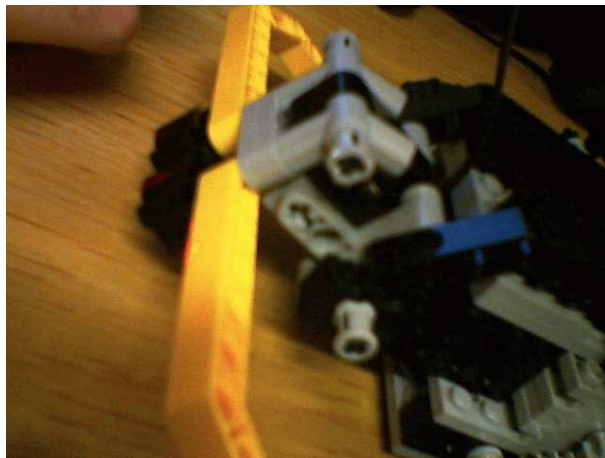


M. Bessone 128714
D. Longobardi 121674
E. Richiardone 126892

Laboratorio di Automazione Discreta
Relazione di Progetto



Prof. G. Menga
Politecnico di Torino
Ingegneria informatica - Terza Facoltà
AA 2004/2005

Indice

1	Introduzione	4
2	Ideazione del progetto	5
3	Realizzazione fisica	7
3.1	Componenti	7
3.2	Montaggio	7
3.2.1	Collegamento RCX	11
4	Programma di controllo	12
4.1	API di controllo	12
4.1.1	'controllomotori'	13
4.1.2	'controllosensori'	14
4.1.3	'controlloazione'	15
4.1.4	Calibrazione	16
4.2	Algoritmi di guida	16
4.2.1	Algoritmo 'quadrato'	16
4.2.2	Algoritmo 'zenzero'	18
5	Conclusione	20

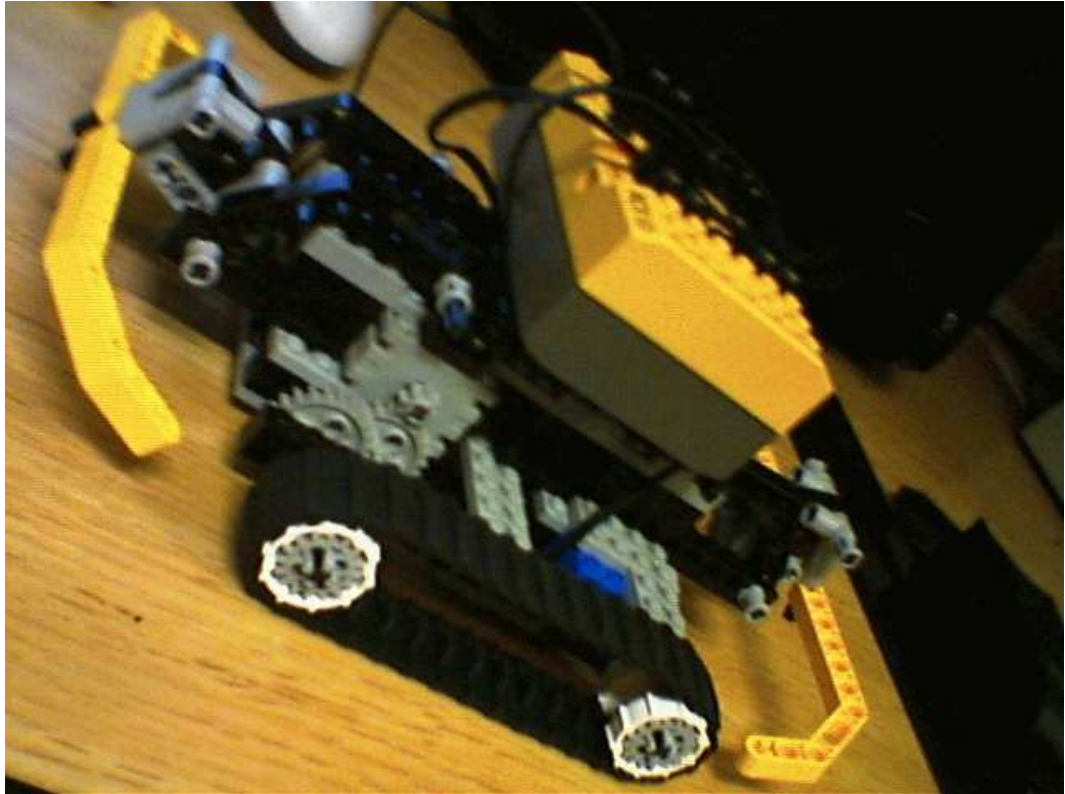


Figura 1: Veduta di 'zenzero'

Capitolo 1

Introduzione

L'obiettivo del laboratorio consiste nella realizzazione di un robot costruito utilizzando Lego Mindstorm che svolga una determinata funzionalità espressa mediante un'automa a stati.

La costruzione del robot è essenzialmente identica a quella di un normale lego, mentre per quanto riguarda movimenti e controllo sono utilizzati motori e sensori. Motori, sensori e altri dispositivi per ciò che riguarda il movimento e il controllo sono appunto la peculiarità dei Lego Mindstorm che li differenzia dai normali lego da costruzione.

Per il funzionamento e l'alimentazione di questi è fondamentale il ruolo svolto dalla base RCX nella quale si inserisce il codice necessario al controllo di motori e sensori.

Il codice per il funzionamento del robot è stato generato tramite il linguaggio Lego Statetra che permette la rappresentazione dell'automa a stati, mediante diagrammi a blocchi, e la conversione in un linguaggio comprensibile alla base RCX.

Nel diagramma è possibile rappresentare i vari stati con le transizioni e le varie condizioni che permettono i passaggi di stato e quindi il funzionamento del robot in base a quello che avviene nell'ambiente che lo circonda. La scrittura del codice prodotto dal software all'interno della base RCX avviene tramite un torretta ad infrarossi collegata al pc tramite interfaccia USB. Tramite questa torretta è anche possibile controllare il funzionamento dei componenti collegati al RCX (sensori e motori) utilizzando un processo di tipo non Lego, inserito nel diagramma a blocchi, che legge tramite la porta infrarossi, puntata verso il robot, i valori assunti istante per istante.

Capitolo 2

Ideazione del progetto

L'idea principale da cui si era partiti era quella di un veicolo che si potesse muovere su un territorio e potesse interagire con l'ambiente evitando o seguendo ostacoli che poteva eventualmente incontrare.

Per poter riconoscere gli ostacoli si è pensato di utilizzare sensori di contatto la cui sensibilità è poi stata aumentata tramite una barra paraurti per poter riconoscere un ostacolo su tutta la larghezza del mezzo.

Per fare in modo che il robot esegua qualche funzionalità è stata definita una sequenza di passi che questo deve svolgere.

Il robot secondo la nostra definizione esegue i seguenti passi:

1. Partenza da ostacolo posto alle spalle del mezzo (serve per iterare)
2. Avanzamento fino al primo ostacolo
3. Superamento primo ostacolo:
 - (a) Inversione di marcia per circa 10cm
 - (b) Rotazione di 90 gradi in senso orario
 - (c) Avanzamento per circa 25cm
 - (d) Rotazione di 90 gradi in senso anti orario
 - (e) Avanzamento di circa 60 cm (per il superamento dell'ostacolo)
 - (f) Rotazione di 90 gradi in senso anti orario
 - (g) Avanzamento di 25cm
 - (h) Rotazione di 90 gradi in senso orario
4. Avanzamento fino al prossimo ostacolo
5. Superamento secondo ostacolo (come per il precedente)
6. Avanzamento fino al prossimo ostacolo
7. Inversione di marcia
8. Avanzamento inverso fino al prossimo ostacolo

9. Superamento primo ostacolo (come i casi precedenti ma con superamento ostacolo in modo opposto)
10. Avanzamento inverso fino al prossimo ostacolo
11. Superamento del secondo ostacolo
12. Iterazione del percorso fino allo spegnimento del mezzo

Per meglio comprendere il percorso del robot ed il funzionamento si può osservare l'immagine rappresentata in figura 2.1.

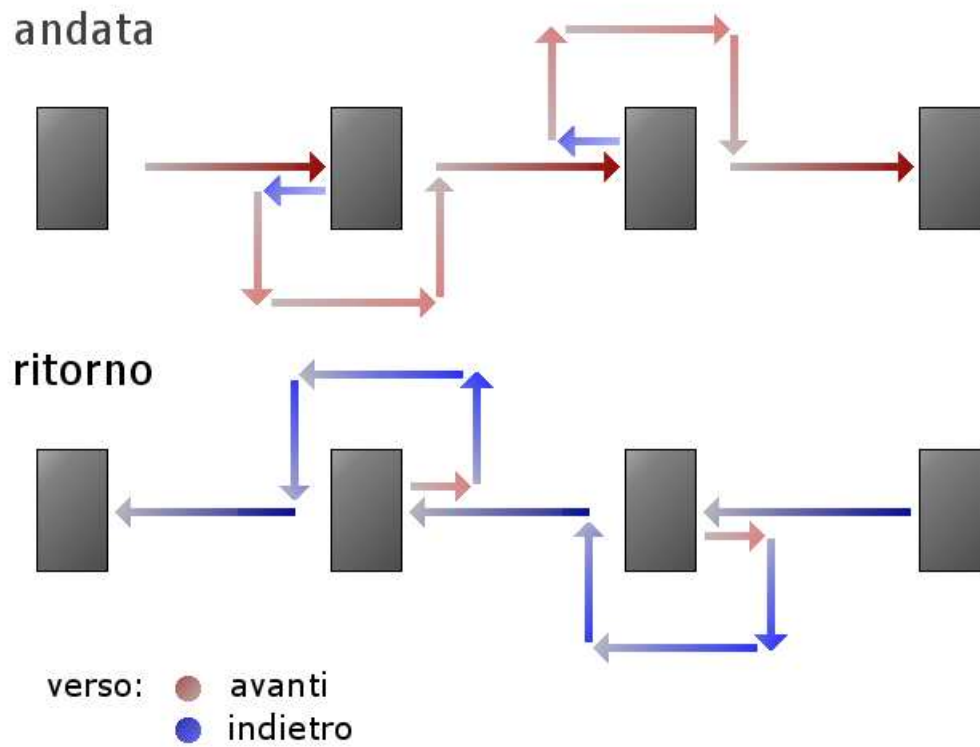


Figura 2.1: Percorso robot

Capitolo 3

Realizzazione fisica

3.1 Componenti

I componenti impiegati per la realizzazione del veicolo sono di tre tipi:

1. telaio:
struttura in lego che costituisce la base, su cui sono montati gli altri componenti.
2. attuatori:
la trazione è realizzata da due motori connessi ai cingoli tramite un paio di ingranaggi. La scelta della trazione è stata limitata ai cingoli a causa della loro unica disponibilità. E' stato necessario utilizzare una coppia di motori per poter sterzare il veicolo su se stesso utilizzando i cingoli.
3. sensori:
sono stati utilizzati due sensori di rotazione, uno per cingolo, e due di contatto, per la rilevazione degli ostacoli anteriore e posteriore. L'informazione su quale dei due sensori è stato premuto, viene ricavato dal verso di marcia; quindi abbiamo collegato i due sensori sullo stesso contatto.

3.2 Montaggio

Individuate le componenti necessarie, siamo passati al montaggio del veicolo 'zenzero'; l'operazione si è svolta in circa due ore, senza incontrare particolari difficoltà.

In figura 3.1 riportiamo uno schema di massima, con ubicazione delle varie componenti.

Osservando l'eccessiva velocità di rotazione dei motori, che causa difficoltà nel controllo del mezzo, sono stati inseriti ingranaggi per diminuire la velocità ed aumentare la precisione del veicolo. Questi sono composti da due rapporti di riduzione per ogni motore, come si può vedere per il motore destro in figura 3.2.

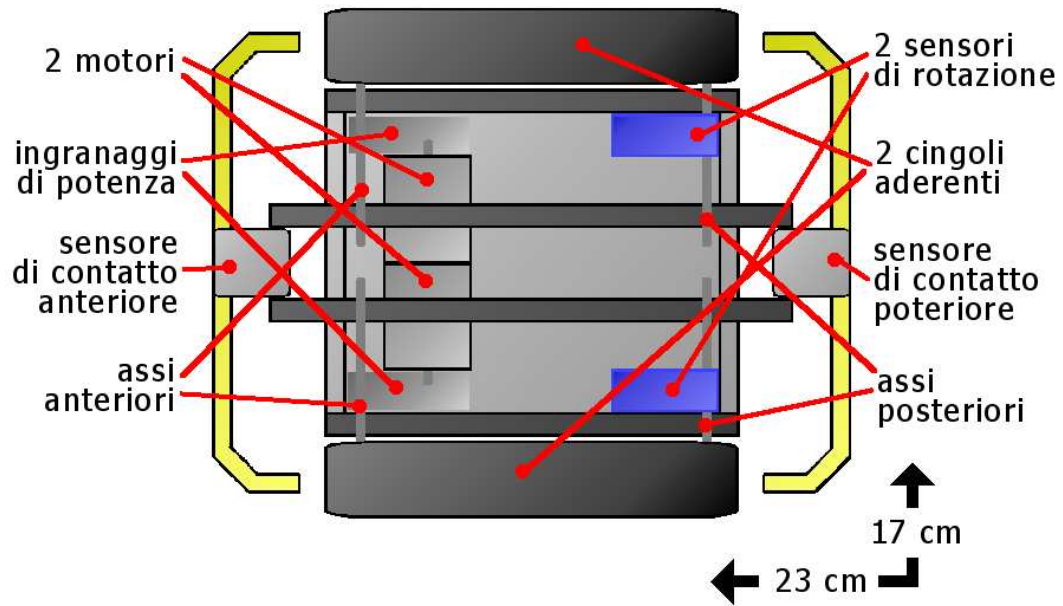


Figura 3.1: Schema veicolo

I sensori di rotazione sono stati posti sull'asse libero che tiene il cingolo dalla parte opposta del motore, per non trasmettere l'informazione di coppia direttamente; si è quindi cercato il più possibile di trasmettere l'informazione del cingolo a terra e non dei movimenti del motore.

Tutto questo assume importanza maggiore se consideriamo che il cingolo ha un certo grado di libertà (circa 2mm) dal rotore su cui è appoggiato. E' possibile vedere in figura 3.3 i due assi del cingolo destro, con connessi il sensore e gli ingranaggi.

Per la costruzione dei paraurti collegati ai sensori di contatto, abbiamo tratto spunto da un manuale di montaggio Lego Mindstorm. I due paraurti anteriore e posteriore sono identici: in figura 3.4 vediamo quello anteriore.

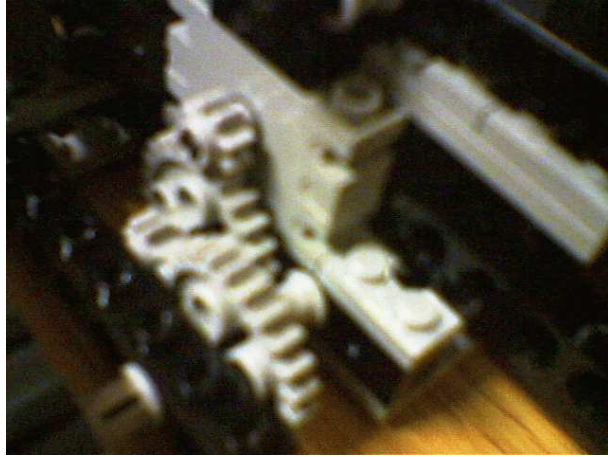


Figura 3.2: Dettaglio ingranaggi

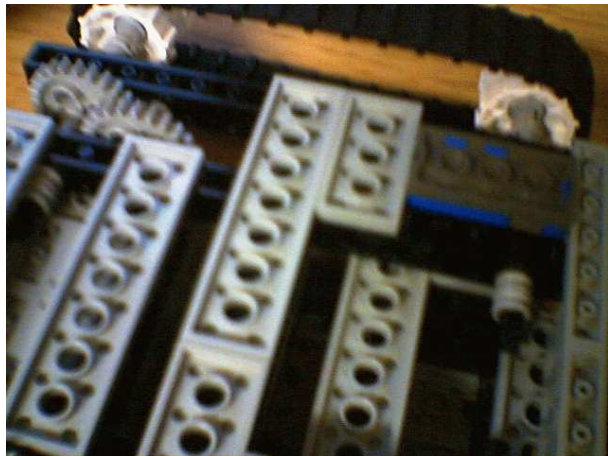


Figura 3.3: Particolare sensori di rotazione e cinghia



Figura 3.4: Dettaglio sensore di contatto anteriore

Le dimensioni finali del robot, senza porre sopra l'RCX, sono:

larghezza	17 cm
lunghezza	23 cm
altezza	5.5 cm

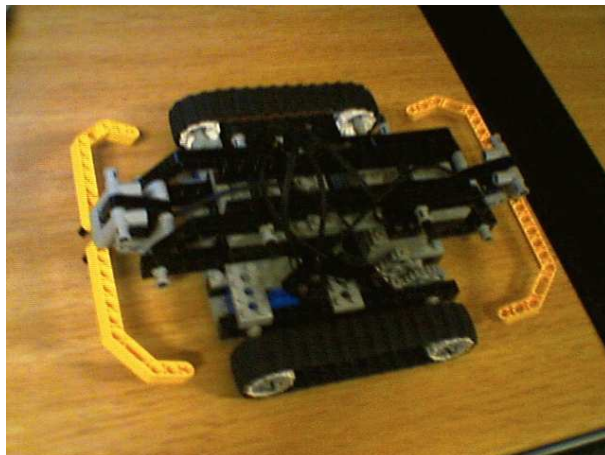


Figura 3.5: Veduta aerea senza RCX

3.2.1 Collegamento RCX

Posando sopra il veicolo il modulo RCX, l'altezza diventa 9 cm.

Enumeriamo l'assegnazione dei contatti dell'RCX disponibili ai diversi componenti:

componente	assegnazione RCX	assegnazione LegoStatetra
motore destro	C	2
motore sinistro	A	0
rotore destro	3	2
rotore sinistro	2	1
sensori di contatto	1	0

Il collegamento dei due sensori di contatto è sovrapposto per guadagnare una porta. In figura 3.6 si possono vedere i collegamenti sull'RCX montato.

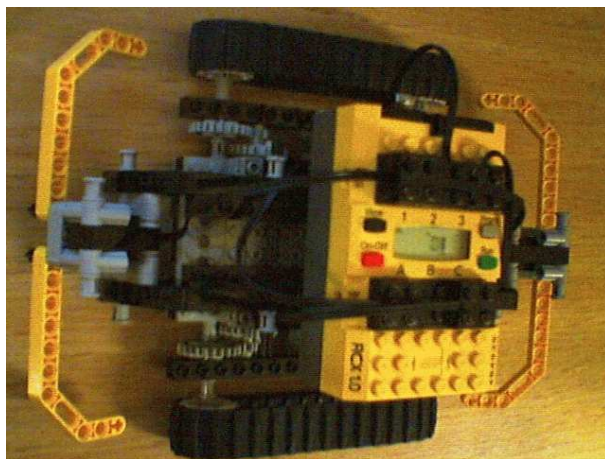


Figura 3.6: Veduta aerea con RCX

Capitolo 4

Programma di controllo

Ora analizziamo il software di controllo creato per gestire il comportamento del veicolo; è composto da un microcodice generato da LegoStatetra che viene scritto nella memoria volatile del modulo RCX, e che può successivamente interfacciarsi con un Non-Lego Process del calcolatore.

4.1 API di controllo

E' stato deciso di vedere le funzionalità di base del robot (avanti/indietro, sterzo e urto) come una API, in modo da poterla utilizzare per diversi obiettivi. Elenchiamo la descrizione e il relativo diagramma per i tre processi Statetra che implementano questa API, utilizzata dai processi che dovranno impartire le azioni da intraprendere.

4.1.1 'controllomotori'

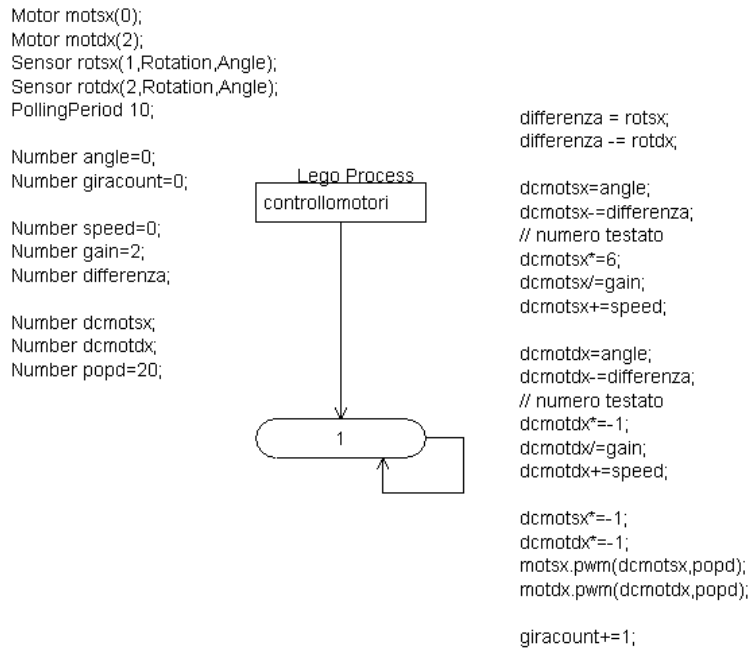


Figura 4.1: Processo controllomotori

Questo processo è un ciclo ripetuto indefinitamente (con PollingPeriod a 10) che imposta i comandi dei motori, con le correzioni di traiettoria ed il contatore dei giri. I motori sono comandati tramite un'onda quadra, impostando un'ampiezza fissa e variando opportunamente il duty cycle dei due motori basandosi sulla differenza fra le rotazioni rilevate dai due sensori. Di conseguenza il processo interagisce con i due motori e i due sensori di rotazione.

4.1.2 'controllosensori'

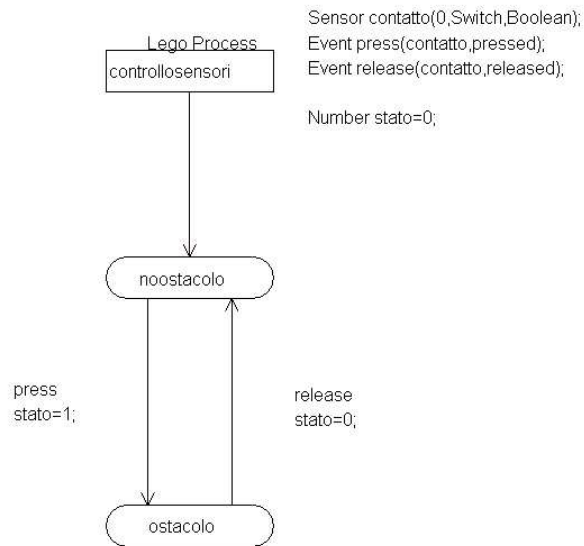


Figura 4.2: Processo controllosensori

Il processo che intercetta l'urto del robot, tramite uno dei sensori di pressione, è costituito da due stati: l'ostacolo è presente o l'ostacolo è assente. La transizione fra i due avviene tramite la pressione ed il rilascio di uno degli switch.

4.1.3 'controlloazione'

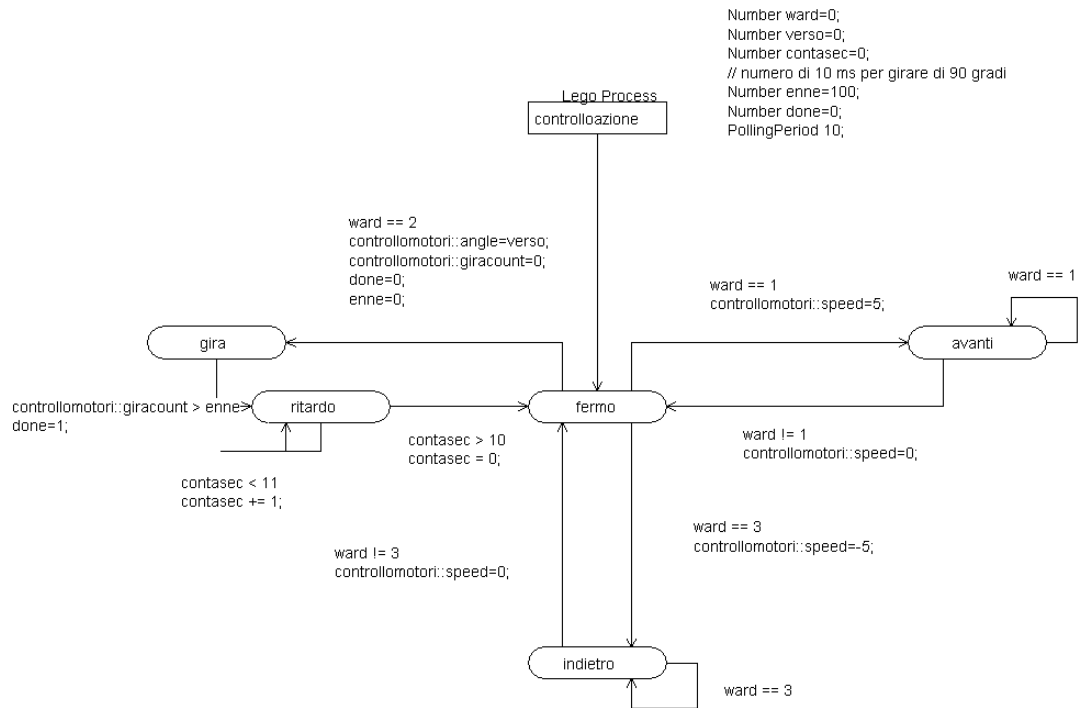


Figura 4.3: Processo controlloazione

'controlloazione' si occupa di gestire il verso del senso di marcia, con relativa velocit  e di gestire l'operazione di sterzo, lavorando con le variabili del processo controllomotori.

La variabile chiave di questo processo, livello di accesso alla API,   'ward'; a seconda del valore che assume, esegue l'operazione voluta (avanti/indietro e sterzo). Questo lavora in PollingPeriod per l'accesso al contatore di 'controllomotori'.

4.1.4 Calibrazione

E' stato utilizzato un algoritmo 'avanti e gira' per testare l'API precedentemente definita.

In questa fase abbiamo dovuto correggere pesantemente le API a causa di problemi sintattici; infatti sono state incontrate molte differenze presenti tra LegoStateTra e StateTra.

Appena siamo riusciti a sistemare i problemi riscontrati abbiamo potuto osservare che il veicolo tendeva sempre a curvare verso destra, molto probabilmente per le grosse differenze di costruzione e per l'usura dei motori.

Per andare quindi dritto evitando il piu' possibile di correggere la traiettoria in modo drastico ogni pochi cm, abbiamo effettuato la calibrazione dei due motori (in controllomotori). I valori ricavati sperimentalmente sono stati implementati tramite le istruzioni riportate qui di seguito:

gain	=	2
dcmotx	*=	6
dcmotdx	*=	-1

Per quanto riguarda la differenza da imporre nel caso in cui si deve curvare di 90 gradi abbiamo definito la seguente calibrazione per curvare a destra:

verso	+=	56
-------	----	----

Per curvare a sinistra invece:

verso	-=	56
-------	----	----

4.2 Algoritmi di guida

4.2.1 Algoritmo 'quadrato'

Per provare l'effettivo funzionamento delle calibrazioni e testare il robot abbiamo creato il seguente algoritmo che fa percorrere al veicolo un quadrato di dimensione di circa 60 cm di lato.

Il diagramma dell'algoritmo sviluppato è rappresentato in figura 4.4.

Eseguito questo test odometrico abbiamo potuto verificare la bontà del robot, che è risultata essere ottima; difatti abbiamo avuto uno spostamento di circa 3 cm sull'intero quadrato.

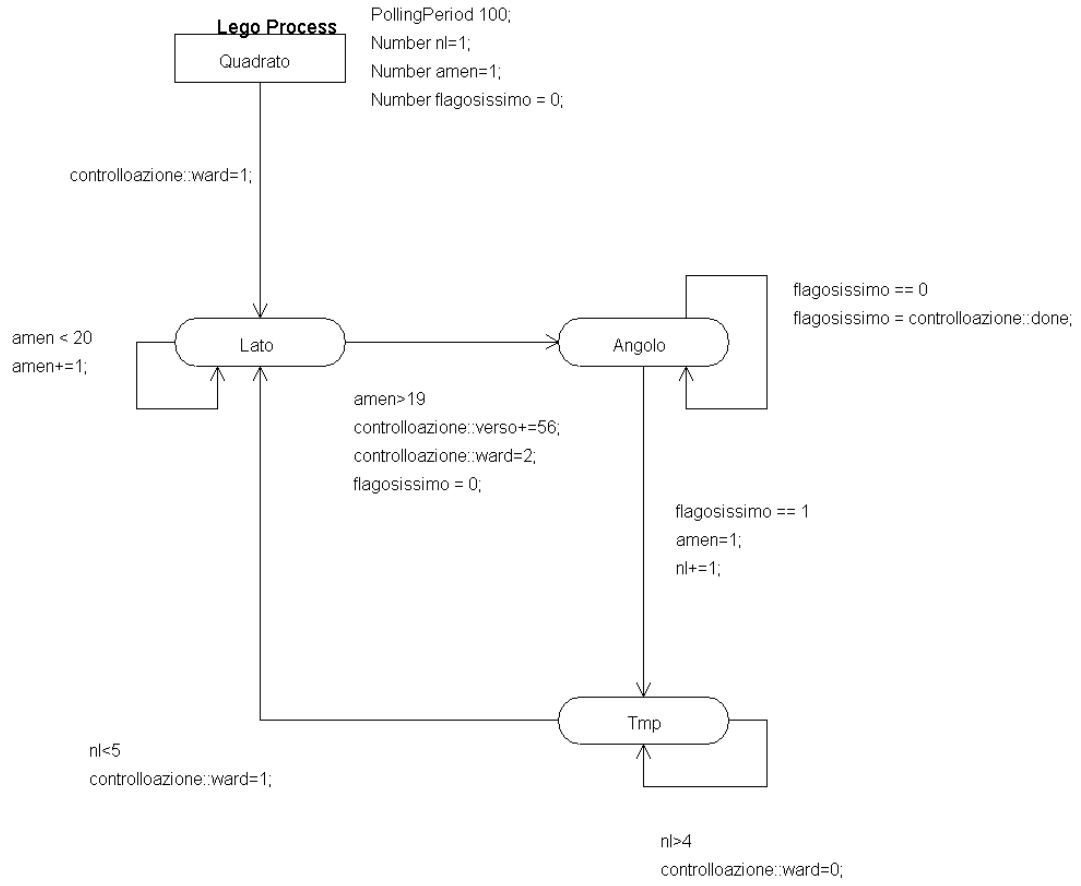


Figura 4.4: Test odometrico

4.2.2 Algoritmo 'zenzero'

Questa macchina a stati implementa l'algoritmo che impartisce le istruzioni per eseguire il percorso descritto nel secondo capitolo. Vediamo in figura 4.5 il processo.

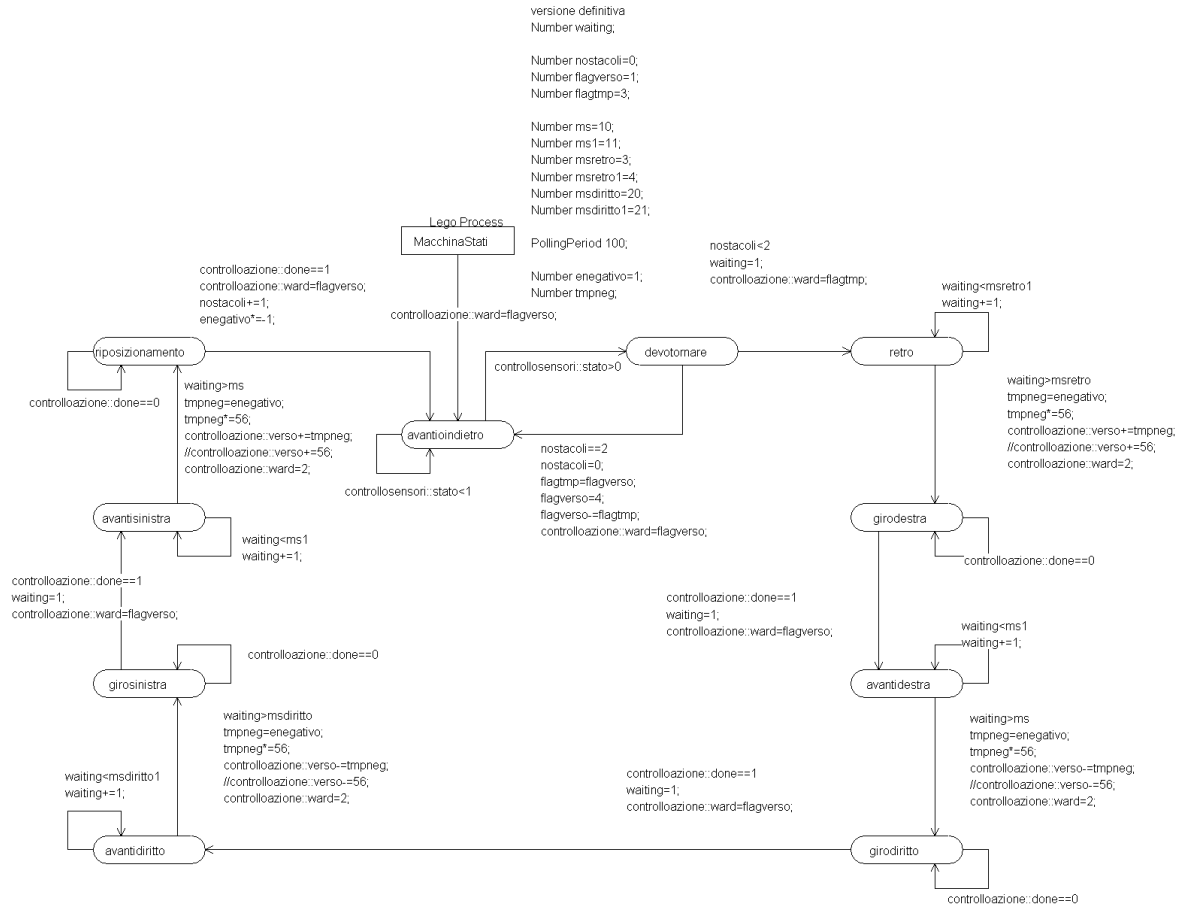


Figura 4.5: Processo MacchinaStati

Lo stato normale è la marcia in avanti oppure indietro, a seconda del segno di 'flagverso', che indica se si sta percorrendo l'andata oppure il ritorno del percorso. Nel momento in cui 'controllosensori' rileva un urto, diventa vera la condizione sulla transizione che esce dallo stato di marcia normale, e viene eseguita la catena di stati necessari al superamento dell'ostacolo. Per primo si controlla se l'urto è il terzo incontrato sul percorso, nel cui caso si esegue il percorso al rovescio invertendo gli opportuni flag.

Altrimenti si passa all'esecuzione dell'anello composto da 8 stati che serve per aggirare l'ostacolo; gli stati corrispondono alla sequenza di azioni:

1. Inversione di marcia per circa 10cm
2. Rotazione di 90 gradi in senso orario
3. Avanzamento per circa 25cm
4. Rotazione di 90 gradi in senso anti orario
5. Avanzamento di circa 60 cm (per il superamento dell'ostacolo)
6. Rotazione di 90 gradi in senso anti orario
7. Avanzamento di 25cm
8. Rotazione di 90 gradi in senso orario

Facendo delle prove sul sistema finale abbiamo notato che i tempi necessari al superamento degli ostacoli erano errati. Essendo questi troppo brevi, non vi era il tempo di oltrepassare gli ostacoli; si nota come gli ostacoli debbano essere di dimensioni note a priori!
 Il diagramma finale con tutte le macchine a stati (API e controllore), funzionanti correttamente è rappresentato in figura 4.6.

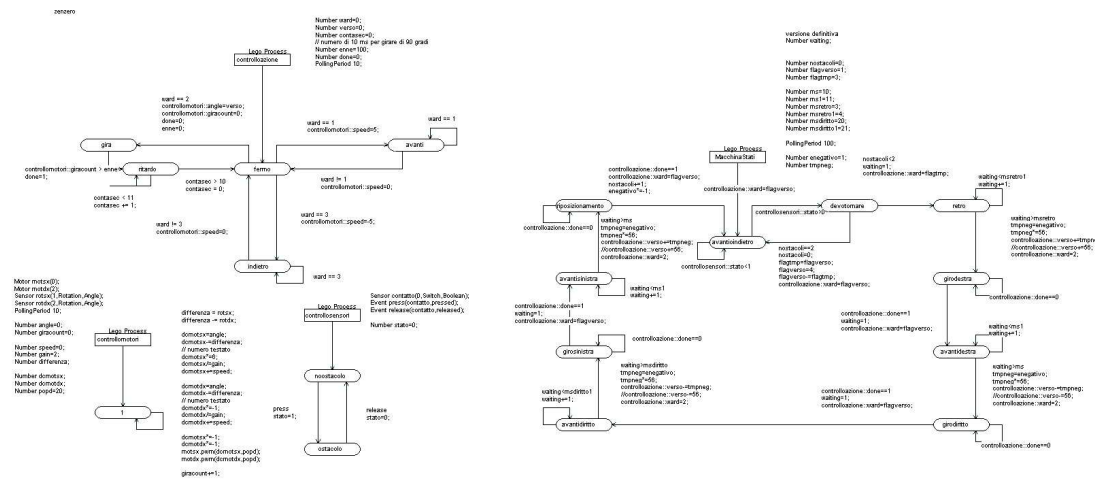


Figura 4.6: Tutto il progetto 'zenzero'

Capitolo 5

Conclusione

Siamo soddisfatti dal lavoro svolto in quanto gli obiettivi prefissati sono stati raggiunti.

Abbiamo però incontrato parecchie difficoltà nella calibrazione del mezzo, dovute alla bassa precisione dei cingoli, alla superficie su cui viaggia il veicolo ed allo stato delle batterie del modulo RCX.