



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

Laboratorio di Robotica Avanzata **Advanced Robotics Laboratory**

Corso di Robotica Mobile
(Prof. Riccardo Cassinis)

Survivor, sedia che cammina.

Elaborato di esame di:

Tosoni Cristian

Consegnato il:

08 Luglio 2009

Sommario

“Survivor è una macchina che utilizza l'approssimazione per ottenere un modello comportamentale riconoscibile.”

Lo scopo di questo progetto è la riattivazione di una seconda macchina costruita sullo stesso modello dell'originale sedia Survivor. In particolare si vuole: raccogliere il materiale relativo al progetto, identificare gli strumenti adatti alla programmazione della macchina e la stesura di un nuovo programma in linguaggio C, che mantenga le funzionalità del programma originale scritto in linguaggio assembler.

1. Introduzione

*“L’opera **Sedia che cammina** fa riferimento al progetto **il soldato perfetto** ideato dal fotografo Giovanni Diffidenti che da un decennio documenta i sopravvissuti delle mine anti-uomo. Nel 2000 è iniziata la campagna di sensibilizzazione e il sito internet www.theperfectsoldier.com racconta il progetto nella sua globalità.*

*Nel 1999 nasce l’idea **Sedia che cammina** come riflessione artistica sul problema delle vittime.*

Un oggetto quotidiano e rassicurante come una sedia diventa un ordigno antropomorfo che si muove lentamente, evita qualsiasi contatto, un po’ bomba-carnefice, un po’ amputato-vittima.

Si porta addosso la memoria di un meccanismo inarrestabile e perverso.”

Così Laura Morelli, curatrice del progetto **Sedia che cammina**, introduce all’opera.

Da un punto di vista non artistico è possibile descrivere brevemente la sedia come segue.

La *sedia* prende forma da una sedia metallica con seduta in legno, simile a quelle presenti normalmente nelle scuole elementari. Le sue gambe anteriori sono state tagliate appena sotto la seduta e modificate in modo da permettere loro di effettuare dei goffi passi, e alla sedia di camminare. Il movimento delle due *gambe* avviene tramite l’azionamento di due motori elettrici, che in origine muovevano il meccanismo dei tergicristalli di un’automobile e che possono essere comandati tramite PWM (Pulse Wave Modulation). La sedia presenta inoltre una serie di 8 sensori agli infrarossi applicati alla seduta per misurare la distanza da eventuali ostacoli in 8 direzioni. Il funzionamento della sedia e la simulazione di un comportamento “umano” sono gestiti da un PIC 16F876 applicato a una scheda montata sul retro dello schienale, su tale scheda sono anche presenti, e controllabili dal PIC, un buzzer per l’emissione di suoni e un led di colore rosso.

Le dimensioni approssimative della sedia prima dell’applicazione delle modifiche sono: lunghezza 40 cm, larghezza 30 cm, altezza 84 cm e peso 3,2 kg.

2. Il problema affrontato

Le problematiche affrontate dallo studente possono essere suddivise in due gruppi, la comprensione del funzionamento della meccanica e dei dispositivi che compongono la sedia, e l’analisi del programma in linguaggio assembler precedentemente caricato nel PIC di controllo unitamente alla stesura di un nuovo programma in linguaggio C che ricalchi le funzionalità del precedente applicativo.

2.1. La meccanica e i dispositivi

La componente meccanica della sedia risulta essere di semplice comprensione, infatti, le uniche parti mobili della sedia sono le due gambe anteriori azionate dai due motori per tergicristalli.

Com’è possibile intuire dalle immagini seguenti a un giro del motore corrisponde un passo effettuato dalla gamba. Nell’immagine di sinistra è visualizzata la gamba destra della sedia nella posizione di riposo, mentre nell’immagine a destra la stessa gamba durante l’esecuzione di un passo.



Fig. 1 - Meccanica delle gambe.

È necessario precisare che l'unico controllo che si ha sul movimento dei motori consiste nella scelta della velocità di rotazione, ciò è possibile tramite la generazione di un'onda PWM, un'onda quadra, inviata ai motori. Inoltre, l'unico feedback che è possibile ottenere dai motori proviene da un sensore che indica quando il motore raggiunge una determinata posizione utilizzata come *posizione di riposo*. Il raggiungimento di tale posizione indica che la gamba della sedia ha terminato di eseguire un passo.

Per quanto riguarda i sensori agli infrarossi è possibile dire che essi hanno un raggio d'azione effettivamente utilizzabile che va da un minimo di 10 cm a un massimo di 80 cm. Il segnale che il sensore mostra come risposta alle rilevazioni effettuate è aggiornato ogni 32ms ed ha una corrispondenza non lineare con la distanza misurata, come è evidente dal grafico seguente.

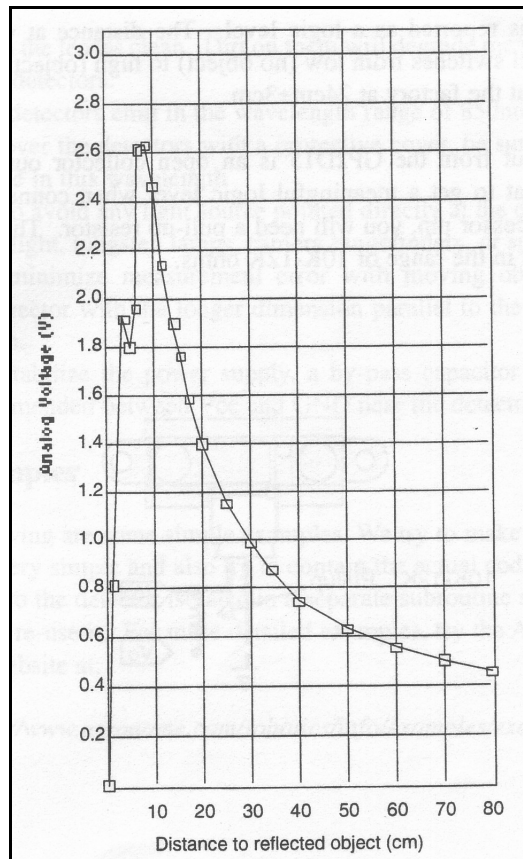


Fig. 2 - Risposta dei sensori IR.

La componentistica che richiede più attenzione e impegno per essere analizzata risiede nella circuiteria di controllo e nell'analisi del PIC16F876 che rappresenta il cuore del funzionamento della sedia.

Di seguito è riportato lo schema circuitale che permette al PIC di interfacciarsi con i motori, gli 8 sensori IR, il buzzer ed il led rosso.

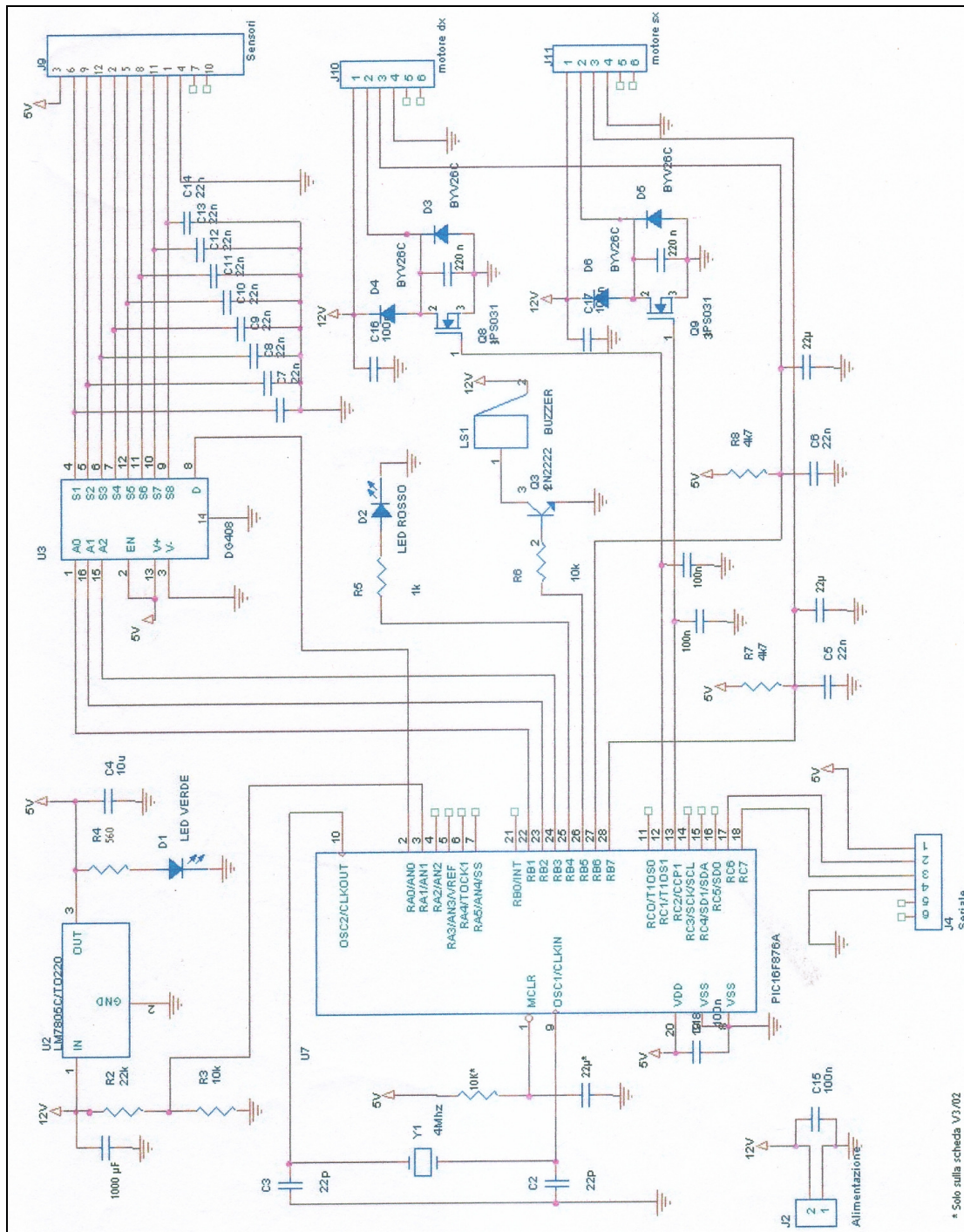


Fig. 3 - Schema circuitale.

In particolare si può notare come gli 8 sensori siano collegati al PIC tramite un Multiplexer che permette di utilizzare tre uscite del PIC (piedini 22, 23 e 24) per selezionare il sensore da interrogare ed effettuare la lettura da un unico ingresso (piedino 2). Al contrario gli altri elementi sono connessi direttamente al

PIC, i motori destro e sinistro sono controllati rispettivamente tramite i piedini 12 e 13, mentre i relativi feedback possono essere rilevati sui piedini 27 e 28. Il led rosso è controllato tramite l'uscita corrispondente al piedino 25, mentre il buzzer tramite il 26. Inoltre, è possibile eseguire misurazioni sullo stato di carica della batteria tramite letture sul piedino numero 3.

2.2. Dal Programma in assembler al nuovo programma in C

Un secondo insieme di problematiche, che sono state affrontate, abbandona la meccanica e l'elettronica per concentrarsi sulla logica che muove la sedia in modo che simuli una serie di comportamenti intelligenti.

Questa parte del lavoro prende il via dal listato del vecchio programma che è stato scritto in linguaggio assembler originariamente da Belleri e Lucchini nell'agosto del 2003, e in seguito modificato dal professor Riccardo Cassinis nel settembre dello stesso anno.

In primo luogo sarà necessario analizzare il programma in linguaggio assembler per comprendere la logica e le funzionalità da esso implementate; in un secondo momento si passerà alla scrittura di un programma in linguaggio C che simuli gli stessi comportamenti dell'originale ma che risulti di maggior comprensione e manipolazione da chi successivamente vorrà lavorare con la sedia Survivor.

3. La soluzione adottata

Anche in questa sezione si distinguerà tra il lavoro effettuato sulla meccanica e sulla componentistica della sedia ed il lavoro di stesura del nuovo software in ingaggio C.

3.1. La meccanica e i dispositivi

Per quanto riguarda la struttura fisica della sedia, dopo un confronto tra le fotografie della sedia originale e quella in nostro possesso, si è notato che l'assetto dei due esemplari era diverso. Credendo che tale differenza potesse influire sulle capacità di movimento della sedia è stato deciso di portare la nostra a un confronto faccia a faccia con l'opera originale. Ne è conseguita la necessità di smontare la piastra, posta sotto la seduta della sedia e che funge da supporto per i motori, e rimontarla al contrario, ossia nella posizione corretta. Dopo quest'operazione sono state necessarie solo piccole modifiche alle distanze tra gli snodi delle gambe per dare alla sedia l'assetto corretto.

Consideriamo ora i dispositivi collegati alla sedia, in particolare i sensori agli infrarossi e la sezione di circuito per la misurazione del livello di carica della batteria.

Dai manuali tecnici e dallo schema circuitale di Figura 3 si nota come il livello di tensione che il PIC riceve al piedino 2 può variare da un minimo di 0V ad un massimo di 5V, mentre al piedino 3 si andrà da un minimo di 0V ad un massimo di 4V. Tali piedini corrispondono nel nostro caso agli ingressi che effettuano la codifica analogico-digitale, sarà quindi possibile all'interno del programma ottenere le letture delle tensioni sui piedini. Bisogna però tener conto che tali valori di tensione saranno rappresentati come numeri interi positivi a 8 bit, all'interno del PIC, e che di conseguenza assumeranno valori compresi tra 0 e 255. Inoltre il PIC ha bisogno di due riferimenti per effettuare la codifica analogico-digitale, ossia i valori di tensione da associare ai numeri 0 e 255; tali riferimenti vengono dati rispettivamente tramite i piedini 19 e 20. Dalla Figura 3 è possibile vedere che a tali piedini sono associati nell'ordine i valori di tensione di 0V e di 5V.

Per tanto, per interpretare correttamente una lettura effettuata dal programma, è necessario tenere in considerazione la seguente proporzione: **valore-analogico : 5 = valore-digitale : 255**.

3.2. Dal Programma in assembler al nuovo programma in C

Per la struttura del nuovo programma in C è stato preso spunto dall'originale programma in codice assembler, leggendo il listato è stato possibile identificare le principali routine utilizzate e quindi le

principali azioni che sono necessarie per il funzionamento della sedia, tra queste sono di particolare importanza quella per l'acquisizione dei dati dai sensori, quella per la gestione dei motori e quelle per la gestione degli interrupt.

Tra queste ultime è rilevante la routine che viene eseguita quando è generato l'interrupt legato all'overflow del timer1. Il PIC16F876 è dotato di tre timer, in particolare il timer1 viene incrementato ad ogni ciclo di clock del microcalcolatore e memorizzato in un registro a 16 bit, quindi una volta raggiunto il valore 65535 ($2^{16}-1$) il contatore va in overflow e viene generato un apposito interrupt. Tenendo conto della frequenza di clock del PIC si può calcolare che tale interrupt viene generato circa ogni 65ms. Da questo dato è possibile scandire il tempo all'interno del flusso del programma.

Una volta analizzate e riscritte in C le funzioni principali, il lavoro si è concentrato nell'analisi dei comportamenti che la sedia mostra e che sono codificati nel codice assembler. I comportamenti principali venuti alla luce sono cinque:

- la sedia si blocca quando identifica un ostacolo troppo vicino da poter essere evitato;
- cerca di mantenersi ad una certa distanza dagli ostacoli;
- se qualcosa si avvicina rapidamente la sedia va in panico e fugge a tutta velocità;
- se qualcosa si allontana, incuriosita, cerca di inseguirlo;
- ad ogni passo la sedia è più affaticata, in particolar modo quando è in panico, quindi dopo un certo tempo deve fermarsi e riprendere fiato.

4. Modalità operative

Questo paragrafo descrive come utilizzare gli strumenti per la programmazione di Survivor. L'ambiente di sviluppo per la creazione del codice C e del codice compilato, e il programmatore che attraverso un software bootloader pre-installato permette di caricare i programmi compilati sul PIC.

4.1. Componenti necessari

L'ambiente di sviluppo e il programma utilizzato per la programmazione del PIC esistono sia per i sistemi operativi Linux che Windows, tuttavia i driver in nostro possesso per il funzionamento dell'adattatore seriale-USB sono nativi per ambiente Windows, in particolar modo per i sistemi operativi fino a Windows XP. Per chi non avesse la possibilità mettere in funzione l'adattatore senza tali sistemi operativi è consigliabile l'utilizzo di un ambiente di virtualizzazione, quale VMware. Il software da utilizzare non richiede un calcolatore dalle elevate prestazioni, quindi la virtualizzazione è una soluzione adatta.

Per lo sviluppo del programma in linguaggio C è stato utilizzato l'ambiente di programmazione "PCW C compiler IDE". Dopo la stesura del codice, la compilazione può essere eseguita tramite il bottone evidenziato nella figura seguente.

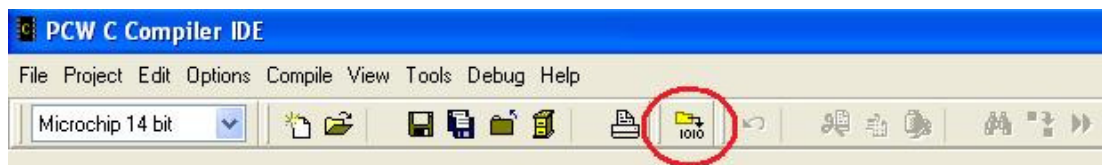


Fig. 4 - PCW C Compiler IDE.

Se la procedura viene eseguita senza errori vengono prodotti alcuni file nella cartella in cui è situato il file sorgente. Tra questi, il file con estensione "HEX" contiene i dati da caricare sul PIC con l'ausilio del programma "PIC Bootloader+".

Per fare ciò è necessario connettere il computer alla sedia, questo è possibile attraverso l'adattatore, etichettato "survivor RS232 adapter", esso si presenta come una scatola di plastica nera da cui escono due cavi, uno termina con un connettore per la linea seriale mentre l'altro con uno a sei contatti che va collegato alla scheda della sedia.

Se il proprio computer non possiede una porta seriale è possibile utilizzare l'adattatore seriale-USB, di cui è presente il CD-ROM con i driver per l'installazione.



Fig. 5 - Survivor RS232 adapter, con adattatore seriale-USB.

Una volta effettuato il collegamento computer-sedia è possibile avviare il programma "PIC Bootloader+". Gli unici parametri da impostare sono la porta attraverso la quale si è connessi alla sedia e il BaudRate, il primo dipende dal proprio sistema operativo mentre il secondo va impostato su 9600. Indicato al programma il file .HEX da caricare si preme il pulsante "Write".

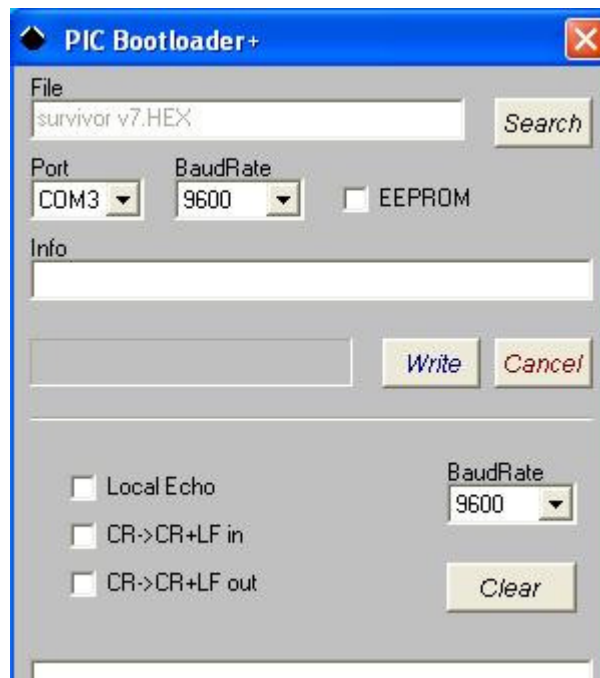


Fig. 6 - PIC Bootloader+.

Subito dopo, con la sedia alimentata, è necessario premere il pulsante presente sulla scheda elettrica in modo da dare il via alla riprogrammazione del PIC (il pulsante è evidenziato nella figura seguente). Tramite il pulsante si resetta il PIC che, identificando il programma “PIC Bootloader+” in attesa, darà il via all’aggiornamento con il nuovo programma utente. Al termine di questa procedura la sedia comincia a eseguire il nuovo software.



Fig. 7 - Scheda presente sullo schienale, evidenziato il pulsante per il reset.

5. Conclusioni e sviluppi futuri

La sedia survivor ha ripreso a funzionare simulando correttamente i comportamenti programmati. Il nuovo programma in linguaggio C si presta ad analisi e possibili evoluzioni meglio del progenitore in linguaggio assembler.

In futuro sarà necessario studiare con maggiore attenzione la meccanica della sedia che non risulta efficiente come quella del modello originale, riducendo la capacità di deambulazione del robot. Particolarmente evidente risulta che il motore che aziona la gamba sinistra è “più debole” di quello destro; in alcune occasioni, quando le batterie tendono a scaricarsi, la gamba sinistra fatica ad eseguire il passo, mentre con la medesima onda PWM la destra lo conclude senza intoppi.

Un’altra questione da approfondire riguarda il mal-funzionamento di alcune funzioni della libreria matematica “math.h”, è stato notato che funzioni come *abs()* e *floor()* non si comportano in modo adeguato. Questo richiede sicuramente un’indagine approfondita se si desidera sfruttare la libreria matematica in progetti futuri.

I comportamenti messi in atto dalla sedia sono regolati da un insieme di costanti arbitrate definite nelle prime righe del codice del programma C; un possibile lavoro futuro potrebbe cercare di rendere tali valori modificabili attraverso un terminale collegato alla porta seriale della sedia, in modo da non dover riprogrammare la sedia ogni volta che si vuole modificare tali parametri.

Bibliografia

- [1] Microchip: "PIC 16F876 Data Sheet", ed Microchip Technology Inc., 2001.
- [2] "C Compiler Manual", July 2003, fonte *PICJuly2003.pdf*.
- [3] Acroname easier robotics: "The Sharp GP2D12, GP2D15, GP2Y0D02YK, GP2Y0A02YK and GP2D120 Infrared Object Detectors", 04/2002.

Indice

SOMMARIO	1
1. INTRODUZIONE	2
2. IL PROBLEMA AFFRONTATO	2
2.1. La meccanica e i dispositivi	2
2.2. Dal Programma in assembler al nuovo programma in C	5
3. LA SOLUZIONE ADOTTATA	5
3.1. La meccanica e i dispositivi	5
3.2. Dal Programma in assembler al nuovo programma in C	5
4. MODALITÀ OPERATIVE	6
4.1. Componenti necessari	6
5. CONCLUSIONI E SVILUPPI FUTURI	8
BIBLIOGRAFIA	9
INDICE	10