



UNIVERSITÀ DI BRESCIA
FACOLTÀ DI INGEGNERIA
Dipartimento di Elettronica per l'Automazione

Laboratorio di Robotica Avanzata
Advanced Robotics Laboratory

Corso di Robotica
(Prof. Riccardo Cassinis)

Progetto
Neogongolo

Elaborato di esame di: **Marco Merigo, Luca**
Alberici, Nicola Gatta

Consegnato il: **15 luglio 2002**

Sommario

Il progetto Neogongolo, come il titolo lascia presagire, consiste in una “reingegnerizzazione” del robot Gongolo.

Più precisamente questo progetto costituisce la prima fase della ristrutturazione di Gongolo. Ovvero l’implementazione di un nuovo sistema di controllo.

Il sistema di controllo originario si basava su un microcontrollore PIC.

Scopo del progetto è quello di sostituire il PIC con un microcontrollore più complesso il MC68HC11(E2) di Motorola, mantenendo le stesse funzioni e le stesse regole comportamentali già attuate dal PIC.

Il 68HC11, comunque, permette ulteriori sviluppi futuri di comportamenti più articolati e un’interfacciabilità col mondo esterno più completa, da qui le ragioni della “migrazione” dal sistema PIC al sistema di Motorola.

Il progetto si articola, fondamentalmente, su due piani: il piano hardware e quello software.

Dal punto di vista hardware gli interventi eseguiti sono la modifica dell’interfaccia elettrica per la nuova scheda 68HC11-labe, il montaggio di questa sul robot e la costruzione di una nuova scatola comandi.

Per quanto concerne il piano software, si tratta di programmare il nuovo microcontrollore mediante linguaggio assembly 68HC11, per eseguire routine di pilotaggio dei motori e di risposta alla presenza di eventuali ostacoli percepiti dai sensori posti sulla parte frontale del robot.

1. Introduzione

1.1. Caratteristiche di Gongolo

Gongolo è un robot in grado di muoversi, mediante ruote, in ambienti con ostacoli, è dotato di un sistema sensoriale a “baffi” che permette di percepire un ostacolo prima di una possibile collisione e di un sistema di controllo con il quale è possibile prendere un’iniziativa adeguata per evitare un ostacolo frontale.

Il sistema di controllo era originariamente implementato mediante un microcontrollore PIC. Il micro generava un segnale ad onda quadra in grado di pilotare i motori a passo per il movimento rettilineo del robot. Per la partenza da fermo era prevista una rampa di accelerazione (onde quadre a periodo uniformemente decrescente) che permetteva ai motori di vincere l’inerzia del robot. I motori a passo, infatti, necessitano di rampe di accelerazione per partire da posizione stazionaria.

Il sistema sensoriale a “baffi” è costituito da due fili di ferro ricurvi posti davanti al robot in modo tale da ricoprire la parte frontale e connessi a un sensore di vibrazione il quale è in grado di fornire un segnale logico (per esempio di 5 volt) in corrispondenza di un urto, e quindi di una leggera vibrazione (ad alta frequenza), da parte del filo di ferro.

In risposta al segnale proveniente da uno dei due baffi anteriori (sintomo della presenza di uno o più ostacoli) il PIC faceva indietreggiare il robot modificando il bit di direzione, in ingresso agli integrati di interfaccia con i motori, successivamente prevedeva una manovra di rotazione assiale di circa 60 gradi e una ripartenza.

1.2. Struttura della relazione

Lo schema con cui è scritta questa relazione è molto semplice.

Nel capitolo successivo è descritto il problema affrontato che costituisce l’obiettivo del progetto. In questo capitolo sono elencate le specifiche richieste per la modifica del sistema di controllo del robot e non si parla, volutamente, della soluzione adottata.

La descrizione del metodo risolutivo usato costituisce l’oggetto del terzo capitolo.

Il quarto capitolo è invece un manuale d’uso per l’utente, ovvero una descrizione dettagliata di come valutare i risultati ottenuti dal progetto.

Una sintesi dei possibili sviluppi futuri è raccolta nel capitolo conclusivo di questo elaborato.

2. Il problema affrontato

Il problema da risolvere consisteva, come già anticipato nel sommario, nel sostituire il sistema di controllo basato sul PIC con un microcontrollore più completo e adatto a sviluppi futuri. Per ottenere questo obiettivo bisognava togliere la vecchia scheda su cui era installato il PIC e montarne una nuova dotata di un microcontrollore dalle potenzialità maggiori.

Il PIC possedeva una scatola comandi semplicissima costituita dal semplice interruttore di accensione dell'intero sistema, bisognava quindi prevedere una possibile modifica della scatola comandi e dell'interfacciamento elettrico tra controllore, gli attuatori e i sensori del robot.

Inoltre bisognava reimplementare le routine software che regolano il comportamento del robot, ovvero che gestiscono il movimento e la "sensorialità" di Gongolo.

I problemi da risolvere sono stati quindi molteplici, in sintesi era necessario:

- Capire il funzionamento del sistema di controllo ovvero come si comportava il PIC nei confronti degli attuatori (motori a passo) e dei sensori (baffi)
- Scegliere un microcontrollore adeguato ai requisiti del progetto (nella fattispecie, la scelta era limitata alla versione del 68HC11 che meglio rispondeva alle esigenze di programmazione)
- Collegare elettricamente il nuovo microcontrollore ai sensori e agli attuatori, il collegamento fisico dipendeva da una scelta preliminare dei port da utilizzare in ingresso e in uscita.
- Il montaggio sul robot della scheda contenente il microcontrollore e i relativi chip di interfaccia (buffer del RS-232, jumper di configurazione, connettore a 40 poli per i port del micro).
- Implementare le stesse funzionalità con il nuovo microcontrollore, il che implicava una programmazione di basso livello in linguaggio assembly del micro.

L'ultimo punto necessitava di un'ulteriore scomposizione in sottofasi:

- Progetto delle routine di generazione delle onde quadre a fase variabile per le rampe di accelerazione dei motori a passo.
- Progetto delle routine di ispezione periodica (polling) dei baffi e di risposta ad eventuali ostacoli
- Studio delle modalità di programmazione da utilizzare: interrupt o polling
- Progetto di un ciclo principale che includeva le routine precedenti
- Scrittura del codice compilazione e inserimento nella EEPROM del microcontrollore.
- Test del robot in ambiente libero e con ostacoli.

3. La soluzione adottata

Il Motorola MC68HC11 è il microcontrollore più adatto a questo scopo. Nella fattispecie, la versione E2 da noi scelta in fase di progettazione, è dotata di un core più performante, di una più ampia possibilità di interfacciamento con l'esterno e di dispositivi di memoria più capienti.

Questa soluzione, grazie alla più elevata interfacciabilità e alle prestazioni migliori offerte dal microcontrollore della motorola, permette una upgrade futuro delle funzionalità del robot, l'implementazione di comportamenti più "intelligenti" e il potenziamento della "sensorialità".

Come specificato nel precedente capitolo, le due macrofasi in cui l'implementazione del progetto si divideva erano: montaggio e collegamento elettrico della scheda 68HC11-labe con relativa scatola comandi e la programmazione di basso livello del microcontrollore.

3.1. Montaggio e interfacciamento

La soluzione adottata per il montaggio della scheda 68HC11-labe è stata influenzata dal suo discreto ingombro. Infatti la scheda è in formato standard Europa (100x160mm) dotata di una superficie millefori per eventuali aggiunte future di componenti elettronici. Questa dotazione la rende più ingombrante del dovuto e ci ha costretti ad adottare un montaggio, per così dire a "wafer" dove la scheda di controllo costituiva il livello superiore di una collocazione delle schede a tre livelli. Con questa soluzione, infatti, abbiamo potuto mantenere la disposizione originaria delle altre schede e della batteria di alimentazione del robot. La scheda di controllo è stata montata sopra la scheda di pilotaggio dei motori la quale a sua volta è montata sopra la scheda di interfaccia con i baffi.

La struttura del robot risulta più alta e come conseguenza di ciò, abbiamo costruito una nuova carrozzeria per la copertura di gongolo.

L'interfacciamento elettrico tra 68HC11-labe con attuatori e sensori è stato implementato mediante connettore a vaschetta DB9, come quello usato dallo standard RS-232.

Dei 9 pin messi a disposizione dal connettore ne sono stati usati 8.

I fili saldati al connettore sono i seguenti:

- Due fili costituiscono gli ingressi del port C, per ricevere i 2 segnali logici dei baffi.
- Due fili costituiscono le uscite degli output compare (port A) del microcontrollore utilizzati per la generazione dell'onda quadra (uno per motore).
- Due fili sono i flag di direzione connessi alle uscite del port B (uno per motore)
- Un filo costituisce l'enable per i motori (port B).
- Un filo è usato per la massa comune alle tre schede (controllo, motori, sensori).

L'esigenza di costruire una nuova scatola di comando è stata dettata dalle differenti modalità di funzionamento del 68HC11. I comandi sono descritti in dettaglio nel paragrafo 4.1 (Scatola comandi).

3.2. Programmazione

Il microcontrollore MC68HC11/E1 è stato da noi programmato mediante linguaggio assembly per CPU 68HC11. Essendo il core di tipo CISC (derivante dalla famiglia 68000 di motorola) il linguaggio assembly relativo dispone di istruzioni piuttosto "ingombranti" che a volte occupano diversi byte in memoria.

Prima di scrivere il codice è stato fatto un progetto di massima delle routine mediante un flowchart (si veda lo schema riportato di seguito).

Inoltre, con l'ausilio di un oscilloscopio, sono state fatte diverse prove per valutare le forme delle onde quadre generate dal port A del microcontrollore e sono stati analizzati i segnali uscenti dalla scheda di interfaccia dei baffi.

In sostanza i problemi da affrontare in fase di programmazione, come specificato nel precedente capitolo, erano legati alla generazione dell'onda quadra a periodo variabile (per le rampe di accelerazione) e l'acquisizione del segnale proveniente dai baffi.

La soluzione adottata è stata quindi:

- Generazione di segnale periodico mediante l'uso di interrupt proveniente dall'output compare. Questo "lancia" un interrupt software nell'istante in cui il timer raggiunge un valore prefissato e specificato in un registro dedicato. Incrementando ad ogni interrupt il registro specifico con un certo valore si ottiene un'uscita sul port A periodica (simile ad un'onda quadra). La variazione del periodo dell'onda si ottiene modificando di volta in volta il valore sommato nel registro dell'output compare. In particolare per compensare l'accelerazione con andamento esponenziale dei motori a passo è stata implementata una routine che aumenta di volta in volta questo valore seguendo una legge approssimabile ad una curva logaritmica (in realtà per semplicità di programmazione la curva è una spezzata con andamento simile a quello di una logaritmica).
- Polling del segnale proveniente dai baffi in ingresso sul port C. Ogni 100 millisecondi circa, viene ispezionato lo stato del port C (port bidirezionale configurato, nel nostro caso, come ingresso)

Nel caso uno dei due baffi è stato colpito da un ostacolo (che si traduce in un 1 logico in ingresso su uno dei due bit del port C) la routine di controllo prevede:

- una rapida decelerazione del robot (dilatazione repentina del periodo delle onde quadre sui due motori)
- una manovra di retromarcia di qualche decina di centimetri (manovra eseguita mediante negazione del bit di direzione e generazione di onde quadre a periodo costante identiche per i due motori)
- una manovra di rotazione assiale di circa 60 gradi (questa routine prevede la generazione di onde quadre a periodo costante identiche per i due motori, la rotazione avviene grazie alla differente direzione in cui ruotano i motori dato che un bit di direzione è posto a 0 e l'altro a 1 con un criterio che dipende da quale baffo è stato urtato ovvero dalla posizione dell'ostacolo davanti al robot)

- una ripartenza gestita dalla stessa routine usata per l'avvio del robot (routine che, quindi, genera onde quadre a periodo variabile che costituiscono le rampe di accelerazione per i motori a passo).

Una copia cartacea del sorgente in linguaggio assembly, contenente l'intero programma da noi scritto per il progetto Neogongolo, è stata allegata a questa relazione.

Il sorgente in formato .asc è stato compilato mediante il programma assemblatore per 68HC11 ed è stato scritto nella EEPROM del microcontrollore mediante il programma di monitor-debugger PCBUG11 della Motorola. Le routine riportate nell'allegato sono, di conseguenza, usate dal sistema di controllo del robot. Per averne una dimostrazione si legga preventivamente il capitolo successivo (Modalità Operative).

Segue l'immagine del flow-chart che descrive il comportamento di base del robot.

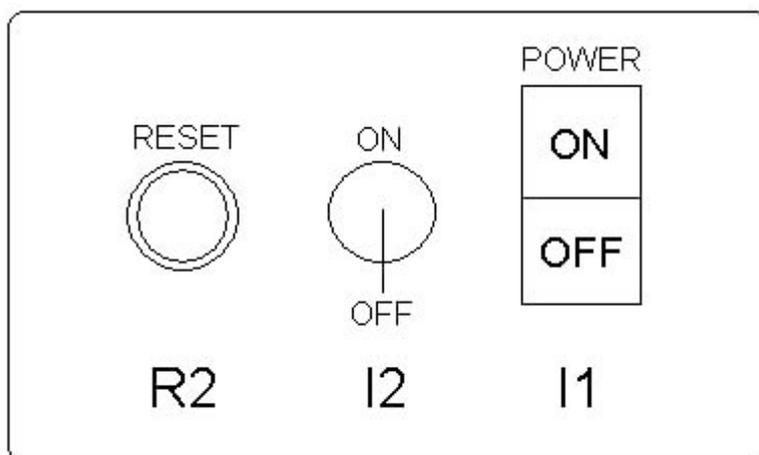
4 .Modalità operative – manuale dell'utente

4.1. Scatola comandi

L'immagine sotto mostra la nuova scatola comandi del sistema di controllo di Gongolo.

Sono presenti due interruttori e un pulsante:

- “Power” – Interruttore I1 - (posizioni possibili ON - OFF): chiude il circuito di alimentazione del robot (motori, sistema di interfaccia dei motori, scheda di interfaccia dei sensori, sistema di controllo).
- “Switch della modalità operativa” – Interruttore I2 – (posizioni possibili ON – OFF): Il microcontrollore MC68HC11-E2 prevede due modalità di funzionamento, “Single chip” e ”Special bootstrap. Il micro deve essere avviato in modi “Single chip” (posizione di I2 su OFF) e poi, per l'esecuzione del programma in eeprom deve essere posto in modi “Special bootstrap” (posizione di I2 su ON).
- “Reset” – Pulsante R1 – : Il microcontrollore deve essere resettato dopo il cambiamento di modalità operativa.



4.2. Avvio del Robot

Grazie alla nuova scatola comandi realizzata nel corso del progetto, per avviare il roboto basta seguire le istruzioni seguenti:

- 1) Assicurarsi che l'interruttore argentato (I2), posto sulla scatola comandi, sia posizionato su OFF.
- 2) Accendere il Robot mediante l'interruttore generale (I1) posto sulla scatola comandi.
- 3) Spostare l'interruttore argentato (I2) nella posizione ON (modalità special-bootstrap del microcontrollore)
- 4) Premere brevemente il pulsante (R1) (reset del microcontrollore, posto anch'esso sulla scatola comandi) per due volte.

A questo punto il Robot parte e, dopo una breve fase di accelerazione (dovuta alle rampe di accelerazione generate per i motori passo) prosegue con movimento rettilineo uniforme fintanto che non incontra un ostacolo.

4.3. Uso del Robot

Una volta avviato il Robot non necessita, in generale, di intervento umano. Il sistema sensoriale costituito dai baffi anteriori, gli permette di rilevare un ostacolo ed il sistema di controllo, da noi implementato, prende le dovute iniziative per fermare il robot e tentare un cambiamento di direzione adeguato per evitare nuovamente l'ostacolo.

Può succedere che la presenza di ostacoli non percettibili dai baffi forzi un intervento dall'esterno per ripristinare la posizione del robot.

4.4. Arresto del robot

Per arrestare il Robot basta posizionare l'interruttore generale (I1) su OFF.

E consigliabile posizionare, successivamente allo spegnimento del robot, l'interruttore argentato (I2) su OFF per un successivo riavvio più rapido (si veda: 4.1. Accensione del Robot).

5. Conclusioni e sviluppi futuri

L'obiettivo stesso di questo progetto è quello di dare la possibilità a sviluppi futuri.

Il potenziamento, per così dire, del sistema di controllo passando dal microcontrollore PIC, più limitato in termini di interfacciamento e memoria a disposizione, al sistema MC68HC11, più performante e maggiormente interfacciabile, permetterà in seguito di migliorare la sensorialità di Gongolo. Usando sensori a lama di luce, sonori o, persino, basati su GPS, il microcontrollore della motorola può ricevere informazioni più precise sull'ambiente circostante e rispondere con routine più adeguate alla presenza di eventuali ostacoli.

Questo può essere reso possibile dalle più ampia connettività del 68HC11 anche se la memoria EEPROM a disposizione del programmatore può rivelarsi insufficiente per routine troppo complesse.

A tal proposito la nostra esperienza in laboratorio testimonia che anche un comportamento "semplice" come quello implementato attualmente su Gongolo, può richiedere l'impiego di numerose risorse in termini di tempo di programmazione e di memoria usata per la scrittura del programma.

Suggeriamo quindi, per eventuali lavori futuri su Gongolo, di fare un preliminare studio di fattibilità relativo a miglorie del sistema sensoriale del robot.

Non è esclusa la necessità di una versione del microcontrollore in cui si facci uso della modalità expanded multiplexing per usufruire di una memoria esterna più capiente.

Bibliografia

- [1] A. Flammini : “Progettazione a microprocessore”, Laboratorio di Elettronica Digitale A.A. 1998-99
- [2] Manuale del microcontrollore MC68HC11, Motorola, 1985

Indice

SOMMARIO	1
1. INTRODUZIONE.....	2
1.1. Caratteristiche di Gongolo	2
1.2. Struttura della relazione	2
2. IL PROBLEMA AFFRONTATO.....	3
3. LA SOLUZIONE ADOTTATA.....	4
3.1. Montaggio e interfacciamento	4
3.2. Programmazione	5
4. MODALITÀ OPERATIVE – MANUALE DELL’UTENTE	9
4.1. Scatola comandi	9
4.2. Avvio del Robot	9
4.3. Uso del Robot	10
4.4. Arresto del robot	10
5. CONCLUSIONI E SVILUPPI FUTURI.....	11
BIBLIOGRAFIA.....	12
INDICE	13