



# ROBONOVA-I



Manuale

## Index

## --Tavola dei Contenuti--

## Specifica di ROBONOVA-I - 3

## I. Prima d'incominciare

1. Note sull'assemblaggio e funzionamento - 4
2. Sicurezza di gestione batterie - 5

## II. Note d'assemblaggio

1. elenco delle parti - 6
2. Note su servo-motori e staffe - 8
3. Strumenti necessari per l'assemblaggio - 11
4. Specifiche per HSR-8498HB - 11

## III. Kit d'assemblaggio

1. Adattamento del servo a squadretto - 12
2. Allacciamento di un supporto ad un servo - 14
3. Assemblaggio delle gambe - 16
4. Assemblaggio delle gambe - 28
5. Assemblaggio del corpo - 30
6. Attaccare gambe e braccia in assemblaggio col corpo - 35
7. Attaccare la testa al corpo - 37
8. Attaccare la copertura frontale del corpo - 38
9. Allegare il regolatore al robot e controllo della portata di spostamento - 39
10. Disposizione del cavo del servo - 41
11. Assemblaggio finale dell'hardware - 4
12. Installare il telecomando e il sensore IR - 52

## IV. Preparativi prima del funzionamento di ROBONOVA-1

1. impostazione del movimento di base e applicazioni - 56
  - (1) Inizializzazione di ROBOBASIC per ROBONOVA - 56
  - 1) Settaggio porte e selezione del regolatore - 56
  - 2) Spiegazione dei settaggi iniziali (zero) e procedure - 56
- (2) lista dei comandi trovati nel programma di modello (template) - 56
  - 1) comando "goto AUTO" - 56
  - 2) tabella per l'impostazione del telecomando e azioni - 57

## V. manuale del software di ROBONOVA -

1. installazione e operazione di ROBOBASIC V2.5 - 58
  - (1) informazioni su ROBOBASIC - 58
  - (2) installazione di ROBOBASIC - 58
  - (3) impostazione iniziale di ROBOBASIC - 60
  - (4) programmare in ROBOBASIC - 63
  - (5) impostazione iniziale (zero) di ROBOBASIC - 64
  - (6) controllo del servo in tempo reale - 68
  - (7) ROBOBASIC ROBONOVA-1 controllo del servo motore - 70
    - 1) controllo in tempo reale del servo motore - 70
    - 2) ROBONOVA-1 controllo del servo motore - 72
    - (8) controllo linea diretta - 73

## 2. funzionamento di ROBOSCRIPT V2.5 - 74

- (1) informazioni su ROBOSCRIPT - 74
- (2) impostazione iniziale di ROBOSCRIPT - 74
- (3) programmare ROBOSCRIPT - 76

## 3. funzionamento di ROBOREMOCON V2.5 - 78

- (1) informazioni su ROBOREMOCON - 78
- (2) programmare e operazioni di ROBOREMOCON - 78
- (3) come usare ROBOREMOCON in ROBOBASIC - 80

## VI. applicazione di programma - 82

1. installazione e operazione del programma di modello (template) - 82
  - (1) installazione del programma di modello (template) - 82
  - (2) modi di cambiare il programma di modello (template) - 82
2. applicazioni della programmazione - 86
  - (1) blocco del movimento e suo inserimento - 86
  - (2) come IMPORTARE un file di ROBOSCRIPT - 89
  - (3) conversione AD - 90
  - (4) settaggi del giroscopio - 90
  - (5) come usare I2C - 93
  - (6) impulso del Protocollo HML - 94
  - (7) impostazione di avvertimento di basso-voltaggio - 95

## Manuale d'Istruzioni di Comando di ROBOBASIC (incluso nel CD)

- Capitolo 1 Sommario dei Comandi di ROBOBASIC - 3
- Capitolo 2 Grammatica generale di ROBOBASIC - 10
- Capitolo 3 Spiegazione di dichiarazione dei comandi in ROBOBASIC - 21
- Capitolo 4 Spiegazione del comando del controllo di flusso - 25
- Capitolo 5 Spiegazione del segnale digitale in entrata e uscita in ROBOBASIC - 44
- Capitolo 6 Spiegazione dei Comandi relativi alla memoria - 55
- Capitolo 7 Controllo del modulo LCD in ROBOBASIC - 61
- Capitolo 8 Spiegazione dei Comandi di controllo del motore in ROBOBASIC - 72
- Capitolo 9 Comandi di ROBOBASIC per il Controllo Musica - 103
- Capitolo 10 Comandi di ROBOBASIC per le comunicazioni esterne - 114
- Capitolo 11 descrizione del Comando di processione del segnale analogico in ROBOBASIC - 126
- Capitolo 12 Comandi di Processione e altri in ROBOBASIC - 138
- Capitolo 13 Descrizione di Comandi in ROBOBASIC - 142

## 1. Facile da Montare

Il suo ROBONOVA-1 è stato progettato per un assemblaggio facile in 6-8 ore utilizzando solamente un cacciavite.

## 2. Servo Motori

Servo motori progettati esclusivamente per operazioni robot

Questo uomo-meccanico completamente articolato, alto 12", è controllato da 16 potenti servo motori digitali HSR-8498HB costruiti specificamente per ROBONOVA-1 dalla Hitec. Questi servo motori fatti su misura sono caratterizzati da un "set di perni" di chiusura delle servo-braccia per un facile montaggio, una protezione per il sovra-voltaggio della corrente, treni di gomme in Carbonite super-forte e una tecnologia "feedback" per una programmazione semplificata.

## 3. Pannelli di Controllo, MR-C3024

Pannello Micom con Interfaccia Flessibile

Il pannello di controllo è il cuore di ROBONOVA-1. Assicurato al retro del robot sotto una resistente protezione di plastica, il pannello di controllo può operare fino a 24 servo motori e 16 moduli accessori. Periferiche opzionali includeranno eventualmente giroscopi, sensori d'accelerazione, moduli di sintesi di linguaggio e periferiche operative come controllori Bluetooth e trasmettenti e riceventi R/C.

Questa sofisticata e flessibile interfaccia permette all'utilizzatore di personalizzare ROBONOVA-1 fino ad ottenere il robot dei suoi sogni. La microprogrammazione (firmware) può essere aggiornata automaticamente connettendosi con un PC.

## 4. Struttura Metallica

Gli originali supporti dorati in metallo anodizzato dei servi hanno la funzione di un forte ma leggero esoscheletro. ROBONOVA-1 dispone inoltre di componenti di protezione del corpo in plastica duratura che proteggono il pannello di controllo e la batteria da un eventuale danneggiamento.

## 5. Batteria e Caricatore

ROBONOVA-1 è alimentato da una batteria ricaricabile NiMH a cella 5 rispettosa dell'ambiente che assicura un tempo d'operazione di un'ora. Il caricatore incluso D/C a D/C (corrente continua) dispone di un circuito di protezione che aiuta a prevenire incidenti di caricamento della batteria.

## 6. La Chiave - "Easy To Operate" ("Facile da Far Operare")

Con ROBONOVA-1, Programmazione Robot Facile, gli utenti di questo eccitante pacchetto robot hanno l'opportunità di imparare la tecnologia robotica di prima mano. La programmazione è facile con i programmi forniti RoboScript e RoboBasic grazie al cavo di interfaccia PC incluso.

## 1) RoboScript &amp; RoboRemocon

I principianti della tecnologia di programmazione robot gradiranno l'utilizzo del software di programmazione fornito RoboScript. Senza conoscere assolutamente alcun linguaggio di programmazione, gli utenti possono creare sottoprogrammi operativi con un semplice "click del mouse". Usate il programma RoboRemocon per controllare il vostro ROBONOVA-1 con i dati operativi creati con RoboScript.

## 2) RoboBasic

RoboBasic è uno strumento di programmazione basato sul linguaggio di programmazione BASIC ed è fornito per gli utenti più avanzati.

Contenendo comandi specifici per semplici operazioni del robot, RoboBasic può anche essere usato in congiunzione con RoboScript per aumentare le performance del pannello di controllo Micom fornito.

## 3) HMI (Interfaccia Multi-protocollo Hitec)

- Aggiornamento di microprogrammazione (firmware) & vari settaggi connettendosi con il connettore del servo motore
- Impostazioni dei parametri (2 tipi di Gain, banda Morta, ecc.) e cambio del valore d'impostazione durante l'operazione
- Compatibile con PWM (Modulazione d'Ampiezza d'Impulso) utilizzato nell'attuale controllo radio
- Controllare un massimo di 128pcs di servo motori con l'interfaccia seriale a catena Daisy e un PC
- In grado di avere un feedback per il voltaggio, la corrente e la posizione usando l'HMI

## 4) Funzione Prendi &amp; Usa (Catch &amp; Play)

Tecnica di Programmazione Facile

La maniera più facile per programmare ROBONOVA-1 è utilizzando la funzione "Catch & Play". Usando RoboScript o RoboBasic, semplicemente muovete il robot in una posizione qualsiasi e cliccate sul mouse per "catturare" quella posizione. Muovete il robot in un'altra posizione e ripetete il procedimento.


Il programma a questo punto collega queste posizioni "catturate" e una volta attivato trasmette in maniera armonica i movimenti del robot attraverso queste posizioni programmate.


### |. Prima d'incominciare


- questo manuale contiene una spiegazione dei componenti di ROBONOVA-1 e il loro assemblaggio, il robo-regolatore MR-C3024 e il programma usato per programmare il robot.
- si prega di leggere accuratamente questo manuale e di fare attenzione alle funzioni e all'utilizzo dei prodotti. Questo manuale è anche fornito in formato file PDF sul CD cosicché possa essere salvata una copia su hard-disk o stampata, nel caso sia richiesto.
- Le specifiche possono essere cambiate senza notifica per migliorare la prestazione del prodotto o per altre ragioni. Controllare l'homepage di Hitec per gli aggiornamenti più recenti, (<http://www.hitecrobotics.com>)

### Precauzioni di Sicurezza

Le precauzioni sono divise in due categorie, Avvertimenti e Cautela, a seconda del loro rapporto di sicurezza. Assicurarsi di leggere le note di Avvertimento e Cautela prima di tentare l'assemblaggio del kit.

 **! PERICOLO:** Applicato quando il rischio di Morte, ferite gravi, o la possibilità di un danno essenziale possano capitare se le istruzioni non sono seguite.

 **! AVVERTIMENTO:** Applicato quando sia il danno dell'utente sia quello dell'equipaggiamento possono capitare se le istruzioni non sono seguite.

 **! CAUTELA:** Applicato quando vi sia pericolo di lesione o danno dell'equipaggiamento.1.

### Note sull'assemblaggio e operazione

Questo manuale contiene informazioni sulla sicurezza per prevenire lesioni corporali o danni essenziali.



### ! Pericolo

- Prestare attenzione alle condizioni di sicurezza quando il robot sta operando.
- La prestazione e l'operazione non sono garantite dal momento che si tratta di un kit non assemblato e se assemblato in maniera scorretta si può incorrere in lesioni o danni.
- Assemblare il kit in un'area con ventilazione adeguata.



### ! Avvertimento

- Tenere lontano dalla portata dei bambini. Anche se il prodotto può sembrare un giocattolo, potrebbe ferire un bambino se lasciato incontrollato.
- Spegnerne e scollegare la batteria immediatamente se capita un problema. Se il prodotto si rompe, o è esposto a liquidi, fiamme, o calore, potrebbe capitare un incidente.
- Non disassemblare o modificare il caricatore della batteria o i cavi.  
(la riparazione del cavo è fornita dal servizio clienti)
- Scollegare il caricatore della batteria quando non in uso.
- Non disassemblare o modificare mai i pannelli di circuito dei servo motori
- non usare in condizioni di calore, bagnato o freddo. Il kit consiste di componenti di precisione. Se esposto a condizioni estreme potrebbero incorrere problemi.
- Se assemblato in maniera scorretta, potrebbero capitare danni o lesioni.
- assicurarsi sempre che la spina del carica batterie sia al sicuro durante il caricamento. Rimuovere immediatamente quando il processo di carica sia completo.
- per favore leggere attentamente il manuale. Fare attenzione alla direzione dei servo motori e dei supporti durante l'assemblaggio. Se assemblati scorrettamente, disassemblare e rimontare correttamente.



### ! Cautela

- I servo motori inclusi nel kit richiedono una periodica manutenzione per mantenere una prestazione ottimale.
- una migliore prestazione si consegue utilizzando il robot su una superficie larga, liscia e piatta. Se la superficie è irregolare o troppo piccola, il robot potrebbe cadere e un danno potrebbe verificarsi.
- non imbracciare il robot durante l'accensione o durante l'operazione.
- non spegnere il regolatore MR- C3024 durante l'aggiornamento della microprogrammazione (firmware). Il programma potrebbe non riuscire o danneggiarsi (contattare il servizio riparazioni).

### 2. Gestione sicura della batteria

- questo kit contiene una batteria NiMH per la sorgente elettrica. La batteria NiMH è una batteria ricaricabile ad alto voltaggio e richiede un attento impiego durante l'utilizzo, riposo e carica
- inserire il carica batteria in una sorgente AC e connetterlo al regolatore, la batteria è caricata attraverso il regolatore. Quando il caricatore è connesso, il LED si accenderà di rosso. Quando il ciclo di caricamento è completo, il LED del caricatore si accenderà di verde.
- Per caricare la batteria fuori dal regolatore, si richiede un cavo di carica opzionale (Parte numero 77102)



### ! Avvertimento

#### 1) Gestione della batteria

- la batteria inclusa in questo kit ha una carica minima. Una batteria NiMH dev'essere conservata con un po' di energia rimanente della batteria.
- Se la batteria NiMH è completamente scarica e lasciata per un lungo tempo, la prestazione della batteria diminuirà.
- la batteria dev'essere rimossa dal pannello o dal carica batterie quando non usata. Conservare la batteria in un luogo fresco e asciutto.



### ! Avvertimento

#### 2) Tempo di carica

- Sono richiesti all'incirca 70 minuti per una carica completa delle batterie. Questo tempo varia a seconda della potenza rimanente della batteria.
- durante il caricamento non lasciare la batteria inosservata. Interrompere il processo di caricamento se la batteria diventa esageratamente calda.
- un sovraccaricamento può risultare dannoso al pacchetto.



### ! Cautela

#### 3) Maneggiamento della batteria

- Non disassemblare o modificare il connettore o l'impianto elettrico della batteria.
- Assicurarsi che nessun oggetto esterno entri in contatto con la spina del connettore e che non ci siano cavi scoperti esposti.
- Non sottoporre la batteria a estreme temperature o ad un ambiente umido. Conservare la batteria in un luogo fresco e asciutto.
- Tenere alla larga altre sorgenti conduttive durante il trasporto o lo stoccaggio.
- Se i cavi della batteria diventano logori o rovinati, sostituire il pacchetto della batteria.



### ! Cautela

#### 4) Primo soccorso e disposizione

- Nel caso che la batteria si danneggi e la spaccatura causi il contatto tra gli elettroliti e un'area di pelle o gli occhi, sciacquare l'area colpita con acqua.
- Nel caso di contatto con gli occhi, consultare uno specialista il prima possibile.
- L'elettrolita della batteria è una sostanza tossica. Non è soltanto dannosa per il corpo umano ma potrebbe danneggiare anche la casa e il mobilio.
- Se le batterie NiMH non mantengono più la carica, gettarle negli appropriati raccoglitori o in accordo con le leggi e regole del vostro paese.
- Non gettarle nell'inceneritore.

## II. Note d'assemblaggio

### ROBONOVA-I Elenco della parti

6 PCS	4 PCS	8 PCS	2 PCS	1 PCS	1 PCS	1 PCS
HR1B-0001	HR1B-0002	HR1B-0003	HR1B-0004	HR1B-0005	HR1B-0006	HR1B-0007
Tapped U Type Universal Bracket	Non-Tapped U Type Universal Bracket	I-Type Universal Bracket	H-Type Bracket	Back Body Frame	Front Body Frame	Top Body Frame
1 PCS	1 PCS	1 PCS	1 PCS	1 PCS	2 PCS	2 PCS
HR1C-0001	HR1C-0002	HR1C-0003	HR1C-0004	HR1C-0005	HR1C-0006 (Top Hand)	HR1C-0007 (Bottom Hand)
Front Body Cover	Back Body Cover	Goggle Cover	Front Head Cover	Back Head Cover	Top Hand Cover	Bottom Hand Cover
1 PCS	3 PCS	1 PCS	3 PCS	1 PCS	1 PCS	1 PCS
HSR-8498HB1R200 (Sticker No.1)	HSR-8498HB1R300 (Sticker No.2)	HSR-8498HB1L200 (Sticker No.3)	HSR-8498HB1L300 (Sticker No.4)	HSR-8498HB2R200 (Sticker No.5)	HSR-8498HB2R400 (Sticker No.6)	HSR-8498HB2L200 (Sticker No.7)
130 PCS	12 PCS	6 PCS	2 PCS	40 PCS	12 PCS	4 PCS
PH/T-2 2*4 NI	PH/T-2 2*5NI	PH/T-2 2*9 NI	PH/T-2 2*26 BK	PH/M 2*4 NI	PH/M 2.6*4 NI	PH/M 3*4 NI
Pan Head Tapping Screw	Pan Head Tapping Screw	Pan Head Tapping Screw	Pan Head Tapping Screw	Pan Head Screw	Pan Head Screw	Pan Head Screw



\* Gli accessori sopra sono inclusi nel servo HSR-8498HB

1 PCS	1 PCS
Pin Cover	Battery Wire Protector
For MR-C3024	

1 PCS	2 PCS	2 PCS	2 PCS	2 PCS	1 EA
HR1B-0008	HR1B-0009	HR1B-0010	HR1B-0011	HR1B-0012	MR-C3024
Bottom Body Frame	Shoulder Back Universal Bracket	Shoulder Front Universal Bracket	Hand Bracket	Foot Bracket	Controller
1 PCS	1 PCS	1 PACK	1 EA	1 EA	1 EA
HR1C-0008	HR1C-0009	Ni-Mh Battery (1,000mAh/ 6.0V/ 5 Cell)	CD-ROM	Interface Cable	Quick Charger
Right Foot Cover	Left Foot Cover	6.0 V 5Cell	ROBOBASIC & ROBONOVA-I User Manual	MR-C3024 Serial Interface Cable	6V 1,000mAh / 100-240V
1 PCS	1 PCS	1 PCS	1 PCS	1 PCS	2 PCS
HSR-8498HB2L400 (Sticker No.8)	HSR-8498HB3R200 (Sticker No.9)	HSR-8498HB3L200 (Sticker No.10)	HSR-8498HB2R300 (Sticker No.11)	HSR-8498HB2L300 (Sticker No.12)	Insert Bolt
4 PCS	28 PCS	1 PCS	8 PCS	28 PCS	2 PCS
Support 3*5mm	FW 6*2.2*0.5 NI	HSR8498HA2	Cable Tie	Cable Clamp	Lug
Support	Flat Washer	Wheel Horn			

\*Altri viti e dati si possono anche comprare da un negozio di zona.

Differenza tra viti PH/M e PH/T.



PH/M (Machined Screw)

PH/T (Tapping Screw)



## 2. Note su servo-motori e staffe

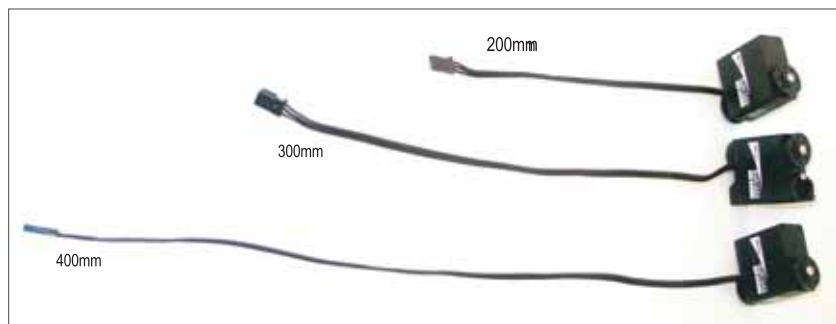
La custodia dei servo motori HSR-8498HB, gli squadretti, la lunghezza dei cavi, e la direzione dei cavi sono ottimizzati per l'operazione e l'assemblaggio di ogni giunto. Controllare le immagini nel manuale.

- Il cablaggio del servo motore è composto da 3 cavi, 2 neri e 1 grigio. Non disassemblare il connettore dal cavo.
- Durante l'assemblaggio, rifarsi alle immagini allegate per orientarsi al meglio.

Sono inclusi nel kit 3 differenti tipi di custodie



Sono inclusi nel kit 3 cavi di lunghezza diversa

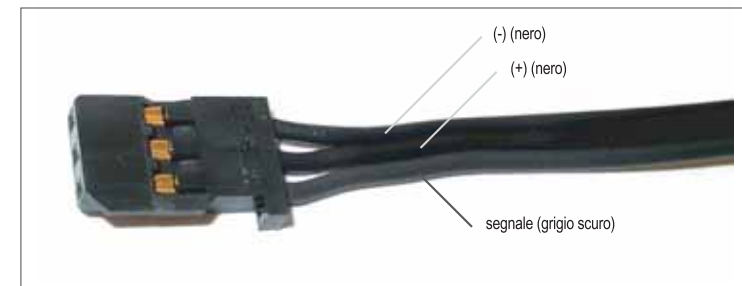


Sono inclusi nel kit 3 diverse orientazioni dei cavi

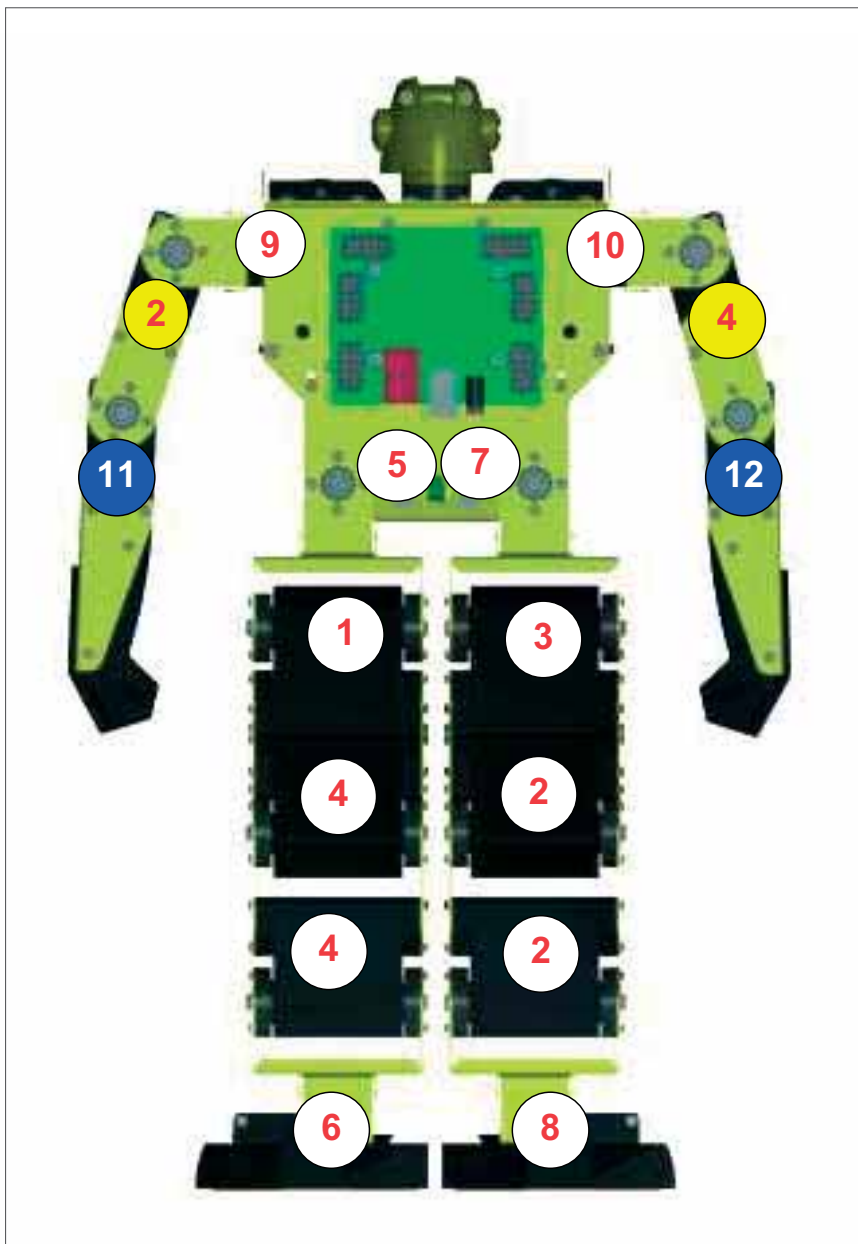


Orientazione destra "senza marchio CE" | Orientazione sinistra "con marchio CE"

Il cavo del servo motore è composto da 1 filo grigio scuro (segnale) e 2 fili neri

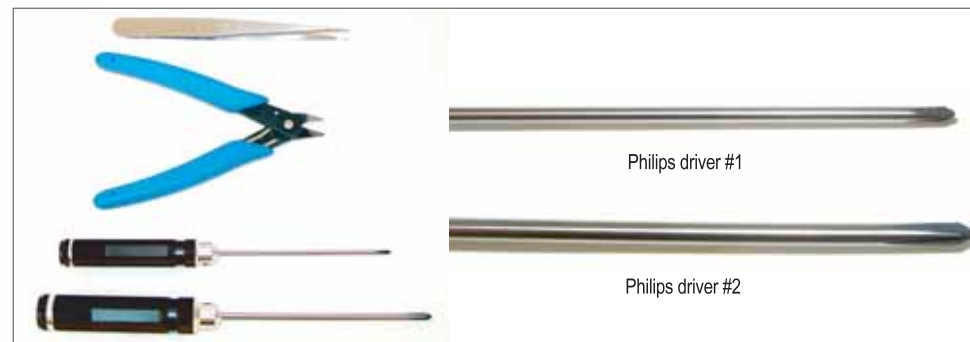


- piazzamento del servo HSR-8498HB a seconda del numero di etichetta  
(Tutti i servo motori HSR-8498HB sono numerati con etichette a seconda della custodia, lunghezza cavo e direzione per un assemblaggio semplice.)



### 3. Strumenti necessari per l'assemblaggio

Cacciaviti d'alta qualità Philips (PH #1 e PH #2), pinzette, tronchesi e blocca viti.



### Requisiti minimi di sistema del computer

- sistema Windows
- AMD o Pentium 300Mhz
- 60MB di spazio libero su hard-disk
- 128MB di RAM

### 4. Specifiche per HSR-8498

#### -- SPECIFICHE

- Interfaccia : Protocollo HMI, PWM
- voltaggio operativo: 6,0V\*
- velocità massima: 0,20sec/60° a 6,0V
- Torsione di stallo: 10kg.cm (138,87oz.in) a 6,0V
- angolo operativo: massimo 180°
- peso: 55g (1,94oz)
- dimensione: 40 X 20 X 47mm

#### -- SPECIFICHE D'IMPULSO

- controllo impulso
- neutrale : 1500µsec, 0 - 180°: +/- 1100 - 1900µsec
- ciclo d'impulso: 12-26mSec (comune : 21mSec)



### III. Kit d'Assemblaggio

#### 1. Adattamento del servo a squadretto

Quattro diversi tipi di servo a squadretto sono forniti con l'HSR-8498HB (due diversi "splined" e due diversi inattivi ["idler"]).

Numeri sono stampati in rilievo sullo Squadretto a Ruota Splined.



I numeri si possono trovare sullo squadrette splined vicino alla vite



Le squadrette splined sono ideate per un facile ed accurato assemblaggio.

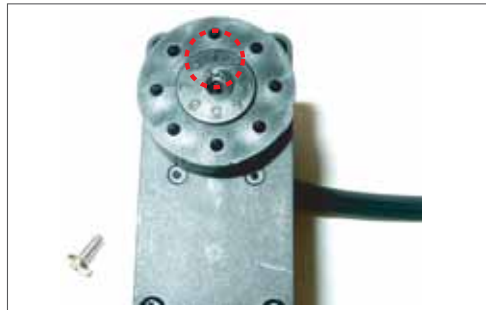


Foto del servo con le squadrette splined rimosse. Viene usata una vite BH/T 2,6X6mm per annettere lo squadrette.



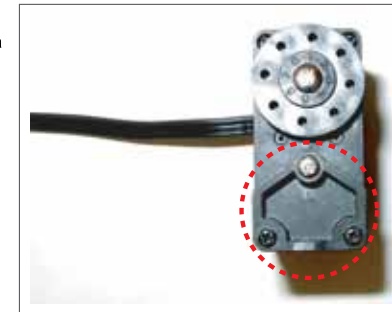
La squadretta inattivo ("idler") gira liberamente.



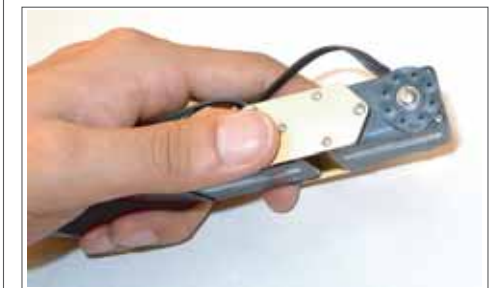
Foto della squadretta idler rimosso. Vengono usate una vite BH/T 2,6X6mm e una rondella piatta (2,8X7,6mm) per annettere lo squadrette.



I servo HSR-8498HB1 ha una montatura a staffa di forma pentagonale.



Per annettere una staffa, svitare le tre viti nel pentagono e attaccare la staffa con quelle.

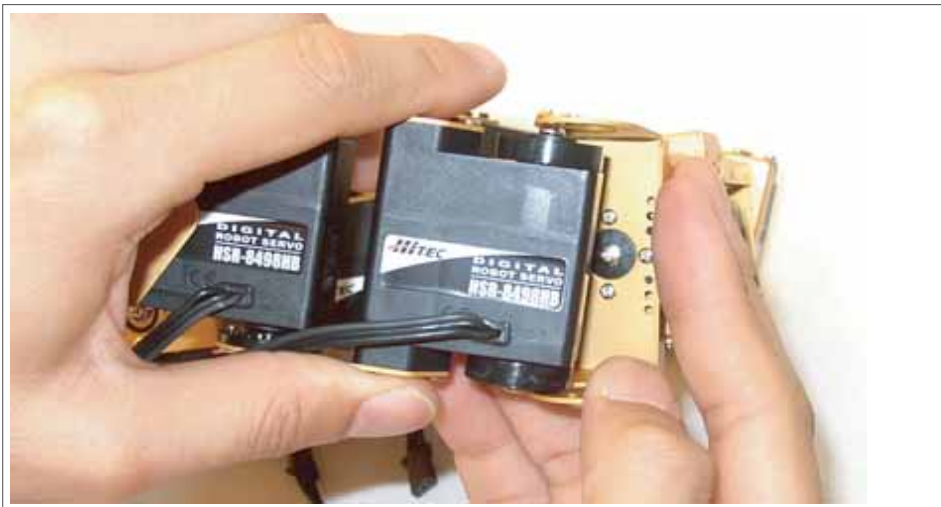


## 2. Allacciamento di un supporto ad un servo.

(1) Porre il supporto sopra le due squadrette con cura, il supporto può essere piegato lievemente per adattarsi sopra le due squadrette.

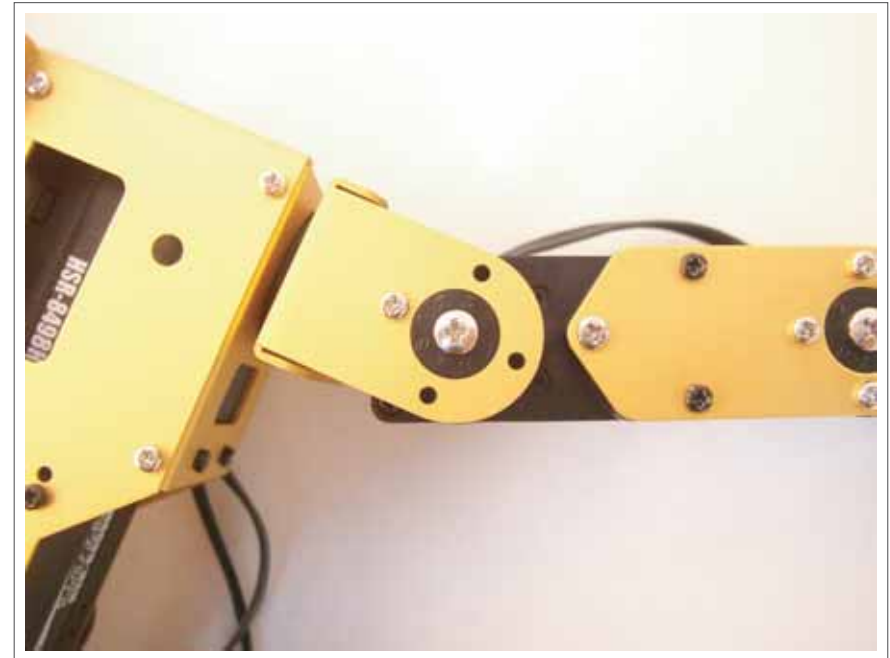


Quando le viti sono attaccate il supporto si ripiegherà nella forma.



## (2) Controllo della gamma di movimento

Controllate la gamma di movimento muovendo i servi con la mano.  
Rifarsi alla sezione "controllo della gamma di spostamento" per l'effettiva gamma di ogni giunto.





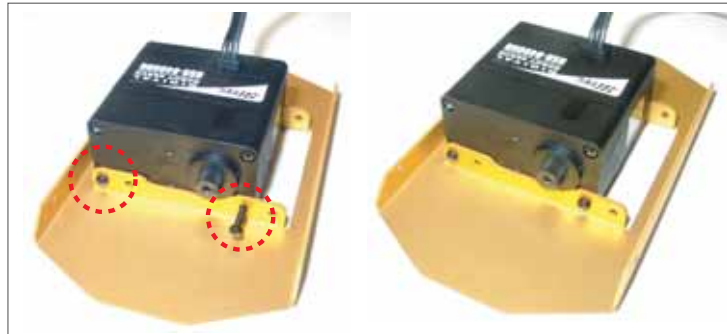
### 3. Assemblaggio delle gambe

#### (1) Assemblaggio del piede destro e del sinistro

Rimuovere le due viti nere dal davanti, e le due viti argentate da dietro del servo 2L400 (Parte # 8, piede destro) e del 2R400 (Parte # 6, piede sinistro) dove si collegheranno al supporto del piede R1B-0012. Rimuovere la vite dalla squadretta frontale e dalla squadretta inattivo del retro. Staccate queste squadrette dai servi. Notare le posizioni delle viti nere e argento dal momento che non sono intercambiabili.



Attaccare un supporto R1B-0012 al servo 2L400 (parte # 8) come mostrato in figura e re-inserite le quattro viti attraverso il foro nel supporto. Riattaccate i servi a squadretto al servo. Una volta finito il filo del servo deve essere in cima al servo.



Utilizzare gli stessi passaggi menzionati precedentemente per assemblare il piede sinistro.



Quando completato, entrambi i piedi dovrebbero trovarsi come nella figura.

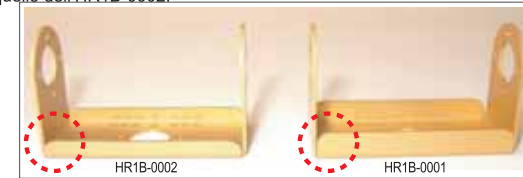


#### (2) Assemblaggio del supporto coscia e caviglia. (4 parti)

Per sapere la differenza tra i due tipi di supporti, guardare l'immagine. I quattro supporti HR1B-0002 sul lato sinistro dell'immagine non presentano fori. Gli otto supporti HR1B-0001 sul lato destro dell'immagine hanno quattro fori ognuno.



Un'altra maniera per differenziare i due tipi di supporto è quella di confrontare i due bordi laterali. Il supporto HR1B-0001 ha un bordo metallico più grande di quello dell'HR1B-0002.



Si richiedono quattro viti PH/M 2X4mm per attaccare un supporto HR1B-0001 e uno 0002 insieme.



Assemblare il supporto HR1B-0001 e lo 0002 come mostrato in figura utilizzando le quattro viti PH/M 2X4mm per congiungerli. Prestare particolare attenzione alle posizioni delle viti. Assemblare quattro parti.





### (3) Assemblaggio caviglia destra e sinistra.

Assemblare la caviglia destra:

Fissare le caviglie destra e sinistra ai piedi precedentemente assemblati. Utilizzare un supporto caviglia HR1B-0001 e fissarlo alla squadretta frontale e a quella posteriore dei servi 2L400 (piede destro) e 2R400 (piede sinistro). Sono richieste otto viti PH/T 2X4mm.



Usando la figura come esempio, posizionate il supporto della caviglia sopra i due squadretti con attenzione. Se necessario il supporto della caviglia può essere piegato lievemente per posizionarsi al meglio sopra gli squadretti. Una volta fissato agli squadretti con le viti, il supporto riprenderà la sua forma.



Controllate l'orientazione del filo del servo e la forma del supporto. Usate la figura come esempio



Fissate il supporto alla squadretta frontale utilizzando quattro viti PH/T 2X4mm come mostrato in figura. Prestare attenzione alla posizione della squadretta frontale e del supporto. La posizione #1 sulla squadretta deve essere a ore 3 (90°) per il piede destro e a ore 9 (270°) per il sinistro. Vedere immagine.



-posizioni delle viti per il piede destro-

Usare altre quattro viti PH/T 2x4mm per fissare il supporto alla squadretta inattivo.

Dopo l'assemblaggio, i piedi e le caviglie dovrebbero apparire come mostrato in figura. Completato questo, i fili di ogni servo dovrebbero correre lungo la suola interna di ogni piede.

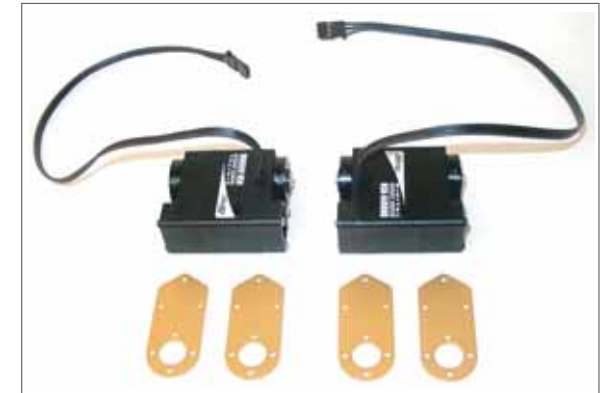


Assemblaggio caviglia sinistra

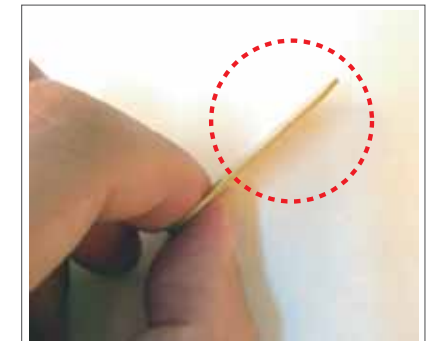
-posizioni delle viti per il piede sinistro-

### (4) Assemblaggio degli stinchi

Parti richieste per l'assemblaggio degli stinchi:  
4- supporti HR1B-0003 e uno per il servo 1L300 (parte #4, stinco sinistro) e uno per l'1R300 (parte #5, stinco destro).



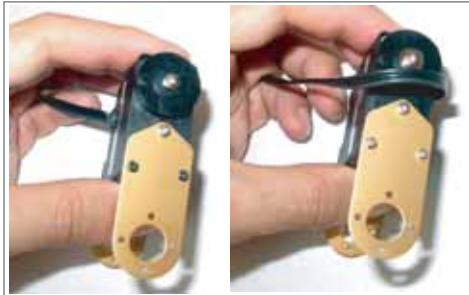
il supporto HR1B-0003 ha un lato con bordo morbido e uno con bordo sottile. Per sicurezza durante l'operazione il lato con bordo morbido del supporto dovrebbe essere assemblato verso la parte esterna del robot.



Rimuovere le sei viti fissate alle parti pentagonali del servo 1L300. Ci sono tre viti argento dietro e una vite argento e due nere davanti. Mettete i due supporti HR1B-0003, uno per lato, sui servi e rifissate le viti.



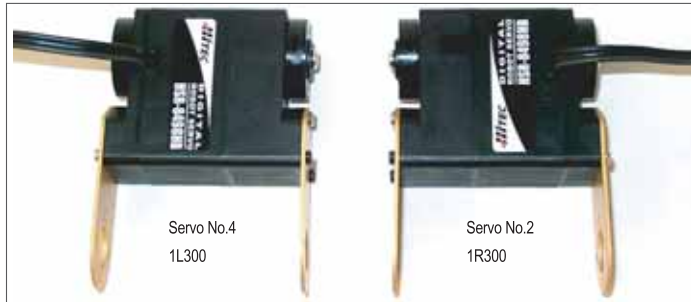
Seguite gli esempi nelle figure per un piazzamento corretto delle viti. Notare le posizioni delle viti nere e argento.



Davanti del servo

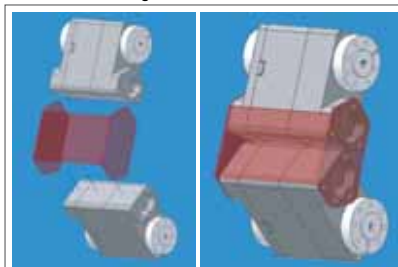
dietro del servo

Costruite l'altro stinco usando il servo 1R300. entrambi gli stinchi sono assemblati nella stessa maniera.

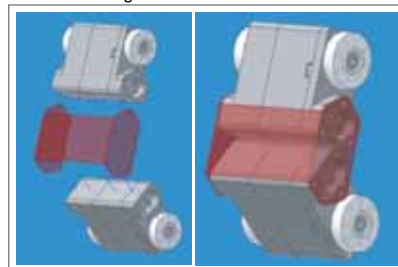


(5) Assemblaggio delle ginocchia

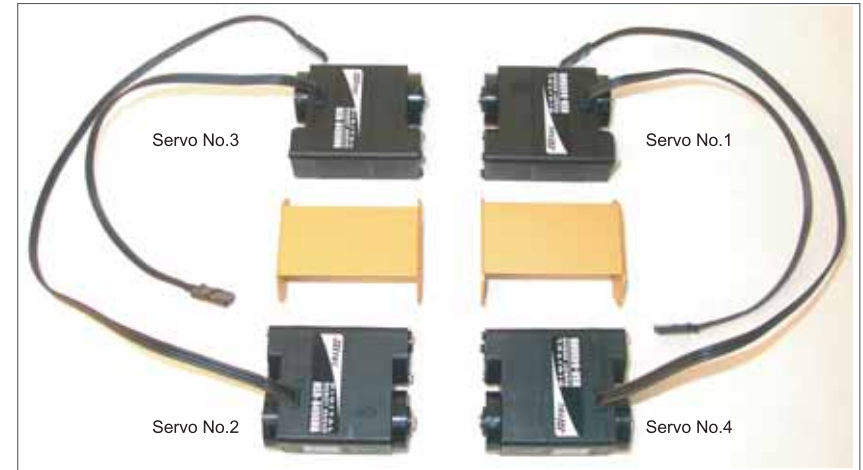
ginocchio destro



ginocchio sinistro



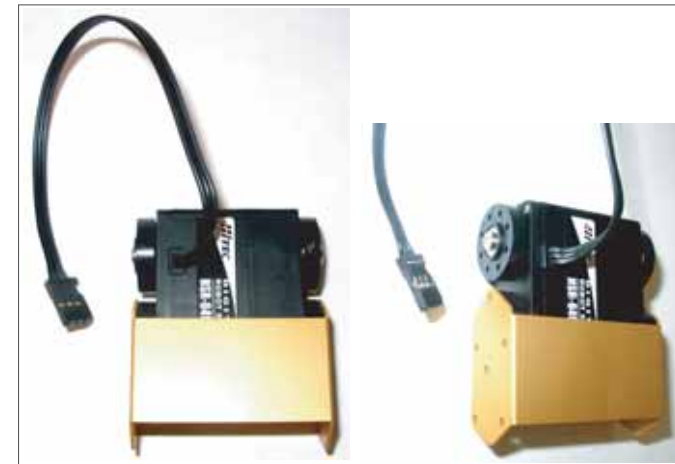
Parti richieste per l'assemblaggio: 1- 1L200 (parte #3, servo superiore del ginocchio destro), 1- 1R300 (parte #2, servo inferiore del ginocchio destro), 1- 1R200 (parte #1, servo superiore del ginocchio sinistro), 1- 1L300 (parte #4, servo inferiore del ginocchio sinistro) e 2- HR1B-0004 supporti delle ginocchia.



Rimuovete le sei viti fissate nelle parti pentagonali di entrambi i servi superiori davanti e dietro in preparazione dell'assemblaggio delle ginocchia.



Guardando la figura, fissare un supporto del ginocchio HR1B-0004 ai servi n.3 e n.1 e bloccate con le viti precedentemente rimosse facendo attenzione alla posizione delle viti nere e argento.



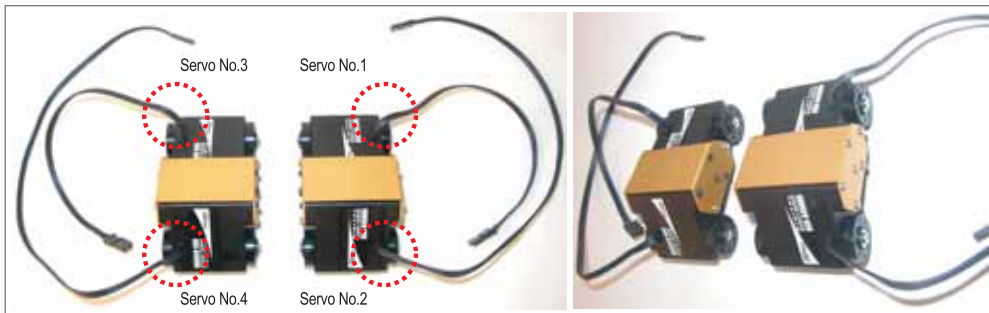
Rimuovete le viti dai settori pentagonali davanti e dietro dei servi 1R300 (parte #2, ginocchio inferiore destro) e 1L300 (parte #4, ginocchio inferiore sinistro). Dopo aver osservato attentamente la figura, fissate i servi 1R300 e 1L300 al supporto HR1B-0004 assicurandovi che i servi siano posizionati sotto al ginocchio nella corretta orientazione.



Bloccate con le viti precedentemente rimosse prestando attenzione al piazzamento delle viti in base al colore.

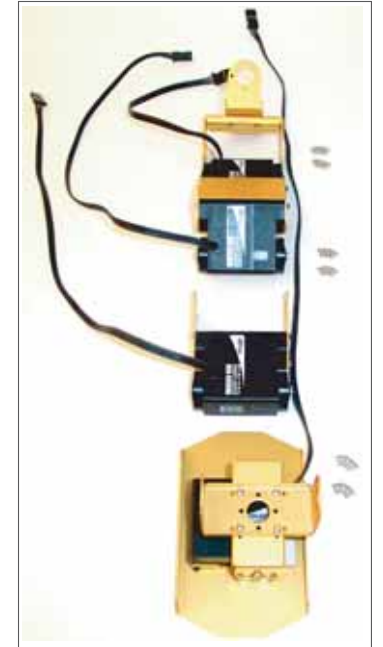
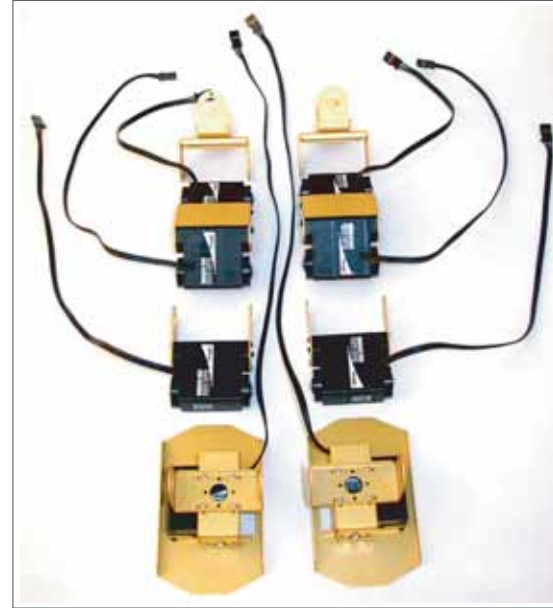


Assicurarsi che la direzione del cavo di ogni servo sia come quella in figura.

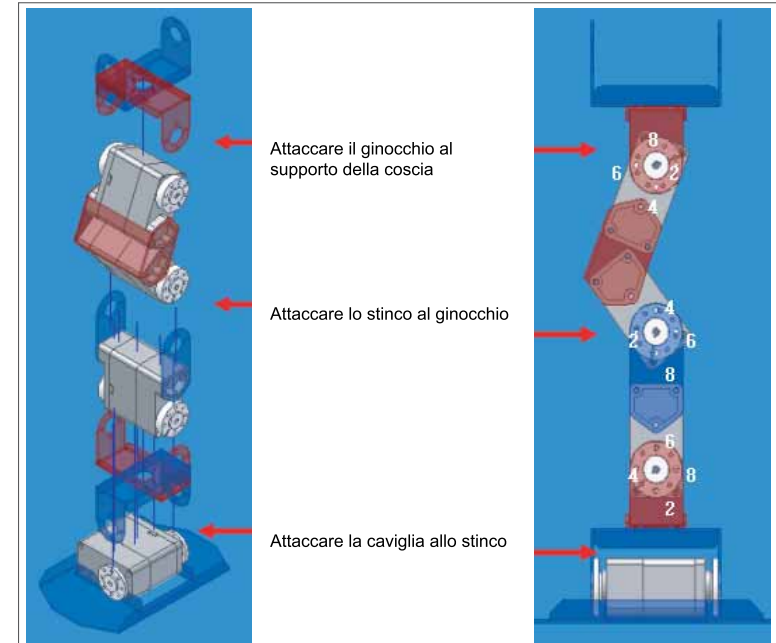


## (6) Assemblaggio di tutta la gamba

Assemblare prima la gamba destra. Sistemate le parti come mostrato in figura prima di procedere col processo d'assemblaggio.



Posizionare i supporti e servi a squadretta frontali seguendo i numeri impressi sui servi a squadretta.





### Fissare piede e stinco

- fissate lo squadretta del servo dello stinco destro (1R300, #2) al supporto del piede destro.
- seguendo la figura per una posizione corretta dello squadretta, bloccate il supporto allo squadretta frontale con quattro viti PH/T 2X4mm.
- Bloccate con quattro viti PH/T 2X4mm il supporto al servo inattivo a squadretta.



### Assemblare stinco e ginocchio

- fissate il supporto dello stinco HR1B-0003 agli squadretti del servo (1R300, #2) al ginocchio inferiore.
- seguendo la figura per una corretta posizione dello squadretta, bloccate il supporto allo squadretta frontale con quattro viti PH/T 2X4mm.
- bloccate il supporto al servo inattivo a squadretta con quattro viti PH/T 2X4mm.



### Assemblare ginocchio e coscia

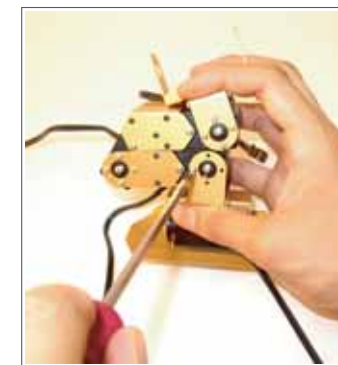
- fissate il supporto della coscia costruito precedentemente alla squadretta del servo (1L200, #3) al ginocchio superiore.
- seguendo la figura per una corretta posizione della squadretta, bloccate il supporto alla squadretta frontale con quattro viti PH/T 2X4mm.
- bloccate il supporto alla squadretta posteriore con quattro viti PH/T 2X4mm.



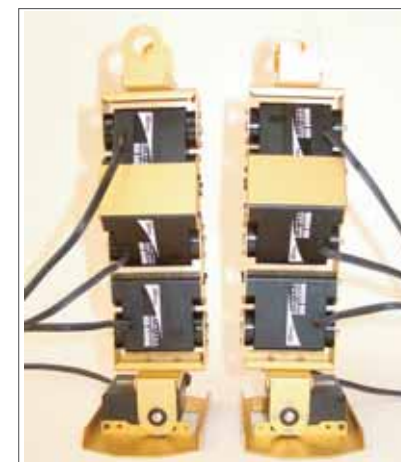
Cautela! Le gambe sono montate correttamente se i servi hanno una gamma completa di movimento.



Se non potete piegare completamente le gambe come mostrato in figura, rimuovete le viti dai supporti e modificate la posizioni delle squadrette. Poi rifissate il supporto con le viti.

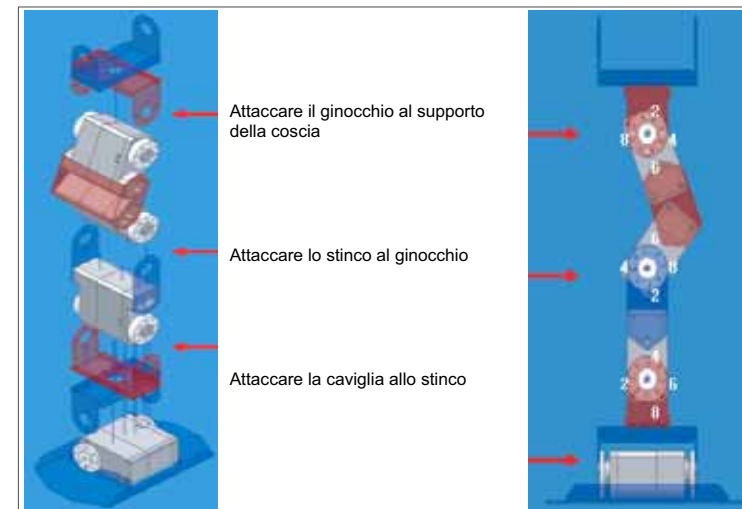


Visione frontale delle gambe assemblate.



Posizionare i supporti e le squadrette frontali seguendo i numeri impressi sulle squadrette.

La gamba sinistra è assemblata esattamente come quella destra. Fare attenzione alla posizione dei numeri sulle squadrette frontali durante l'assemblaggio.



## (7) Disposizione del filo del piede e assemblaggio della Copertura del Piede

Questa parte serve a disporre i servo-cavi e per installare le coperture dei piedi di ROBONOVA-1.



preparate due morse per filo (fasce trasparenti) e due rondelle 2.2X0.5mm.

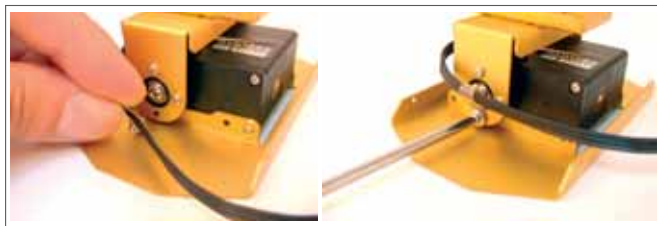


Assemblare prima il piede destro. Rimuovere la vite che si trova in basso della squadretta inattivo.

Piegate la morsa per filo attorno al filo.



Inserite la vite rimossa, con la rondella, attraverso i fori nella morsa del filo e fissatela alla squadretta inattiva.



Forma del piede con la morsa del filo attaccata.



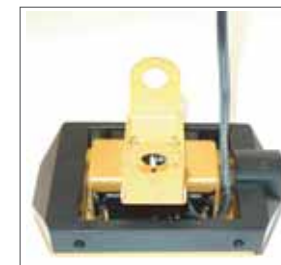
Le coperture del piede HR1C-0008(lato destro), 0009(lato sinistro) adesso possono essere fissate ai piedi con quattro viti PH/M 2X4mm ognuna.



Fate scivolare le coperture del piede sopra le gambe. Il bordo piano della copertura del piede dovrebbe guardare all'interno e il bordo smussato dovrebbe essere rivolto verso la parte esterna.



Controllare la direzione del cavo per ogni piede. Non fare contatto con la copertura del piede.



Bloccate il piede con 4 viti PH/M 2X4mm.



Piedi completati.

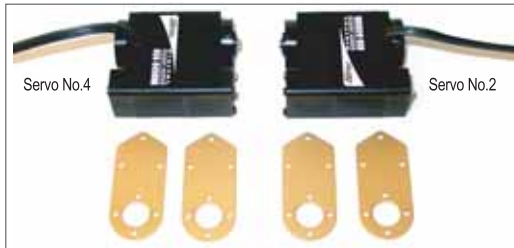




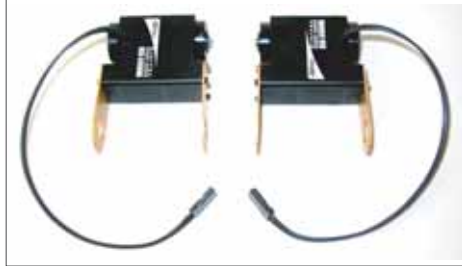
## 4. Assemblaggio delle Braccia

### (1) Assemblaggio della spalla

rimuovete le sei viti fissate nelle parti pentagonali dei servi 1R300 (parte #2, lato sinistro) e 1L300 (parte #4, lato destro) e attaccate i due supporti HR1B-0003 con le viti rimosse. Notare la posizione delle viti nere e argento.



Assemblare le due parti, spalla sinistra e spalla destra sono assemblate nella stessa maniera.



### (2) Assemblaggio dell'avambraccio

Rimuovete quattro viti da entrambi i servi 2R300 (parte #11) e 2R300 (parte #12). Le viti sono posizionate in fondo ai servi. Non rimuovere le viti vicino alle squadrette. Fissate i supporti H1B-0011 e bloccateli con le viti rimosse. Fare attenzione alla posizione delle viti nere e argento quando si riassume. Assicurarsi che la direzione dei servo-cavi di entrambi i servi sia verso l'esterno del



Per assemblare l'avambraccio destro, unite l'HR1C0006 il 0007 e bloccateli al supporto 0011 con quattro viti PH/T 2X5mm.



L'avambraccio sinistro viene montato allo stesso modo. L'HR1C-0006 e 0007 sono usati anche sul lato sinistro.

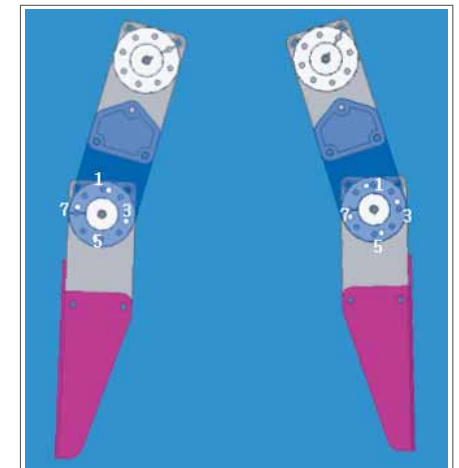


### (3) Assemblaggio dell'intero braccio

Unite la spalla montata precedentemente e gli avambracci usando otto viti PH/T 2X4mm per braccio.



I cavi dei servi per le spalle e gli avambracci devono essere rivolti verso l'esterno. Usando la figura come riferimento, modificate lo squadretta frontale in maniera che i numeri impressi corrispondano alla figura e fissatelo al supporto con quattro viti PH/T 2X4mm. Fissate le squadrette inattive con quattro viti PH/T 2X4mm.



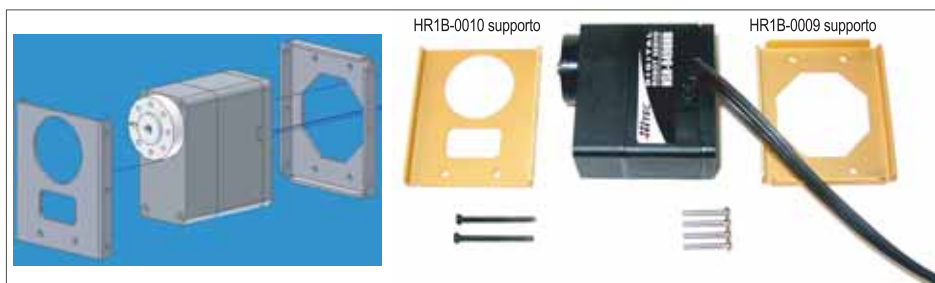
Quando completate le braccia dovrebbero assomigliare all'immagine.



## 5. Assemblaggio del corpo

(1) Attaccare la spalla all'interno del corpo

Rimuovete i bulloni dai servi 3R200(n.9) e 3L200(n.10). Poi disponete i supporti HR1B0009 e 0010 sul servo motore.



Usando la figura come riferimento, rimuovete le due viti nere dalla parte frontale dei servi 3R200(n.9) e 3L200(n.10). Non rimuovete le due viti vicine allo squadretta. Fissate il supporto HR1B-0010 al servo reinserendo le viti. Per installare i supporti HR1B-0009 sui servi, rimuovete le quattro viti di colore argento dalla parte posteriore di ogni servo e bloccate il supporto reinserendo le quattro viti. Assemblate i due.



## (2) Assemblaggio dell'addome frontale

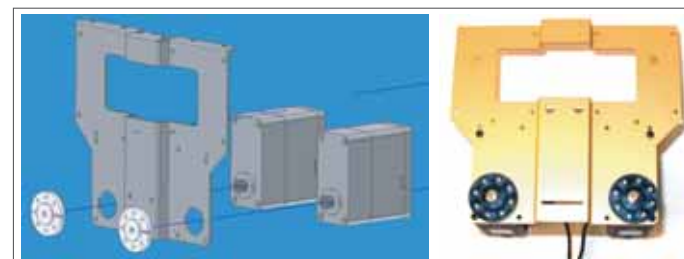
Predisponete i servi 2R200 (parte #5, addome sinistro) e 2L200 (parte #7, addome destro) per l'assemblaggio.



Rimuovete le squadrette frontali da ogni servo e rimuovete le due viti nere dal bordo esterno frontale.



Fissate i servi al telaio del corpo HR1B-0006 reinserendo le viti e le squadrette. I cavi dei servi devono essere rivolti verso l'interno del corpo.



Fissate le due spalle al telaio frontale del corpo con quattro viti PH/M 2X4mm ognuna.



Notate la direzione dei servo-cavi. Entrambi devono essere rivolti verso l'interno del corpo.

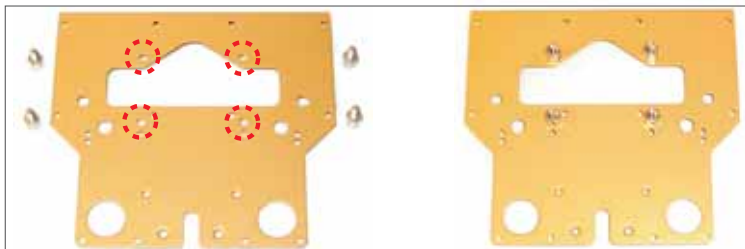


Forma completata.

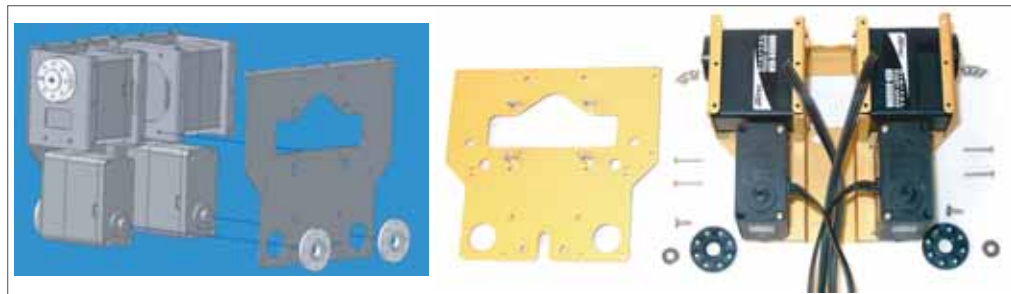


(3) Assemblaggio dell'addome posteriore

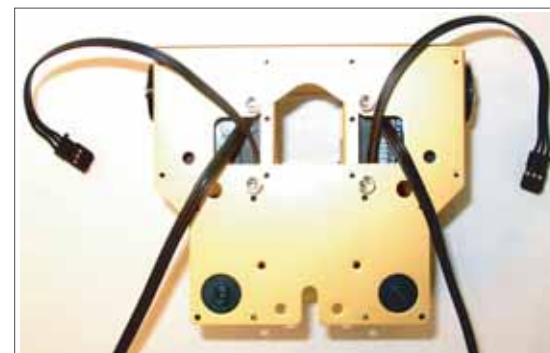
Fissate il supporto a forma di "dado" 5mm-3 al telaio del corpo posteriore HR1B0005 per supportare il regolatore MR-C3024.



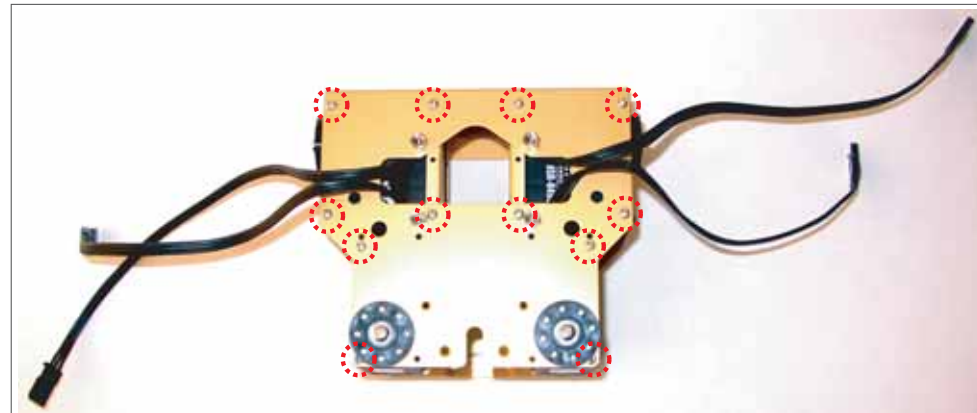
Rimuovete le squadrette inattive dai servi 2R200 (n.5) e 2L200 (n.7).  
Rimuovete le due viti situate sul bordo esterno del servo.



Usando la figura come riferimento, piazzate il telaio del corpo posteriore HR1B-0005 contro il corpo principale.

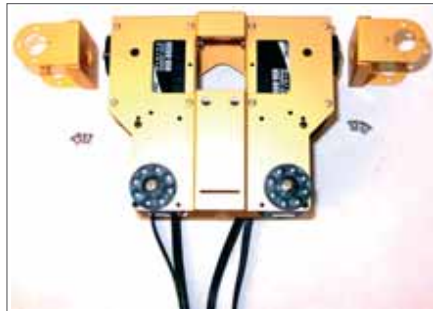


Usare otto bulloni PH/M 2X4mm per fissare il telaio del corpo posteriore HR1B0005 alle spalle e reinserite le quattro viti precedentemente rimosse dall'addome. Riattaccate le squadrette inattive.

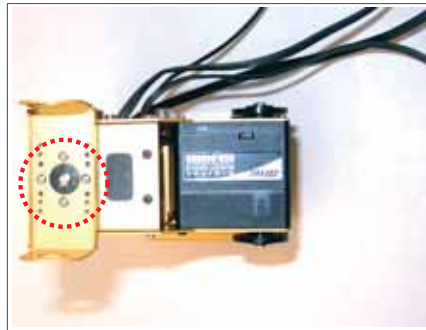




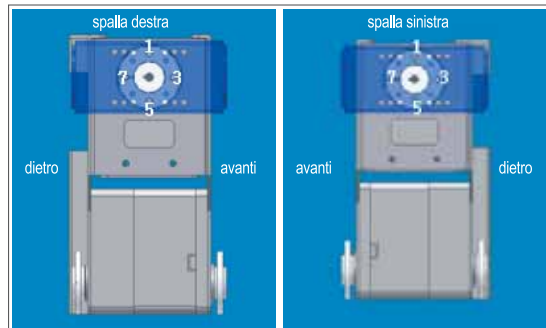
(4) Assemblaggio della spalla  
Fissate un supporto HR1B-0002 su ognuno dei servi 3L200 (parte #10, spalla destra) e 3R200 (parte #9, spalla sinistra) installati sul corpo.



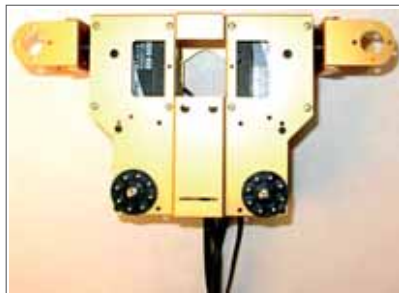
Usando la figura come riferimento, bloccate il supporto su ogni squadretta con quattro viti PH/T 2X4mm.



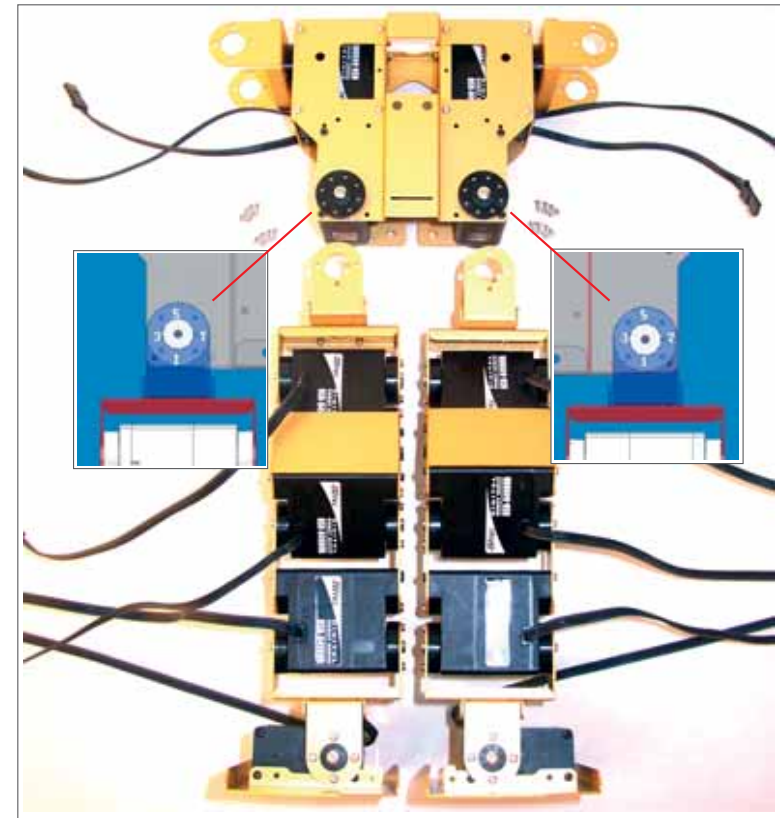
- quest'immagine mostra in dettaglio la posizione della squadretta sul supporto.



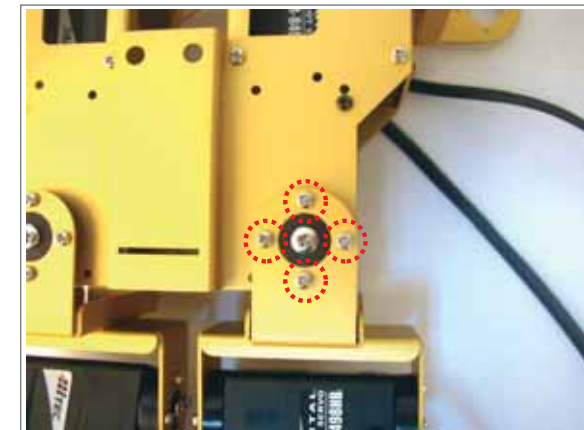
la forma completata del corpo.



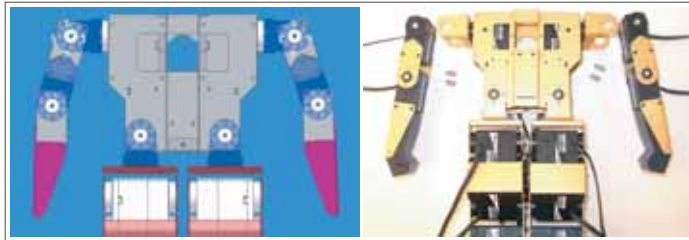
6. Fissare le gambe e le braccia al corpo  
(1) Assemblaggio del corpo delle gambe



Rifersi all'immagine per una posizione corretta dello squadretta durante l'annessione delle gambe al corpo. Fissate il supporto ad ogni squadretta (Scanalato e Inattivo) con quattro bulloni PH/T 2X4mm.



## (2) Assemblaggio del corpo e delle braccia



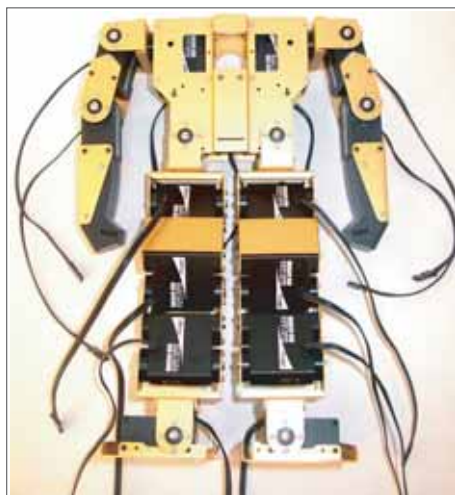
rifarsi all'immagine per le posizioni corrette delle squadrette durante l'installazione delle braccia sul corpo. Fissate il supporto ad ogni squadretta (Scanalato e Inattivo) con quattro bulloni PH/T 2X4mm.



dopo aver fissato le squadrette al supporto, controllate la gamma di moto muovendo i servo con la mano. Le braccia sono attaccate correttamente quando è possibile una rotazione di 180°.



Forma completa.



## 7. Assemblaggio della testa e del corpo



Fissate la squadretta del servo HSR8498HA2 al telaio del corpo superiore HR1B-0007 con quattro viti PH/T 2X8mm.



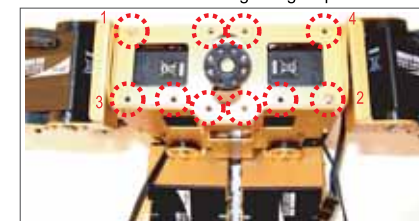
annettete il pannello del LED al visore (parte # HR1C-0003) con due viti PH/T 2X4mm



fissate il supporto HR1B-0007 al copro usando dieci bulloni PH/T 2.6X4mm.



per posizionare l'HR1B-0007 correttamente iniziate con una vite in ogni angolo posizionate diagonalmente una dall'altra. A questo punto inserite le viti rimanenti.





Bloccate la parte frontale della testa sullo squadretta con due viti PH/T 2X8mm. Non stringete eccessivamente.



fissate la parte posteriore della testa alla parte frontale della testa con due viti PH/T 2X5mm.



La forma del robot



8. Fissare il rivestimento frontale del corpo.

Il rivestimento frontale protegge il corpo



la parte frontale del corpo HR1C-0001 è attaccata al corpo con due viti PH/T 2X4mm. Questa figura mostra la posizione dei fori delle viti nel rivestimento.



Utilizzando un lungo e sottile cacciavite Phillips, avvitate le due viti PH/T 2X4mm attraverso il retro del robot sul rivestimento frontale del corpo.



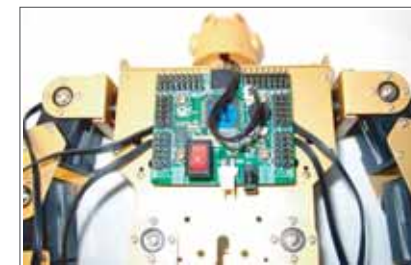
Assemblaggio completato del rivestimento frontale del corpo.



9. Fissare il regolatore al robot



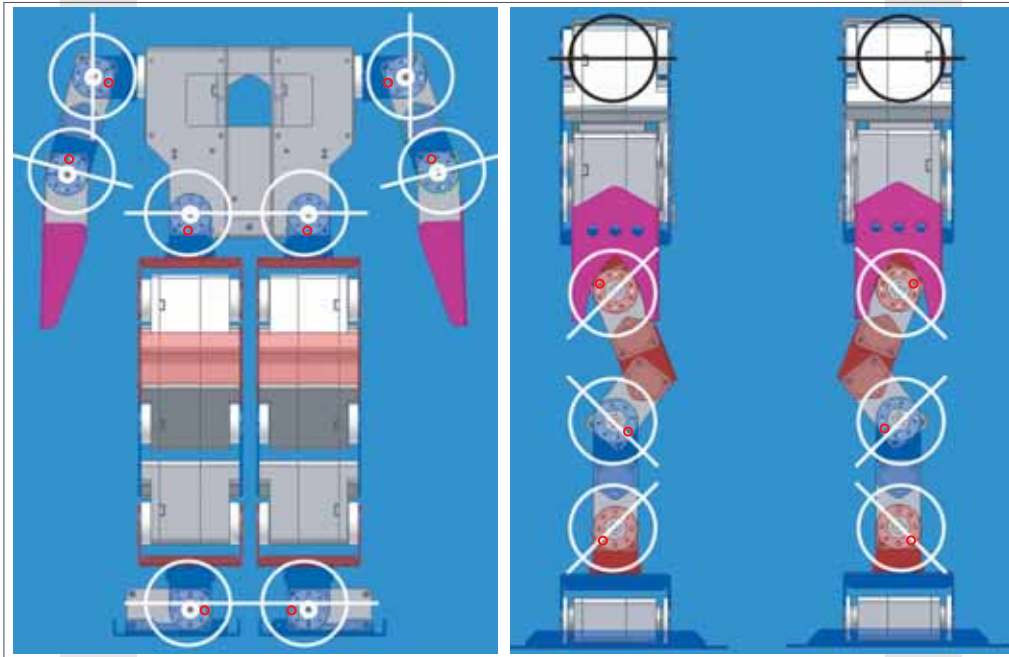
Fissate l'MR-C3024 al telaio posteriore del corpo con quattro bulloni PH/M 3X4mm.



## Controllo della Gamma di Movimento

\*Come in figura, i servi hanno 180 gradi di movimento.

\*Muovete ogni giunto del robot con la mano per controllare il movimento corretto dei servi.



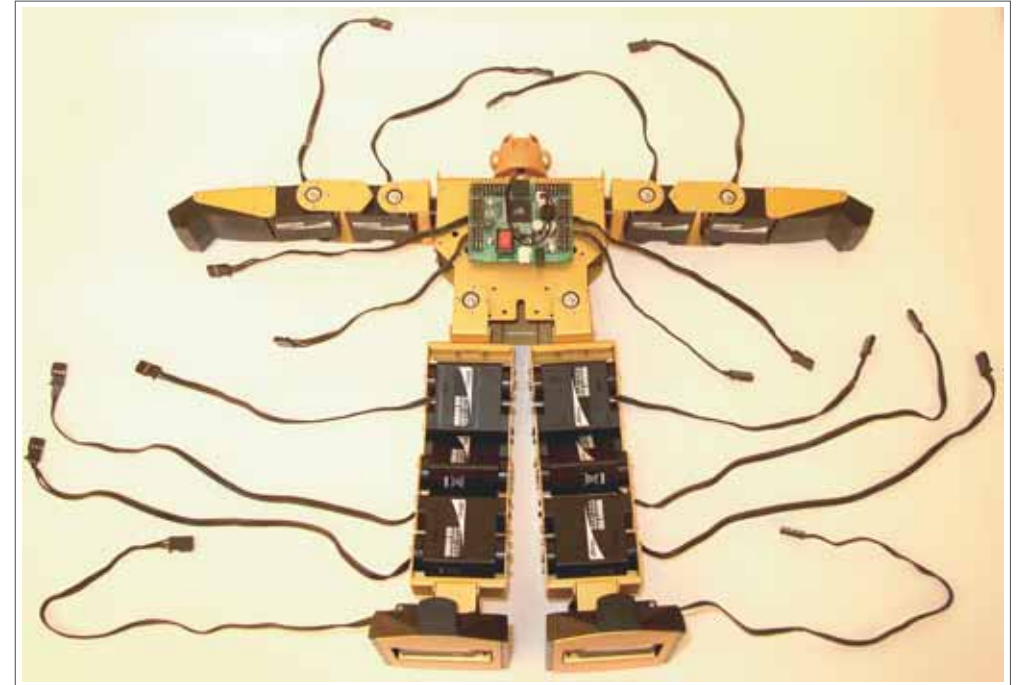
## 10. Disposizione del servo-cavo

Robonova-1 ha 16 cavi che si connettono col regolatore.

È importante per il funzionamento del robot che questi cavi siano fissati ordinatamente al corpo. Connessione del cavo alla Porta MR-C3024 e posizioni della morsa del cavo e della fascetta ferma-cavo.

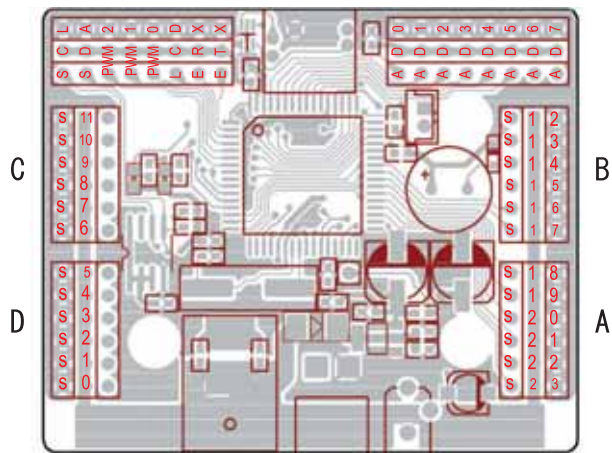
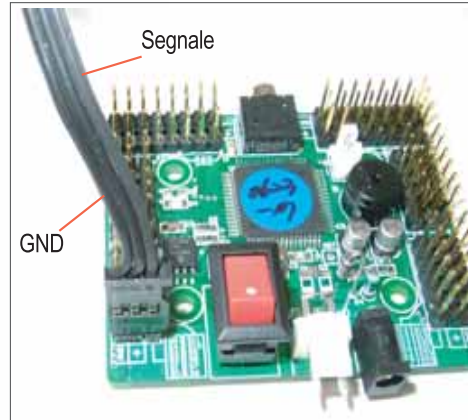
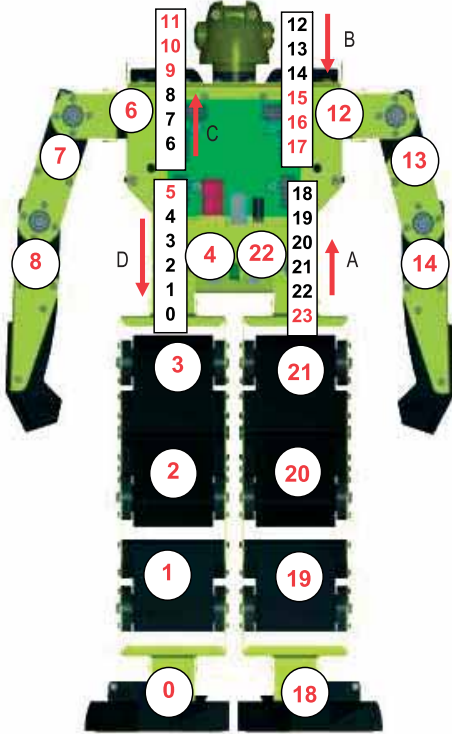
### (1) Disposizione del Cavo con fascetta ferma-cavo

Disponete i cavi in preparazione per la loro installazione sul regolatore



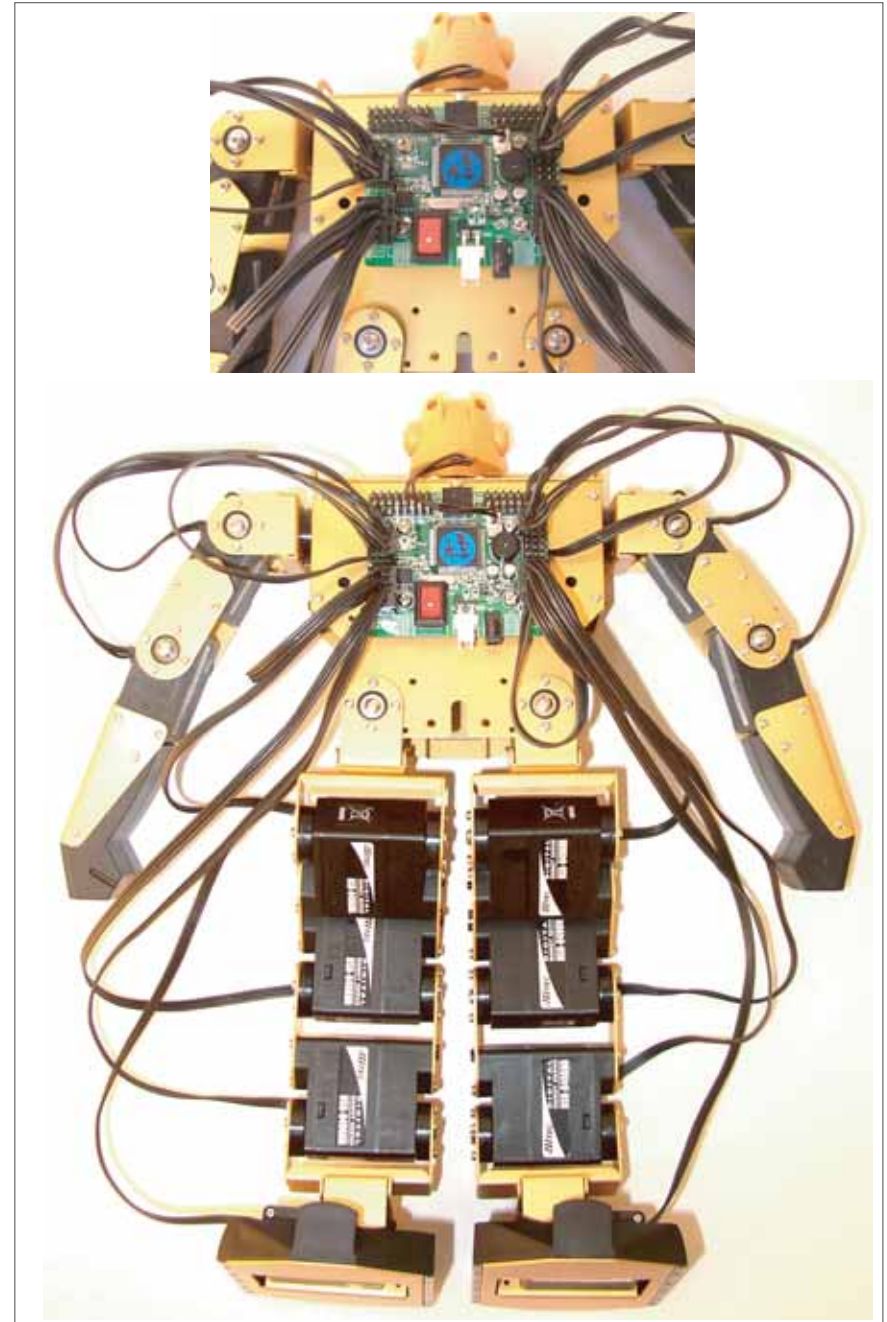


Vedere il diagramma del servo per un corretto piazzamento dei connettori del servo sul regolatore.  
 Notare che il cavo grigio è il cavo del segnale.



Questo diagramma mostra il pinout per il pannello del regolatore MR-C3024.

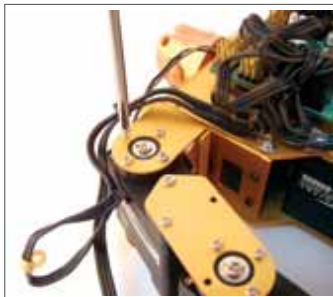
Collegate i servi sulle rispettive spine del pannello del regolatore.  
 Rifarsi al diagramma per un piazzamento corretto delle spine



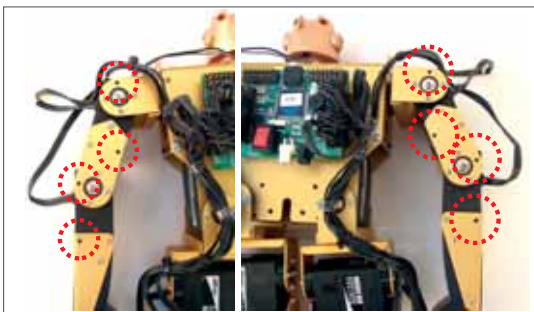
## 2) Installare le morse del filo elettrico

Installare le morse del filo eviterà che i cavi interferiscano con i movimenti del robot. Vedere la sezione uno per informazioni sull'attacco delle morse.

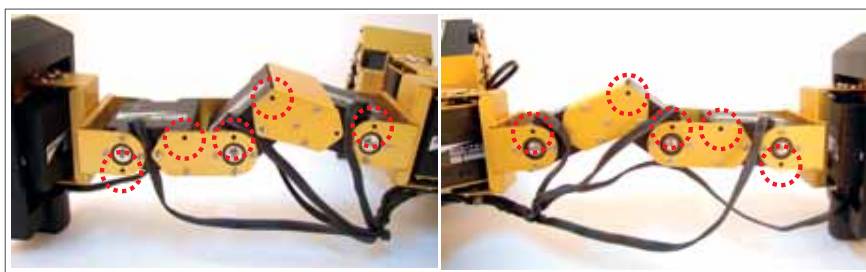
Usando la figura come riferimento, rimuovete le viti per installare le morse del filo.



l'immagine mostra le posizioni delle morse del filo sulle braccia.



quest'immagine mostra le posizioni delle morse del filo sulle gambe.



parti richieste per fissare una morsa ad una squadretta.



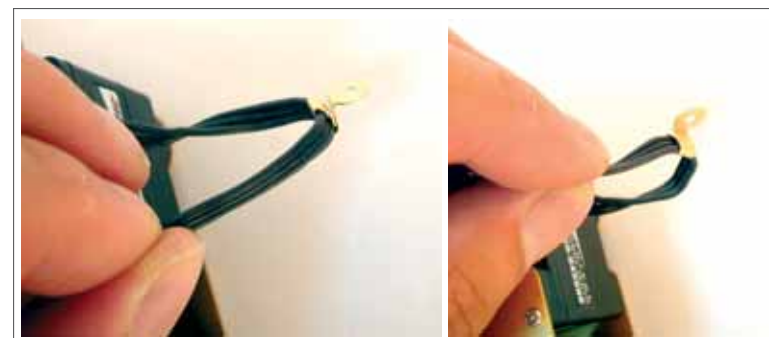
Installate una morsa del filo elettrico sul corpo nelle posizioni mostrate nelle figure.



per i servi nelle braccia n.2 e n.4, usate un'aletta (parte di colore oro chiaro) e una morsa del filo elettrico.

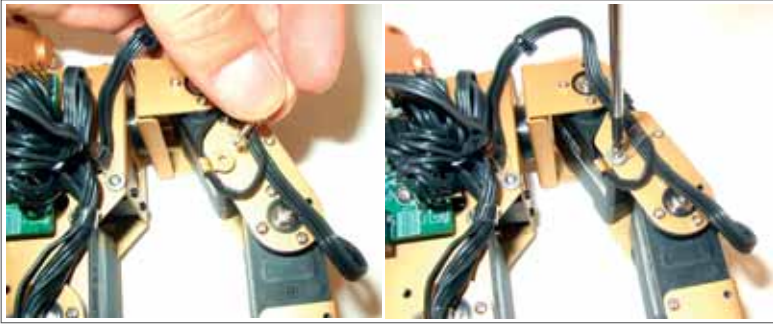


Prima piegate l'aletta attorno al cavo.





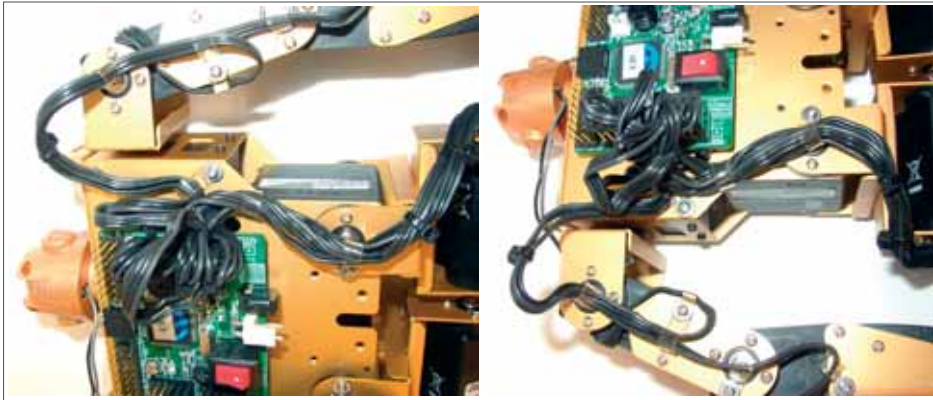
Usate l'immagine come esempio per collegare l'aletta e la morsa del filo.



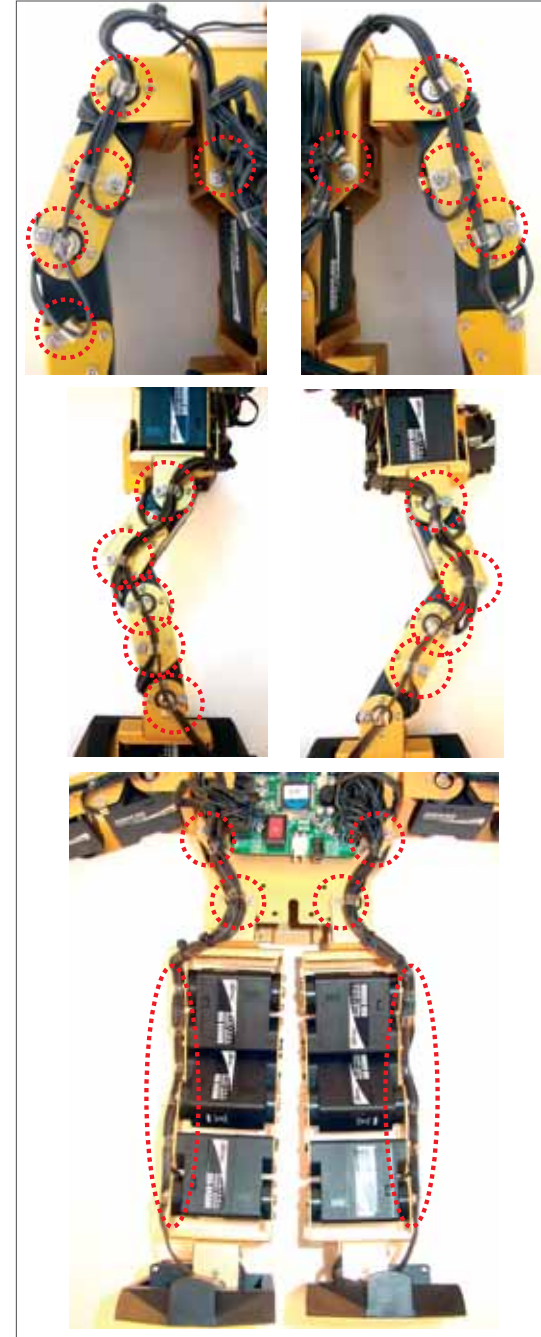
Assemblaggio completato della morsa e dell'aletta.



Attaccate le morse ad ognuno degli squadretti dell'addome come mostrato in figura.



Assemblaggio delle morse completo.

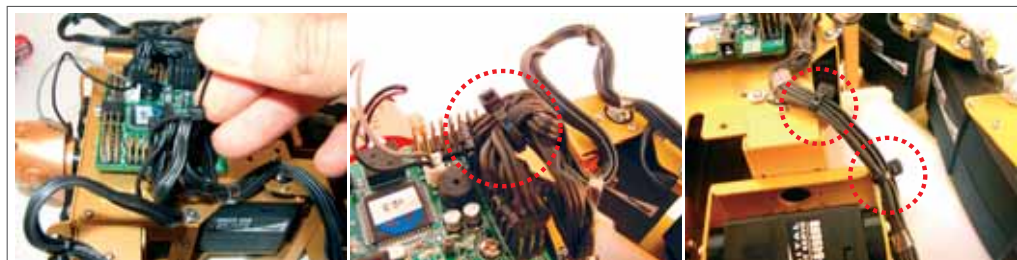




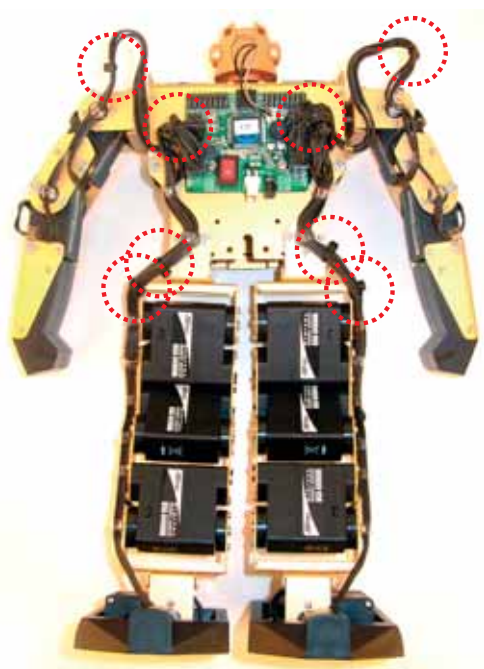
Innanzitutto, estraete i cavi che vengono fuori dal braccio e legateli con una fascetta ferma-cavo. Allentate i bulloni del regolatore MR-C3024 e spingete dentro 3-4 dei cavi più lunghi sotto il fianco dell'MR-C3024 poi stringete nuovamente i bulloni.



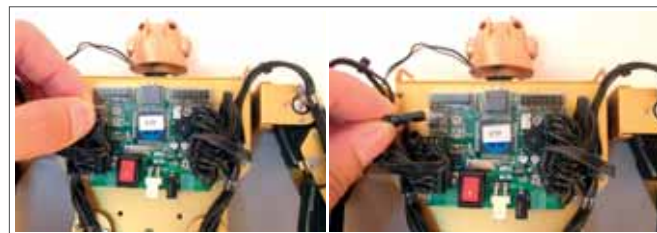
Sistamate i restanti cavi attorno al regolatore come in figura. Legate i cavi che connettono gambe e corpo con una fascetta ferma-cavo per ridurre l'attrito col supporto.



Assemblaggio delle fascette ferma-cavo completato.



(3) fissare il rivestimento spine (PIN) e LED. Disconnettere il servo superiore sinistro dal regolatore e inserire il rivestimento trasparente delle spine sul regolatore. Riconnettere il servo.



Collegate il connettore del LED nel regolatore come mostrato in figura.



11. Assemblaggio finale dell'hardware

Ponete il rivestimento dorsale del corpo sopra il regolatore MR-C3024. Il rivestimento s'incasta con il rivestimento frontale. Assicurandovi che non ci siano cavi impigliati, inserite due viti PH/T 2X26mm nella parte alta del rivestimento e due viti PH/M 2.6X4mm nella parte bassa.



Assemblaggio del rivestimento dorsale completo.



## (2) Installazione della Batteria

Preparare la batteria Ni-Mh 6V 1000mAh, due viti a testa zigrinata 3X4mm e il telaio posteriore del corpo HR1B-0008 e la protezione del cavo della batteria.



Prima di tutto installate la protezione del cavo-batteria sul telaio posteriore del corpo come in figura. Inserite la batteria nella parte bassa del robot. Prestate attenzione all'orientazione del pacchetto.



Fissate il rivestimento della batteria e bloccatelo con le due viti a testa zigrinata 3X4.



Connettete il connettore della batteria nel regolatore.



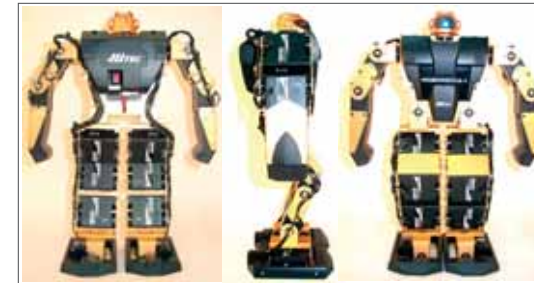
Per caricare la batteria, inserite il connettore della carica nel regolatore e inserite la spina del caricatore in una presa del muro.



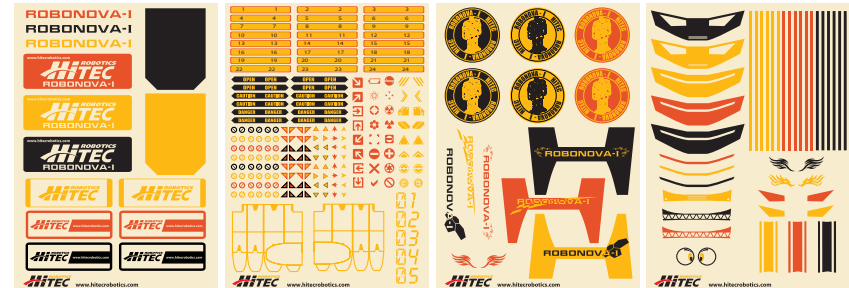
Il caricatore ha un LED rosso per indicare che sta caricando. Quando ha finito la luce del LED è verde.

## (3) Conclusione

Una volta completato RoboNova-1 dovrebbe essere come l'esempio in figura.



Il kit Robonova-1 è fornito di 4 diversi set di adesivi per personalizzare il vostro robot.



Sono inclusi un numero di adesivi per una rapida designazione dei cavi.





12. Installare il Remocon (telecomando) e il Sensore IR.

\*Il Remocon e il Sensore IR non sono inseriti in tutte le versioni di Kit.

ROBONOVA-1 può essere controllato con gli esclusivi Remocon e Sensore IR. Per installare il Sistema Remocon, sono richiesti il Sensore IR e del nastro bi-adesivo.



Aprire il rivestimento posteriore del corpo di RN-1



Inserite il connettore del Sensore IR nella porta "AD7" situata sul lato destro in alto del regolatore MR-C3024



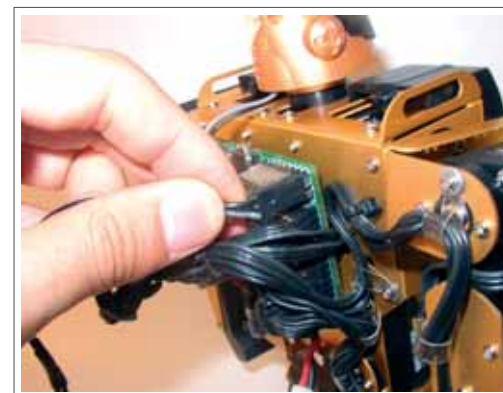
Prima di inserire il connettore, rimuovete il rivestimento delle spine dall'MR-C3024 e tagliate via parte del rivestimento spine.



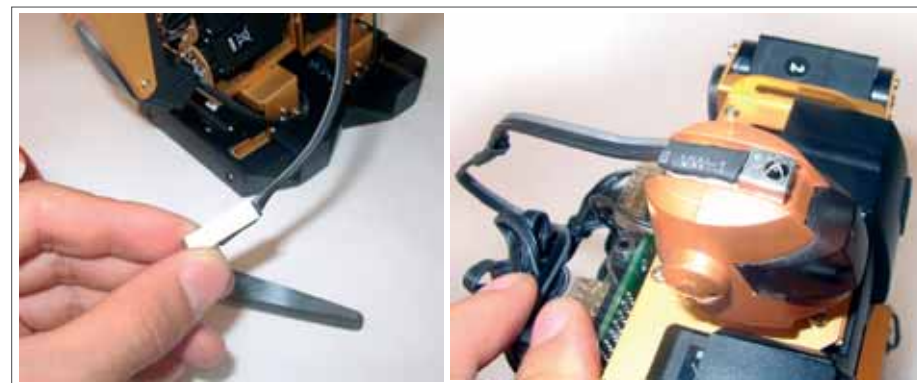
Riattaccate il rivestimento spine.



Connettete il Sensore IR all'MR-C3024. Il cavogrigio scuro del connettore deve stare in basso.



Attaccate un po' di nastro bi-adesivo 15X8mm sul retro del Sensore IR. Attaccate il Sensore IR in cima a RN-1.







Riattaccate il rivestimento posteriore del corpo a RN-1

Il prossimo passo è la messa punto della programmazione del Remocon. Aprite roboBASIC e create anche un nuovo file o aprite "action-auto.bas" che trovate nella cartella di template del CD.  
 Nel programma di modello (template) trovate 'A = REMOCON(0) e cambiatelo in A = REMOCON(1)

```

MAIN1 :
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN, K1, K2, K3, K4, K5, K6, K7, K8, K9, K10, K11, K12, K13, K14, K15, K16, K17, K18, K19, K20
GOTO MAIN
    
```

Connettete ROBONOVA-1 al PC con il cavo seriale d'interfaccia.



Cliccate il pulsante "Run All" (esecuzione integrata) per caricare sul regolatore.

```

01 | ' action_01
02 | '== Bow motion =====
03 |
04 | DIM A AS BYTE
05 |
06 | PTP SETON
07 | PTP ALLON
08 |
09 | '== motor direction setting =====
10 | DIR G6A,1,0,0,1,0,0
11 | DIR G6B,1,1,1,1,1,1
12 | DIR G6C,0,0,0,0,0,0
13 | DIR G6D,0,1,1,0,1,0
14 |
15 | '== motor start position read =====
16 | GETMOTORSET G6A,1,1,1,1,1,0
17 | GETMOTORSET G6B,1,1,1,0,0,0
18 | GETMOTORSET G6C,1,1,1,0,0,0
19 | GETMOTORSET G6D,1,1,1,1,1,0
    
```

Il Remocon può essere identificato con un ID tra 1 e 4. Questo permette fino a quattro ROBONOVA-1 di essere controllati da differenti remocon in contemporanea senza interferenza.  
 Accendete ROBONOVA-1 e con il Remocon puntato verso il Sensore IR, schiacciate il pulsante "P1" situato sul lato sinistro in alto del Remocon. Poi premete il numero di ID desiderato (scegliete tra i pulsanti "1-4"). Tenete premuti entrambi i pulsanti per 2 secondi.  
 Quando rimpiazzate le batterie, il numero di ID dovrebbe resettarsi.



## IV. Preparativi prima del funzionamento di ROBONOVA-1

\*RoboBasic deve essere installato prima del funzionamento di ROBONOVA-1

'Vedi "(2) Installazione di RoboBasic" nella sezione "1. Installazione ed esecuzione di RoboBasic".

1. Messa a punto dei movimenti di base e applicazioni.

(1) Messa a punto iniziale di RoboBasic per ROBONOVA-1

E' importante mettere a punto il programma di RoboBasic per un funzionamento corretto.

1) Impostazioni della Porta per il regolatore e il PC. Selezione del regolatore ' Vedi "(3) Esecuzione di RoboBasic" nella sezione "1.

Installazione ed esecuzione di RoboBasic". Il regolatore sarà automaticamente impostato per ROBONOVA-1.

2) Spiegazione delle impostazioni iniziali (zero settings) e procedure ' Vedi "(5) Impostazioni iniziali di RoboBasic" nella sezione "1.

Installazione ed esecuzione di RoboBasic".

(2) La lista di comandi trovabile nel programma di modello (template).

1) Comando "goto AUTO" (manuale di comando di RoboBasic pag.44 nel CD)

## goto AUTO

Va al programma di modello.

### Struttura della frase

- Frase in Inglese : goto AUTO

### Spiegazione del Comando

Comando per iniziare il programma di modello incluso.

### Esempio di Comando

Goto AUTO 'va al programma di modello.

Nota: Questo comando vale solo per il regolatore MR-C3024 e il Robot Robonova-1.

```

0009 GOTO AUTO
0010 FILL 255,10000
0011
0012
0013
0014 DIM RR AS BYTE
0015 DIM A AS BYTE
0016 DIM A16 AS BYTE
0017 DIM A26 AS BYTE
0018
0019 CONST ID = 0 ' 1:0, 2:32, 3:64, 4:96,
0020
0021 '** Action command check (50 - 82)
0022 IF RR > 50 AND RR < 83 THEN GOTO action_proc
0023
0024 RR = 0
0025
0026 PTP SETON
0027 PTP ALLON
0028
0029 '** motor direction setting *****
0030 DIR G6A 1.0.0.1.0.0
0031 DIR G6B 1.1.1.1.1.1
0032 DIR G6C 0.0.0.0.0.0
0033 DIR G6D 0.1.1.0.1.0
0034
0035
    
```

(il comando "GOTO AUTO" si trova in alto nel programma di modello)

Aprire il file "Overall Template rogram.bas" nella cartella "Template Program for roboBASIC" del CD.  
-> Vedi 1. Programma di modello nella sezione VI. Applicazioni del programma

Cliccate su "RUN ALL" (esecuzione integrata) per caricare il file sul regolatore.

Dopo il processo di caricamento. L'altoparlante a Piezo emetterà un suono e ROBONOVA si metterà in una posizione eretta.

- se Robonova non resta eretto correttamente sono richieste modifiche alle impostazioni iniziali. ' Vedi (5) impostazione iniziale di RoboBASIC nella sezione V. Come usare il software per ROBONOVA-1.

- se c'è una grande discrepanza tra la posizione eretta di ROBONOVA e la figura sottostante, assicuratevi che i servi, gli squadretti, e i supporti siano assemblati correttamente.

- In seguito ad un doppio controllo dell'assemblaggio, accendete l'interruttore. Se la posizione effettuata è la stessa delle immagini

(Vedi (5) Impostazioni iniziali di RoboBasic), le 29 posizioni base contenute nel programma di modello possono essere effettuate.

(Vedi la Tabella d'Assegnazione Chiavi di REMOCON). Controllate l'impostazione del programma di RoboRemocon o IR

REMOCON (Vedi (4) Usare ROBOREMOCON con RoboBasic nella sezione 1. Installazione di RoboBasic).

2) Tabella per l'impostazione e le azioni di remocon.

- i movimenti che sono installati nell'MR-C3024 possono essere effettuati con il RoboRemocon o con un IR REMOCON (non disponibile in alcune versioni del kit).

(Vedi la sezione "11. Installazione dell'IR REMOCON" per l'installazione e le impostazioni dell'IR REMOCON)

(Vedi la sezione "3. Installazione ed impostazioni di RoboRemocon" per l'utilizzo del programma di RoboRemocon)

Tabella d'Assegnazione Chiavi di REMOCON

ACTION	key	Motion	Variable	Code
0	power	ON : motor on-)Basic Position OFF: Sitting Position-) motor off	A16	16
1	1	Bow -) Basic Position		1
2	2	Raise arms -) Basic Position		2
3	3	Sit -) Basic Position		3
4	4	Sit -) Raise arms -) Basic Position		4
5	5	Raise a leg -) Basic Position		5
6	6	Spread the legs -) Extend arms -) Right-left tilt -) Basic Position		6
7	7	Flap arms like a bird		7
8	8	Kick		8
9	9	Handstand		9
10	0	Walk fast		10
11	*	Left turnabout		22
12	#	Right turnabout		24
13	▲	Forward		11
14	◀	Left move		14
15	■	Sit(-)Stand up	A26	26
16	▶	Right move		13
17	▼	Reverse		12
18	△	Front tumbling		21
19	◁	Left cartwheel		28
20	◻	Front attack		29
21	▷	Right cartwheel		30
22	▽	Rear tumbling		31
23	A	Left attack		15
24	B	Right attack		20
25	C	Left front jab		17
26	D	Right front jab		27
	E,F,G	27,28,29 Spare 18,32,23		18

## V. Manuale del software per ROBONOVA-1

### 1. Installazione e Funzionamento di roboBASIC v2.5

#### Marchio Registrato

Windows è un marchio registrato della Microsoft Corporation

#### Avviso

Questo manuale serve per l'installazione e il funzionamento di roboBASIC. Questa versione potrebbe essere diversa dalle precedenti versioni e potrebbe, lungo il manuale, essere cambiata senza notifica in maniera di migliorare la prestazione.

RoboBasic è un programma registrato, perciò è illegale riprodurre, pubblicare, spedire, trasmettere o distribuire questo manuale o programma senza permesso.

HITEC Robotics <http://www.hitecrobotics.com>

#### (1) Informazioni su roboBASIC

RoboBASIC è basato sul linguaggio di programmazione BASIC e specificamente sviluppato per comunicare con la famiglia di pannelli di controllo robot MR-C.

RoboBASIC è un linguaggio educativo che si aggiunge al linguaggio di programmazione BASIC la possibilità di far funzionare dei robots.

RoboBASIC è compatibile con MS Window 98, ME, 2000 e XP

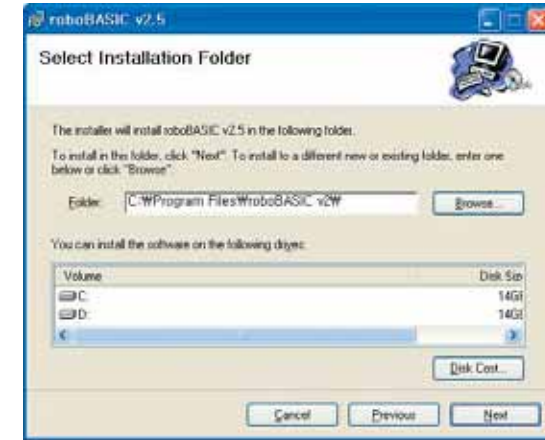
#### (2) Installazione di roboBASIC

- il software roboBASIC può essere installato dal dischetto o CD incluso col robot RN-1, o scaricato dalla homepage della HITEC Robotics (<http://www.hitecrobotics.com>)

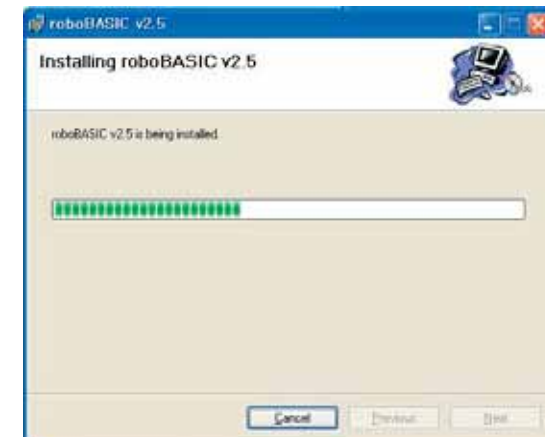
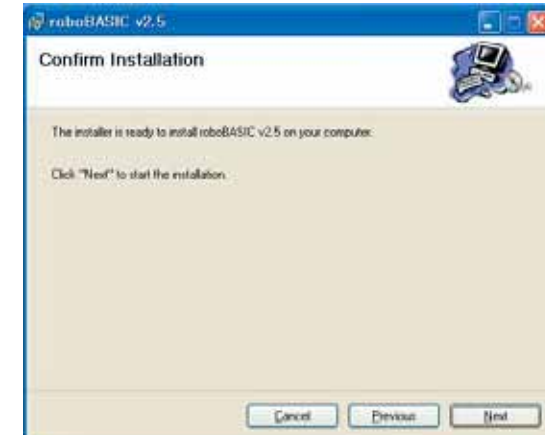
\*al momento dell'installazione di ROBObASIC, i programmi ROBOSCRIPt e ROBOREMOCON vengono installati nello stesso momento.



Clickate su "SETUP.EXE" per avviare il programma d'installazione

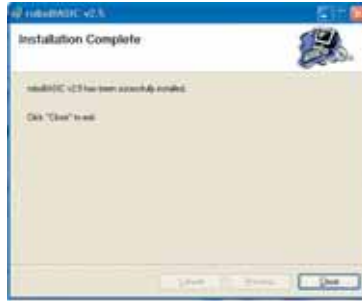


Selezionate la cartella d'installazione



L'installazione di ROBObASIC incomincia



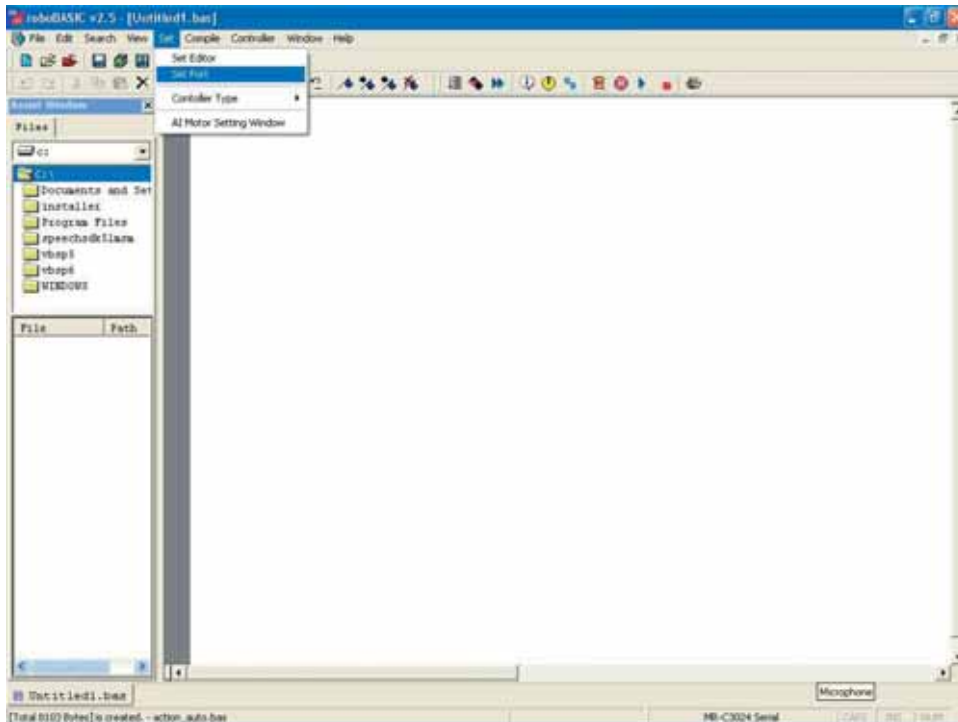


Installazione completata

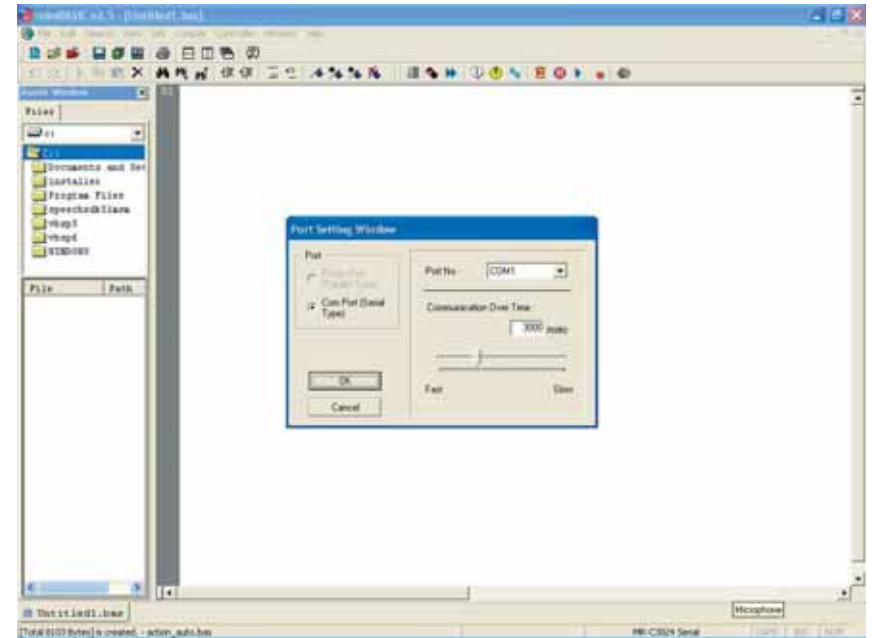
### (3) Impostazione iniziale di roboBASIC

Connetete il cavo di download e accensione al regolatore e accendete l'unità.  
 Aprite roboBASIC


Selezionare il tipo di regolatore.  
 Selezionate il regolatore MR-C3000. Se avete l'MR-C3024, selezionate il regolatore MR-C3024.

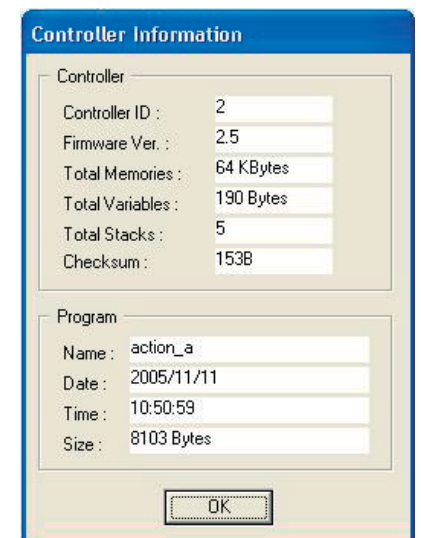


Selezionare il numero della porta seriale. Selezionare la porta seriale che verrà utilizzata.  
 Per confermare quale porta seriale verrà usata, aprite la gestione periferiche che trovate nel sistema operativo di Windows. [pannello di controllo-> sistema -> gestione periferiche -> porta (COM e LPT)]

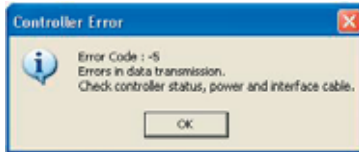


quando si utilizza roboBASIC per la prima volta, deve essere effettuata un'impostazione iniziale per il regolatore e il sistema. Una volta che l'impostazione è completa, questa operazione non viene richiesta successivamente a meno di cambi fatti al sistema. Per confermare che il regolatore sta comunicando con il PC, aprite la finestra informativa del regolatore.

Clickate  [Controller(C)-> Controller Information o l'icona .  
 Se la comunicazione è stabilita, le informazioni sul regolatore sono mostrate nella finestra "informazioni regolatore".



Se avviene un fallimento di comunicazioni, apparirà un messaggio d'errore. Cliccate il pulsante OK. In questo momento non è visualizzabile nessuna informazione nella finestra "informazioni regolatore"

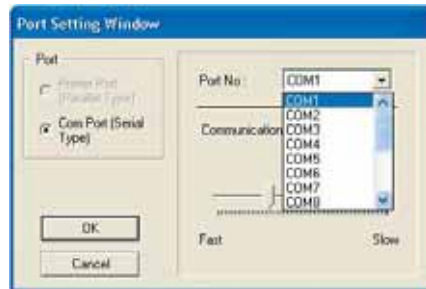


(Messaggio d'Errore del Regolatore)



(Nessuna informazione visualizzata nella finestra)

Controllate approfonditamente che il cavo di download sia attaccato in maniera sicura e che c'è corrente nel regolatore. Iniziate di nuovo la procedura d'impostazione.

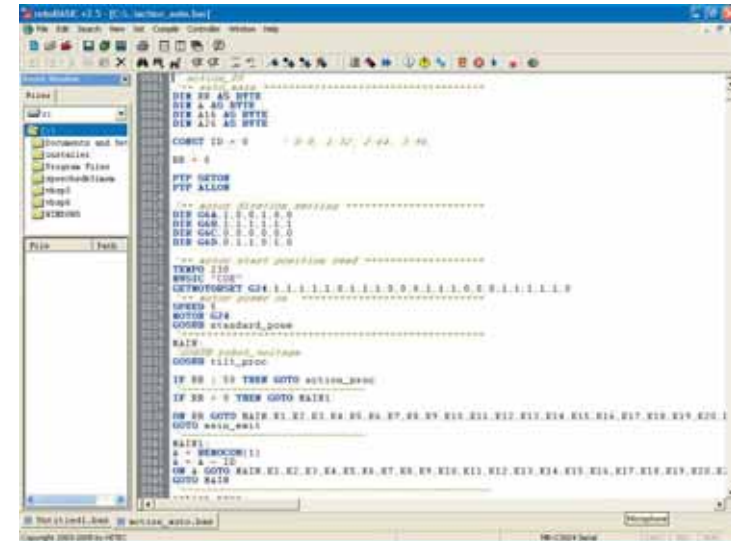


Cliccate [Set up (T) -> Port set up

La porta può essere cambiata a seconda della necessità.

#### (4) Programmare in RoboBASIC


Con l'installazione e l'impostazione completate, è tempo di iniziare la parte di programmazione. Create un nuovo programma o aprite un file esistente. Tutti i file sono salvati con l'estensione [\*.bas].

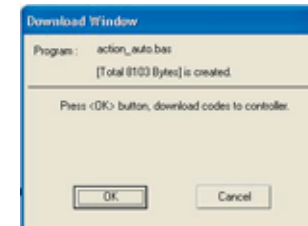



Per caricare un programma esistente o uno appena creato sul regolatore, bisogna convertirlo in un codice d'oggetto tramite il processo di compilazione. Solo a questo punto sarà possibile caricarlo sulla ROM del regolatore. I processi di compilazione e caricamento possono essere effettuati insieme o separatamente

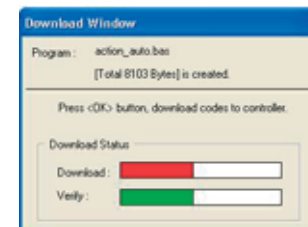
Cliccate su l'icona  "make object code" (crea codice oggetto), F2 o Make Object Code dal menu. [Compile (O) -> Make object code (F2)]



Cliccate su l'icona  "download", (F6) o selezionate Download dal menu. [File (O) -> Download (F6)]



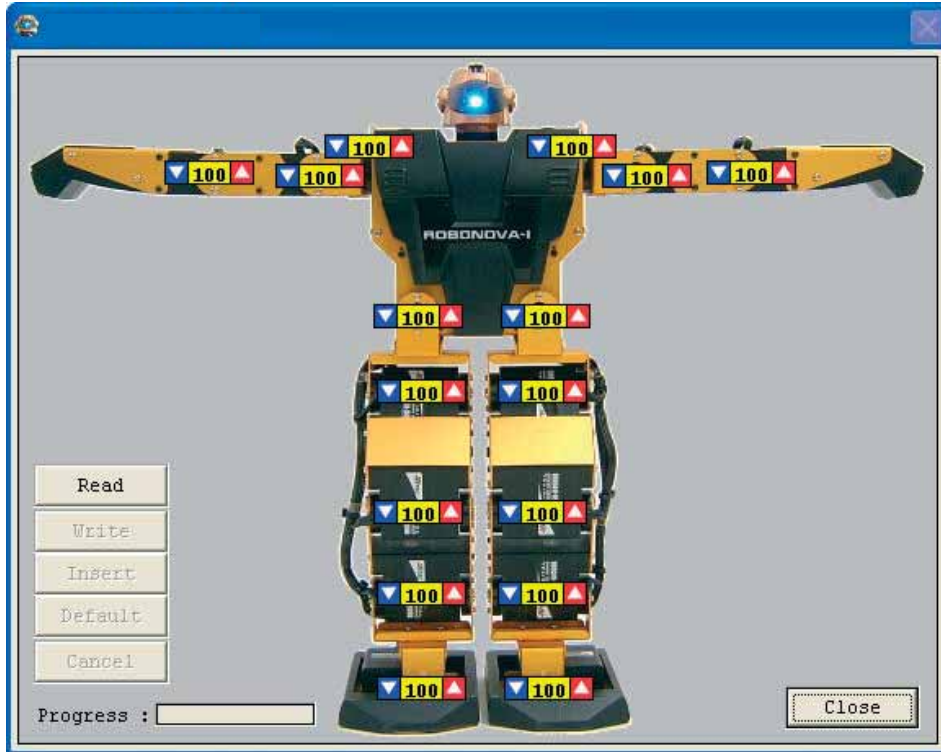
Cliccate su l'icona  di "integrated execution" (esecuzione integrata), (F9) o selezionate Run All dal menu. [File (O) -> Run All (F9)]



## (5) Impostazione iniziale (zero) di RoboBasic

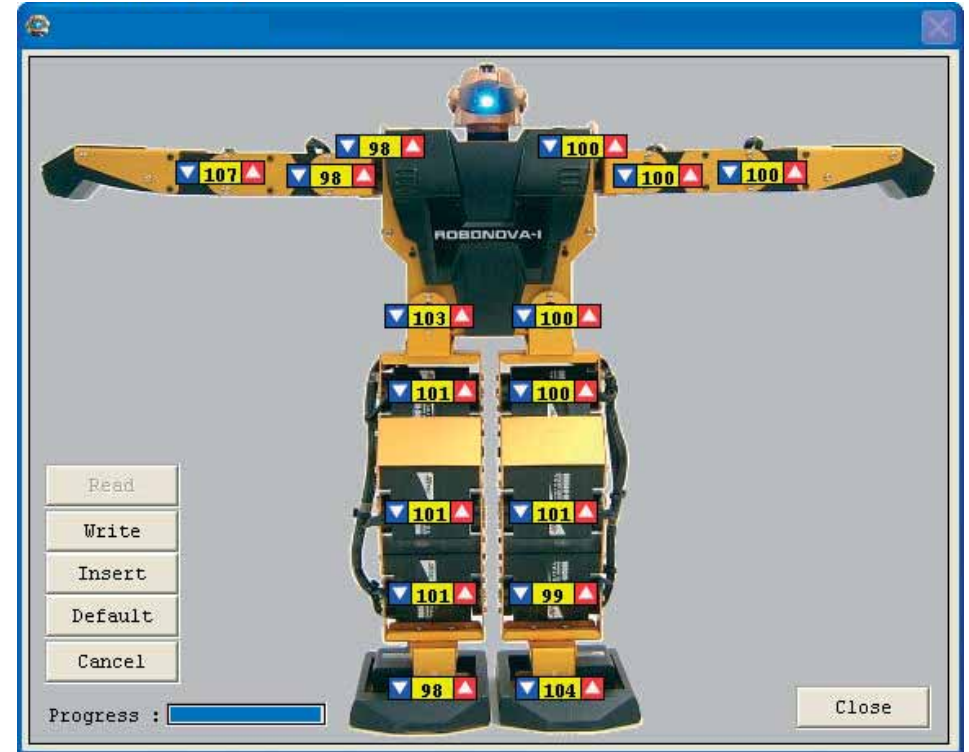
Anche se il robot fosse assemblato correttamente, i servi potrebbero essere lievemente fuori-centro. Un'impostazione iniziale è necessaria per fissare le posizioni neutrali dei servi.  
Le impostazioni iniziali sono controllate tramite il software RoboBASIC.

- per accedere alle impostazioni iniziali dal menu in robobasic, andare a "compile" Set Zero Point.



- una piccola finestra si aprirà con un'immagine di ROBONOVA. Di fianco ad ogni servo c'è la sua posizione neutrale iniziale. (Vedi foto RN-1). Per fare muovere il robot nella stessa posizione della foto, cliccare sul pulsante "Read".

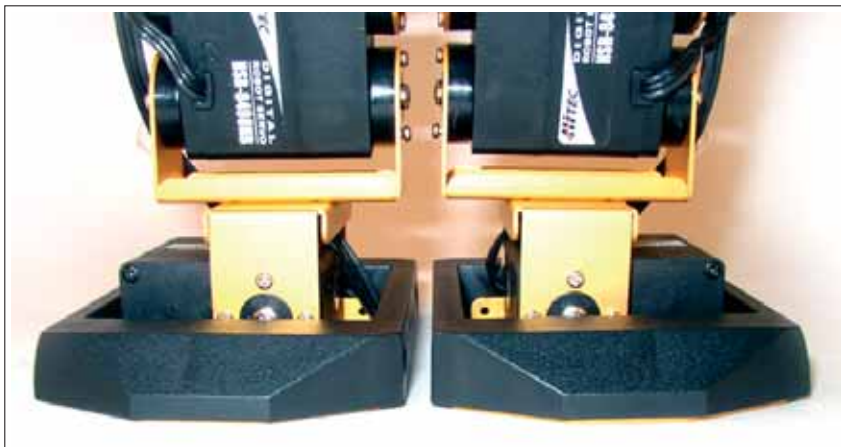
\* (Attenzione) Quando cliccate il pulsante "Read", le braccia e le gambe si muoveranno per raggiungere la posizione come nella foto sopra. Se il robot è tenuto in mano in quel momento c'è la possibilità di ferirsi. È suggerito che prima di cliccare su "Read" il robot sia posizionato su di una superficie piatta.



(le impostazioni iniziali di ogni servo possono essere modificate usando le frecce giù e su)



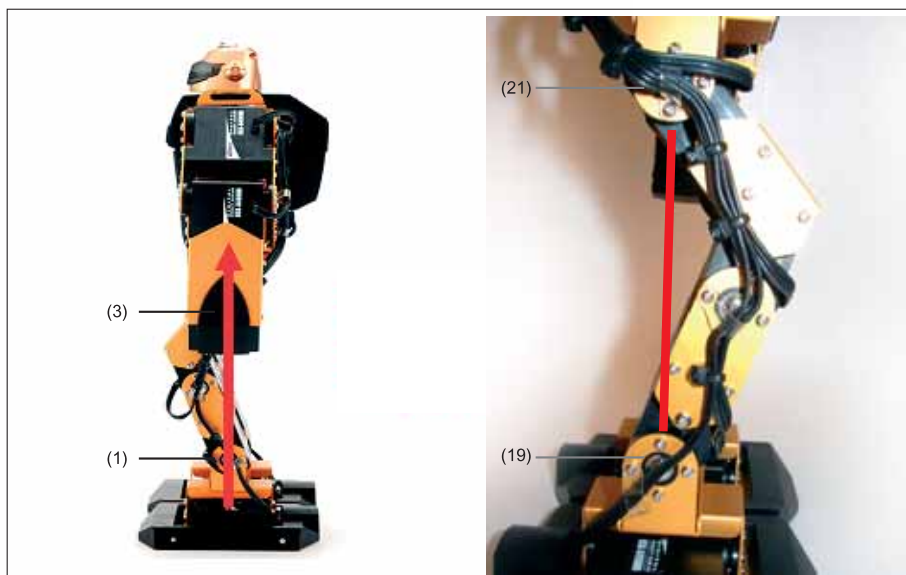
Posizioni iniziali corrette.



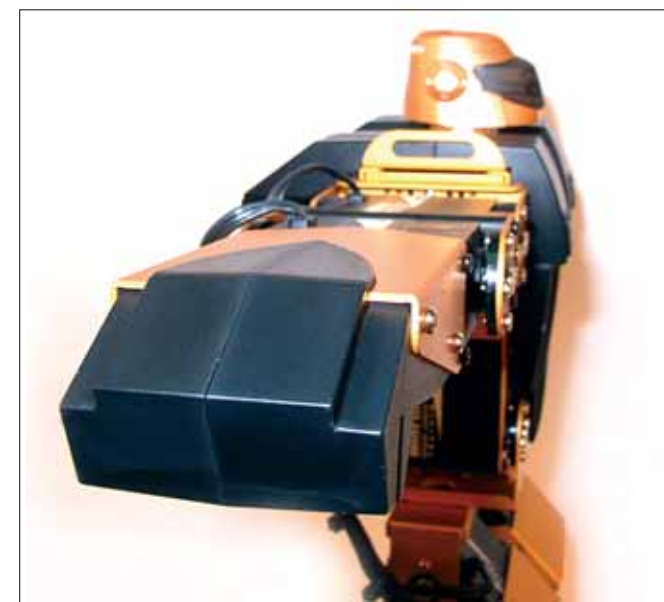
I due piedi devono essere piatti sulla superficie.



Lo spazio tra le gambe è di circa 8mm



I servi n.1 e n.3 della gamba sinistra (n.19 e n.21 della gamba destra) sono in linea retta.




Le braccia sono in una posizione orizzontale piatta come nella foto sopra.

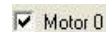
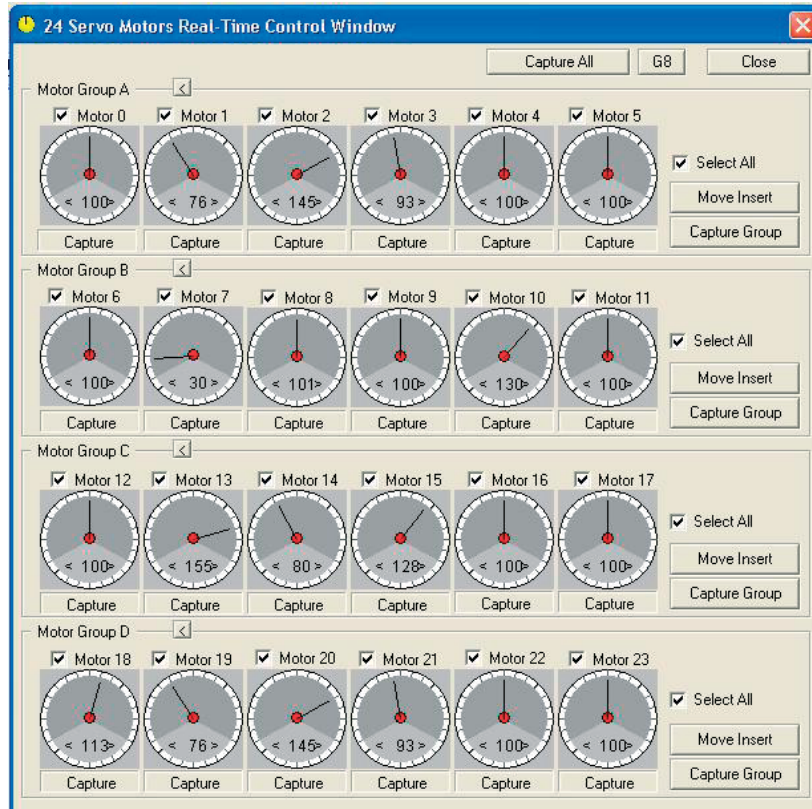
- le impostazioni iniziali possono essere modificate con le frecce giù e su.  
Per salvare le impostazioni in RoboBASIC, cliccate sul pulsante "Write". Questo inserirà le nuove impostazioni nel programma. Per caricare le impostazioni iniziali sul robot immediatamente, cliccate il pulsante "Insert". Questo inserisce le nuove impostazioni iniziali sul regolatore MR-C3024 bypassando il normale caricamento del software RoboBasic. Per vedere le nuove impostazioni, spegnete e riaccendete il robot.

## (6) Controllo in tempo reale dei servo motori

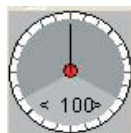
Con il controllo in tempo reale dei servo motori, potete controllare il movimento del robot in maniera facile e veloce.

Connettete l'MR-C3024 col PC e fate partire il programma ROBObASIC.

Cliccate l'icona  controllo in tempo reale dei servo motori o cliccate [Controller(C) ' Servo motor realtime control(F7)]



Servo motore da controllare



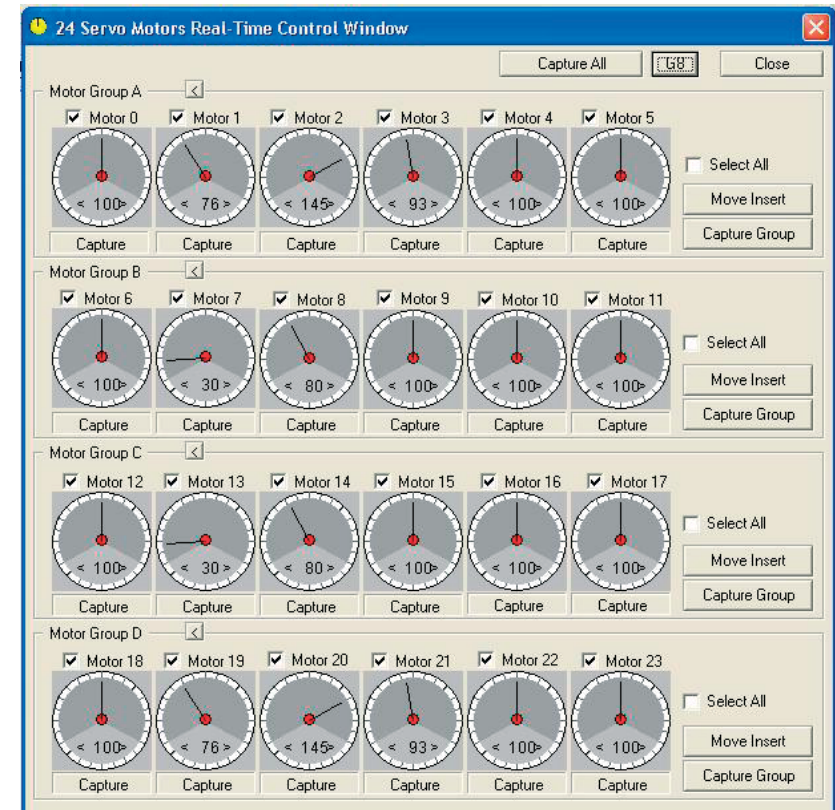
La posizione del servo motore, la gamma è 10-190



Apri e chiude il gruppo di servo motori

G8

24 finestre di manopola e nomina 6 unità sul gruppo dei servo motori.



Move Insert

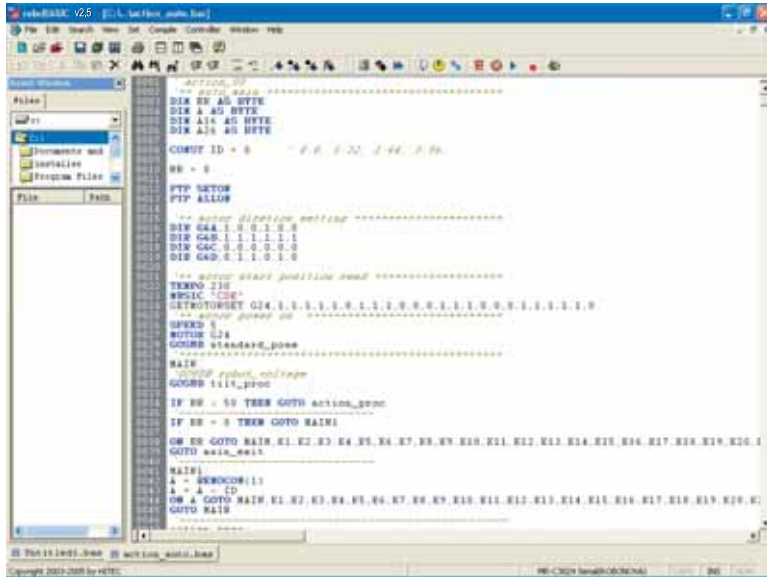
Inserisce il comando "MOVE" nell'afinestra del programma.  
Es) MOVE G8A, 100, 100, 100, 100, 100, 100, 100, 100

Capture Group

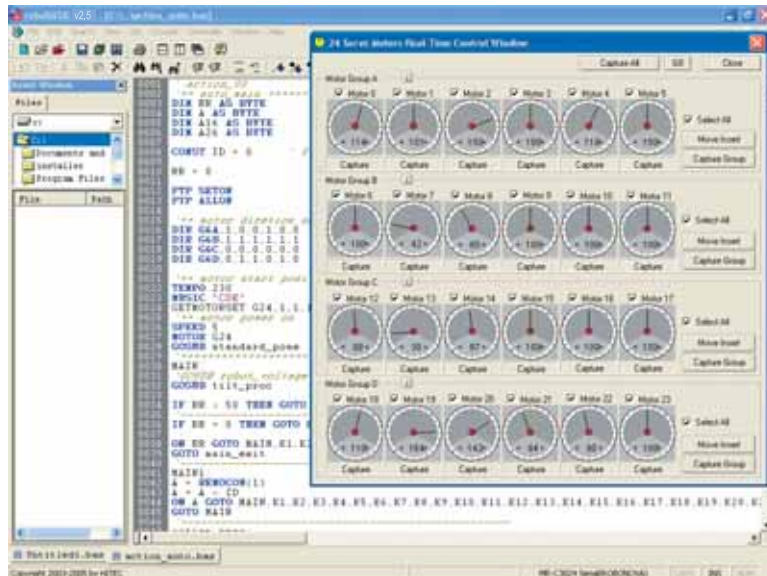
Chiude il controllo in tempo reale dei servo motori



(7) RoboBasic ROBONOVA-1 controllo del servo motore  
 1) Controllo in Tempo Reale del Servo Motore.



(Connettere ROBONOVA al Pc tramite un Cavo Seriale d'Interfaccia e accendete l'interruttore. Muovete il cursore all'interno del programma fino alla linea dove verranno inseriti i comandi di "MOVE".)



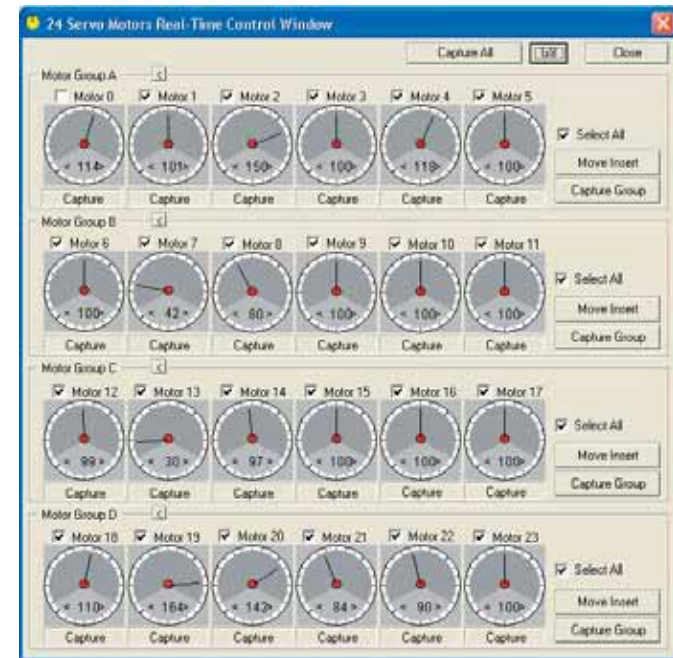
(Cliccare sul Menu ' controller ' controllo in tempo reale del servo motore. Il controllo del servo si aprirà.)  
 - Ci sono due maniere per usare il controllo in tempo reale del servo motore. Un metodo è quello di muovere i servi con la mano. Un altro metodo è cliccare le frecce ai lati dei numeri per incrementare o diminuire il valore.

Metodo di modifica a mano



- depennate il servo #0 nel gruppo A. Questo servo adesso si muoverà liberamente.
- muovete il servo nella posizione richiesta. Per salvare questa posizione, tenete il servo in posizione e cliccate la finestrella di scelta.
- per inserire questa posizione nel programma RoboBasic, cliccate il pulsante "MOVE INSERT" (inserisci movimento), se c'è alcuna finestrella di scelta libera, la posizione non sarà inserita nel programma, solo uno spazio bianco.

Metodo di modifica col mouse

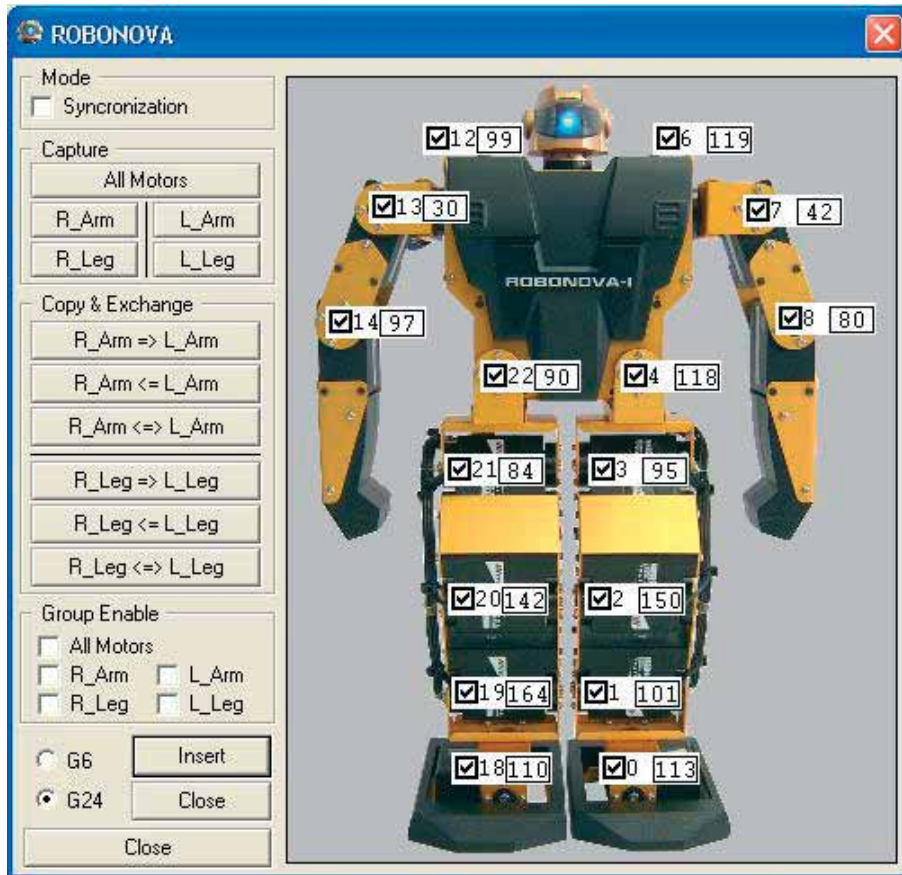


- assicurarsi che la finestrella di scelta sia selezionata e poi muovete anche il calibro per il servo richiesto a destra e sinistra. Come le modifiche sono cambiate, così lo specifico servo si muoverà. Una volta che assume la posizione desiderata, cliccare il pulsante "MOVE INSERT" per inserire il valore in RoboBasic.
- per selezionare tutte le finestre di un gruppo, cliccare il pulsante "Select All" (seleziona tutti).
- Per depennare tutte le finestre di un gruppo, cliccare il pulsante "Select All".



## 2) controllo del servo motore in ROBONOVA-1

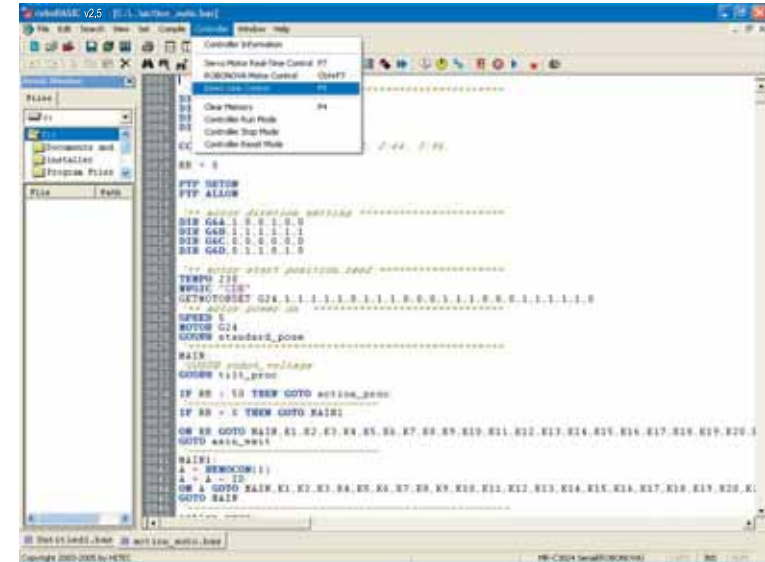
- i servi possono essere anche controllati con il controllo motore di ROBONOVA-1.  
Menu 'controller' controllo motore RoboNova



- finestra di controllo servo motore: selezionare finestra/numero del servo/valore di numero (angolo del servo)  
Ogni servo può essere controllato cliccando la finestra di controllo servo motore.

## (8) Controllo linea diretta

- questo è usato per fare andare una sola linea nel programma.



## 2. Funzionamento di RoboScript v2.5

Marchio Registrato  
Windows è un marchio registrato della Microsoft Corporation

### Avviso

Questo manuale e RoboScript possono essere cambiati senza notifica per migliorare il rendimento del prodotto. RoboScript è un software registrato, perciò è illegale riprodurre, pubblicare, spedire, trasmettere o distribuire questo manuale o software senza permesso.

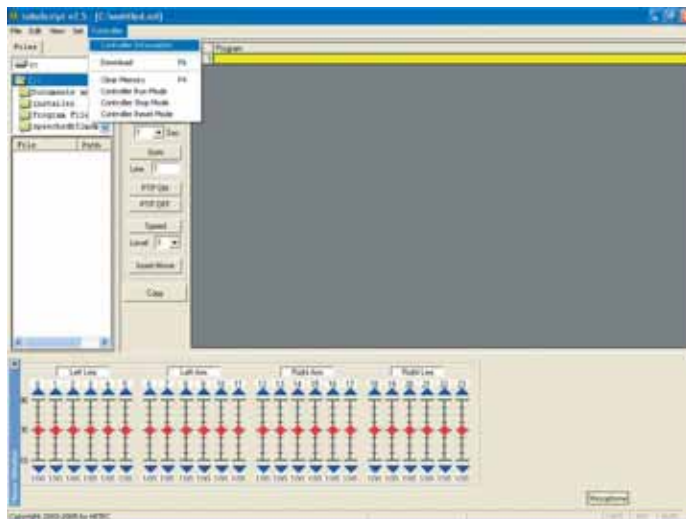
Hitec Robotics <http://www.hitecrobotics.com>

### (1) Informazioni su RoboScript

RoboScript è un ambiente di scrittura sviluppato per semplificare la modifica e la programmazione di RoboBASIC. RoboScript è compatibile con MS Windows 98, ME, 2000 e XP.

### (2) messa a punto iniziale di RoboScript

Connettere il cavo di download e corrente e accendere l'unità. Aprire RoboScript. La finestra di messa a punto iniziale apparirà. Per confermare se RoboScript sia installato correttamente, controllate le informazioni del regolatore. Cliccate [Controller(C) -> Informazioni Controller] o l'icona 



Se la comunicazione è stabilita, le informazioni sul regolatore sono visualizzate sulla finestra "informazioni del regolatore".



(l'informazione sul regolatore viene visualizzata)

Se c'è un fallimento di comunicazione, un messaggio d'errore apparirà. Cliccate il pulsante OK. Nessuna informazione verrà visualizzata nella finestra di "controllo regolatore" a questo punto.

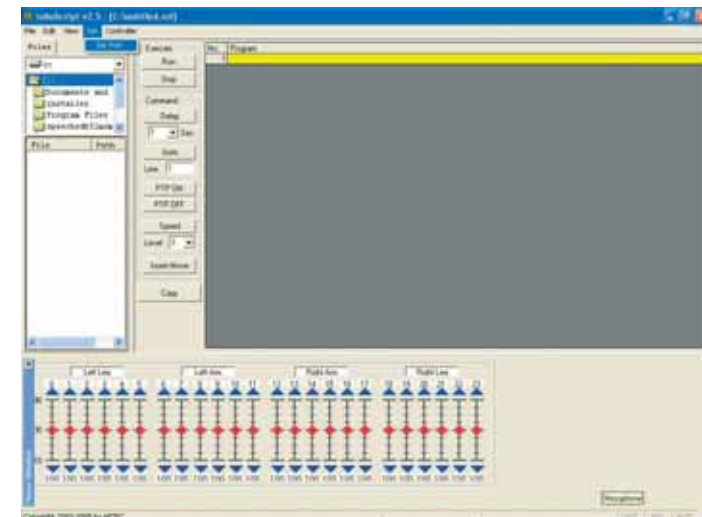


(messaggio d'errore del regolatore)

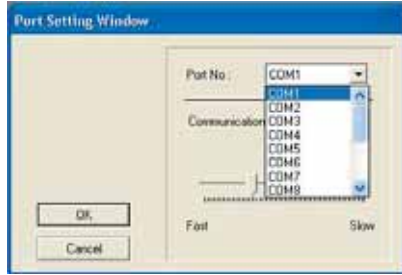


(non c'è informazione sul regolatore nella finestra)

Controllate attentamente che il cavo di download sia connesso in maniera sicura e che c'è corrente nel regolatore. Re-iniziate la procedura di messa a punto un'altra volta. Cliccare [Setup(T) -> messa a punto della porta]

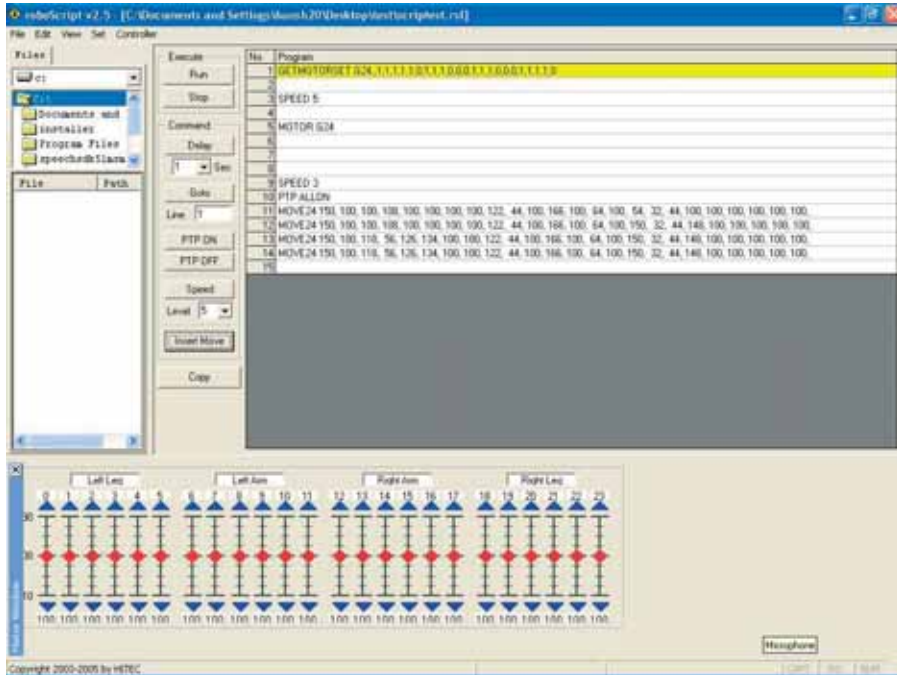


La porta può essere cambiata a seconda della necessità.

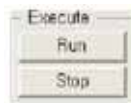


(3) programmazione di RoboScript

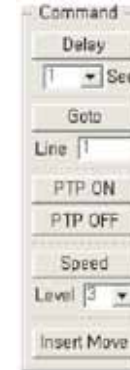
Con l'installazione e la messa a punto completate, è giunto il momento di iniziare la programmazione. Create un nuovo programma o aprite un file esistente. Tutti i file saranno salvati con l'estensione [\*.rsf].



In RoboScript è possibile testare il movimento del robot senza caricare un programma nel regolatore. Se richiesto può essere modificato e re-incollato in RoboBASIC.



Avvia/Ferma il programma creato o modificato.



Delay - imposta un tempo di ritardo tra 0,5sec e 30sec.

Goto - Inserisce una linea di comando goto nel programma.

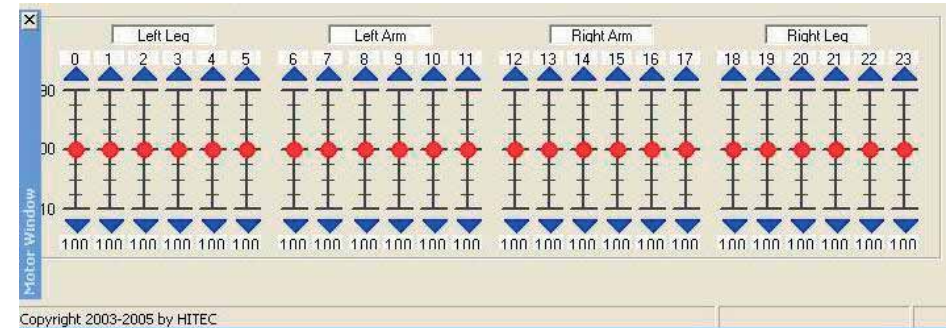
PTP - accende o spegne la funzione PTP.

Speed - imposta la velocità del servo da 1 a 15.

Insert Move - inserisce un comando "Move" nel programma

Copy

Copy - Copia il programma da voi creato con la chiave "copy" in modo di incollarla in RoboBASIC con [Ctrl+V] o [Paste]



Finestra di Controllo Servo - ogni servo può essere controllato in tempo reale muovendo la levetta corrispondente.

Dopo aver finito la posizione del servo, inseritela nel programma con la chiave "Insert Move".



### (3) Funzionamento di RoboREMOCON v2.5

Marchio Registrato

Windows è un marchio registrato della Microsoft Corporation

Avviso

RoboRemocon è un software registrato, perciò è illegale riprodurre, pubblicare, spedire, trasmettere o distribuire questo manuale o software senza permesso.

Hitec Robotics <http://www.hitecrobotics.com>

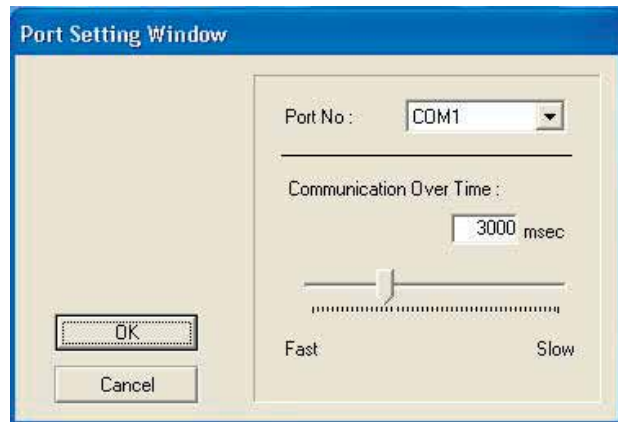
### (1) Informazioni su RoboRemocon

RoboRemocon permette il funzionamento wireless (senza cavo) del robot. Per il funzionamento di RoboRemocon è richiesta l'installazione di un programma del programma precedentemente creato RoboScript. Il telecomando non richiede la conoscenza dei concetti del software RoboRemocon.

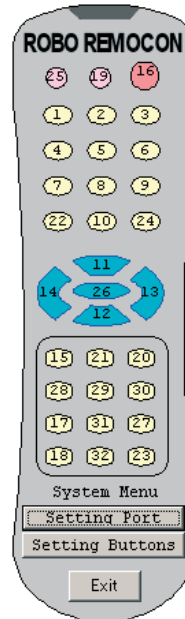
RoboRemocon è compatibile con MS Windows 98, ME, 2000 e XP

### (2) Programmazione e funzionamento di RoboRemocon

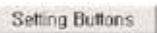
Per avere conferma dell'avvenuta corretta impostazione di RoboRemocon, controllare le informazioni del regolatore



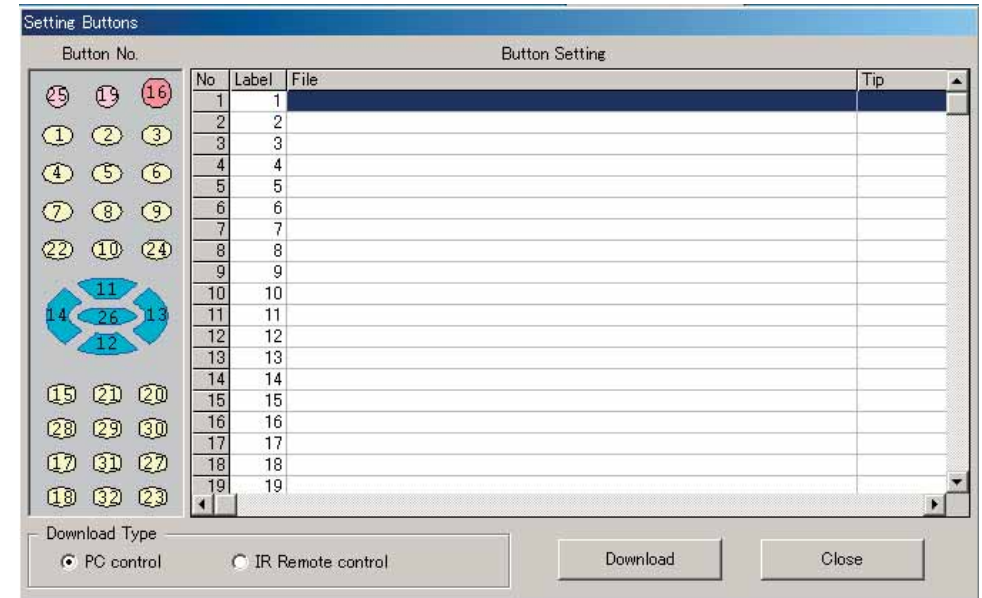
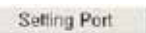
Imposta la porta di comunicazione



Cliccate il pulsante icona



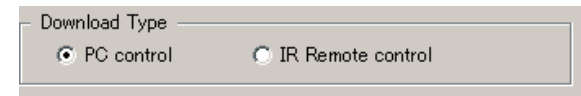
Cliccate icona della porta



Cliccare ogni numero di pulsante due volte sul remocon e impostare la funzione



Questa figura è un esempio di impostazione del pulsante #0. Imposta l'etichetta, il file di RoboScript e la targa del pulsante #0



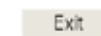
Nel menu Download Type (tipo di download). Cliccate "PC Control" se volete scegliere il programma del PC ROBOREMOCON.



Cliccate "IR Remote Control" se volete scegliere Handheld IR Remocon (telecomando IR a mano)



Scarica il programma sul regolatore.



Chiude il programma

(3) Come usare RoboREMOCON in RoboBASIC

- aprite il file "Overall Template Program.bas".

```

0001 ' action_00
0002 '== auto_main =====
0003 DIM RR AS BYTE
0004 DIM A AS BYTE
0005 DIM A16 AS BYTE
0006 DIM A26 AS BYTE
0007
0008 CONST ID = 0      ' 0:0, 1:32, 2:64, 3:96,
0009
0010 RR = 0
0011
0012 PTP SETON
0013 PTP ALLON
0014
0015 '== motor direction setting =====
0016 DIR G6A,1,0,0,1,0,0
0017 DIR G6B,1,1,1,1,1,1
0018 DIR G6C,0,0,0,0,0,0
0019 DIR G6D,0,1,1,0,1,0
0020
0021 '== motor start position read =====
0022 TEMPO 230
0023 MUSIC "CDE"
0024 GETMOTORSET G24,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
0025 '== motor power on =====
0026 SPEED 5
0027 MOTOR G24
0028 GOSUB standard_pose
0029 '-----
0030 MAIN:
0031 'GOSUB robot_voltage
0032 GOSUB tilt_proc
0033
0034 IF RR > 50 THEN GOTO action_proc
0035 '-----
0036 IF RR = 0 THEN GOTO MAIN1
0037
0038 ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15
0039 GOTO main_exit
0040 '-----
0041 MAIN1:
0042 A = REMOCON(1)
0043 A = A - ID
0044 ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15

```

DIM RR AS BYTE // La dichiarazione RR è usata come una variabile di ROBOREMOCON

RR = 0 // Inizializza RR come 0

MAIN:

IF RR> 50 THEN GOTO action\_proc  
IF RR=0 THEN GOTO MAIN1 // Salta a MAIN1 da RR = 0

MAIN1:  
A = REMOCON(1)  
A = A - ID  
ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8.....  
GOTO MAIN

Da quando la variabile RR viene inizializzata come 0, il REMOCON HITEC può essere usato. Quando un segnale da RoboRemocon viene ricevuto, il valore di RR cambierà inviando il programma alla sub-routine MAIN.

```

MAIN:
'GOSUB robot_voltage
GOSUB tilt_proc

IF RR > 50 THEN GOTO action_proc
'-----
IF RR = 0 THEN GOTO MAIN1

ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12
GOTO main_exit
'-----

```

Eseguite ogni valore (k1,k2...) con ON RR GOTO.

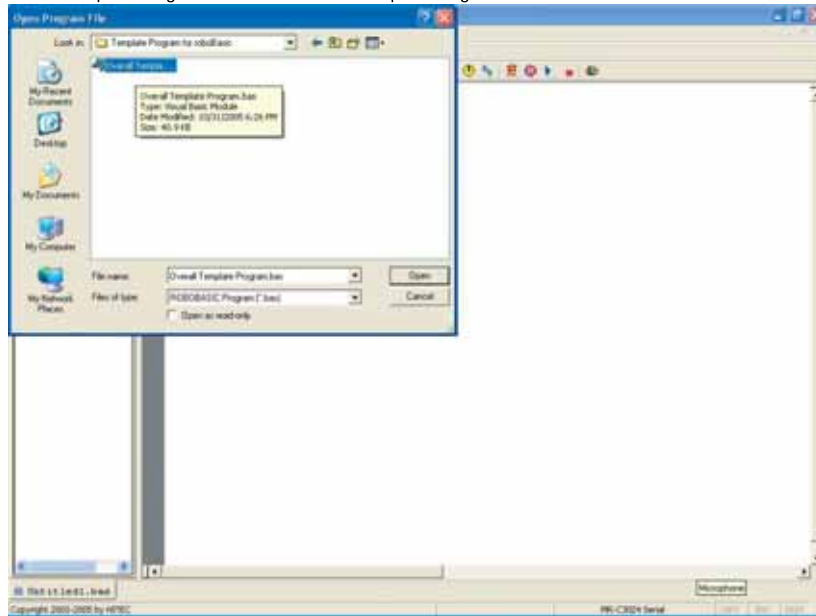
(vedi la Tabella nelle operazioni di IR REMOCON)

## VI. Applicazione del Programma

### 1. Installazione e Funzionamento del Programma di Modello (Template)

#### (1) Installazione del Programma di Modello

Aprire il file "Overall Template Program.bas" nella cartella "Template Program for roboBASIC" del CD.



(Aprire il file di Modello)

#### (2) modi di cambiare il Programma di Modello

Il programma è diviso in tre parti:

- 1) impostazioni Basic del Programma

```

0000 ** auto basic
0001 GOTO AUTO
0002 FILL 255,10000
0003
0004
0005 DIM RR AS BYTE
0006 DIM A AS BYTE
0007 DIM A16 AS BYTE
0008 DIM A32 AS BYTE
0009
0010 CONST ID = 0      0 0 1 32 2 64 3 96
0011
0012 RR = 0
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029
0030
0031
0032
0033
0034
0035
0036
0037
0038
0039
0040
0041
0042
0043
0044
0045
0046
0047
0048
0049
0050
0051
0052
0053
0054
0055
0056
0057
0058
0059
0060
0061
0062
0063
0064
0065
0066
0067
0068
0069
0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106
0107
0108
0109
0110
0111
0112
0113
0114
0115
0116
0117
0118
0119
0120
0121
0122
0123
0124
0125
0126
0127
0128
0129
0130
0131
0132
0133
0134
0135
0136
0137
0138
0139
0140
0141
0142
0143
0144
0145
0146
0147
0148
0149
0150
0151
0152
0153
0154
0155
0156
0157
0158
0159
0160
0161
0162
0163
0164
0165
0166
0167
0168
0169
0170
0171
0172
0173
0174
0175
0176
0177
0178
0179
0180
0181
0182
0183
0184
0185
0186
0187
0188
0189
0190
0191
0192
0193
0194
0195
0196
0197
0198
0199
0200
0201
0202
0203
0204
0205
0206
0207
0208
0209
0210
0211
0212
0213
0214
0215
0216
0217
0218
0219
0220
0221
0222
0223
0224
0225
0226
0227
0228
0229
0230
0231
0232
0233
0234
0235
0236
0237
0238
0239
0240
0241
0242
0243
0244
0245
0246
0247
0248
0249
0250
0251
0252
0253
0254
0255
0256
0257
0258
0259
0260
0261
0262
0263
0264
0265
0266
0267
0268
0269
0270
0271
0272
0273
0274
0275
0276
0277
0278
0279
0280
0281
0282
0283
0284
0285
0286
0287
0288
0289
0290
0291
0292
0293
0294
0295
0296
0297
0298
0299
0300
0301
0302
0303
0304
0305
0306
0307
0308
0309
0310
0311
0312
0313
0314
0315
0316
0317
0318
0319
0320
0321
0322
0323
0324
0325
0326
0327
0328
0329
0330
0331
0332
0333
0334
0335
0336
0337
0338
0339
0340
0341
0342
0343
0344
0345
0346
0347
0348
0349
0350
0351
0352
0353
0354
0355
0356
0357
0358
0359
0360
0361
0362
0363
0364
0365
0366
0367
0368
0369
0370
0371
0372
0373
0374
0375
0376
0377
0378
0379
0380
0381
0382
0383
0384
0385
0386
0387
0388
0389
0390
0391
0392
0393
0394
0395
0396
0397
0398
0399
0400
0401
0402
0403
0404
0405
0406
0407
0408
0409
0410
0411
0412
0413
0414
0415
0416
0417
0418
0419
0420
0421
0422
0423
0424
0425
0426
0427
0428
0429
0430
0431
0432
0433
0434
0435
0436
0437
0438
0439
0440
0441
0442
0443
0444
0445
0446
0447
0448
0449
0450
0451
0452
0453
0454
0455
0456
0457
0458
0459
0460
0461
0462
0463
0464
0465
0466
0467
0468
0469
0470
0471
0472
0473
0474
0475
0476
0477
0478
0479
0480
0481
0482
0483
0484
0485
0486
0487
0488
0489
0490
0491
0492
0493
0494
0495
0496
0497
0498
0499
0500
0501
0502
0503
0504
0505
0506
0507
0508
0509
0510
0511
0512
0513
0514
0515
0516
0517
0518
0519
0520
0521
0522
0523
0524
0525
0526
0527
0528
0529
0530
0531
0532
0533
0534
0535
0536
0537
0538
0539
0540
0541
0542
0543
0544
0545
0546
0547
0548
0549
0550
0551
0552
0553
0554
0555
0556
0557
0558
0559
0560
0561
0562
0563
0564
0565
0566
0567
0568
0569
0570
0571
0572
0573
0574
0575
0576
0577
0578
0579
0580
0581
0582
0583
0584
0585
0586
0587
0588
0589
0590
0591
0592
0593
0594
0595
0596
0597
0598
0599
0600
0601
0602
0603
0604
0605
0606
0607
0608
0609
0610
0611
0612
0613
0614
0615
0616
0617
0618
0619
0620
0621
0622
0623
0624
0625
0626
0627
0628
0629
0630
0631
0632
0633
0634
0635
0636
0637
0638
0639
0640
0641
0642
0643
0644
0645
0646
0647
0648
0649
0650
0651
0652
0653
0654
0655
0656
0657
0658
0659
0660
0661
0662
0663
0664
0665
0666
0667
0668
0669
0670
0671
0672
0673
0674
0675
0676
0677
0678
0679
0680
0681
0682
0683
0684
0685
0686
0687
0688
0689
0690
0691
0692
0693
0694
0695
0696
0697
0698
0699
0700
0701
0702
0703
0704
0705
0706
0707
0708
0709
0710
0711
0712
0713
0714
0715
0716
0717
0718
0719
0720
0721
0722
0723
0724
0725
0726
0727
0728
0729
0730
0731
0732
0733
0734
0735
0736
0737
0738
0739
0740
0741
0742
0743
0744
0745
0746
0747
0748
0749
0750
0751
0752
0753
0754
0755
0756
0757
0758
0759
0760
0761
0762
0763
0764
0765
0766
0767
0768
0769
0770
0771
0772
0773
0774
0775
0776
0777
0778
0779
0780
0781
0782
0783
0784
0785
0786
0787
0788
0789
0790
0791
0792
0793
0794
0795
0796
0797
0798
0799
0800
0801
0802
0803
0804
0805
0806
0807
0808
0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
0819
0820
0821
0822
0823
0824
0825
0826
0827
0828
0829
0830
0831
0832
0833
0834
0835
0836
0837
0838
0839
0840
0841
0842
0843
0844
0845
0846
0847
0848
0849
0850
0851
0852
0853
0854
0855
0856
0857
0858
0859
0860
0861
0862
0863
0864
0865
0866
0867
0868
0869
0870
0871
0872
0873
0874
0875
0876
0877
0878
0879
0880
0881
0882
0883
0884
0885
0886
0887
0888
0889
0890
0891
0892
0893
0894
0895
0896
0897
0898
0899
0900
0901
0902
0903
0904
0905
0906
0907
0908
0909
0910
0911
0912
0913
0914
0915
0916
0917
0918
0919
0920
0921
0922
0923
0924
0925
0926
0927
0928
0929
0930
0931
0932
0933
0934
0935
0936
0937
0938
0939
0940
0941
0942
0943
0944
0945
0946
0947
0948
0949
0950
0951
0952
0953
0954
0955
0956
0957
0958
0959
0960
0961
0962
0963
0964
0965
0966
0967
0968
0969
0970
0971
0972
0973
0974
0975
0976
0977
0978
0979
0980
0981
0982
0983
0984
0985
0986
0987
0988
0989
0990
0991
0992
0993
0994
0995
0996
0997
0998
0999
1000

```

GOTO AUTO

FILL 255,10000 'archivia il programma di modello all'indirizzo 10000 della memoria del regolatore MR-C3024

Ex)

```

GETMOTORSET G24,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
SPEED 5
MOTOR G24

```

```

ACTION 1
DELAY 1000
ACTION 2
DIM RR AS BYTE

```

'la prima variabile dichiarata in RoboBasic viene usata con RoboRemocon.

```

CONST ID =0
'HITEC IR REMOCON (con funzione di ID) puogestire 4 remocons e 4 RoboNova allo stesso tempo
senza interferenza.
'EX: CONST ID =1 (ID No.1)

```

F RR > 50 AND RR < 83 THEN GOTO action\_proc  
'quando l'immissione di un valore è tra '50 e '83 dovendo essere controllato da RoboRemocon, salta a "action\_proc"  
Per altri comandi usati all'interno del Modello, vedere la lista dei comandi di Modello.

#### 2) MAIN

```

-----
MAIN:
  GOSUB robot_voltage
  GOSUB tilt_proc

  IF RR > 50 THEN GOTO action_proc
  IF RR = 0 THEN GOTO MAIN1

  ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,
  GOTO main_exit
-----
MAIN1:
  A = REMOCON(1)
  A = A - ID
  ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,K
  GOTO MAIN
-----
action_proc:
  A = RR - 50
  ON A GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,K
  RETURN
-----

```

MAIN è una funzione molto importante del software ROBONOVA-1. Il comando 'GOSUB robot\_voltage' è annotato nella prima riga della sub-routine MAIN. Questo comando è per un avvertimento di basso voltaggio. (Vedi la sezione VI. Applicazione di programma2. applicazione di programmazione di ROBOBASIC (7) impostazione Avvertimento di Basso Voltaggio) per usare questo comando cancellate il simbolo (!) e la funzione di avvertimento di basso voltaggio (6,25v) sarà attivata. GOSUB robot\_tilt 'quando il robot cade giù, questo comando lo farà stare su (è richiesto il sensore di tilt) IF RR = 0 THEN GOTO MAIN1 'quando RR non riceve alcun segnale, salta a MAIN1.

```

MAIN1:
A = REMOCON(1)
A = A - ID
ON A GOTO MAIN,
K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16,K17,K18,K19,K20,K21,K22,K23,K24,K25,K26,K27,K28,K29,K30,K31,K32
GOTO MAIN

```

'MAIN1: è la sotto-routine che permette all'HITEC REMOCON di controllare il robot. I valori sono salvati dal remocon nella variabile A. Ma il valore di ID è diverso. La frase A = A-ID permette che il movimento sia applicato ad ogni chiave del remocon.



## 3) Funzione di movimento e chiave di remocon

```

0064 '-----
0065 k1:
0066   GOSUB bow_pose
0067   GOSUB standard_pose
0068   GOTO main_exit
0069 k2:
0070   GOSUB hans_up
0071   DELAY 500
0072   GOSUB standard_pose
0073   GOTO main_exit
0074 k3:
0075   GOSUB sit_down_pose
0076   DELAY 1000
0077   GOSUB standard_pose
0078   GOTO main_exit
0079 k4:
0080   GOSUB sit_hans_up
0081   DELAY 1000
0082   GOSUB standard_pose
0083   GOTO main_exit
0084 k5:
0085   GOSUB foot_up
0086   GOSUB standard_pose
0087   GOTO main_exit
0088 k6:
0089   GOSUB body_move
0090   GOSUB standard_pose
0091   GOTO main_exit
0092 k7:
0093   GOSUB wing_move
0094   GOSUB standard_pose
0095   GOTO main_exit
0096 k8:
0097   GOSUB right_shoot
0098   GOSUB standard_pose
0099   DELAY 500
0100   GOSUB left_shoot
0101   GOSUB standard_pose

```

## 4) Movimento individuale

```

0234   RETURN
0235 '-----
0236 sit_down_pose16:
0237   IF A16 = 0 THEN GOTO standard_pose16
0238   A16 = 0
0239   SPEED 10
0240   MOVE G6A, 100, 151, 23, 140, 101, 100
0241   MOVE G6D, 100, 151, 23, 140, 101, 100
0242   MOVE G6B, 100, 30, 80, 100, 100, 100
0243   MOVE G6C, 100, 30, 80, 100, 100, 100
0244   WAIT
0245 '== motor power off =====
0246   MOTOROFF G24
0247   TEMPO 230
0248   MUSIC "FEDC"
0249   RETURN
0250 '-----
0251 standard_pose16:
0252   TEMPO 230
0253   MUSIC "CDE"
0254   GETMOTORSET G24,1,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,0
0255 '== motor power on =====
0256   MOTOR G24
0257   A16 = 1
0258 '-----
0259   SPEED 10
0260   GOSUB standard_pose
0261   RETURN
0262 '-----
0263 '-----
0264 bow_pose:
0265   MOVE G6A, 100, 58, 135, 160, 100, 100
0266   MOVE G6D, 100, 58, 135, 160, 100, 100
0267   MOVE G6B, 100, 30, 80, , ,
0268   MOVE G6C, 100, 30, 80, , ,
0269   WAIT
0270   DELAY 1000
0271   RETURN

```

Per assegnare un'azione individuale ad una chiave del remocon, create la funzione e inseritela nella chiave di sub-routine per la chiave del remocon scelta.

Es.: creare un'operazione individuale e assegnarla ad una chiave. Il seguente programma farà lampeggiare il LED.

```

LED_TOGGLE:
  OUT 52,1
  DELAY 1000
  OUT 52,0
  DELAY 1000

```

quando il programma (routine) è terminato, collegatelo ad una chiave sul remocon.

Cancellate il contenuto attuale nella sub-routine k18 in "action\_auto.bas". Inserite i comandi seguenti.

K18:

```

GOSUB LED_TOGGLE
GOTO main_exit

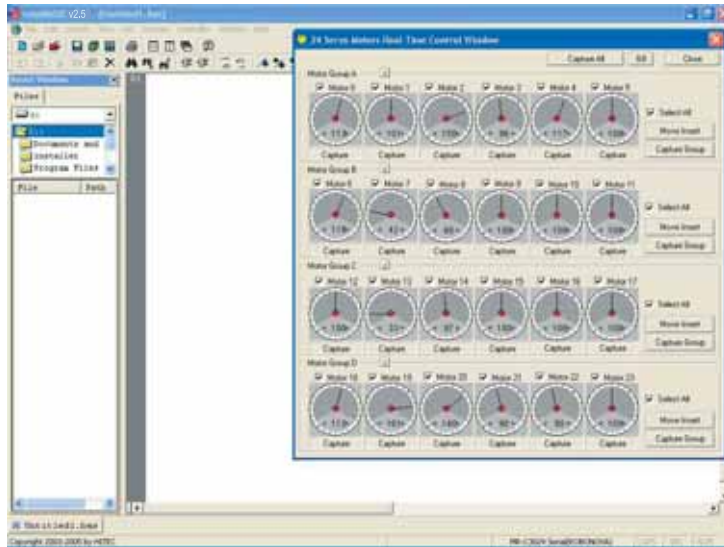
```

Cliccate sul pulsante "Run All" (esecuzione integrata) per caricare sul regolatore.

Spingete il pulsante 'E' sul remocon e la luce comincerà a lampeggiare.

## 2. Applicazioni della Programmazione

### (1) Blocco del movimento e suo inserimento

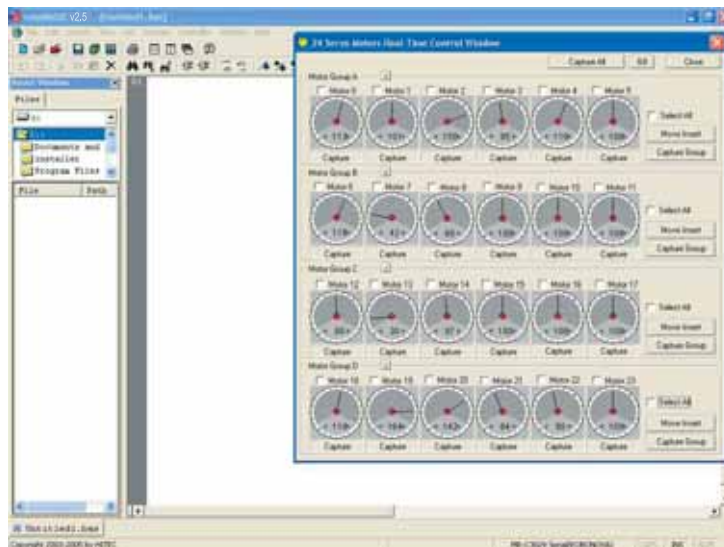


(questa immagine è un esempio dei controlli degli angoli per ogni gruppo di motori)

- Accendete il robot dopo averlo connesso con un cavo seriale e un PC.
- Menu 'Compile' controllo in tempo reale del servo motore.
- Vengono visualizzati gli attuali valori d'angolo dei servi.

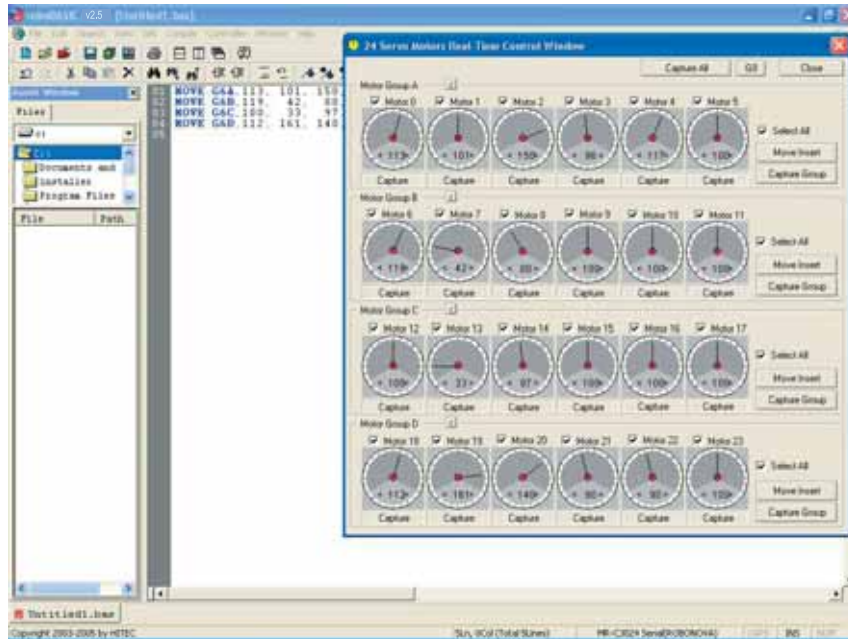


(A questo punto i servi si muovono liberamente. Muoveteli nella posizione desiderata e poi selezionate la finestrella. La nuova posizione sarà impostata e i servi non si muoveranno più liberamente.)



(per creare un movimento, deselegate la finestra di selezione del servo o dei servi interessati al movimento. Per deselegare tutti i servi in un gruppo motore, cliccate il pulsante "Select All".)

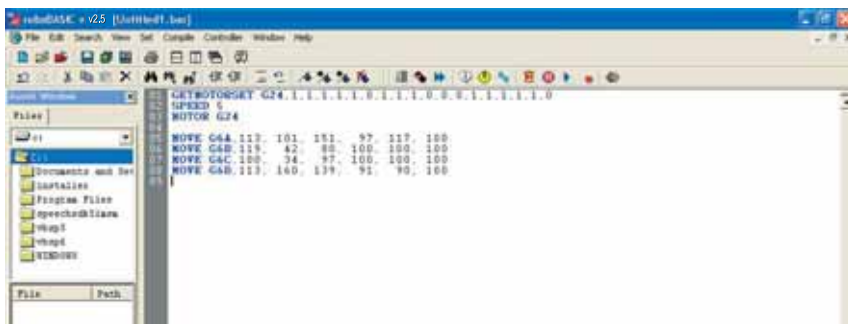




(Cliccate il pulsante "motion insert" per inserire il movimento su RoboBasic.)

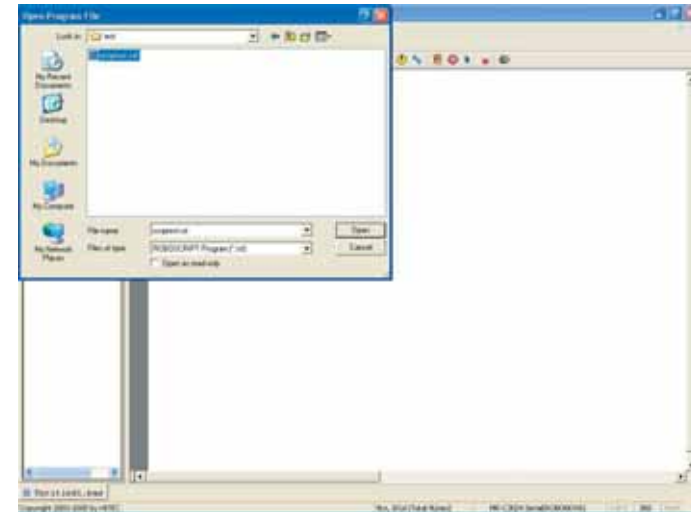
Per eseguire il movimento creato, il seguente comando deve essere inserito prima. Questo comando può essere copiato dal programma di modello e incollato o scritto manualmente dall'utente.

```
GETMOTORSET G24,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
SPEED 5
MOTOR G24
```



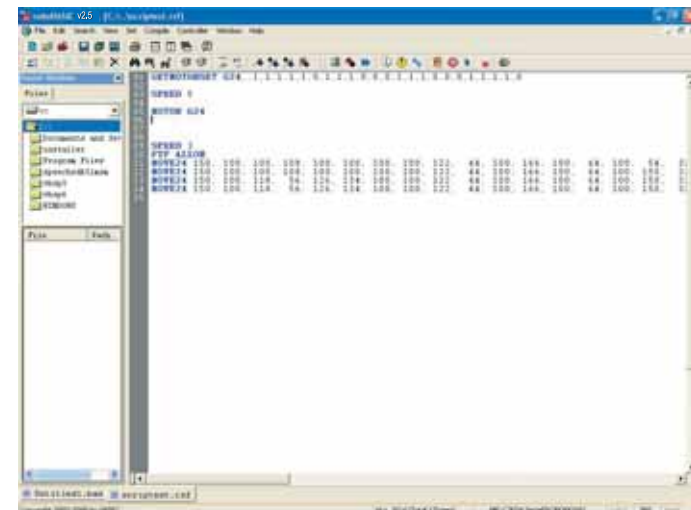
Una volta terminato, cliccate il pulsante "Run All" (esecuzione integrata) e il movimento sarà caricato sul regolatore. Il robot adesso eseguirà il movimento creato.

(2) Come importare un file di RoboScript



(Apre un file \*.rsf)

- Aprire un file di RoboScript (\*.rsf) in RoboBasic.



(Comandi inseriti nel programma RoboBasic)

Aggiungete i seguenti comandi prima dei comandi importati di RoboScript.

```
GETMOTORSET G24,1,1,1,1,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,1,0
SPEED 5
MOTOR G24
```

Cliccate il pulsante "Run All" (esecuzione integrata) per caricare sul regolatore.



(3) Conversione AD  
 il comando AD è nuovamente spiegato in "Capitolo 11. Spiegazione del comando di processione di segnale analogico in RoboBasic" che si trova nel manuale sul CD.

## AD()

Il segnale analogico dalla Porta AD è convertito in un segnale Digitale.

### Struttura della frase

AD ([Porta AD])

### Spiegazione di comando

Nei regolatori di serie MR-C3000 ci sono otto Porte AD (Porte Digitali In-Out da 32 a 39), numerate da zero a sette, che convertono un segnale analogico da periferiche o sensori esterni in un segnale digitale. Per la [Porta AD] sono usate costanti e variabili byte.

### Esempio di comando

Nel seguente esempio, un valore viene emesso ad un modulo LCD dopo aver ricevuto un segnale analogico dalla porta AD n° 1.

```

DIM a AS BYTE      Dichiaro la variabile byte a
LCDINIT            Viene inizializzato l'uso del modulo LCD.
CLS                Tutti i dati sullo schermo dell'LCD sono cancellati.
CSOFF              il cursore scompare.
MAIN:              Viene dichiarata un'etichetta di nome MAIN.
A = AD (1)         Il valore immesso nella Porta AD #1 è salvato come variabile a.
LOCATE 5, 0        Il cursore è posizionato a 5.0 sull'LCD.
PRINT FORMAT(a,DEC,2) Il valore immesso, a, viene emesso al modulo LCD
                  Come due digitali usando il sistema decimale.
GOTO MAIN          Va alla MAIN.
    
```

### (4) Impostazioni Giroscopio

- i giroscopi possono essere connessi alle porte AD 0,1,2,3,4,5,6,7.  
 - quando un giroscopio KRG-1 (kondo) viene usato, deve essere selezionato il metodo normale (normal mode) dai due metodi (Robot Mode/valore iniziale, Normal Mode). Per informazioni dettagliate, richiedete al costruttore del giroscopio.

La funzione del giroscopio è spiegata nuovamente nel Capitolo 11. Spiegazione del comando di processione di segnale analogico in RoboBasic che si trova nel manuale sul CD.

## GYRODIR

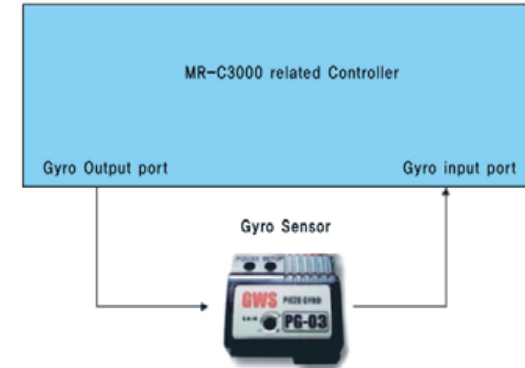
Imposta la direzione dei servo-motori quando supportato da un Giroscopio

### Struttura della frase

GYRODIR [Gruppo], [Direzione Motore] ...

### Esempio di comando

Questo processo controlla la direzione di un gruppo di servo-motori quando un giroscopio è connesso ad una porta AD dei regolatori di serie MR-C3000. Il numero di giroscopi utilizzabili è quattro. Vedere la tabella seguente.



AD Port Number (Digital In and Out Port Number) of MR-C3000 series controller.	Gyro Port
Port #0 (Port #32)	Gyro #1 channel output Port
Port #1 (Port #33)	Gyro #2 channel output Port
Port #2 (Port #34)	Gyro #3 channel output Port
Port #3 (Port #35)	Gyro #4 channel output Port
Port #4 (Port #36)	Gyro #1 channel input Port
Port #5 (Port #37)	Gyro #2 channel input Port
Port #6 (Port #38)	Gyro #3 channel input Port
Port #7 (Port #39)	Gyro #4 channel input Port

Finché un giroscopio è reversibile, la specifica direzione sarà determinata dal contingente valore del servo-motore. La [Direzione Motore] è uno "0" o un "1". Un valore di 1 incrementerà la posizione del servo-motore e uno 0 la diminuirà.

### Esempio di comando

GYRODIR G6A, 1, 1, 0, 0, 1, 0

## GYROSET

Define que giróscopo controlará un grupo determinado de servos.

### Struttura della frase

GYROSET [Gruppo], [Giro N Motore] ...

### Spiegazione di comando

Gyroset determina quale servo-motore in gruppo di servo-motori [Gruppo] viene controllato da un giroscopio specifico. [Giro N Motore] è la specifica porta "giro" usata per ogni servo nel gruppo. Si veda l'esempio sotto.

### Esempio di comando

GYROSET G6B, 1, 1, 2, 2, 0, 0

Il servo #6 riceve il sensore di giroscopio #1 e lo processa.  
 Il servo #7 riceve il sensore di giroscopio #1 e lo processa.  
 Il servo #8 riceve il sensore di giroscopio #2 e lo processa.  
 Il servo #9 riceve il sensore di giroscopio #2 e lo processa.  
 I servo-motori #10 e #11 non usano un sensore di giroscopio.

## GYROSENSE

Imposta la sensibilità di un servo-motore ad un giroscopio.

### Struttura della frase

GYROSENSE [Gruppo], [Giroscopio N Motore Sensibilità] ...

### Spiegazione di comando

Si possono connettere quattro giroscopi al regolatore di serie MR-C3000. GYROSENSE imposta la sensibilità di un singolo servo-motore ad un giroscopio. [Giroscopio N Motore Sensibilità] usa numeri da 0 255 o costanti per controllare la sensibilità di ogni servo-motore di un gruppo. Un'impostazione di "0" non cambierà la sensibilità del servo. Come il valore aumenta così sarà la risposta del giroscopio.

### Esempio di comando

GYROSENSE G6A, 100, 100, 255, 255, 50, 50

I servo-motori #0 e #1 impostano a 100 la sensibilità del Giroscopio.  
 I servo-motori #2 e #3 impostano al massimo (255) la sensibilità del Giroscopio.  
 I servo-motori #4 e #5 impostano a 50 la sensibilità del Giroscopio.

1) come connettere i giroscopi GWS, KRG-1

- \* giro porta #1: AD0, AD4
- \* giro porta #2: AD1, AD5
- \* giro porta #3: AD2, AD6
- \* giro porta #4: AD3, AD7

Es.:

i servo motori #1, #2, #3 nel gruppo A e i servo motori #1, #2, #3 nel gruppo D stanno usando il giroscopio #1  
 i servo motori #1, #2, #3 nel gruppo A e i servo motori #1, #2, #3 nel gruppo D stanno usando il giroscopio con direzione #0  
 i servo motori #1, #2, #3 nel gruppo A e i servo motori #1, #2, #3 nel gruppo D stanno usando il giroscopio con sensibilità 205,200,250

(Codice per un giroscopio GWS)

GYROSET G6A,0,1,1,1,0,0  
 GYROSET G6D,0,1,1,1,0,0

GYROSET G6A,0,0,0,0,0,0  
 GYROSET G6D,0,0,0,0,0,0

GYROSENSE G6A, 0, 250, 200, 250, 0, 0  
 GYROSENSE G6D, 0, 250, 200, 250, 0, 0

(Codice per giroscopio KRG-1)

\* KRG-1  
 (versione normale: per ogni domanda, chiedere al costruttore)  
 \* Write 1  
 (numero di modulo di KRG-1) davanti al numero di canale del giro.  
 (il numero di modulo del giroscopio GWS è 0 ma 0 può essere omesso)  
 \* Write (scrivere) numero di canale del giroscopio dopo il numero di modulo

- \* "10" significa non usare il giroscopio
- \* utilizzando #1 canale giroscopio "11"
- \* utilizzando #2 canale giroscopio "12"
- \* utilizzando #3 canale giroscopio "13"
- \* utilizzando #4 canale giroscopio "14"

GYROSET G8A,11,10,10,10,10,10,10  
 GYROSET G8B,10,10,10,10,10,10,10  
 GYROSET G8C,10,10,10,10,10,10,10  
 GYROSET G8D,10,10,10,10,10,10,10

.....  
 GYROSENSE G8A,255,1,1,1,1,1,1  
 GYROSENSE G8B,1,1,1,1,1,1,1,1  
 GYROSENSE G8C,1,1,1,1,1,1,1,1  
 GYROSENSE G8D,1,1,1,1,1,1,1,1

GYRODIR G8A,1,1,1,1,1,1,1,1

(5) Come usare I2C

Questa è un'interfaccia di una forma di comunicazione. Questa può essere usata quando ci sono 2 porte disponibili a ricevere e trasmettere un segnale.

CONST scl =22  
 CONST sda =23

dim A as byte

'=====

I2C\_SAMPLE:

'=====

a = IN(sda)  
 s1: OUT scl,0  
 OUT scl,1

'=====

s3: OUT sda,0  
 OUT scl,0  
 OUT scl,1

'=====

OUT sda,1  
 OUT scl,0  
 OUT scl,1

OUT sda,0  
 OUT scl,0  
 OUT scl,1

OUT sda,0  
 OUT scl,0  
 OUT scl,1

OUT sda,1  
 OUT scl,0  
 OUT scl,1

'=====

CODE = 0  
 FOR I = 0 TO 7

OUT SCL,0  
 OUT SCL,1  
 A = IN(SDA)

A = A<<1  
 CODE = CODE OR A  
 NEXT I

'=====

RETURN

## (6) impulso del protocollo di HMI

HMI (Interfaccia Multi-Protocollo Hitec) è una nuova interfaccia creata per i servi robot Hitec. (per una programmazione dei servi più dettagliata, viene richiesto un kit di impostazioni opzionali.)

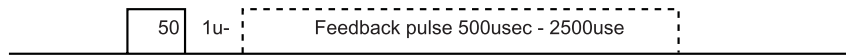
\* le seguenti informazioni possono essere usate da utenti interessati ai dati d'impulso. Questi dati d'impulso sono richiesti per la posizione (angolo) del feedback tra i servi robot Hitec e un PC.

I servi eseguiranno un movimento specifico quando quattro tipi d'impulso saranno immessi da una sorgente esterna.

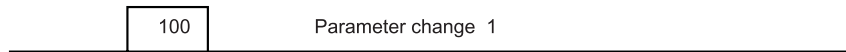
- 1) ampiezza d'impulso 50usec / feedback del valore della posizione (angolo)
- 2) ampiezza d'impulso 100usec / usa il valore di servo a parametro 1 (iniziale)
- 3) ampiezza d'impulso 150usec / usa il valore di servo a parametro 2
- 4) ampiezza d'impulso 200usec / usa il valore di servo a parametro 3

Nota: per usare la posizione (angolo) feedback con un circuito esterno, il terminale del segnale di controllo deve avere funzioni bidirezionali. Per leggere il valore di feedback, deve essere eseguito un processo di risalita nella linea di segnale.

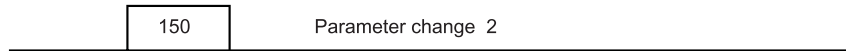
Control signal 50usec



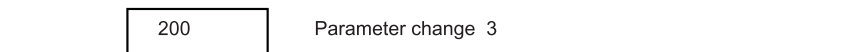
Control signal 100usec



Control signal 150usec



Control signal 200usec



Nota: la funzione di feedback ha un tasso d'errore del 10% poiché questa funzione è usata in congiunzione con il controllo PWM.

## (7) impostazione dell'AVVERTIMENTO DI BASSO-VOLTAGGIO

quando il voltaggio della batteria di ROBONOVA-1 scende sotto un certo livello, il LED di ROBONOVA-1 installato sulla testa lampeggerà per avvertire l'utente che è richiesta immediatamente la ricarica della batteria. Il valore iniziale di basso voltaggio è 5,8V. questo può essere cambiato con la programmazione.

```

0029 '-----
0030 MAIN:
0031 GOSUB robot_voltage
0032 GOSUB tilt_proc
0033
0034 IF RR > 50 THEN GOTO action_proc
0035
0036 IF RR = 0 THEN GOTO MAIN1
0037
0038 ON RR GOTO MAIN,K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K
0039 GOTO main_exit
0040 '-----
    
```

Cancellate l'annotazione di comando (!) da 'GOSUB robot\_voltage trovato nella sub-routine MAIN  
Es.: Cambiare 'GOSUB robot\_voltage in GOSUB robot\_voltage

```

-----
robot_voltage:
DIM v AS BYTE
A = AD(6)
IF A < 148 THEN
    5.8v
FOR v = 0 TO 2
    OUT 52,1
    DELAY 200
    OUT 52,0
    DELAY 200
NEXT v
RETURN
    
```

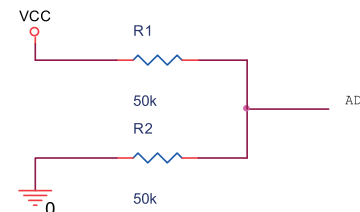
Va alla sub-routine robot\_voltage.  
Per cambiare il valore di basso\_voltaggio, inserite il seguente comando:

IF A < [value] THEN  
(in questo esempio, valore = voltaggio\*256/10)

Es.: se il nuovo valore di basso voltaggio è 6,0V, il [valore] sarà 153,6 (6,0 \* 256/10 = 153,6). Poiché RoboBasic non può calcolare i punti decimali, viene usato 153.

IF A < 153 THEN  
6.0 \* 256/10 = 153.6 = 153 -> IF A < 153 THEN

### 3. schema di basso voltaggio





## ROBOBASIC

### Manuale Istruzioni di Comando

#### v.2.10

#### Marchio Registrato

Windows è un marchio registrato di Microsoft Corporation. ROBOBASIC è un software registrato di miniROBOT, INC.

#### Avviso

Questo manuale spiega i comandi usati da roboBASIC. Hitec non è responsabile di ogni uso improprio che potrebbe essere procurato. Questo manuale può essere soggetto a cambiamenti senza preavviso per migliorare la qualità funzionale del prodotto in questione.

RoboBasic è un software registrato, perciò è illegale la riproduzione, la pubblicazione, la spedizione, la trasmissione o la distribuzione di questo manuale o del software senza permesso.

#### Indice

<b>Capitolo 1:</b>	Sommario dei Comandi per ROBOBASIC	-	2
<b>Capitolo 2:</b>	Grammatica generica per ROBOBASIC	-	5
<b>Capitolo 3:</b>	Spiegazione dei Comandi di dichiarazione in ROBOBASIC	-	9
<b>Capitolo 4:</b>	Spiegazione del Comando di controllo del flusso	-	10
<b>Capitolo 5:</b>	Spiegazione del segnale digitale in entrata e uscita	-	16
<b>Capitolo 6:</b>	Spiegazione dei Comandi relativi alla memoria	-	20
<b>Capitolo 7:</b>	Controllo del modulo LCD in ROBOBASIC	-	22
<b>Capitolo 8:</b>	Spiegazione dei Comandi di controllo del motore	-	26
<b>Capitolo 9:</b>	Comandi di ROBOBASIC per il Controllo Musica	-	36
<b>Capitolo 10:</b>	Comandi per le comunicazioni esterne di ROBOBASIC	-	40
<b>Capitolo 11:</b>	descrizione del procedimento di Comando del segnale Analogico	-	44
<b>Capitolo 12:</b>	Comandi di Procedimento e Altri di ROBOBASIC	-	49
<b>Capitolo 13:</b>	Descrizione di Comando di ROBOBASIC	-	51

## Capitolo 1

### Sommario dei Comandi

#### per roboBasic

#### Sommario di Comando

RoboBASIC è un esclusivo linguaggio di programmazione con funzione di controllo per robot. Con roboBASIC, i comandi che sono necessari per controllare un robot sono stati aggiunti al linguaggio generico di programmazione BASIC.

- (2) Questo simbolo significa che questo comando può essere eseguito solo con il regolatore di serie MR-C2000
- (3) Questo simbolo significa che questo comando può essere eseguito solo con il regolatore di serie MR-C3000

#### Comandi relativi a dichiarazione/definizione

DIM	Dichiara la variabile
AS	Assegna la variabile quando si dichiara la variabile
CONST	Dichiara la costante
BYTE	Assegna come tipo di byte quando si dichiara la variabile
INTEGER	Assegna come tipo di numero intero quando si dichiara la variabile

#### Comandi di controllo del flusso

IF	Inizia l'istruzione condizionale
THEN	Esegue l'istruzione successiva quando la condizione di affermazione condizionale è vera
ELSE	Esegue l'istruzione successiva quando la condizione di affermazione condizionale è falsa
ELSEIF	Inizia un'altra istruzione condizionale
ENDIF	Termina l'istruzione condizionale
FOR	Inizia l'istruzione in circolo
TO	Assegna la serie ripetitiva per un'istruzione in circolo
NEXT	Finisce l'istruzione in circolo
GOTO	Divide il flusso del programma
GOSUB	Richiama una sub routine
RETURN	Ritorno da una sub routine
END	Termina l'esecuzione del programma
STOP	Ferma l'esecuzione del programma
RUN	Esegue il programma continuamente
WAIT	Attende finché il programma è completato
DELAY	Ritarda l'esecuzione del programma per un determinato periodo di tempo
(2) BREAK	Mette in pausa l'esecuzione del programma e passa alla modalità messa a punto

#### Comando del segnale digitale in entrata e uscita

IN	Legge il segnale dalla porta d'entrata
OUT	Invia il segnale alla porta d'uscita
BYTEIN	Legge il segnale byte dalla porta d'entrata
BYTEOUT	Invia il segnale byte alla porta d'uscita
(2) INKEY	Codice in entrata dalla porta d'entrata
STATE	Situazione della porta d'uscita
PULSE	Invia segnale pulsante alla porta d'uscita
TOGGLE	Ribalta la situazione della porta d'uscita
(3) KEYIN	Riceve una tastiera analogica in entrata

**Comando per la memoria**

PEEK	Legge i dati dal regolatore RAM
POKE	Scrivete i dati al regolatore RAM
ROMPEEK	Legge i dati dall'esterna EEPROM RAM del regolatore
ROMPOKE	Scrivete i dati al regolatore EEPROM RAM

**Comando per l'LCD**

LCDINIT	Inizializza il modulo LCD
CLS	Cancella tutti i caratteri sul modulo LCD
LOCATE	Definisce il posto della lettera sul modulo LCD
PRINT	Mostra la lettera sul modulo LCD
FORMAT	Definisce il tipo di formato mostrato sul modulo LCD
CSON	Fa apparire il cursore sul modulo LCD
CSOFF	Nasconde il cursore sul modulo LCD
CONT	Definisce il contrasto delle lettere del modulo LCD
DEC	Emissione numerale decimale dell'LCD
HEX	Emissione numerale esadecimale dell'LCD
(3) BIN	Emissione in numero binari dell'LCD

**Operazione relativa all'operante**

AND	Usa l'espressione logica condizionale AND
OR	Usa l'espressione logica condizionale OR
MOD	Calcola il modulo per un'operazione aritmetica
XOR	Usa l'espressione logica condizionale XOR
(3) NOT	Inverte tutti i bit

**Comando di controllo del motore**

ZERO	Definisce il punto-0 (angolo neutrale) del servo
MOTOR	Accende la porta d'uscita del servo
MOTOROFF	Spegne la porta d'uscita del servo
MOVE	Comanda diversi servi nello stesso tempo
SPEED	Definisce la velocità del servo
(2) ACCEL	Definisce l'accelerazione del servo
DIR	Definisce la direzione del motore del servo
PTP	Accende/spegne la simultanea operazione di controllo
SERVO	Controlla il servo
PWM	Definisce il controllo dell'ampiezza di pulsazione di un motore DC
(2) FASTSERVO	Comanda il servo alla massima velocità
(3) HIGHSPEED	Accende/spegne la modalità veloce del servo
(3) MOVEPOS	Muove il gruppo servo dichiarato da POS
(3) POS	Definisce una posa specifica per il robot
(3) FPWM	Cambia l'ampiezza di pulsazione e la frequenza
(3) MOVE24	Comanda tutti i 24 servi nello stesso tempo
(3) INIT	Definisce la posizione iniziale di movimento
(3) MOTORIN	Legge il valore dell'attuale posizione del servo
(3) AIMOTOR	Modalità per usare motore AI
(3) AIMOTOROFF	Annulla motore AI
(3) AIMOTORIN	Legge il valore dell'attuale posizione del motore AI
(3) SETON	Modalità per usare la "funzione modalità"
(3) SETOFF	Annulla la "funzione modalità"
(3) ALLON	Funzioni modalità per tutti i servi
(3) ALLOFF	Annulla la funzione modalità per tutti i servi
(3) GETMOTORSET	Legge l'attuale valore del servo e mantiene la posizione corrente

**Parametri da assegnare al gruppo motore**

(3) G6A	Assegna i servi #0-#5 al gruppo A
(3) G6B	Assegna i servi #6-11# al gruppo B
(3) G6C	Assegna i servi #12-#17 al gruppo C
(3) G6D	Assegna i servi #18-#23 al gruppo D
(3) G6E	Assegna i servi #24-#29 al gruppo E
(3) G8A	Assegna i servi #0-#7 al gruppo A
(3) G8B	Assegna i servi #8-#15 al gruppo B
(3) G8C	Assegna i servi #16-#23 al gruppo C
(3) G8D	Assegna i servi #24-#31 al gruppo D
(3) G12	Assegna i servi #0-#11
(3) G16	Assegna i servi #0-#15
(3) G24	Assegna i servi #0-#23
(3) G32	Assegna i servi #0-#31

**Comando per il controllo del suono**

(2) BEEP	Emette un suono d'avvertimento con PIEZO
(2) SOUND	Emette un suono di frequenza con PIEZO
(2) PLAY	Suona una canzone con PIEZO
(3) MUSIC	Suona musica con PIEZO
(3) TEMPO	Definisce un ritmo sonoro

**Comando per le comunicazioni esterne**

(2) RX	Riceve un segnale RS-232 attraverso la porta RX
(2) TX	Trasmette un segnale RS-232 attraverso la porta TX
(2) MINIIN	Riceve un segnale minibus attraverso la porta di comunicazione mini
(2) MINIOUT	Trasmette un segnale minibus attraverso la porta di comunicazione mini
(3) ERX	Riceve un segnale RS-232 attraverso la porta RX
(3) ETX	Trasmette un segnale RS-232 attraverso la porta TX

**Comandi per processare il segnale analogico**

(3) AD	Legge il segnale analogico dalla porta AD
(3) REMOCON	Legge un valore di codice da un telecomando a infrarossi
(3) SONAR	Legge la distanza dalla porta ad onda di ultrasuoni
(3) RCIN	Legge il valore d'entrata da un telecomando RC
(3) GYRODIR	Definisce la direzione di un giroscopio
(3) GYROSET	Assegna un giroscopio ad un servo
(3) GYROSENSE	Definisce l'intensità di un giroscopio

**Comando di processione**

ON...GOTO Salta secondo il valore di una variabile

**Altri comandi**

RND	Crea un numero casuale
REMARK	Fa un'ingresso nel testo

**Comandi d'intenzione**

(3) \$DEVICE	Definisce che il regolatore sia processato dal programma in azione in quel momento
(3) \$LIMIT	Delimita l'estensione di spostamento del servo

## Capitolo 2

### Grammatica generica

#### per roboBASIC

Dato che la grammatica di roboBASIC si basa sul linguaggio generico di programmazione BASIC, grossa parte di roboBASIC è simile o uguale a BASIC. In questo capitolo verrà spiegata la grammatica generica di roboBASIC

#### Settaggio del carattere

Il settaggio del carattere grammaticale di roboBASIC è composta dalle lettere dell'alfabeto Inglese (A-Z, a-z), dai numeri (0-9) e da simboli speciali. I simboli che sono elencati nella tabella seguente in roboBASIC hanno un significato speciale.

Simbolo	Descrizione
+	Simbolo d'addizione
-	Simbolo di sottrazione
*	Simbolo di moltiplicazione
/	Simbolo di divisione
%	Simbolo di rimanenza
.	Simbolo di assegnazione di bit
&	Simbolo numerico
??	Simbolo di testo
??	Simbolo di stringa di caratteri
:	Simbolo d'etichetta
=	Simbolo di segno uguale o di sostituzione
<	Simbolo di disuguaglianza
>	Simbolo di disuguaglianza
<<	Simbolo di spostamento di bit a sinistra
>>	Simbolo di spostamento di bit a destra

#### Formula e operatore

Le formule possono essere composte da un valore che calcolato da integranti invariabili, variabili, e numeriche con ognuna utilizzando degli operatori. Un operatore esegue operazioni aritmetiche o logiche per un dato valore. In roboBASIC, gli operatori possono essere classificati come nella tabella sottostante.

Classificazione	Funzione
Operatore aritmetico	Esegue un calcolo aritmetico
Operatore relazionale	Confronta i valori numerici
Operatore logico	Confronta condizioni combinate o pratica la manipolazione di bit
Operatore di bit	Manipola i bit o esegue operazioni per i bit

#### Operatori aritmetici

Un operatore aritmetico è un simbolo che esegue un calcolo. Come nel generico linguaggio BASIC, addizione (+), sottrazione (-), moltiplicazione (\*), divisione (/), e modulo (% o MOD) possono essere usati in roboBASIC. Ma ci sono alcuni punti differenti tra roboBASIC e BASIC generico.

#### Prima di tutto non c'è precedenza nell'operatore

In roboBASIC le parentesi ( ) non possono essere usate

Esempio: A = 1, B = 2, C = 3

BASIC generico:

$A + B * C = 1 + 2 * 3 = 1 + 6 = 7$  (in BASIC, le parentesi sarebbero usate se l'equazione d'addizione avesse

precedenza sulla moltiplicazione)

RoboBASIC:

$$A + B * C = 1 + 2 * 3 = 3 * 3 = 9$$

In secondo luogo, calcoli matematici complessi possono causare errori inaspettati

In questo caso il calcolo matematico deve essere diviso in 2 o 3 calcoli.

Esempio:

$$D = A * B + C \quad (\text{Accettato})$$

$$F = A * B / C * D + E \quad (\text{Evitare calcoli aritmetici complessi})$$

In più roboBASIC supporta solo tipi di byte o tipi di numeri interi, perciò un punto decimale nel risultato sarà ignorato.

Le operazioni del modulo sono il simbolo "%" o "MOD" e l'emissione sarà un modulo.

#### Operatori relazionali

Un operatore relazionale è usato per comparare due valori. L'emissione può essere "TRUE" o "FALSE". Questa emissione è usata per controllare il flusso di un programma in una frase IF.

Operatore	Relazione	Espressione
=	Uguale a	X = Y
<>	Diverso	X <> Y
<	Minore di	X < Y
>	Maggiore di	X > Y
<=	Minore o uguale a	X <= Y
>=	Maggiore o uguale a	X >= Y

Quando un operatore aritmetico e uno logico sono combinati in un'unica formula, l'operatore aritmetico sarà eseguito prima dell'operatore logico.

#### Operatori logici

Un operatore logico è usato per comparare condizioni combinate. Il risultato dei calcoli riporta sia "TRUE" che "FALSE". Questa emissione è usata per controllare il flusso di un programma in una frase IF.

Operatore	Significato
AND	E
OR	Disgiunzione
XOR	Disgiunzione esclusiva

Ogni operatore ha un'emissione come nella tabella sottostante. Nella tabella, "V" significa vero, "F" significa falso.

Valore di X, Y		Emissione		
X	Y	X AND Y	X OR Y	X XOR Y
V	V	V	V	F
V	F	F	V	V
F	V	F	V	V
F	F	F	F	F

#### Operatori di bit

Un operatore di bit esegue dei calcoli per ogni variabile usata nel regolatore del robot facilitando il controllo dei bit attraverso le porte di entrata/uscita.

Ci sono bit di somma (OR), bit di produzione (AND) e speciali bit di somma per calcoli di tutti i bit. In roboBASIC,



il i simboli di calcolo, sinistra (<<), destra (>>) e “.”, sono usati per muovere un bit in un punto specifico.

Se il valore di A è 33 (numero binario 00100001) e il valore di B è 15 (numero binario 00001111), avremo i seguenti risultati quando usati gli operatori menzionati.

Operatore	Emissione
A AND B	1 (00000001)
A OR B	47 (00101111)
A XOR B	46 (00101110)
A << 1	66 (01000010)
A >> 1	16 (00010000)
A.0	1 (bit-zero di A)

Quando svariati operatori sono usati nello stesso comando, l'operazione sarà eseguita nel seguente ordine.

- 1) operatore aritmetico/ operatore di Bit
- 2) operatore realzionale
- 3) operatore logico

### Figura, Variabile/Costante e altre spiegazioni grammaticali

Dato che roboBASIC è stato sviluppato per controllare hardware, roboBASIC non supporta variabili o costanti relative alle stringhe usate nel BASIC generale.

#### Tipo di figura

Ci sono figure di tipo byte e figure di tipo numero intero. La gamma realtiva al tipo di figura usata è mostrata sotto.

Tipo di Figura	Dimensione	Gamma
BYTE	1 byte (8bit)	0-255
INTEGER (numero intero)	2 byte (16bit)	0-65535

*RoboBASIC non supporta i numeri negativi.* Perciò quando un simbolo “+” o “-” viene aggiunto di fronte ad un numero, l'operazione darà come risultato un errore.

Le dichiarazioni devono essere di un tipo di numero adatto.

#### Antilogaritmo

Dato che roboBASIC è progettato per controllare hardware, è più conveniente usare un'espressione esadecimale o un altro tipo che una di tipo decimale. In roboBASIC possono essere usati numeri binari (Bin), numeri ottonari (Oct), numeri decimali (Dec), e numeri esadecimale (Hex).

Antilogaritmo	Dichiarazione	Figura Utilizzabile	Esempio
Numero binario	&B	0, 1	&B111101
Numero ottonario	&O	0, ... , 7	&O75
Numero decimale	N/A	0, ... , 9	61
Numero esadecimale	&H	0, ... , 9, A, ... F	&H3D

#### Costante e variabile

Una costante non cambia durante l'esecuzione del programma. RoboBASIC può definire la costante come un numero di tipo byte o un numero di tipo intero. Il tipo di costante è definito automaticamente secondo la gamma del numero. Una volta che la costante è definita, non può essere ri-definita. Definire una costante non ha effetto sulle dimensioni del programma. Una modifica del programma può essere più conveniente quando il numero usato frequentemente viene definito come una costante.

Esempio

```
CONST OFF = 0
CONST motr_1 = 3
CONST motor_1 speed = 200
```

Variabile è il nome di una posizione di memoria nei dati usata nel programma. Nel regolatore del minirobot, il numero di variabili è limitato, così la dichiarazione variabile deve essere destinata per minimizzare il formato delle variabili in conformità con l'oggetto.

```
DIM motor_1_delay AS INTEGER
DIM sensor_left AS BYTE
```

Quando si dichiara una costante o una variabile, si seguano le regole sotto elencate.

Primo: l'Inglese o il Coreano devono essere usati nella prima lettera. In Coreano (Cinese) o Inglese, le figure e “\_” possono essere usati per il nome di una variabile o di una costante.

Secondo: il nome di una variabile o di una costante non può superare la lunghezza di 64 caratteri.

Terzo: il nome di una variabile o di una costante non può essere dichiarato due volte con lo stesso nome e non c'è distinzione tra letter maiuscole e minuscole.

Quarto: Nel dichiarare un punto costante più grande di 65535, che è la limitazione della gamma di numero intero, può risultare un errore.

#### Indicare bit

In roboBASIC le variabili possono essere utilizzate come unità di bit. Per utilizzare le variabili come unità di bit, è usato l'operatore indicatore di bit (“.”). quando si usa l'operatore indicatore di bit sono possibili i bit 0~7 (variabile byte) e i bit 0~15 (variabile numero intero). Solo una figura o una costante possono essere usate con questo operatore.

Esempio

```
DIM A AS INTEGER
CONST BIT_2 = 2
```

```
A.1 = 1
A.BIT_2 = 0
A.3 = IN(1)      'legge il valore dalla porta #1 e mette questo valore nel terzo bit del numero intero
variabile A
```

```
OUT 2, A.1      'estrae alla porta #2 il valore del primo bit del numero intero variabile A
```

#### Dichiarazione di spiegazione

Le spiegazioni di codice dovrebbero essere sparpagliate nel programma per un'efficiente amministrazione e creazione. Per inserire una dichiarazione di spiegazione viene usato il simbolo (\*) o il comando “REMARK”. La disposizione di una dichiarazione nel programma non ha effetto sull'esecuzione del programma.

#### Dichiarazione di sostituzione

Una dichiarazione di sostituzione è usata per sostituire un valore con una variabile. Viene usato il simbolo “=” . Il valore è sempre alla sinistra del simbolo di sostituzione (=) e la variabile, stringa di caratteri, formula di calcolo o funzione è alla destra del simbolo di sostituzione (⇒).

Esempio

```
A = B      'sostituisce ogni variabile
A.1 = 1    'sostituzione di bit indicatore
A = ADIN(0) 'sostituzione di funzione
A = 3 * 2 - 1 Formula di calcolo numerica sostitutiva
```



```
[dichiarazione quando la condizione 1 è vera]
ELSEIF [condizione 2] THEN
[dichiarazione quando la condizione 2 è vera]
ELSE
[dichiarazione quando la condizione 1 e la condizione 2 sono false]
ENDIF
```

**Spiegazione di comando**

Quando una condizione IF ... THEN viene eseguita, la condizione IF verrà studiata. Se la condizione è TRUE (vera), la dichiarazione THEN sarà eseguita. Quando la circostanza è FALSE (falsa), ogni condizione seguente di ELSEIF sarà studiata ed eseguita, al contrario la dichiarazione condizionale ELSE sarà eseguita. Qui l'ELSEIF può essere inclusa oppure no a seconda della necessità.

In roboBASIC, la frase (IF ... THEN) è essenziale per operare in accordo con una periferica esterna e salvare il valore esterno come una variabile. La dichiarazione condizionale giudica il valore della variabile e permette al robot di muoversi a seconda del valore.

**Esempio di comando**

1. l'esecuzione della condizione e dichiarazione è molto semplice. Entrambe possono essere incluse nella stessa linea.

```
IF A > 0 THEN B = 5
IF A > 5 THEN B = 0 ELSE B = 1
```

1. la formula condizionale di una frase IF può usare 2 tipi di condizioni quando si usa un operatore relazionale.

```
IF A > 0 AND A < 5 THEN B = 3
IF A = 7 OR A = 9 THEN B = 1
```

1. esempio di utilizzo di una complicata frase IF

```
IF A = 1 THEN
  B = 2
  C = 3
ELSEIF A = 3 AND A = 5 THEN
  B = 1
  C = 2
ELSEIF A = 8 THEN
  B = 6
  C = 0
ELSE
  B = 0
  C = 0
ENDIF
```

**FOR ... NEXT**

Cicla un numero fisso di volte

**Struttura della frase**

frase in Inglese: FOR [variabile del ciclo] = [inizio] TO [fine]  
[dichiarazione del ciclo]  
NEXT [variabile del ciclo]

**Spiegazione di comando**

[Variabile del ciclo] conta il numero di cicli. [Inizio] è il valore iniziale del ciclo e [fine] è l'ultimo valore del ciclo della variabile. Una figura, una costante, o una variabile può essere usata per i passaggi di [inizio] e [fine].

In roboBASIC il passaggio [Fine] deve essere più grande del passaggio [inizio]. roboBASIC aumenta incrementalmente la variabile del ciclo. Esistono regole per usare le frasi FOR ... NEXT.

1. una frase FOR ... NEXT può essere usata dentro un'altra frase FOR ... NEXT.

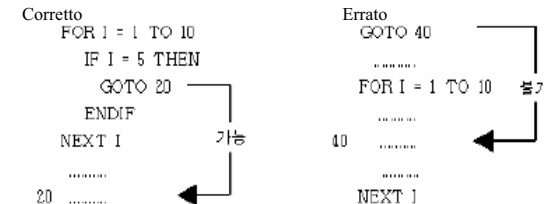
```
FOR I = 1 TO 10
  FOR J = 1 TO 5
  .....
  NEXT J
NEXT I
```

2. Quando si usano numerosa frasi FOR ... NEXT, l'ordine del NEXT [variabile del ciclo] non deve essere cambiato.

```
FOR I = 1 TO 10
  FOR J = 1 TO 5
  .....
  NEXT I
NEXT J
```

Anche se l'ordine della variabile di ciclo è cambiato nell'esempio errato, non si avrà un errore quando si creerà il codice obiettivo, ma una volta caricato sul regolatore del minirobot si otterranno risultati inaspettati.

3. è possibile uscire da dentro una frase FOR ... NEXT ma non è possibile entrare in una frase FOR ... NEXT da fuori.



4. Il valore di una variabile usata nel [variabile di ciclo], [Inizio] e [Fine] non deve essere cambiato arbitrariamente durante l'esecuzione di una frase FOR ... NEXT.

**Esempio di comando**

Connettere L.E.D. alla porta #0 del regolatore e farlo lampeggiare 5 volte.

```
DIM A AS BYTE 'dichiara la variabile da usare nellafrase in ripetizione
```

```
FOR A = 1 TO 5 'il numero di ripetizioni è 5 volte
OUT 0, 0 'spegne il L.E.D. connesso alla porta #0
  DELAY 100 'ritardo di 100
OUT 0, 1 'spegne il L.E.D. connesso alla porta #0
  DELAY 100 'ritardo di 100
```

```
NEXT A
```

**GOTO**

Muove in una posizione specifica

**Struttura della frase**

frase in Inglese: GOTO [etichetta di linea]

**Spiegazione di comando**

Il comando GOTO cambia il flusso del programma saltando ad una specifica linea del codice. Usare il comando GOTO eccessivamente complicherà il programma perciò non usarlo troppo frequentemente.



**Esempio di comando**

```

DIM I AS INTEGER
DIM J AS BYTE

I = 7
IF I = 6 THEN GOTO L1
.....
L1:  J = 1
      OUT I, J

```

**GOSUB ... RETURN**

Richiama un sottoprogramma e ritorno.

**Struttura della frase**

Frase in Inglese: GOSUB [etichetta di linea]

```

[etichetta di linea]: .....
RETURN

```

**Spiegazione di comando**

Il comando GOSUB richiama un sottoprogramma (sub routine) usato frequentemente e poi ritorna. In questo modo GOSUB permette al programma di essere più piccolo e più efficiente.

È possibile chiamare un secondo sottoprogramma da dentro il sottoprogramma originale. Questo è limitato a 4 volte con la serie MR-C2000 e a 5 volte con la serie MR-C3000. Superare il limite causerà errori.

**Esempio di comando**

```

DIM LED_PORT AS INTEGER
LED_PORT = 1
START: .....
.....
GOSUB LED_TOGGLE
.....
GOTO START
END

```

```

LED_TOGGLE:
TOGGLE LED_PORT
RETURN

```

**END**

Conclude l'esecuzione del programma

**Struttura della frase**

frase in Inglese: END

**Spiegazione di comando**

Dopo 2 secondi dall'accensione del regolatore del minirobot, verrà eseguito un programma salvato in EEPROM. Se il comando END non è usato alla fine di un sottoprogramma o di una frase di esecuzione del programma, il programma funzionerà continuamente. Per prevenire ciò inserire sempre il comando END alla fine dei sottoprogrammi o delle frasi di esecuzione.

**Esempio di comando**

- concludere l'esecuzione di un programma dopo averlo eseguito.

```

DIM AS BYTE
START: A = IN (0)
      IF A = 1 THEN END
      .....
      GOTO START

```

2. Può essere creato un sottoprogramma strutturale.

```

DIM A AS BYTE

A = BYTEIN(0)
IF A = 1 THEN
  GOSUB L1
ELSEIF A = 3 THEN
  GOSUB L2
ELSEIF A = 4 THEN
  GOSUB L3
ELSE
  GOSUB L4
ENDIF
.....
END

```

```

L1: .....
    RETURN
L2: .....
    RETURN
L3: .....
    RETURN
L4: .....
    RETURN

```

**STOP/RUN**

Ferma/Lancia l'esecuzione di un programma

**Struttura della frase**

frase in Inglese: STOP/RUN

**Spiegazione di comando**

Questo comando farà fermare o lancerà il programma continuamente. Quando il programma si sarà fermato, usando il comando RUN inizierà di nuovo.

**WAIT**

Aspetta fino a che il programma è finito

**Struttura della frase**

frase in Inglese: WAIT

**Spiegazione di comando**

L'OS di controllo installato nel regolatore del robot ha l'ultimo programma di controllo in tempo reale (REAL-TIME).

Quando un programma sarà eseguito, il programma successivo verrà eseguito allo stesso tempo senza fermare il programma precedente. Se il programma successivo richiede di essere lanciato solo dopo che il programma corrente ha finito, allora viene usato il comando WAIT.

**Esempio di comando**

Es 1:  
uscita dalle porte #7 e #8 dopo aver mosso sei motori.

```

MOVE 120, 100, 140, 90, 70, 150
WAIT
OUT 7, 1
OUT 8, 1

```

Es 2.  
uscita #8 dopo aver fatto funzionare sei motori ma uscita #7 allo stesso tempo.

MOVE 120, 100, 140, 90, 70, 150  
OUT 7, 1  
WAIT  
OUT 8, 1

**DELAY**

Ritarda l'esecuzione del programma per un tempo stabilito.

**Struttura della frase**

frase in Inglese: DELAY [tempo di ritardo]

**Spiegazione di comando**

Questo comando ritarderà l'esecuzione del programma per un tempo stabilito. Il tempo di ritardo per i regolatori della serie MR-C2000 è di 10ms e di 1ms per i regolatori della serie MR-C3000.

Per il [tempo di ritardo] possono essere usate una figura, una costante o una variabile.

**Esempio di comando**

Regolatore serie MR-C2000:

DELAY 10 Ritardo di 100ms. (10ms \* 10 = 100ms = 0.1sec)

Regolatore serie MR-C3000:

DELAY 500 Ritardo di 500ms. (1ms \* 500 = 500ms = 0.5sec)

**BREAK**

Mette in pausa l'esecuzione del programma e cambia in modalità messa a punto (debug).  
Serie MR-C2000.

**Struttura della frase**

frase in Inglese: BREAK

**Spiegazione di comando**

Mette in pausa l'esecuzione del programma e passa alla modalità messa a punto (debug). Quando il programma è in pausa, la condizione di memoria del regolatore del minirobot viene inviata al PC. Assicurarsi che il regolatore e il PC siano connessi tra loro. Altrimenti, il programma s'arresterà e rimarrà in quella condizione. Informazioni più dettagliate sono incluse in "Spiegazione di programma roboBASIC".

Il comando BREAK non funzionerà con i regolatori della serie MR-C3000. Se si sta usando un regolatore della serie MR-C3000 il progress del programma potrà essere seguito sistematicamente con la modalità messa a punto (debug).

**Esempio di comando**

```
.....
BREAK ..... 'Mette in pausa l'esecuzione del programma
.....
```

**ACTION no**

Esegue i movimenti base prescritti di "templet" seguendo il numero del valore.

**Struttura della frase**

frase in Inglese: ACTION [numero]

**Spiegazione di comando**

Esegue un movimento prescritto dal template seguendo il numero del movimento. Sono possibili un massimo di 32 movimenti.

**Esempio di comando**

```
ACTION 3 'esegue il movimento num.3.
ACTION 5 'esegue il movimento num.5.
ACTION 23 'esegue il movimento num.23.
```

**Nota:**

Questo comando riguarda solo il regolatore MR-C3024 e il Robot Robonova-I.

**goto AUTO**

passa al programma di template

**Struttura della frase**

frase in Inglese: goto AUTO

**Spiegazione di comando**

Comando per iniziare l'incluso programma di template.

**Esempio di comando**

```
Goto AUTO 'passa al programma di template
```

**Nota:**

Questo comando riguarda solo il regolatore MR-C3024 e il Robot Robonova-I.

**Capitolo 5****Spiegazione****Del segnale in entrate e uscita****In roboBASIC**

Nel MR-C2000 ci sono 12 porte digitali I/O. E nel MR-C3000 ci sono 40 porte digitali I/O. queste porte esguono diverse funzioni. Vedere la "spiegazione del regolatore" per maggiori informazioni sull'eporte digitali I/O.

**IN()**

segnale digitale dalla porta.

**Struttura della frase**

comando in Inglese: IN(*numero della porta*)

**Spiegazione di comando**

Un valore di segnale in entrata attraverso una porta viene salvato come variabile. I valori entrati sono 1 o 0. Possono

essere usate variabili di tipo byte o numero intero. Attualmente solo l'ultimo valore di bit è disponibile. Il modo più efficiente è quello di usare una variabile di tipo byte.

**Esempio di comando**

```
DIM AS BYTE
A = IN(0) 'legge il segnale (interruttore o sensore) da #0 e lo fissa come variabile A
```

**OUT()**

Invia un segnale digitale ad una porta.

**Struttura della frase**

frase in Inglese: OUT [numero della porta], [valore d'uscita]

**Spiegazione di comando**

Invia un segnale dal regolatore attraverso una porta. Quando si invia un valore 0 (LOW, basso), uscirà un segnale di 0V. Quando si invia un valore 1 (HIGH, alto), uscirà un segnale di +6V. Numeri (0 o 1), costanti e variabili possono essere usati per il [valore d'uscita]. Possono essere usate regolazioni bit per il [valore d'uscita] perché soltanto 0 o 1 sono disponibili nella porta d'uscita.

**Esempio di comando**

Questo esempio è stato creato per testare le porte di entrata/uscita. Un pulsante è connesso alla porta #0 e un LED alla porta #3.

```
DIM AS INTEGER
A = 0 'inizializza la variabile A
START: A = IN(0) 'legge la condizione del pulsante
'A.0 = IN(0) è disponibile
IF A = 1 THEN 'quando il pulsante non è spinto
  OUT 3, 0 'spegne il LED
ELSE 'altrimenti,
  OUT 3, 1 'accende il LED
ENDIF
GOTO START 'controlla il pulsante di nuovo
```

**BYTEIN()**

Legge il segnale dall'unità byte della porta d'entrata

**Struttura della frase**

frase in Inglese: BYTEIN([numero della porta byte])

**Spiegazione di comando**

La porta del regolatore del robot può entrare/uscire come un'unità di un bit come il comando IN/OUT. In certi casi il segnale deve essere in entrata/uscita come un'unità (qui, porta di byte dell'unità di porta #8)

**Regolatore serie MR-C2000:**

porta byte	Porta
0	Porta #0-#7 (porta #0 è l'ordine inferiore della porta byte)
1	Porta #8-#11 (porta #8 è l'ordine inferiore della porta byte)

**Regolatore serie MR-C3000:**

porta byte	Porta
0	Porta #0-#7 (porta #0 è l'ordine inferiore della porta byte)
1	Porta #8-#15 (porta #8 è l'ordine inferiore della porta byte)
2	Porta #16-#23 (porta #16 è l'ordine inferiore della porta byte)
3	Porta #24-#31 (porta #24 è l'ordine inferiore della porta byte)
4	Porta #32-#39 (porta #32 è l'ordine inferiore della porta byte)

Usando il comando BYTEIN, un valore di segnale in entrata attraverso la porta d'entrata byte viene salvato come

una variabile. Il tipo di variabile deve essere dichiarato come un byte o come numero intero.

**Esempio di comando**

```
A = BYTEIN(0) 'tutti i segnali dalle porte #0-#7 sono immessi come variabile A
```

**BYTEOUT**

invia un segnale d'uscita ad una porta come un'unità di byte

**Struttura della frase**

frase in Inglese: BYTEOUT [numero della porta byte], [valore d'uscita]

**Spiegazione di comando**

Fa uscire i valori di segnale attraverso la porta di unità byte. Numeri, costanti o variabili possono essere usati per il [numero di porta byte]. Per il [valore d'uscita] possono essere usati numeri tra 0-255, costanti o variabili di byte.

**Esempio di comando**

```
BYTEOUT 0, 255 'invia il valore(5v) 1 alle porte #0-#7
BYTEOUT 0, &h10101010
'invia il valore(5v) 1 alle porte #1, #3, #5, #7
'invia il valore(0v) 0 alle porte #0, #2, #4, #6
```

**INKEY**

Inserisce il valore chiave attraverso la porta d'entrata **-2000-**

**Struttura della frase**

frase in Inglese: INKEY ([numero della porta])

**Spiegazione di comando**

Quando un interruttore viene spinto una volta, è in realtà spinto elettricamente e meccanicamente centinaia o migliaia di volte. Questo fenomeno viene chiamato "chattering" (vibrazione, battimento). Il chattering può causare errori perciò è necessario un circuito di protezione aggiunto. Nel regolatore del robot è inserita nel software una funzione di protezione dal chattering. Per utilizzare questo software, usare il comando INKEY.

**Esempio di comando**

Salva lo stato di schiacciamento di un pulsante connesso alla porta #0 come variabile A.

```
DIM A AS BYTE
```

```
A = INKEY(0)
```

**STATE()**

Legge il valore attuale della porta d'uscita

**Struttura della frase**

frase in Inglese: STATE ([numero della porta])

**Spiegazione di comando**

Se viene richiesto il valore dello stato di una porta d'uscita dopo aver inviato un segnale attraverso la porta, si usi la funzione STATE. Non usare la funzione IN.

**Esempio di comando**

Questo è un programma campione per testare lo stato d'uscita della porta #1.

```
DIM A AS BYTE
```

```
OUT 1, 1
A = STATE(1) 'A = 1
OUT 1, 0
```



A = STATE(1) 'A = 0

### PULSE

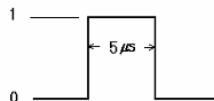
Invia un segnale pulsante ad una porta d'uscita.

#### Struttura della frase

frase in Inglese: PULSE [numero della porta]

#### Spiegazione di comando

Invia un segnale pulsante ad una porta d'uscita per 5µs. Il segnale pulsante viene usato per fornire un segnale ad una periferica esterna.



Per il [numero della porta] possono essere usati numeri, costanti o variabili.

#### Esempio di comando

PULSE 3 'Invia un segnale pulsante alla porta #3.

### TOGGLE

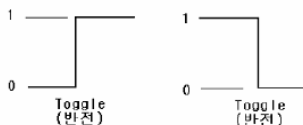
Inverte il segnale della porta d'uscita.

#### Struttura della frase

frase in Inglese: TOGGLE [numero della porta]

#### Spiegazione di comando

Inverte il segnale d'uscita di una porta d'uscita. Se il segnale della porta è 0 (low, basso), il segnale sarà invertito in 1 (HIGH, alto).



Per il [numero della porta] possono essere usati numeri, costanti e variabili.

#### Esempio di comando

OUT 3, 1 'Invia il segnale "1" alla porta #3.  
TOGGLE 3 'Inverte il segnale della porta #3.

### KEYIN()

Inserisce numerose chiavi (tastierino numerico) -3000-

#### Struttura della frase

frase in Inglese: KEYIN ([numero della porta analogica], [il numero delle chiavi])

#### Spiegazione di comando

Questo comando legge i valori di 16 pulsanti attraverso le porte di conversine AD (porte analogiche) nel regolatore della serie MR-C3000.

Possono essere usati per la [porta analogica] numeri (0-7), costanti e variabili byte.

Numeri (1-16), costanti e variabili byte possono anche essere usati per [il numero della chiave].

La gamma di valore di un pulsante è da 0 a 16. 0 significa che la chiave non è spinta. I numeri da 1 a 16 significano che la chiave è spinta.

#### Esempio di comando

DIM K AS BYTE

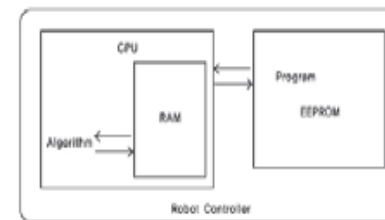
K = KEYIN(0, 16) 'Inserisce i valori delle 16 chiavi connesse alla porta AD #0 come K

## Capitolo 6

### Spiegazione dei comandi

#### Relativi alla memoria

Il regolatore del robot ha una CPU e una memoria, per questo può essere descritto come un microcomputer. L'esecuzione della memoria ha un ruolo importante nei programmi di salvataggio e di calcolo. La memoria esterna per il regolatore del robot, nella forma di EEPROM, viene usata per i programmi applicativi di salvataggio. La memoria per i calcoli esecutivi è situata nella CPU.



La memoria interna è chiamata RAM ma anche registro per via delle speciali caratteristiche della CPU del regolatore del robot. La memoria interna è legata al numero di variabili utilizzabili nel momento in cui viene creato un programma. I regolatori della serie MR-C2000 hanno 30bytes di spazio variabile. I regolatori della serie MR-C3000 hanno 256bytes di spazio variabile. Gli altri settori di memoria sono delegati all'utilizzo interno del regolatore del robot.

La memoria esterna è legata alla dimensione del programma creato. La serie MR-C2000 ha 4kbytes e la serie MR-C3000 ha 12k, 32k, 64k bytes di memoria a seconda del modello.

### PEEK()

Legge il contenuto della memoria interna.

#### Struttura della frase

frase in Inglese: PEEK ([sette della RAM])

**Spiegazione di comando**

La funzione PEEK richiama i dati dalla memoria interna. Non usare questa funzione se non si conosce l'esatta struttura della memoria interna.

**Regolatore di serie MR-C2000:**

possono essere usati numeri tra 0-255, costanti o variabili byte.

**Regolatore di serie MR-C3000:**

possono essere usati numeri tra 0-65535, costanti o variabili byte.

**Esempio di comando**

DIM A AS BYTE

A = PEEK(43)                    'porta il valore del settore di RAM di indirizzo 43 alla variabile A

**POKE**

Scriva i dati nella memoria interna.

**Struttura della frase**

frase in Inglese: POKE [settore di RAM], [dati]

**Spiegazione di comando**

Il comando POKE può essere usato per scrivere i dati nella memoria interna. Nell'MR-C2000 per il [settore di RAM] possono essere usati numeri tra 0-255, costanti o variabili byte. Nell'MR-C3000 per il [settore di RAM] possono essere usati numeri tra 0-65535, costanti o variabili byte.

Per i [dati] possono essere usati numeri, costanti o variabili (variabili numero intero).

**Esempio di comando**

POKE &h40, 100                    'Scriva 100 nel settore di RAM di indirizzo 40.

**ROMPEEK()**

Legge i dati dall'esterna EEPROM.

**Struttura della frase**

frase in Inglese: ROMPEEK ([settore di ROM])

**Spiegazione di comando**

Il regolatore del robot usa una EEPROM per salvare i programmi o altri oggetti. Le funzioni ROMPEEK o ROMPOKE che controllano la memoria esterna possono essere usate per salvare dei dati. Se il settore contiene già dei dati, potrebbe accadere un errore fatale. Per il [settore di ROM] possono essere usati numeri, costanti o variabili.

**ROMPOKE**

Scriva i dati nella EEPROM esterna.

**Struttura della frase**

frase in Inglese: ROMPOKE [settore di ROM], [dati]

**Spiegazione di comando**

Il regolatore del robot usa una EEPROM per salvare i programmi o altri oggetti. Le funzioni ROMPEEK o ROMPOKE che controllano la memoria esterna possono essere usate per salvare dei dati. Se il settore contiene già dei dati, potrebbe accadere un errore fatale. Per il [settore di ROM] possono essere usati numeri, costanti o variabili. Per i [dati] possono essere usati numeri tra 0-255, costanti o variabili byte.

**Capitolo 7****Controllo del modulo LCD****In roboBASIC**

Il modulo LCD progettato per l'uso con il regolatore del robot è l'MR-16202. Connettere il modulo LCD con la porta #6 del regolatore della serie MR-C2000. L'MR-C3000 ha una porta LCD specifica. Qui verranno spiegati i comandi per il controllo del modulo LCD e per visualizzare le stringhe di caratteri.



MR-16202 LCD-Modul

**LCDINIT**

Inizializza il modulo LCD.

**Struttura della frase**

frase in Inglese: LCDINIT

**Spiegazione di comando**

Il modulo LCD deve essere inizializzato usando il comando LCDINIT in maniera tale da prevenire la visualizzazione di caratteri inaspettati. Quando il modulo LCD è inizializzato, tutti i caratteri saranno cancellati e il cursore sarà posizionato nell'angolo in alto a sinistra.

**Esempio di comando**

LCDINIT                    'Inizializza il modulo LCD.

**CLS**

Cancella i caratteri dal modulo LCD.

**Struttura della frase**

frase in Inglese: CLS

**Spiegazione di comando**

Usare il comando CLS per cancellare tutti i caratteri visualizzati nel modulo LCD. Quando il comando CLS viene eseguito, tutti i caratteri saranno cancellati e il cursore sarà posizionato nell'angolo in alto a sinistra. Ci sono discrepanze tra LCDINIT e CLS. Con il comando CLS saranno cancellati solo i caratteri, ma con il comando LCDINIT saranno cancellate tutte le informazioni, come le variabili interne.

**Esempio di comando**

CLS 'Cancella quello che viene visualizzato nel modulo LCD

**LOCATE**

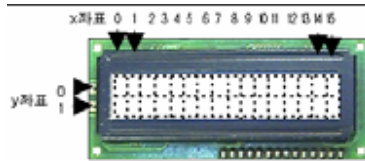
Indica la posizione sul display di un carattere nel modulo LCD.

**Struttura della frase**

frase in Inglese: LOCATE [coordinata x], [coordinata y]

**Spiegazione di comando**

Con il comando LOCATE si puntano le coordinate x e y nel modulo LCD. Le coordinate di un modulo LDC 16x2 sono fissate come nella figura sottostante. Possono essere usati per le coordinate x e y numeri, costanti e variabili, ma il tutto deve cominciare con 0.



**Esempio di comando**

LOCATE 0, 0 'Posiziona il cursore nell'angolo in alto a sinistra del modulo LCD.  
 LOCATE 4, 1 'Posiziona il cursore nelle coordinate (4, 1) del modulo LCD.

**PRINT**

Esponde i caratteri nel modulo LCD.

**Struttura della frase**

frase in Inglese: PRINT "[stringa di carattere]", [numero]/ "[stringa di carattere]", ...

**Spiegazione di comando**

Usare il comando PRINT per esporre un carattere nell'attuale posizione del cursore. La [stringa di carattere] può essere distinta con le doppie virgolette (" "). La gamma di [numero] è tra 1-255 (lo 0 non può essere usato). Nel modulo LCD sarà visualizzato il carattere applicabile ai numeri di codice ASCII.

**Esempio di comando**

CLS  
 PRINT "miniROBOT", 126, „LCD“



I seguenti sono esempi sono di codice ASCII per un modulo LCD di linea 16x2. Il codice di carattere dipende dal tipo di modulo LCD in uso.

Line	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xxxx0001	!	@	#	\$	%	&	'	(	)	*	+	,	-	.	/	:
xxxx0010	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	:	;
xxxx0011	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K
xxxx0100	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xxxx0101	!	@	#	\$	%	&	'	(	)	*	+	,	-	.	/	:
xxxx0110	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	:	;
xxxx0111	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K
xxxx1000	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xxxx1001	!	@	#	\$	%	&	'	(	)	*	+	,	-	.	/	:
xxxx1010	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	:	;
xxxx1011	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K
xxxx1100	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
xxxx1101	!	@	#	\$	%	&	'	(	)	*	+	,	-	.	/	:
xxxx1110	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	:	;
xxxx1111	<	=	>	?	@	A	B	C	D	E	F	G	H	I	J	K

**FORMAT()**

**DEC()**

**HEX()**

**BIN()**

Specifica il tipo di numero da visualizzare sul modulo LCD

**Struttura della frase**

frase di comando: FORMAT ([variabile], [tipo di uscita], [posizione del punto])

**Spiegazione di comando**

Il modulo LCD segue un formato specifico quando produce una variabile. Il comando FORMAT deve essere posizionato dopo il comando PRINT.

[variable type][output type]	Basic position of point	The number expression
Byte type /decimal	3	0 ~ 255
Byte type /hexadecimal	2	00 ~ FF
Byte type /binary	8	00000000 ~ 11111111
Integer type /decimal	5	0 ~ 65535
Integer type /hexadecimal	4	0000 ~ FFFF
Integer type /binary	16	0000000000000000 ~ 1111111111111111

**Esempio di comando**

```

DIM A AS BYTE
DIM B AS INTEGER

LCDINIT
A = 100
B = 20000
LOCATE 0, 0
PRINT FORMAT(A, DEC, 4)
LOCATE 0, 1
PRINT FORMAT(B, HEX)

```

**CSON()**  
**Csoff()**

Mostra/nasconde il cursore sul modulo LCD.

**Struttura della frase**

frase in Inglese: CSON / Csoff

**Spiegazione di comando**

Il comando CSON/Csoff mostrerà o nasconderà il cursore sul modulo LCD. In generale, quando il modulo LCD è inizializzato il cursore è nascosto.

**Esempio di comando**

```

LCDINIT
CSON
PRINT "CURSOR ON"

```



커서(Cursor)

**CONT**

Regola il contrasto del modulo LCD.

**Struttura della frase**

frase in Inglese: CONT [valore del contrasto]

**Spiegazione di comando**

Il modulo LCD è retroilluminato. I caratteri sono visualizzati con il colore nero. Con il comando CONT può essere regolata l'intensità del colore. Per il [valore del contrasto] possono essere usati numeri, costanti e variabili. Come il [valore del contrasto] aumenterà, così il carattere diventerà più spesso. Il valore iniziale è 7.

**Esempio di comando**

```

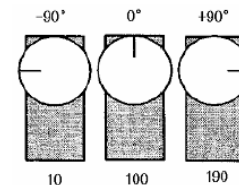
LCDINIT

CONT 10
PRINT "robonova"

```

**Capitolo 8****Spiegazione dei Comandi****di controllo del motore****in ROBOBASIC**

Il regolatore del robot può controllare servo-motori e motori DC. Nel caso dei motori DC, il regolatore può controllare la velocità, la direzione, e fermare il motore usando comandi digitali di entrata e uscita. La gamma di spostamento dei servo-motori è da  $-90^\circ$  a  $+90^\circ$ . Per far funzionare i servo-motori in roboBASIC i gradi sono espressi da numeri tra 10 e 190, dato che il regolatore del robot non usa i numeri negativi.



Vari servo-motori:

**ZERO**

Fissa il punto zero del servo-motore.

**Struttura della frase****Regolatore di serie MR-C2000:**

frase in Inglese: ZERO [punto 0 standard del motore], [punto 1 standard del motore], ..., [punto 5 standard del motore]

**Regolatore di serie MR-C3000:**

frase in Inglese: ZERO [fissaggio di punto di gruppo], [punto n standard del motore], ...

**Spiegazione di comando**

Il punto zero dei servo-motori dipende da ogni singolo servo. Questo a causa di deviazioni del prodotto. Certi punti zero possono essere 99 o 98, altri punti zero possono essere 101 o 102. Questi tipi di errori possono essere rimediati con il comando ZERO. Una volta che il punto zero di ogni servo viene fissato, sarà il punto standard del comando MOVE.

Il punto zero fissato verrà salvato nella EEPROM per evitare che sia cancellato dallo spegnimento.

**Fissare il punto zero nella serie MR-C2000:**

- 1) fissare il punto zero per tutti i servo-motori come 100 per cancellare i vecchi dati.
- 2) fissare la direzione del motore nella direzione normale.



- 3) spegnere e riaccendere.
- 4) portare il motore sul punto zero (centro) usando la funzione in linea.
- 5) salvare questa posizione come il punto zero utilizzando il comando di punto zero.

**ESI:**

eliminare il vecchio valore di punto zero

ZERO 100, 100, 100, 100, 100, 100

DIR 1,1,1,1,1,1 'fissa la direzione del motore nella posizione normale

**ES2:**

fissare un nuovo punto zero. Salvare il punto zero (centro)

DIR 1,1,1,1,1,1

ZERO 100,101,99 'fissa il punto zero del servo- motore a 3

ZERO 102, 100, 100, 99, 101, 100 'fissa il punto zero del servo- motore a 6

Nel regolatore della serie MR-C2000 il settaggio del grado di punto zero deve essere dentro 90-100.

**Fissare il punto zero nella serie MR-C3000:**

Quando si fissano i punti zero nel regolatore MR-C3000 si deve dichiarare un gruppo.

ZERO G8B, 80, 120, 115, 80, 117, 88, 95, 120 'fissa il punto zero del Gruppo8B  
(servo- motori #8-#15)

Nel regolatore della serie MR-C2000 il settaggio del grado di punto zero deve essere dentro 80-120.

**MOTOR**

Fissa il servo-motore da usare.

**Struttura della frase****Regolatore di serie MR-C2000:**

frase in Inglese: MOTOR [numero del motore]

**Regolatore di serie MR-C3000:**

frase in Inglese: MOTOR [numero del motore] / [gruppo da specificare]

**Spiegazione di comando****Fissare il servo-motore nella serie MR-C2000:**

Nei regolatori della serie MR-C2000 ci sono sei porte (#0-#5) per i servo-motori. I numeri dei motori che possono essere specificati sono 0-5. Quando si vogliono usare tutti i servo-motori, si fissa il numero di motore a 6.

Se un numero non è stabilito per il servo-motore, il servo-motore non funzionerà assolutamente. Nel [numero del motore] possono essere usati solo numeri tra 0-6.

Es1: MOTOR 6 'saranno usati tutti (#0-#5) i motori

Es2: MOTOR 2 'sarà usato il motore #2

**Fissare un servo-motore nella serie MR-C3000:**

Nel regolatore della serie MR-C3000 ci sono 32 porte per 32 servo-motori. Ogni motore può essere assegnato ad un [numero di motore]. Gruppi di motori possono essere assegnati a [fissaggio di punto di gruppo].

Per il [numero di motore] possono essere usati numeri, costanti, e variabili byte.

**Esempio di comando**

Es1: MOTOR 0 'sarà usato il servo-motore #0

Es2: MOTOR G6A 'sarà usato il gruppo di servo-motori 6A (#0-#5)

- MOTOR G6C 'sarà usato il gruppo di servo-motori 6C (#12-#17)
- Es3: MOTOR G8A 'sarà usato il gruppo di servo-motori 8A (#0-#7)
- Es4: fissare il servo-motore con una variabile  
DIM I AS BYTE  
FOR I = 0 TO 31 'saranno usati i servo-motori (#0-#31) utilizzando la variabile I'  
MOTOR I  
NEXT I
- Es5: MOTOR G24 'sarà usato il gruppo di servo-motori 24 (#0-#23)
- Es6: MOTOR ALLON 'saranno usati tutti i servo-motori

**MOTOROFF**

Spegne un servo-motore.

**Struttura della frase****Regolatore di serie MR-C2000:**

frase in Inglese: MOTOROFF [numero del motore]

**Regolatore di serie MR-C3000:**

frase in Inglese: MOTOROFF [numero del motore] / [fissaggio di punto di gruppo]

**Spiegazione di comando**

Il comando MOTOROFF è lo stesso del comando MOTORON.

**MOVE**

Fa funzionare diversi servo-motori allo stesso tempo.

**Struttura della frase****Regolatore di serie MR-C2000:**

frase in Inglese: MOVE [angolo0del motore], [angolo1 del motore], ..., [angolo5del motore]

**Regolatore di serie MR-C3000:**

frase in Inglese: MOVE [fissaggio di punto di gruppo], [angolo n del motore]

**Spiegazione di comando****Comando di movimento (move) con il regolatore di serie MR-C2000:**

Il comando MOVE fa funzionare un servo motore nello specifico angolo desiderato. Mentre lo si usa, la funzione PWM è disabilitata. La gamma dell' [angolo di motore] è tra 10-190.

Quando si vorrà fissare i servo-motori #1, #3 e #4, la frase sarà come questa:  
MOVE 60, 100, 120

Quando si vorrà fissare solo il motore #2, sarà come in questo esempio.  
MOVE, 140

Questo processo può risultare molto complicato, specialmente quando si devono far funzionare 6 motori allo stesso tempo. Questo processo sarebbe più semplice se fosse usato il "controllo istantaneo del servo" (servo real time controlling).

Se viene usato un motore DC, 100 significa 'ferma', 190 significa massima velocità con rotazione al contrario e 10 significa massima velocità con normale rotazione.

Se il motore deve essere usato dopo l'operazione precedente, si usi il comando WAIT.

**Esempio di comando**

MOVE 100, 50, 140, 120, 80, 40

MOVE 120, , , 160

MOVE , 70, 100

MOVE , , , , 100

**Comando di movimento nel regolatore di serie MR-C3000:**

Nel regolatore di serie MR-C3000 le porte per i servo-motori sono diverse da quelle per PWM. Perciò il comando di movimento e il comando PWM possono essere usati allo stesso tempo.

**Esempio di comando**

EX 1)

```
MOVE G6A, 85, 113, 72, 117, 115, 100
MOVE G6C, 75, , 96, 123, , 122
MOVE G8A, 85, 113, 72, 117, 115, 100, 95, 45
```

EX 2)

```
MOVE G24, 85, 113, 72, 117, 115, 100
```

Is the same as:

```
MOVE24 85, 113, 72, 117, 115, 100
```

**SPEED**

Stabilisce la velocità di un servo-motore

**Struttura della frase**

frase in Inglese: SPEED [velocità del motore]

**Spiegazione di comando**

Il comando SPEED stabilisce la velocità di un servo-motore fatto funzionare dal comando MOVE.

Nel regolatore di serie MR-C2000 per la [velocità del motore] possono essere usati numeri o costanti tra 1-15.

Nel regolatore di serie MR-C3000 possono essere usate variabili byte.

Il settaggio di norma è 3. Una velocità troppo elevata è pericolosa tanto per il robot quanto per l'utilizzatore.

**Esempio di comando**

Es 1:

```
SPEED 7          'fissa la velocità del motore come 7
```

Es 2:

```
DIM STEP_SPEED AS BYTE      'dichiara STEP_SPEED (variable)
STEP_SPEED = 15             'fissa la STEP_SPEED (variabile) come 15
SPEED STEP_SPEED            'fissa la STEP_SPEED come STEP_SPEED
```

**ACCEL**

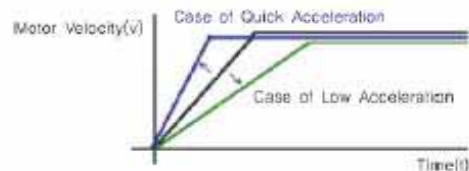
Stabilisce l'accelerazione di un servo-motore.

**Struttura della frase**

frase in Inglese: ACCEL [accelerazione del motore]

**Spiegazione di comando**

Il comando ACCEL stabilisce il tasso d'accelerazione del servo-motore da 0 alla velocità fissata.



L' [accelerazione del motore] utilizza numeri o costanti tra 0 e 15, il valore di norma è 3. Un numero più grande aumenta l'accelerazione del servo-motore.

Quando si fa funzionare un servo-motore per la prima volta, il servo ruota fino al punto fissato rapidamente. Per

diminuire questa velocità iniziale è raccomandabile usare i comandi ACCEL e SPEED. Nel regolatore di serie MR-C3000 il comando ACCEL non può essere usato.

**Esempio di comando**

```
ACCEL 7          'fissa l'accelerazione del servo-motore come 7
```

**DIR**

Fissa la direzione di rotazione di un servo-motore.

**Struttura della frase****Regolatore di serie MR-C2000:**

frase in Inglese: DIR [direzione 0 del motore], [direzione 1 del motore], ..., [direzione 5 del motore]

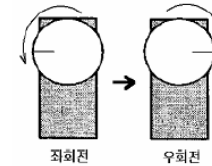
**Regolatore di serie MR-C3000:**

frase in Inglese: DIR [fissaggio di punto di gruppo], [direzione n del motore], ...

**Spiegazione di comando**

Un servo-motore girerà a sinistra quando l'angolo fissato è più piccolo di 100 (angolo standard) e girerà a destra quando l'angolo fissato è più grande di 100 (angolo standard).

Nella figura in basso la rotazione del servo-motore è a sinistra (direzione normale) di 10 gradi.



La [direzione del motore] usa costanti o numeri come 0 (rotazione al contrario/gira a sinistra) o 1 (rotazione normale/gira a destra). Il valore di base è 0. Per esempio, quando vengono usati 4 servo-motori, se il servo #3 viene omesso (come "DIR 0, 1, , 0"), quello girerà nella direzione standard.

**Esempio di comando****Esempio per il regolatore di serie MR-C2000:**

```
DIR 0, 1, 1, 0, 1, 0
```

```
DIR , , 0
```

**Esempio per il regolatore di serie MR-C3000:**

```
DIR G8A, 0, 1, 0, 0, 1, 0, 0, 0
```

```
DIR G8B, 1, 0, 1, 1, 0, 1, 1, 1
```

**PTP**

Fissa la funzione On/Off per il controllo simultaneo dei servo-motori

**Point to Point****Struttura della frase****Regolatore di serie MR-C2000:**

frase in Inglese: PTP [fissa il valore]

**Regolatore di serie MR-C3000:**

frase in Inglese: PTP [SETON/SETOFF/ALLON/ALLOFF]

**Spiegazione di comando**

Nel caso di molteplici movimenti e movimenti ad angoli differenti, i tempi di chiusura dei servo-motori sono diversi l'uno dall'altro. Quindi, nel caso di un Braccio Robot o di un altro robot servo-funzionante, i movimenti potrebbero essere instabili.

Nell'Ingegneria Robotica c'è una teoria chiamata "Point-to-Point" che può calcolare il tempo di chiusura dei servo-motori e finisce tutti i movimenti simultaneamente permettendo un'esecuzione più morbida.

I regolatori delle serie MR-C usano questo metodo di controllo Point-to-Point usando il comando PTP.

#### Controllo PTP nell'MR-C2000

Il [valore di settaggio] usa 0 (annulla) o 1 (fissa) (numero o costante) quando vengono usati due servo-motori (n°1 e n°2). Vedere l'esempio sotto:

- esempio di moto teatrale:

```
PTP 0
MOVE 100, 100
MOVE 110, 120
```

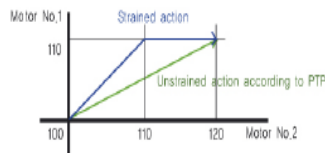
Descrizione: il motore n°1 si muove di 10 gradi e il n°2 di 20 gradi. Viaggiando entrambi alla stessa velocità si muoveranno di 10 gradi insieme, e poi il n°2 si sposterà dei rimanenti 10 gradi da solo.

- moto morbido con la funzione PTP

```
PTP 0
MOVE 100, 100
MOVE 110, 120
```

Descrizione: il motore n°1 si muove di 10 gradi e il n°2 di 20 gradi, ma il n°1 si muove a metà della velocità del n°2. Entrambi i servo-motori si muovono e si fermano nello stesso tempo.

Si guardi il grafico sottostante per una comparazione tra l'utilizzo del comando "PTP" (freccia verde) e no (freccia blu).



Si noti che nel moto normale il motore n°1 e il n°2 si muovono fino all'angolo 110 simultaneamente, ma poi solo il n°2 si muove in seguito fino al 120. Tuttavia, usando il comando "PTP", con il calcolo tra l'aspettato movimento all'angolo 10 del n°1 e al 20 del n°2, si compie un morbido finale di movimento.

#### Controllo PTP nel regolatore di serie MR-C3000:

con i regolatori delle serie MR-C3000 si possono usare molteplici servo-motori. La funzione di PTP può essere adeguata al controllo di tutti i servo-motori o di gruppi individuali-

PTP SETON (fissaggio del PTP): funzione di fissaggio del PTP come gruppi.  
 PTP SETOFF (annulla il PTP): funzione di annullamento del PTP come gruppi.  
 PTP ALLON (fissaggio del PTP su tutti): funzione di fissaggio del PTP per tutti i servo-motori.  
 PTP ALLOFF (annulla il PTP su tutti): funzione di annullamento del PTP per tutti i servo-motori.

Nel regolatore di serie MR-C3000, usando il comando "WAIT" alla fine di un movimento per ogni gruppo, tutti i servo-motori di quel gruppo finiranno i movimenti nello stesso tempo.

#### SERVO

Viene fatto funzionare un servo-motore

#### Struttura della frase

frase in Inglese: SERVO [n° del motore], [angolo del motore]

#### Spiegazione di comando

Questo comando fissa l'angolo del motore desiderato. Nel caso del regolatore di serie MR-C2000, la funzione PWM

è cancellata. Il [n° del motore] è il n° della porta del motore sul regolatore. L'[angolo del motore] è compreso tra 10 e 190, ed è possibile usare numeri, costanti o variabili byte.

#### Esempio di comando

Es1: SERVO 1, 130 'fa funzionare il motore n°1 alla posizione 130

Es2: DIM I AS BYTE

```
FOR I = 10 TO 190
SERVO 4, I
DELAY 100
NEXT I
```

#### PWM

Controllo dell'Ampiezza di Pulsazione (Pulse Width Control)

#### dell'Ampiezza di Pulsazione

#### Struttura della frase

frase in Inglese: PWM [n° del motore], [valore d'ampiezza di pulsazione]

#### Spiegazione di comando

##### Controllo del PWM nell'MR-C2000:

Nei regolatori di serie MR-C2000 le porte di controllo dei servo-motori e le porte PWM sono usate in congiunzione. Quindi, il [n° del motore] è da 0 a 5.

Se il comando PWM è impostato con un valore d'ampiezza di pulsazione la funzione servo viene annullata.

(Attenzione: in un programma i comandi SERVO o MOVE e il comando PWM non possono essere usati insieme.)

Tasso di utilizzo	Valore ampiezza pulsazione	Tasso di utilizzo	Valore ampiezza di pulsazione
0	0	60	153
10	25	70	178
20	51	80	204
30	77	90	229
40	102	100	254
50	127		

PWM 3, 127 'PWM imposta al motore n°3 una gamma di utilizzo del 50%

#### Controllo del PWM nell'MR-C3000:

Il controllo del servo-motore e il PWM non usano le stesse porte. Ci sono 3 porte PWM (n°0-n°3) installate nei regolatori di serie MR-C3000 (454Hz di frequenza PWM nei serie MR-C3000).

PWM 0, 120 'Uscita della pulsazione di 120 di gamma di utilizzo alla porta PWM n°0

#### FASTSERVO

Fa funzionare un servo-motore ad una velocità maggiore **-2000-**

#### Struttura della frase

frase in Inglese: FASTSERVO [n° del motore], [angolo del motore]

#### Spiegazione di comando

Questo comando accelera un servo specifico all'angolo desiderato il più velocemente possibile.

Soltanto i regolatori di serie MR-C2000 usano questo comando.

Il [n° del motore] è la porta del servo-motore e l'[angolo del motore] è l'angolo desiderato. Per l'[angolo del motore] possono essere usati numeri e costanti compresi tra 10 e 190.

#### Esempio di comando

FASTSERVO 2, 190 'invia il motore n°2 ad un angolo di 190 il più velocemente possibile

**HIGHSPEED**

Fissa un servo-motore in modalità alta velocità **-3000-**

**Struttura della frase**

frase in Inglese: HIGHSPEED [*SETON/SETOFF*]

**Spiegazione di comando**

Questo comando fissa o annulla la modalità alta-velocità di un servo-motore nei regolatori di serie MR-C3000. L'alta velocità è circa 3 volte più veloce della velocità normale.

HIGHSPEED SETON: imposta la modalità alta-velocità nei regolatori di serie MR-C3000.  
HIGHSPEED SETOFF: annulla la modalità alta-velocità nei regolatori di serie MR-C3000.  
(Ritorna alla modalità di velocità normale).

**Esempio di comando**

HIGHSPEED SETON                    'imposta la modalità alta velocità

**MOVEPOS****POS**

Stabilisce la posizione di movimento o la posizione del motore **-3000-**

**Move position  
Motor position**

**Struttura della frase**

frase in Inglese: MOVEPOS [etichetta di linea]

.....  
[etichetta di linea]: POS [nomina del gruppo], [angolo n del motore]

.....  
[etichetta di linea]:

.....  
[etichetta di linea]:

**Spiegazione di comando**

Quando la posizione del robot (consistente nel comando 'move') nel regolatore di serie MR-C3000 è portata da altri comandi, viene maneggiata con il comando 'POS' (posizione del motore) e l'[etichetta di linea] con il comando 'MOVEPOS'. Con i comandi 'MOVEPOS' e 'POS', si può facilmente modificare e scrivere il programma roboBasic.

**Esempio di comando**

.....  
MOVEPOS POS01                    '*comando di movimento 'POS' parte della posizione  
d'etichetta 'POS01'*

.....  
POS01: POS G6A, 10, 32, 15, 120, 78, 93  
POS02: POS G6A, 67, 47, 32, 153, 23, 33  
POS03: POS G6A, 34, 37, 122, 162, 84, 28

**FPWM**

Fa uscire un segnale PWM (la frequenza può essere variabile) **-3000-**

**Frequency pulse**

**Struttura della frase**

frase in Inglese: FPWM [*port*], [*duty rate*]

**Spiegazione di comando**

La frequenza del PWM è cambiata e la pulsazione è uscita nei regolatori di serie MR-C3000.  
Porta: Da 0 a 2  
Frequenza: da 1 (bassa frequenza)- 5 (alta frequenza)  
Tasso di utilizzo: da 0 a 255

**Esempio di comando**

FPWM 0, 1, 127                    'fa uscire il segnale PWM (al 50% di tasso di utilizzo (127) e a bassa frequenza nella porta

PWM n°0 del regolatore di serie MR-C3000)

**MOVE24**

Muove tutti i 24 servo-motori **-3000-**

**Struttura della frase**

frase in Inglese: MOVE24 [angolo 0 del motore], ..., [angolo 23 del motore]

**Spiegazione di comando**

Con i regolatori di serie MR-C3000 si possono far funzionare gruppi di servo-motori con il comando "MOVE". I comandi "MOVE24" e "MOVE G24" sono usati per far funzionare 24 servo-motori simultaneamente. Questo comando è utile per fare andare un robot da 16 a 24 servo-motori.

**Esempio di comando**

MOVE24 100, 45, 67, 44, 132, 122, , , , 76, 81, 90

**INIT**

Fissa la posizione iniziale del robot **-3000-**

**Iniziale**

**Struttura della frase**

frase in Inglese: INIT [nomina del Gruppo], [angolo n del motore], ...

**Spiegazione di comando**

Quando il regolatore di serie MR-C3000 viene installato in un robot, tutti i servo-motori sono fissati in una posizione iniziale di "100" durante l'accensione. E' possibile che si incappi in un danno per il robot.

Per prevenire il danno, la posizione iniziale d'accensione può essere fissata ad una posizione diversa da "100".

Usare il comando "INIT" nel caso di un servo-motore analogico (serie-HS).

Usare il comando "GETMOTORSET" nel caso di un servo-motore digitale (serie-HSR).

**Esempio di comando**

INIT G8A, 100, 45, 67, 44, 132, 122, 76, 81

**MOTORIN()**

Legge l'attuale tasso del servo-motore. **-3000-**

**Dati in entrata del Motore ()**

**Struttura della frase**

frase in Inglese: MOTORIN ([n° del motore])

**Spiegazione di comando**

Con questo comando è possibile leggere la posizione attuale di ogni servo robot (serie-HSR) connesso al regolatore di serie MR-C3000 e controllarlo.

Per il [n° del motore] è possibile usare i numeri da 0 a 31 come costante.

Connesso al regolatore: può essere letto un angolo di motore da 0 a 190.

Non connesso: angolo del motore 0.

**Esempio di comando**

DIM S0 AS BYTE

MOTOR 0                    'usa il servo motore n°0  
S0 = MOTORIN(0)                    'salva il valore del servo-motore n°0 nella variabile S0



**AIMOTOR****Motore AI pronto**

Usa il motore AI

**Struttura della frase**

frase inInglese: AIMOTOR SETON/SETOFF/[n° del motore] / [nommina del gruppo]

**Spiegazione di comando**

Il motore AI è prodotto dalla Megarobotics. Un microchip di controllo è installato nel motore AI che può comunicare con i regolatori di serie MR-C tramite RS232. il motore AI è controllato dai regolatori di serie MR-C come un normale servo-motore.

- controllare l'angolo del motore
- presentare lo stato del motore e controllare lo stato di torsione tramite il controllo PDI (PGAIN, DGAIN)

I motori AI possono connettersi alle porte da n°0 a n°30 (31 porte totali).

[n° del motore]: nomina ogni motore,

[nomina di gruppo]: nomina i motori in gruppo

Il processo del comando è simile al comando "MOTOR". È possibile usare per il [n° del motore] un numero, una costante o una variabile byte.

Quando si usa un motore AI, è essenziale dichiarare "uso motore AI", ma non è richiesto per gli altri servo-motori.

- AIMOTOR SETON: approntamento usando un motore AI.
- AIMOTOR SETOFF: annullamento usando un motore AI.
- AIMOTOR INIT: sposta uniformemente il motore AI nella sizione iniziale

**Esempio di comando**

AIMOTOR INIT 'Inizializza il motore AI  
AIMOTOR SETON 'dichiara l'utilizzo del motore AI  
AIMOTOR 0 'utilizza il motore AI n°0  
AIMOTOR G6B 'usa il gruppo di motore 6B (n°6-11)

**AIMOTOROFF**

Annulla il motore AI

**-3000-**

**Annullare il motore AI****Struttura della frase**

frase in Inglese: AIMOTOROFF [n° del motore] / [nomina di gruppo]

**Spiegazione di comando**

Annullamento del comando "AIMOTOR".

La stessa funzione di comando di "MOTOROFF".

**Esempio di comando**

AIMOTOROFF 'annulla il motore AI n°0.  
AIMOTOROFF G6B 'annulla tutto il gruppo di motori 6B (n°6-11).  
AIMOTOR SETOFF 'dichiara spento il motore AI.

**AIMOTORIN()**

Legge il valore attuale di un motore AI

**-3000-**

**Dati in entrata del motore AI****Struttura della frase**

frase in Inglese: AIMOTORIN ([n° del motore])

**Spiegazione di comando**

Con questo comando è possibile leggere la posizione attuale e controllare un motore AI connesso ai regolatori della serie MR-C3000.

È possibile usare per il [n° del motore] numeri da 0 a 30 come una costante.

Connesso al regolatore: viene letto un angolo di motore da 10 a 90.

Non connesso: angolo del motore 0.

**Esempio di comando**

DIM AI5 AS BYTE

AIMOTOR INIT

AIMOTOR SETON

AIMOTOR 5

AI5 = AIMOTORIN(5)

'stabilisce l'utilizzo del motore AI n°5

'salva il valore del motore AI n°5 come variabile AI5

**GETMOTORSET****Fissare i dati in entrata di un motore**

legge il tasso attuale di un servo-motore e mantiene quello stato.-3000-

**Struttura della frase**

frase in Inglese: GETMOTORSET [nomina di gruppo], [nomina d'entrata del motore n], ...

**Spiegazione di comando**

Con un sistema di comunicazione interattivo, è possibile leggere il valore della posizione attuale di un servo robot digitale (serie-HSR) connesso ad un regolatore della serie MR-C3000.

Quando il regolatore della serie MR-C3000 è installato in un robot, tutti i servo-motori sono riportati alla posizione iniziale di "100" durante l'accensione, e si potrebbe avere un danno per il robot come risultato.

Per prevenire il danno, la posizione iniziale d'accensione può essere sistemata in una posizione diversa da "100".

Leggere la posizione attuale prima dell'iniziale d'accensione e poi l'azione nominata uniformemente in accordo con il comando "move".

Per la [nomina d'entrata del motore n] si usi "0" o "1". si legge il valore attuale del servo-motore selezionato e si mantiene lo stato attuale del robot. Nel caso di "0", si sposta all'iniziale valore del regolatore che è 100.

**Esempio di comando**

GETMOTORSET G8A, 1, 1, 1, 1, 0, 0, 0

'i servo-motori n° 0, 1, 2, 3 mantengono l'attuale valore all'accensione.

'i servo-motori n°4, 5, 6, 7 si spostano al valore iniziale 100 d'accensione.

**Capitolo 9****Comandi di roboBASIC****per il Controllo Musica**

I regolatori di serie MR-C hanno la capacità di suonare un "beep" d'avvertimento e musica. Per utilizzare questa funzione è necessario avere un normale piezo separatamente. Lo spinotto esterno al piezo consiste di un terminale rosso (+) e un terminale bianco di segnale.

Se si desidera un suono potente e pulito si può connettere un altoparlante, ma è richiesto un circuito driver corrente o AMP.

**Connessione al regolatore della serie MR-C2000:**

il piezo è connesso con la porta n°8 del regolatore di serie MR-C2000. il terminale (+) del piezo è connesso al VCC della porta n°8 e il terminale (-) è connesso al SIG (segnale) della porta n°8.

**Connessione al regolatore della serie MR-C3000:**

Il piezo è connesso alla porta n°28 del regolatore di serie MR-C3000. il terminale (+) è connesso al VCC della porta n°28 e il terminale (-) è connesso al SIG (segnale) della porta n°28.

Nota: il regolatore MR-C3024 ha un proprio piezo interno.

**BEEP**

suono d'avvertimento con piezo

-2000-

**Struttura della frase**

frase in Inglese: BEEP

**Spiegazione di comando**

Usare il comando "BEEP" per fare un suono d'avvertimento con i regolatori della serie MR-C2000. Utilizzare un cicalino come suono d'avvertimento è possibile solamente collegandosi alla normale porta d'uscita usando il comando "OUT".

Il regolatore di serie MR-C3000 usa il cicalino con il comando "OUT".

**Esempio di comando**

BEEP 'produce un suono d'avvertimento

**SOUND**

Produce un suono con piezo

-2000-

**Struttura della frase**

frase in Inglese: SOUND [tono], [lunghezza], [tono], [lunghezza], ...

**Spiegazione di comando**

Nei regolatori di serie MR-C2000 possono essere impostati la frequenza del segnale e il tempo di ritardo nel piezo. Valore d'impostazione: da 1 a 254

Rifarsi alla tabella sottostante:

Entrata	Frequenza (Hz)	Entrata	Frequenza (Hz)	Entrata	Frequenza (Hz)
1	38,86k	70	800	160	389
2	23,81k	80	775	170	365
5	11,11k	90	689	180	344
10	5,88k	100	621	190	327
20	3,00k	110	565	200	311
30	2,00k	120	518	210	295
40	1,54k	130	478	220	283
50	1,23k	140	444	230	270
60	1,00k	150	413	240	260

Tempo	Lunghezza (11msec)
0,5 sec	45
1 sec	90
2 sec	180

**Esempio di comando**

SOUND 60, 90, 60, 180, 30, 45 'genera una frequenza di 1KHz per un 1sec, 2sec e genera una frequenza di 2KHz per 0.5sec.

**PLAY**

suona una musica con piezo

-2000-

**Frase di comando**

frase in Inglese: PLAY "[linea suona musica]"

**Spiegazione di comando**

Il programma roboBASIC e i regolatori di serie MR-C forniscono una funzione di suono di musica. Per suonare la musica i dati devono essere aggiunti alla [linea suona musica]. Rifarsi alla tabella sottostante:

Linea di Suono Musica	Descrizione
C	„Do“
D	„Re“
E	„Mi“
F	„Fa“
G	„Sol“
A	„La“
B	„Si“
T	Controlla il tempo, il battito e la velocità
L	Sceglie l'ottava bassa
M	Sceglie l'ottava media
H	Sceglie l'ottava alta
#, +	Alza di un semi-tono(#)
\$, -	Abbassa di un semi-tono (b)
P, . (pausa)	Pausa
<	Abbassa un'ottava
>	Alza un'ottava

T significa tempo e il valore di tempo base è 7.

Si può modificare da 1 a 0 (0 significa 10). 1 è il tempo più veloce e 0 il più lento.

Sono disponibili 3ottave con il regolatore di serie MR-C2000, sono chiamate ottava bassa, media e alta.

Per modificare la lunghezza di una nota, usare i numeri da 0 a 9 e rifarsi alla tabella sottostante:

Se la lunghezza di un tono non è scelta, sarà suonata l'ultima lunghezza.

"4CDEF8G" significa suonare Do, Re, Mi e Fa e una semiminima di Solcome ottava nota.

Il valore standard del comando "PLAY" è una ottava media. Semiminima, Do e tempo 7 hanno lo stesso valore di ottava e tempo dell'ultima modifica.

- 1.) La lunghezza del tono è fissata [4Do 8Mi 6Fa].
- 2.) Il simbolo # o altri segni sono fissati di fronte al tono [#Do \$Mi +Fa -Sol 4#Do #4Do].

Usare il comando "MUSIC" invece che "PLAY" quando si utilizzano regolatori della serie MR-C3000

**Esempio di comando**

PLAY „M4GGAA GGE GGEED“

PLAY „M4GGAA GGE GEDEC“

**MUSIC**

Suona una musica con piezo

-3000-

**Frase di comando**

frase in Inglese: MUSIC "[linea suona musica]"

**Spiegazione di comando**

Il programma roboBASIC e i regolatori di serie MR-C forniscono una funzione di suono di musica.

Per suonare la musica i dati devono essere aggiunti alla [linea suona musica].

Rifarsi alla tabella sottostante:

Linea di Suono Musica	Descrizione
C	„Do“
D	„Re“
E	„Mi“
F	„Fa“
G	„Sol“
A	„La“
B	„Si“
[	Accorcia il tono 1,5 volte per un gruppo di note
]	Allunga il tono 1,5 volte per un gruppo di note
O	Sceglie ottava
M	Sceglie ottava 3
.	Allunga il tono 1,5 volte
#, +	Alza di un semitono (#)
\$, -	Abbassa di un semitono (b)
P, (rest)	Pausa
<, L	Scende di un'ottava
>, H	Sale di un'ottava

Con il regolatore di serie MR-C3000 sono possibili 7 stadi di ottave.

Per modificare la lunghezza di un tono si usino i numeri da 0 a 9. Rifarsi alla tabella sottostante:

Nel caso di note puntate, si scriva come un numero o “.” nella [linea suona musica].

Si deve usare il comando “PLAY” invece che “MUSIC” se si usano regolatori della serie MR-C3000.

Il comando “TEMPO” è disponibile separatamente con i regolatori della serie MR-C3000.

#### Esempio di comando

MUSIC „O34GGAA GGE GGEED“  
MUSIC „O3GGA4.A GGE GEDEC“

#### TEMPO

Imposta il tempo della musica

-3000-

#### Struttura della frase

frase in Inglese: TEMPO [imposta il valore]

#### Spiegazione di comando

Impostare il tempo della musica usando il comando “MUSIC” quando si utilizzano regolatori della serie MR-C3000.

## Capitolo 10

### Comandi

#### Per le comunicazioni esterne di roboBASIC

##### Comunicazione esterna del regolatore di serie MR-C2000

Con i regolatori della serie MR-C2000 possono essere usati due tipi di comunicazioni esterne. Una è la comunicazione seriale RS232 e l'altra è la miniBUS.

Nelle comunicazioni seriali, è possibile avere una comunicazione interattiva con un personal computer (RS232 compatibile) o un altro regolatore MR-C. in aggiunta, si può comunicare sia tramite cavo o con tecnologia wireless usando un modulo RF. Sono necessari tre cavi per la comunicazione RS232 (trasmittente di connessione (TX), ricevente (RX) e terra (GND)).

TX	TX	PC
RX	RX	Micro regolatore
GND	GND	Regolatore robot
Regolatore robot		

Quando ci si connette ad un computer, si dovrebbe usare un circuito di conversione di voltaggio (MAX232).

Con la miniBUS, il segnale BUS e la terra (GND) sono usati per la comunicazione interattiva.

MINIBUS	MINIBUS
GND	GND
Regolatore Robot	Modulo relativo per regolatore

La miniBUS usa solo un cavo per la comunicazione interattiva. Dovrebbe essere seguite regole specifiche. Il modulo LCD è un esempio di trasmissione di dati usando la minibus. Qui il regolatore costruito nell'LCD è impostato solo per ricevere.

##### Comunicazione esterna con il regolatore di serie MR-C3000:

Le comunicazioni ad alta velocità RS232 con equipaggiamento da esterno sono possibili con i regolatori di serie MR-C3000.

Tuttavia, la comunicazione miniBUS non è possibile con la serie MR-C3000.

Entrambi i tipi di regolatore hanno una velocità massima di connesse RS232 di 115,200bps.

#### RX

La porta RX riceve un segnale RS232.

-2000-

#### Struttura della frase

RX [velocità porta], [variabili ricevute], [etichetta di errore di processo ricevuta]

#### Spiegazione di comando

Usando la porta n°9 dei regolatori di serie MR-C2000 si possono ricevere i dati tramite RS232.

La [velocità porta] è rappresentata da numeri da 1 a 4, dove i numeri corrispondono ad una specifica velocità e impostazioni di porta che sono spiegate di sotto.

Numero	Impostazione Porta
1	1200bps, 8Bit Daten, keine Parität, 1 Stop bit
2	2400bps, 8Bit Daten, keine Parität, 1 Stop bit
3	2400bps, 8Bit Daten, keine Parität, 1 Stop bit
4	4800bps, 8Bit Daten, keine Parität, 1 Stop bit

[Variabili Ricevute] è la variabile di ricezione dei dati. Sono permesse solo le variabili byte di formato dichiarato.

[Etichetta di Errore di Processo Ricevuta] è l'etichetta per gli errori che capitano durante la ricezione. Come nel caso del buffer di comunicazione vuoto. Tutti i programmi che aspettano dati attraverso la Porta RS232 possono usare la struttura di frase sottostante.

Retry:

RX 4, A, Retry

Per ricevere segnali RS232 con i regolatori di serie MRC-C3000, deve essere usato l'ERX (ricezione dati).

#### Esempio di comando

In questo esempio il codice ASCII, &h80 (16 in analogico), è ricevuto tramite la connessione RS232 da un terminale esterno, il LED nella Porta Zero è acceso, tutti gli altri sono spenti.

DIM A AS BYTE

```

Retry: RX 4, A, Retry
      IF A = &h80 THEN
          OUT 0, 0
      ELSE
          OUT 0, 1
      ENDIF
      GOTO Retry

```

#### **TX**

La Porta TX trasmette un segnale RS232 -2000-

#### Struttura della frase

TX [*velocità porta*], [*dati*]

#### Spiegazione di comando

Usando la Porta n°10 del regolatore di serie MR-C2000, i dati possono essere trasmessi tramite RS232.

La [*velocità porta*] è rappresentata da numeri da 1 a 4, dove i numeri corrispondono ad una specifica velocità di impostazioni di porta che sono spiegate di sotto.

Numero	Impostazione Porta n
1	1200bps, dati 8Bit, No parità, 1 bit Stop
2	2400bps, dati 8Bit, No parità, 1 bit Stop
3	2400bps, dati 8Bit, No parità, 1 bit Stop
4	4800bps, dati 8Bit, No parità, 1 bit Stop

[Data] è un valore di dati trasmesso attraverso la Porta TX. Possono essere usati numeri, costanti e variabili. Nel caso si volesse trasmettere la lettera "A", andrebbe inviato il codice ASCII relativo alla lettera "A". Rifarsi all'esempio di sotto.

```

DIM I AS BYTE
I = "A"
TX 4, I

```

Se code di lettere vengono messe dentro le variabili, il valore ASCII della lettera nella coda è messo dentro le variabili.

Per trasmettere un segnale RS232 con i regolatori della serie MR-C3000, andrebbe usato il comando "ETX" (trasmissione dati) al posto del comando "TX".

#### Esempio di comando

Il seguente è un esempio dove il valore di chiave della porta byte "0" sta continuamente trasmettendo attraverso l'RS232 ad un terminale esterno.

DIM A AS BYTE

```

Main:
      A = BYTEIN(0)
      TX 4, A
      GOTO Main

```

#### **MINIIN**

Riceve segnali Minibus attraverso la porta minibus -2000-

#### Struttura della frase

MINIIN

#### Spiegazione di comando

I dati del miniBUS sono ricevuti usando una delle sei porte miniBUS nel regolatore di serie MR-C2000 oper tutto il tempo che i dati "0" non sono ricevuti. La struttura del programma può essere vista di sotto.

DIM A AS BYTE

```

Retry:
      A = MINIIN
      IF A = 0 THEN GOTO Retry

```

#### **MINIOUT**

la Porta miniBUS trasmette segnali miniBUS -2000-

#### Struttura della frase

MINIOUT [Dati], [Dati]...

#### Spiegazione di comando

I dati del miniBUS sono trasmessi usando la porta miniBUS n°6 dei regolatori di serie MR-C2000. I protocolli di comunicazione del minibus sono simili a quelli dell'RS232. Numeri, costanti e variabili sono usati per i [Dati]. Una illimitata quantità di dati può essere trasmessa, ma non si può trasmettere il numero "0".

#### Esempio di comando

MINIOUT 100, 20, 76, 65

#### **ERX**

La Porta ERX riceve segnali RS232 -3000-

#### Frase di comando

ERX [*Velocità Porta*], [*Variabili Ricevute*], [*Etichetta di Errore di Processo Ricevuta*]

#### Spiegazione di comando

I Dati vengono ricevuti utilizzando la Porta ERX dei regolatori di serie MR-C3000. Le costanti della [*Velocità Porta*] sono elencate sotto.



Numero	Impostazioni porta
2400	2400bps, 8Bit data, No Parity, 1 Stop bit
4800	4800bps, 8Bit data, No Parity, 1 Stop bit
9600	9600bps, 8Bit data, No Parity, 1 Stop bit
14400	14400bps, 8Bit data, No Parity, 1 Stop bit
19200	19200bps, 8Bit data, No Parity, 1 Stop bit
28800	28800bps, 8Bit data, No Parity, 1 Stop bit
38400	38400bps, 8Bit data, No Parity, 1 Stop bit
57600	57600bps, 8Bit data, No Parity, 1 Stop bit
76800	76800bps, 8Bit data, No Parity, 1 Stop bit
115200	115200bps, 8Bit data, No Parity, 1 Stop bit
230400	230400bps, 8Bit data, No Parity, 1 Stop bit

[Variabili Ricevute] sono le variabili che immagazzinano i dati ricevuti, possono essere usate solo variabili byte dichiarate.

[Etichetta di Errore di processo Ricevuta] è l'etichetta di posizione dei dati che non sono ancora stati ricevuti.

Retry:

ERX 9600, A, Retry

Per ricevere segnali RS232 da un regolatore di serie MR-C2000, si dovrebbe usare il comando "RX" invece che il comando "ERX" (ricezione Dati).

## ETX

La Porta ETX trasmette segnali RS232.

-3000-

### Struttura della frase

Inglese: ETX [Velocità Porta], [Dati]

### Spiegazione di comando

I Dati sono trasmessi attraverso la Porta ETX del regolatore della serie MR-C3000

Le costanti della [Velocità Porta] sono elencate sotto.

Numero	Impostazioni porta
2400	2400bps, 8Bit data, No Parity, 1 Stop bit
4800	4800bps, 8Bit data, No Parity, 1 Stop bit
9600	9600bps, 8Bit data, No Parity, 1 Stop bit
14400	14400bps, 8Bit data, No Parity, 1 Stop bit
19200	19200bps, 8Bit data, No Parity, 1 Stop bit
28800	28800bps, 8Bit data, No Parity, 1 Stop bit
38400	38400bps, 8Bit data, No Parity, 1 Stop bit
57600	57600bps, 8Bit data, No Parity, 1 Stop bit
76800	76800bps, 8Bit data, No Parity, 1 Stop bit
115200	115200bps, 8Bit data, No Parity, 1 Stop bit
230400	230400bps, 8Bit data, No Parity, 1 Stop bit

[Dati] è il valore da trasmettere attraverso la Porta ETX. Numeri, variabili e costanti possono essere usati per i [Dati]. Notare l'esempio sottostante.

DIM I AS BYTE

I = "A"

ETX 9600, I

Se si deve trasmettere la lettera "A", il codice ASCII per la lettera "A" andrà inviato. Una lettera da inserire in una variabile dovrà sempre essere inivata prima in codice ASCII.

Per trasmettere segnali RS232 da regolatori di serie MR-C2000, il comando "TX" andrebbe usato invece che il comando "ETX" (Trasmissione Dati).

## Capitolo 11

### descrizione del procedimento

#### di Comando del segnale

#### Analogico di ROBOBASIC

### AD()

Il segnale analogico dalla Porta AD è convertito in un segnale -3000- Digitale

### Struttura della frase

AD ([Porta AD])

### Spiegazione di comando

Nei regolatori di serie MR-C3000 ci sono otto Porte AD (Porte Digitali In-Out da 32 a 39), numerate da zero a sette, che convertono un segnale analogico da periferiche o sensori esterni in un segnale digitale. Per la [Porta AD] sono usate costanti e variabili byte.

### Esempio di comando

Nel seguente esempio, un valore viene emesso ad un modulo LCD dopo aver ricevuto un segnale analogico dalla porta AD n°1.

DIM a AS BYTE

Dichiara la variabile byte "a".

LCDINIT

Viene inizializzato l'uso del modulo LCD.

CLS

Tutti i dati sullo schermo dell'LCD sono cancellati.

CSOFF

il cursore scompare.

MAIN:

Viene dichiarata un'etichetta di nome MAIN.

A = AD (1)

Il valore immesso nella Porta AD #1 è salvato come variabile "a".

LOCATE 5, 0

Il cursore è posizionato a 5.0 sull'LCD.

PRINT FORMAT(a,DEC,2)

Il valore immesso, a, viene emesso al modulo LCD come due digitali usando il sistema decimale.

GOTO MAIN

Va alla MAIN.

### REMOCON()

Legge i valori di un telecomando a infrarossi dalla Porta di -3000- Trasformazione AD #7

### Struttura della frase

REMOCON ([Remocon(#)])

### Spiegazione di comando

La Porta n°7 viene utilizzata per un telecomando ad infrarossi. [Remocon(#)] è assegnato al numero 1, ma si possono

utilizzare altri numeri a seconda della versione di regolatore MR-C3000 (3024) in uso. A riguardo di particolari più specifici, rifarsi agli esempi dei capitoli seguenti.



Remocon(1) IR Remocon (tipo ID)

#### Esempio di comando

```

DIM a AS BYTE          Dichiaro i valori di variabile ricevuti.
MAIN:                  l'etichetta MAIN riceve costantemente il valore di remocon.
A = REMOCON(0)         Il valore di remocon è messo nella variabile "a".
ON a GOTO MAIN,KEY1,KEY2,KEY3,KEY4  Va alla MAIN senza che esista un valore di ricezione.
GOTO MAIN              Va alla MAIN.
END
KEY1:                  Lo processa quando un valore di ricezione è uno.F
.....
GOTO MAIN              Va alla MAIN.
KEY2:                  lo processa quando un valore di ricezione è due.
.....
GOTO MAIN              Va alla MAIN.
KEY3:                  lo processa quando un valore di ricezione è tre.
.....
GOTO MAIN              Va alla MAIN.
KEY4:                  lo processa quando un valore di ricezione è quattro.
.....
GOTO MAIN              Va alla MAIN.
  
```

#### SONAR()

Legge le distanze calcolate da un sensore ad ultrasuoni connesso alla Porta ad Ultrasuoni **-3000-**

#### Struttura della frase

SONAR ([Porta ad Ultrasuoni])

#### Spiegazione di comando

Le Porte Digitali In e Out da 0 a 23 del regolatore di serie MR-C3000 possono essere usate come porte ad Ultrasuoni da 0 a 11. Si osservi la descrizione seguente.

Porta Digitale In e Out del regolatore di serie MR-C3000	Porta ad Ultrasuoni
Porta #0	Uscita Porta Ultrasonica #0
Porta #1	Entrata Porta Ultrasonica #0
Porta #2	Uscita Porta Ultrasonica #1
Porta #3	Entrata Porta Ultrasonica #1
Porta #4	Uscita Porta Ultrasonica #2
Porta #5	Entrata Porta Ultrasonica #2
Porta #6	Uscita Porta Ultrasonica #3
Porta #7	Entrata Porta Ultrasonica #3
Porta #8	Uscita Porta Ultrasonica #4
Porta #9	Entrata Porta Ultrasonica #4

Porta #10	Uscita Porta Ultrasonica #5
Porta #11	Entrata Porta Ultrasonica #5
Porta #12	Uscita Porta Ultrasonica #6
Porta #13	Entrata Porta Ultrasonica #6
Porta #14	Uscita Porta Ultrasonica #7
Porta #15	Entrata Porta Ultrasonica #7
Porta #16	Uscita Porta Ultrasonica #8
Porta #17	Entrata Porta Ultrasonica #8
Porta #18	Uscita Porta Ultrasonica #9
Porta #19	Entrata Porta Ultrasonica #9
Porta #20	Uscita Porta Ultrasonica #10
Porta #21	Entrata Porta Ultrasonica #10
Porta #22	Uscita Porta Ultrasonica #11
Porta #23	Entrata Porta Ultrasonica #11

Il valore della [Porta ad Ultrasuoni] deve essere un numero fisso. Il valore di ritorno da SONAR deve essere all'interno della gamma da 0 a 3000. Se il valore di ritorno è 0, allora la distanza non è percepita, altrimenti viene riportato un valore "XX".

Un sensore ad ultrasuoni disponibile per i regolatori di serie MR-C3000 è il modello SRF04 della ROBOT ELECTRONICS Inc.



#### Esempio di comando

```

DIM A AS INTEGER
A = SONAR(3)
  
```

Dichiara numero fisso la variabile A.

Il valore ricevuto (distanza) è salvato nella variabile A usando la Porta ad Ultrasuoni #3 (Porta Digitale In e Out #6, 7).

#### RCIN()

Immette i valori d'impulso da trasmettitori e ricevitori RC **-3000-**

#### Struttura della frase

RCIN ([Porta di Ricezione RC])

**Spiegazione di comando**

I ricevitori RC sono annessi alla Porta d'Entrata AD del regolatore della serie MR-C3000. Da lì il valore di ricezione del trasmettitore può essere letto. La tabella sottostante mostra la configurazione della [Porta di Ricezione].

Numero di Porta AD del regolatore di serie MR-C3000 (Numero di Porta Digitale In e Out)	Porta di Ricezione RC
Porta #0 (Porta #32)	Porta di Ricezione RC #0
Porta #0 (Porta #32)	Porta di Ricezione RC #1
Porta #0 (Porta #32)	Porta di Ricezione RC #2
Porta #0 (Porta #32)	Porta di Ricezione RC #3
Porta #0 (Porta #32)	Porta di Ricezione RC #4
Porta #0 (Porta #32)	Porta di Ricezione RC #5
Porta #0 (Porta #32)	Porta di Ricezione RC #6
Porta #0 (Porta #32)	Porta di Ricezione RC #7

Per prevenire malfunzionamenti causati da interferenze elettroniche, si dovrebbero usare ricevitori FM più che AM. Se il trasmettitore ha funzioni high-end, si possono realizzare più varie azioni o movimenti.

**Esempio di comando**

DIM A AS BYTE

A = RCIN(0)

Un segnale Ricevuto dalla Porta di Ricezione RC è salvato come variabile A.

**GYRODIR**

Imposta la direzione dei servo-motori quando supportato da un Giroscopio -3000-

**Struttura della frase**

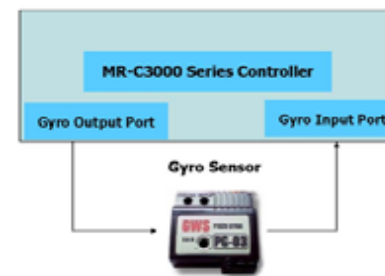
GYRODIR [Gruppo], [Direzione Motore] ...

**Spiegazione di comando**

Questo processo controlla la direzione di un gruppo di servo-motori quando un giroscopio è connesso ad una porta AD dei regolatori di serie MR-C3000.

Il numero di giroscopi utilizzabili è quattro.

Vedere il tabella seguente.



Numero Porta AD (numero di Porta Digitale In e Out) del regolatore di serie MR-C3000	Porta Giroscopio
Porta #0 (Porta #32)	Giroscopio #1 canale d'uscita Porta
Porta #1 (Porta #33)	Giroscopio #2 canale d'uscita Porta
Porta #2 (Porta #34)	Giroscopio #3 canale d'uscita Porta
Porta #3 (Porta #35)	Giroscopio #4 canale d'uscita Porta
Porta #4 (Porta #36)	Giroscopio #1 canale d'entrata Porta
Porta #5 (Porta #37)	Giroscopio #2 canale d'entrata Porta
Porta #6 (Porta #38)	Giroscopio #3 canale d'entrata Porta
Porta #7 (Porta #39)	Giroscopio #4 canale d'entrata Porta

Finché un giroscopio è reversibile, la specifica direzione sarà determinata dal contingente valore del servo-motore. La [Direzione Motore] è uno "0" o un "1". Un valore di 1 incrementerà la posizione del servo-motore e uno 0 la diminuirà.

**Esempio di comando**

GYRODIR G6A, 1, 1, 0, 0, 1, 0

**GYROSET**

determina quale giroscopio controllerà un particolare gruppo di servo-motori.-3000-

**Struttura della frase**

GYROSET [Gruppo], [Giro N Motore] ...

**Spiegazione di comando**

Gyroset determina quale servo-motore in gruppo di servo-motori [Gruppo] viene controllato da un giroscopio specifico. [Giro N Motore] è la specifica porta "giro" usata per ogni servo nel gruppo. Si veda l'esempio sotto.

01,02,03,04 : GWS PG03 11,12,13,14 : KRG-1

21,22,23,24 : reserved 31,32,33,34 : reserved

**Esempio di comando**

GYROSET G6B, 1, 1, 2, 2, 0, 0

Il servo #6 riceve il sensore di giroscopio #1 e lo processa.

Il servo #7 riceve il sensore di giroscopio #1 e lo processa.

Il servo #8 riceve il sensore di giroscopio #2 e lo processa.

Il servo #9 riceve il sensore di giroscopio #2 e lo processa.

I servo-motori #10 e #11 non usano un sensore di giroscopio.

**GYROSENSE**

Imposta la sensibilità di un servo-motore ad un giroscopio. **-3000-**

**Frase di comando**

GYROSENSE [Gruppo], [Giroscopio N Motore Sensibilità] ...

**Spiegazione di comando**

Si possono connettere quattro giroscopi al regolatore di serie MR-C3000. GYROSENSE imposta la sensibilità di un singolo servo-motore ad un giroscopio.

[Giroscopio N Motore Sensibilità] usa numeri da 0 255 o costanti per controllare la sensibilità di ogni servo-motore di un gruppo. Un'impostazione di "0" non cambierà la sensibilità del servo.

Come il valore aumenta così sarà la risposta del giroscopio.

**Esempio di comando**

GYROSENSE G6A, 100, 100, 255, 255, 50, 50

I servo-motori #0 e #1 impostano a 100 la sensibilità del Giroscopio.

I servo-motori #2 e #3 impostano al massimo (255) la sensibilità del Giroscopio.

I servo-motori #4 e #5 impostano a 50 la sensibilità del Giroscopio.

**Capitolo 12****Comandi di Procedimento e Altri****di ROBOBASIC****ON ... GOTO**

Divergenza condizionale in accordo con il valore delle variabili.

**Struttura della frase**

ON [Variabile] GOTO [Etichetta di Linea], [Etichetta di Linea] ...

**Spiegazione di comando**

"ON... GOTO" è il comando usato quando il programma da processare diverge secondo il valore di una [Variabile]. Quando si usa il comando IF, il comando ON... GOTO può essere ancora usato. Infatti, quando si comaprano i due comandi, ON... GOTO coadiuva la creazione di un codice più piccolo.

La seguente è una comparazione tra IF e ON... GOTO.

A = BYTEIN(0)	.....	A = BYTEIN(0)
IF A = 0 THEN		ON A GOTO 10, 20, 30
GOTO 10		IF A>2 THEN GOTO 40
ELSEIF A = 1 THEN		
GOTO 20		
ELSEIF A = 2 THEN		
GOTO 30		
ELSE		
GOTO 40		
ENDIF		

Seguendo il comando ON... GOTOP c'è l'[Etichetta di Linea]. L'[etichetta di Linea] aumenta numericamente quando la [variabile] è vera.

Numeri e costanti possono essere usati per la [variabile]. Un numero massimo di 255 può essere utilizzato per l'[Etichetta di Linea].

**RND**

Casuale.

**Struttura della frase**

RND

**Spiegazione di comando**

Per creare un programma casuale all'interno dei regolatori MR-C, si usi il comando RND. Questo comando creerà un numero casuale compreso tra 0 e 255.

**Esempio di comando**

DIM A AS BYTE

A = RND

BYTEOUT 0, A

Un valore casuale esce alla Porta Byte 0.

**REMARK**

Pone una dichiarazione all'interno del codice

**Struttura della frase**

REMARK [Descrizione]

**Spiegazione di comando**

È una buona pratica di programmazione quella di inserire dichiarazioni all'interno del codice per spiegare certe procedure. Questo può essere fatto sia con un (\*) o col comando REMARK.

Seguendo il comando, una descrizione è inserita nella stessa linea. REMARK non ha effetti sull'operazione del programma.

**Esempio di comando**

REMARK 8 LED sono accesi

BYTEOUT 0, 0



## **Capitolo 13**

### **Descrizione di Comando**

#### **di ROBOBASIC**

##### **\$DEVICE**

Imposta il regolatore applicabile al programma.

##### **Struttura della frase**

'\$DEVICE [Regolatore]

##### **Spiegazione di comando**

Questo comando imposta il regolatore su cui il programma è caricato dopo la compilazione. Se un regolatore è già stato assegnato, sarà cambiato nel regolatore specificato da [regolatore].

##### **Esempio di comando**

'\$DEVICE MRC2000            I programma attuale usa il regolatore MR-C2000.

##### **,\$LIMIT**

Limita la gamma di movimenti per ogni servo-motore

##### **Struttura della frase**

'\$LIMIT [Numero Motore], [Valore Minimo], [Valore Massimo]

##### **Spiegazione di comando**

Il comando "\$LIMIT" è usato per impostare un angolo totale rotazionale del servo-motore nell'ordine di prevenire un eventuale danno dovuto al sovrautilizzo del servo.

[Numero Motore] è lo specifico servo utilizzato. La gamma è compresa tra 0 e 31. [Valore Minimo] e [Valore Massimo] sono l'angolo minimo e massimo desiderati compresi tra 10 e 190. L'angolo di base del servo-motore è da 10 a 190.

##### **Esempio di comando**

'\$LIMIT 0, 50, 100            L'angolo del servo-motore è limitato tra 50 e 100.

Manuale

# ROBONOVA-I



MULTIPLEX Modellsport GmbH & Co. KG

[www.robonova.de](http://www.robonova.de)

Italiano V1.00