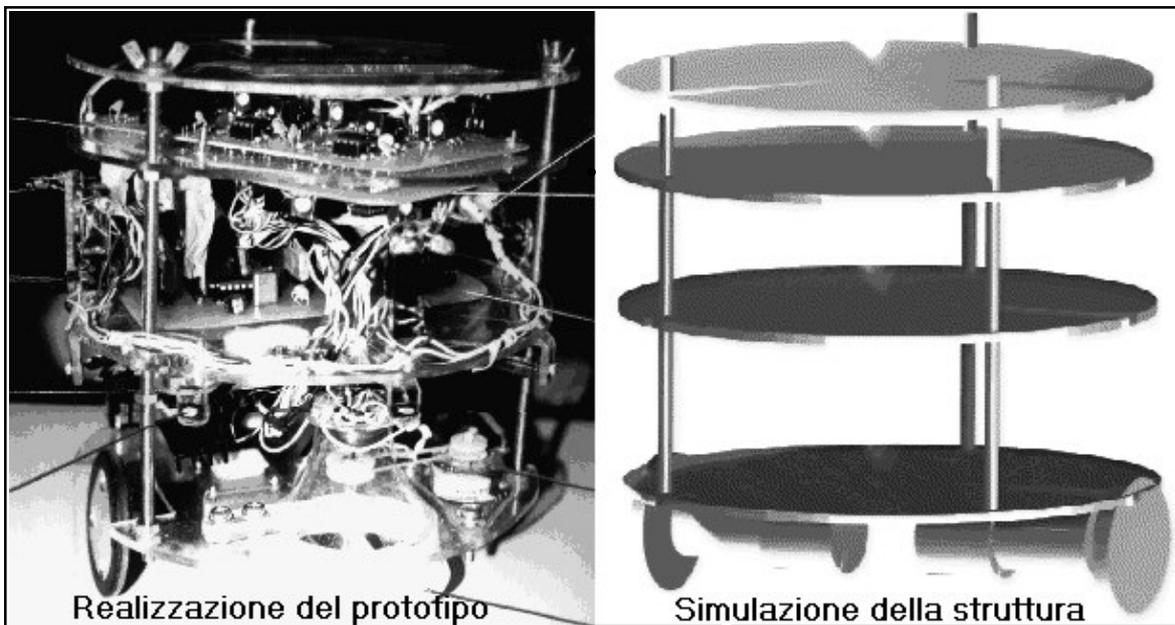


PIC³

Robot semovente, finalizzato alla individuazione di sorgenti (suono, luce e gas), in ambiente piano, con ostacoli casualmente distribuiti



Realizzazione del prototipo

Simulazione della struttura

Esposizione delle fasi di: (1) analisi del problema, (2) definizione dell'architettura, (3) scomposizione in moduli e loro definizione, (4) progettazione e realizzazione dei singoli moduli, (5) verifica funzionale dei moduli, (6) composizione dei moduli, (7) test del sistema finale.

di
Mario Giovanni C. A. CIMINO

Pisa, dicembre 2000.

1. INTRODUZIONE E FINALITÀ

La presente trattazione riguarda il progetto e la realizzazione di un sistema elettronico digitale a logica programmabile, dedicato al controllo di un supporto meccanico automobile su ruote, destinato a raccogliere 4 tipi di segnali, ed effettuare 3 tipi di azioni. In ingresso, misura gli effetti (attraverso i rispettivi trasduttori) di luci (con fotoresistenze), suoni (con microfoni), vapori di alcool (con un rivelatore di particelle di gas), ostacoli presenti nelle immediate vicinanze (con ricevitori ad infrarossi). In uscita (mediante attuatori), governa il proprio moto (con motori elettrici in continua), dirigendosi verso le sorgenti, aggirando eventuali ostacoli (con emettitori ad infrarossi), segnalando la immediata vicinanza ed il tipo di emissione ad un osservatore umano (con led di vari colori).

L'idea e la sollecitudine a realizzarla, provengono da una gara amatoriale tenuta a Pisa ogni anno dalla Scuola Superiore "S. Anna", e dall'Istituto Professionale "Fascetti" separatamente.

La possibilità di progettare secondo una esperienza didatticamente corretta è dovuta invece all'interazione con diverse figure professionali, che ha consentito di acquisire le principali metodologie e conoscenze necessarie: sintesi ed analisi dei dispositivi di comunicazione tra moduli digitali, architetture dei sistemi digitali, microcontrollo, strumenti CAD per circuiti elettronici digitali, teoria del motion control, trasduttori ed attuatori, caratteristiche elettriche dei sistemi digitali, sistemi ausiliari ai circuiti digitali (oscillatori, alimentatori,...), strumenti CAD per circuiti elettronici analogici.

L'analisi delle varie soluzioni progettuali ideate, la risoluzione di numerosi problemi di realizzazione circuitale e meccanica, il 'know how' sull'elettronica applicata, mi è stato trasmesso in maniera diffusa, prima ai ricevimenti dei docenti che conoscevo, poi da alcuni conoscenti al di fuori dell'Università, e infine da qualche professore dell'Istituto Professionale "Fascetti", tutti sotto menzionati, che mi hanno permesso di ridurre sensibilmente i tempi di apprendimento.

Infine, la fase di montaggio delle schede elettroniche, test di verifica dei singoli circuiti, composizione della struttura meccanica, e prime prove di programmazione, è avvenuta in collaborazione con altri quattro studenti, di vari corsi di laurea, con cui ho partecipato alla gara; quest'ultima esperienza ha permesso di conoscere, pur in minima parte, i ruoli necessari a coordinare e incentivare un gruppo di persone con diverse culture accademiche, e diverse personalità.

Ringrazio innanzitutto il professor M. Avvenuti del corso di Sistemi di Elaborazione I, per le competenze atte a raccordare le discipline 'logiche' con quelle 'elettroniche'; così come il prof. R. Saletti del corso di Elettronica dei Sistemi Digitali I, per i consigli sulla stabilizzazione delle tensioni sui circuiti digitali, e sulla simulazione elettrica con SPICE; anche il prof. A. Balestrino di Controlli Automatici I, per la parte relativa alla strategia di ricerca delle sorgenti; ed il prof. A. Landi di Controlli Automatici II.

Ringrazio inoltre sentitamente l'ing. S. Di Pascoli del Corso di Elettronica I, per la disponibilità a ricevermi in qualsiasi orario, e per i contributi sulla parte elettronica del trattamento del segnale sonoro e luminoso, per il circuito di programmazione ed il programmatore del PIC16C84, per i consigli sull'uso di PROTEL sullo sbroglio delle piste, per i metodi di realizzazione dei circuiti stampati.

Altro ringraziamento al prof. M. Ciampi di Fisica Tecnica, per le osservazioni sulla propagazione e la ricezione della luce e del suono.

Fuori dall'Università, ringrazio A. Spinello, titolare dell'azienda "ELETTRONICA ITALIANA", per i consigli sulla scelta, tra gli infiniti prodotti in commercio, del tipo di convertitore analogico digitale, di interfaccia parallela, dei trasduttori, dell'integrato dei motori, dei motori, e per aver permesso il loro acquisto presso la ditta RS Components.

Ringrazio l'ing. S. Roas, professore di Elettronica presso L'Istituto Professionale "Fascetti" di Pisa, promotore della gara, per l'uso del forno ad ultravioletti ed i trapani a colonna, per incidere e forare i circuiti stampati, e tutto quanto ci è servito del laboratorio.

Ringrazio l'ingegnere elettronico Nicola Muto, titolare dell'azienda "DSM INFOTRONICA" per i consigli sui componenti commerciali, sul compilatore e l'assembler del PIC16C84, sulle caratteristiche delle interfacce; e anche l'ingegnere elettronica G. Ruggiero, per il corso di programmazione sull'assembler del PIC16C84, e per la parte di simulazione elettrica con SPICE.

Si ringrazia Mario Consani dei dipartimenti di Fisica, per la realizzazione delle ruote in metallo.

Ringrazio inoltre i partner di sviluppo che hanno collaborato alla costruzione del primo prototipo del robot, che mi ha poi permesso successivamente di rivisitare e perfezionare da principio tutta la progettazione, apportare al robot tutte le modifiche necessarie, e di realizzare tutti gli algoritmi di controllo in assembler: E. Campanelli, studente di Fisica, per la lavorazione ed il montaggio delle parti meccaniche, ed i test su basetta millefori dei circuiti di suono, luce, gas, infrarossi; A. Campana, studente di Informatica, per la discussione sulle strategie possibili, ed alcuni consigli sulla programmazione; P. Imbesi, studente di Ingegneria Elettronica, per il circuito di regolazione dell'offset degli operazionali, i test su basetta millefori dei circuiti di suono, luce, gas, infrarossi, la saldatura dei componenti sul circuito stampato, i fili di collegamento tra schede, lo sbroglio della scheda di alimentazione; L. Piccinini, studente del diploma di ing. elettronica, per la fotoincisione delle basette presensibilizzate, la saldatura dei componenti, la risoluzione di alcuni problemi elettrici dopo il montaggio.

La trattazione seguirà tutta l'evoluzione del progetto, soffermandosi sulle varie scelte adottate, evidenziando i metodi matematici di stima delle grandezze in gioco (i conti con 'carta e matita'), le successive simulazioni con i CAD, una realizzazione con il CAM, e i problemi riscontrati.

2. ANALISI DEL PROBLEMA (tratto dal regolamento della gara)

Il minirobot deve avere dimensioni massime 20x20x25 cm; deve muoversi autonomamente in un campo di esplorazione assegnato, evitare gli ostacoli in posizioni fisse e sconosciute, individuare la posizione delle sorgenti tramite i segnali che emettono: luci, suoni, gas.

Il campo di esplorazione, è una superficie piana di 2x4 m, di colore nero, delimitata da un bordo bianco di altezza 30 cm.

Gli ostacoli, sono dei parallelepipedi di 20x40x30 cm, di colore bianco, poggiati all'interno del campo di esplorazione in una qualsiasi delle proprie superfici laterali, e disposti in modo da formare dei corridoi di larghezza almeno 30 cm.

Le sorgenti, sono inglobate nei bordi o negli ostacoli (luce, suono) o nel pavimento (gas), senza sporgenze; quelle sonore sono poste ad un'altezza di 15 cm, ed emettono un suono direzionale monocromatico ($f=4\text{Khz}$); quelle luminose sono poste ad un'altezza di 10 cm, ed emettono luce bianca direzionale; le sorgenti di gas sono vapori di alcool provenienti dal basso.

La individuazione di una sorgente è valida se il minirobot si arresta per un tempo minimo di tre secondi, ha parzialmente o totalmente attraversato il cerchio di 25 cm di raggio, centrato sulla sorgente, ed ha segnalato la sorgente riconosciuta con un led o con un segnale acustico, di colori o tipi diversi per ogni sorgente.

Le prestazioni del robot vengono valutate con un QM (quoziente macchina), che parte da zero ed incrementa o decrementa a seconda, rispettivamente, che venga riconosciuta correttamente una sorgente, o avvenga un urto:

- 2 urto contro sorgente o ostacolo contenente sorgente
- 1 urto contro ostacolo
- +1 riconoscimento di una sorgente di gas non ancora riconosciuta
- +2 riconoscimento di una sorgente di luce non ancora riconosciuta
- +3 riconoscimento di una sorgente di suono non ancora riconosciuta

3. SCHEMATIZZAZIONE DEL PROBLEMA E DEFINIZIONE DELL'ARCHITETTURA PER RISOLVERLO.

Innanzitutto, prima del filtraggio e l'elaborazione dell'informazione, dovrà avvenire l'incontro tra un sistema meccanico che si muove in un certo modo ed i segnali emessi dalle sorgenti, portatori dell'informazione.

Il percorso completo dell'informazione è il seguente: (1) parte da una sorgente di segnale; (2) viaggia in un campo di esplorazione, venendo riflessa dalle pareti e dagli ostacoli; (3) entra nel sensore; (4) esce come corrente elettrica, dal sensore; (5) viene trattata mediante circuiti (elettronici); (6) viene digitalizzata, ossia campionata e quantizzata, se necessario; (7) entra in un dispositivo di controllo (digitale, o analogico); (8) viene elaborata; (9) entra negli attuatori, generando delle azioni (in questo caso il movimento o la segnalazione luminosa), oppure viene memorizzata nel controllore.

Il sistema quindi, deve misurare le seguenti informazioni:

- livello di intensità luminosa, e sua direzione
 - livello di intensità sonora, e sua direzione
 - livello di concentrazione di alcool
 - livello di distanza dagli ostacoli, e loro direzione;
- e produrre in uscita:
- identificazione di ogni sorgente presente nel campo ed il suo tipo, entro 5 minuti.

I livelli sono delimitati da un minimo, dovuto al rumore di fondo (background level) misurabile sperimentalmente in una posizione del campo lontana da tutte le sorgenti, ed un massimo, (maximum level) misurabile quasi a contatto delle sorgenti.

Si considera, con buona approssimazione, il rumore di fondo pressoché costante entro i cinque minuti di gara, e così pure il massimo livello, dipendendo entrambi dalle condizioni ambientali, successivamente specificate, per ogni tipo di sorgente (luce solare, temperatura, umidità,...)

Anche il segnale vero è proprio è pressoché costante nei cinque minuti, per una data posizione nel campo, poiché è rappresentato dall'ampiezza ricevuta da sorgenti ad emissione costante.

Se immaginiamo di effettuare dei filtraggi mediante circuiti elettrici analogici, che restituiscano solo le ampiezze, allora il segnale ripulito si può sicuramente *campionare e quantizzare*.

Il campionamento è legato al teorema di Shannon, in base al quale tutta l'informazione contenuta in un segnale, in termini di armoniche di Fourier, è recuperabile da campionature effettuate a frequenza almeno doppia della frequenza massima contenuta nel segnale.

La quantizzazione, invece è legata a questioni di errore della funzione ingresso-uscita del sistema, nel senso della teoria del calcolo numerico; infatti un qualsiasi sistema di controllo risponde muovendo degli attuatori, a delle letture ottenute tramite trasduttori; ora, tali attuatori avranno bisogno di una certa precisione, oltre la quale è inutile andare, perché i sistemi fisici hanno un certo attrito e una certa inerzia per cui, sotto un certo livello, l'azione risultata è nulla; allora fissato un limite Δy di risoluzione nell'uscita, se il sistema è stabile BIBO, ossia ha uscite limitate ad ingressi limitati, come deve essere, sicuramente esisterà una variazione Δx nell'ingresso, tale che le corrispondenti Δy siano sempre minori del limite scelto. In altre parole, anche i trasduttori possono permettersi un certo errore di precisione, assorbito dall'errore degli attuatori.

In conclusione, il numero di livelli e di direzioni è limitato, ed il sistema di elaborazione dovrà riconoscere solo un numero finito di alternative possibili.

Ciò significa, dunque, che è possibile *codificare e decodificare l'informazione* di ingresso e di uscita, ed operare sulla sua codifica, tramite un sistema digitale. Operare sulla codifica, vuol dire scindere l'informazione dal mezzo fisico che la porta, perdendo ogni *analogia* con le misure delle grandezze fisiche, ma mantenendo un rapporto biunivoco con esse; in tal modo, si possono realizzare gli algoritmi più sofisticati, praticamente senza alcun limite per la complessità dei programmi, se non quello delle Macchine di Turing.

Si badi bene che ciò è sempre possibile solo nei sistemi di controllo, perché sono progettati per controllare dispositivi reali, quindi con un limite di banda ed una risoluzione limitata; perciò si può far finta, sotto l'ipotesi di stabilità, che non ci siano errori, perché tanto avranno effetti non rilevabili.

Potremmo, in teoria, anche progettare un controllore analogico; e su questo faremo una discussione; ma, come abbiamo detto, questa sarebbe solo una semplificazione matematica di progetto, perché l'informazione computata dal sistema, sarebbe sempre finita nelle alternative possibili.



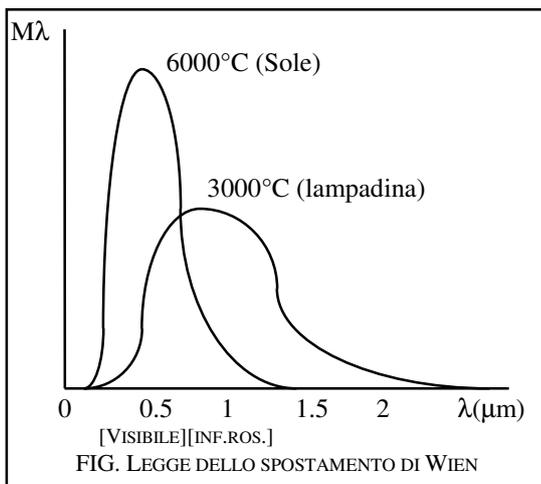
Il sistema dovrà essere necessariamente efficace, ossia individuare tutte le sorgenti entro 5 minuti; le varie soluzioni architettoniche saranno diverse per efficienza, ovvero nella maggiore o minore velocità a raggiungere l'obiettivo; il nostro obiettivo sarà però il rapporto efficienza/costo, oltre che possibilità di realizzazione in breve termine.

E' opportuno sottolineare che identificare una sorgente non vuol dire necessariamente conoscerne la posizione in termini di coordinate; basta segnalarla, in maniera univoca rispetto alle altre, e poi possiamo anche cancellare tutte le tracce di ciò.

Cominciamo ad esaminare il percorso dell'informazione, costruendo man mano le caratteristiche che dovrà avere il sistema per incontrarla. Solo nel prossimo capitolo parleremo di elaborazione.

3.1 Le sorgenti di segnale.

La luce, emessa da lampadine ad incandescenza, è convogliata mediante uno specchio concavo, in un fascio orizzontale, largo 10 cm circa, e situata ad altezza di 10 cm (è simile alle torce elettriche esistenti in commercio). Nell'ambiente è presente una luce solare proveniente dal cielo e diffusa dalle pareti. A noi interessa quindi innanzitutto distinguere la luce della sorgente da quella dell'ambiente.



Possiamo distinguere, di un fascio di luce, la sua distribuzione spettrale, la sua direzione, la sua intensità. Sia la luce del sole che quella della lampadina sono 'bianche' ossia contengono tutte le frequenze dello spettro visibile (in effetti, le lampadine ad incandescenza sono state create apposta per simulare la luce del sole, nelle ore notturne), ma ci sono delle frequenze più intense di altre, per cui se una luce appare rossastra, allora emetterà di più sull'intorno del rosso; se consideriamo la potenza radiante totale M , emessa dall'unità di superficie di un corpo (cioè una densità di energia radiante), in funzione della lunghezza d'onda, allora abbiamo, secondo la legge dello spostamento di Wien, che la frequenza in cui una

sorgente emette il massimo della densità di energia, è linearmente proporzionale con la temperatura. La distribuzione spettrale della luce emessa dalle lampadine ad incandescenza è quella di un corpo nero a 3000°C ; per il sole è a 6000°C ; allora la lampadina avrà il massimo di intensità ad una frequenza doppia di quella del sole; in altri termini, la luce della lampadina emette più energia sul rosso-infrarosso, di quanta ne emette, sulla stessa banda di frequenza, quella del sole, il quale emette di più sull'ultravioletto; allora un filtro polarizzatore rosso (cioè una cartina rossa) filtrerebbe tanto la luce solare, e di meno quella delle torce. Questo è ancora più vero, se si considera che la luce del sole non è diretta, ma riflessa dalle pareti bianche, che in prima approssimazione diffondono con lo stesso coefficiente (minore di uno) tutte le frequenze, o la luce riflessa dal cielo, che trattiene la componente azzurra. Possiamo anche prendere dei trasduttori sensibili di più alle frequenze prossime al rosso.

L'intensità radiante, cioè la potenza emessa dalla sorgente è costante in tutte le direzioni; l'irradianza sul robot (cioè la potenza radiante incidente sopra una superficie unitaria), decade quadraticamente, in prima approssimazione, con la distanza dalla sorgente della superficie considerata (legge dell'inverso del quadrato della distanza), perché la superficie circolare irradiata dal cono di luce varia col quadrato del raggio, il quale varia linearmente con la distanza. Tale dipendenza quadratica però potrebbe essere approssimabile ad una lineare, se il fascio fosse abbastanza stretto; altrimenti, dopo pochi cm la luce decade molto, e quindi difficilmente ci saranno zone illuminate da due sorgenti; possiamo fare l'ipotesi che in ogni regione si possa rilevare nessuna, o una sola sorgente.

La direzione, infine, è fissa a 10 cm; mentre quella ambiente proviene dall'alto; possiamo allora rendere più o meno selettiva la ricerca sulla direzione, mediante dei tubi neri, che permettano alla luce di entrare solo se incide quasi perpendicolarmente, e non obliquamente. Si possono mettere molti trasduttori, e misurare la differenza tra le loro misurazioni, da cui ricavare la direzione.

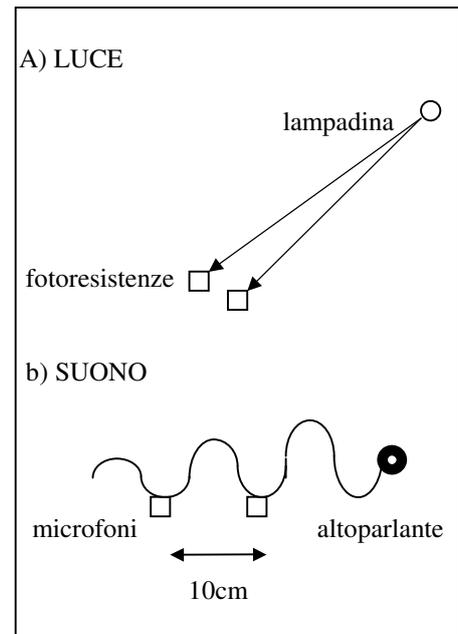
Il suono, proveniente da altoparlanti, è 'monocromatico', cioè ha solo la frequenza dei 4Khz; ma a questo di aggiungono numerosi rumori: pubblico, rumore dei dispositivi meccanici di avanzamento del robot (ingranaggi di riduzione, motorini,), rumore dei dispositivi elettronici. Il filtro sulla frequenza sarà in tal caso effettuato dopo il sensore, con filtri elettronici; in aggiunta si potrebbe mettere tra sensore ed ambiente uno strato di gommapiuma, o altro materiale, che filtri almeno le basse frequenze. Esistono sistema acustici di tubi per filtrare bene le frequenze, ma sono troppo ingombranti. Se il suono viaggia a 360 m/s, e se ha frequenza di 4 Khz, percorre $4000 \cdot \lambda$ in 1 secondo, e quindi $4000 \cdot \lambda = 360$ m, cioè $\lambda = 9$ cm. Quindi ci sono onde sferiche in cui due picchi di compressione o di rarefazione distano 9 cm.

Mentre nel caso della luce, gli effetti della natura ondulatoria sono trascurabili, ai fini della direzione di provenienza, perché le lunghezze d'onda sono dell'ordine dei nm, nel caso del suono abbiamo i tipici comportamenti delle onde: interferenza tra le onde e soprattutto diffrazione. Infatti, secondo il principio di Huygens-Fresnel, ogni punto di una superficie d'onda può considerarsi a sua volta una sorgente di onde con ampiezza decrescente al crescere dell'angolo con la normale al fronte d'onda. Il fenomeno della diffrazione consiste nel fatto che l'onda può aggirare gli ostacoli e propagarsi anche fuori della visuale geometrica, e risulta apprezzabile quando le dimensioni degli ostacoli sono dell'ordine della lunghezza d'onda (si badi NON dell'ampiezza d'onda). Allora, ai fini del suono, tutti gli ostacoli della gara, (dell'ordine dei cm) sono quasi trasparenti, ed il suono proverrà da tutte le parti. Cioè il suono è molto diffuso nell'ambiente, e varierà poco in intensità, con la posizione. Da prove pratiche, si vede

che due microfoni devono almeno distare ad una decina di cm di distanza, per avere differenze apprezzabili. Per raccogliere il suono in maniera direzionale, si può porre un padiglione a trombetta, con curvatura quadratica, divergente verso l'esterno, ed imbottire l'esterno con un po' di gommapiuma. Si ricorda che il legno ha un coefficiente di riflessione fino all'80% sul suono.

Il vapore di alcool infine, proveniente dal basso dai tombini, segue la legge di diffusione di un miscuglio di gas (nella fattispecie alcool ed aria), in base alla quale il numero di particelle di uno dei due gas che si spostano attraverso la superficie unitaria è proporzionale al gradiente di densità del gas.

Le molecole di alcool, escono dalla vaschetta, anche se non vi sono correnti d'aria, spontaneamente e rapidamente, con una velocità media in modulo dei Km/s, muovendosi in tutte le direzioni con moti tipicamente browniani; nell'intorno della vaschetta quindi c'è un'alta concentrazione di vapore e per diffusione questo si allontana dalla vaschetta, prontamente reintegrato da altro alcool che vaporizza. Se supponiamo che l'alcool resti a temperatura costante, e non si esaurisca, le molecole vanno alla deriva con velocità sempre decrescente man mano che ci si allontana, poiché diminuisce il gradiente di concentrazione di particelle per mm cubo d'aria, fino ad esaurirsi molto lontano dalla sorgente; ad un certo istante, la concentrazione parte dal massimo nelle immediate vicinanze, per calare in maniera esponenziale in pochi 20-30 cm dalla sorgente, e poi rimanere costantemente a zero, come una curva a gomito; la velocità di diffusione dipende da: area superficiale, ventilazione, temperatura, peso molecolare della sostanza. Di queste grandezze possiamo modificare la temperatura, e questo già viene fatto dai trasduttori di gas, i quali hanno una resistenza da riscaldamento, che aumenta l'energia cinetica delle molecole, velocizzando il processo di misurazione; e poi la ventilazione. Un sistema con una ventola che aspira verticalmente l'aria, in un condotto terminante con il sensore, è un buon amplificatore meccanico del segnale.



3.2 Il campo di esplorazione.

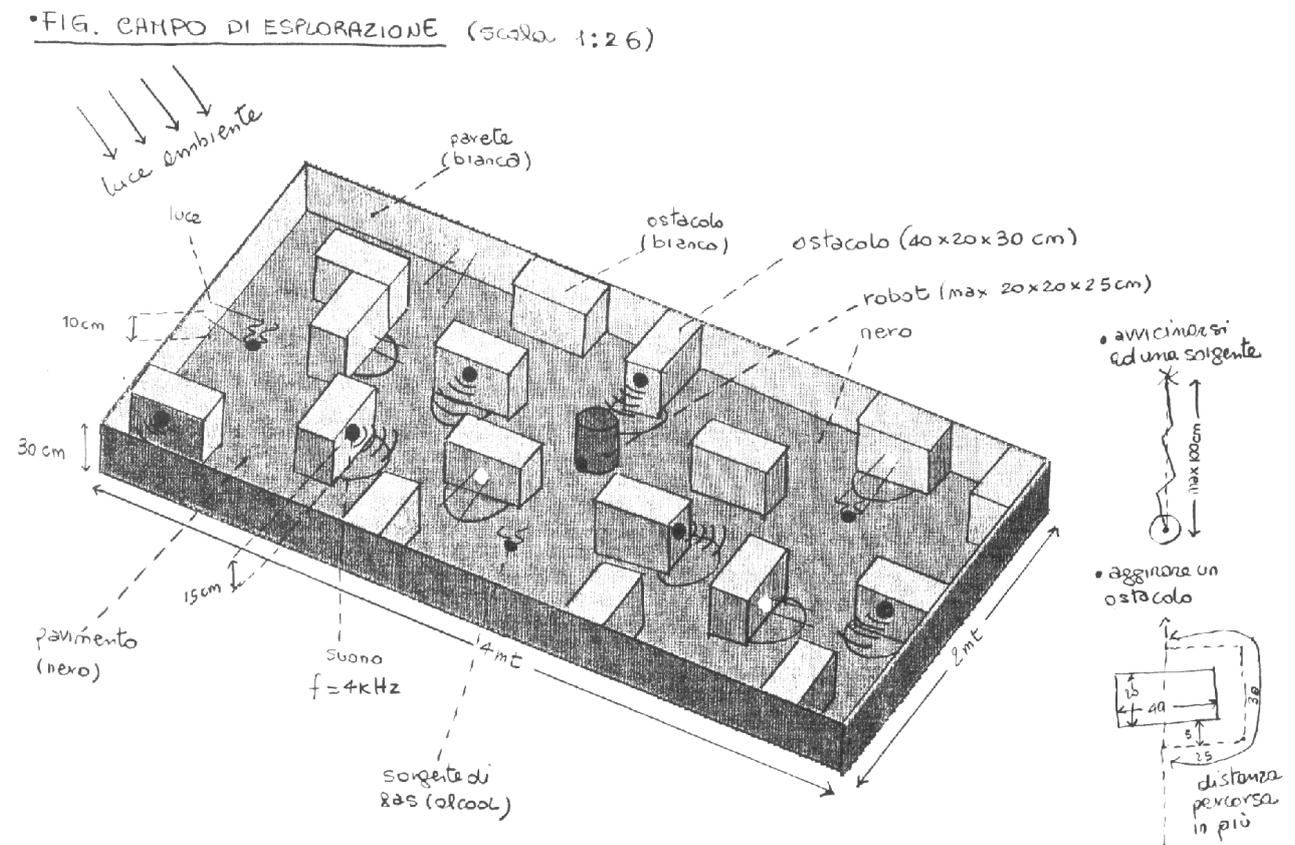
Prendendo informazioni dall' 'analisi del problema', possiamo ricavarne qualche proprietà sul campo e quindi sul sistema. Facciamo un modellino del campo di gara, mediante un CAD di progettazione di ambienti e strutture meccaniche, 3DSTUDIO. Montiamo un tipico campo di gara, dopo aver creato gli ostacoli, il pavimento, ed il contorno, specificando anche le dimensioni, il colore ed il materiale, e visualizzando una qualsiasi inquadratura (v. pag. seguente)

Come risultato, otteniamo le seguenti informazioni:

ci saranno massimo 18 ostacoli, 10 luci, 7 suoni, 4 tombini di gas. Prendiamo un parallelepipedo delle dimensioni massime del robot e collochiamolo dentro: la lunghezza massima percorribile in linea retta, in una zona abbastanza sgombra di ostacoli (le 'piazzuole'), prima di incontrare un ostacolo è di un metro circa; questa è quindi la massima distanza da cui può essere rilevata una qualche luce o suono; incontrando davanti a se un ostacolo, si può aggirarlo a destra o a sinistra, quindi percorrendo il suo semiperimetro (80 cm, considerando la distanza di sicurezza).

Quindi volendo almeno farsi irradiare da tutte le sorgenti presenti, quanti metri si dovranno percorrere in tutto, al massimo? Bisogna (1) individuarle e raggiungerle (max 1 metro) (2) aggirare gli ostacoli presenti (80 cm); nel caso del gas la sorgente si scopre per 'caso', quindi non c'è tempo di ricerca; in tutto $18 \cdot 80 \text{ cm} + (10+7) \cdot 100 \text{ cm} = 31 \text{ metri}$. Il tempo di sosta per ogni sorgente è di 3 sec, quindi il tempo di sosta totale è $(10+7+4) \cdot 3 = 63 \text{ sec}$. Se v è la velocità di avanzamento del robot, allora la prima condizione è $T_{\text{ricerca}} + T_{\text{segnalazione}} \leq 5 \text{ minuti (300 sec)}$, ossia $3100/v + 63 \leq 300$ da cui si ottiene $v \geq 13 \text{ cm/sec}$.

(fig. campo di esplorazione)



3.3 La scelta dei trasduttori di misura.

Il tipo dei trasduttori è già una limitazione alla massima quantità d'informazione che possiamo raccogliere, perché ogni sensore ha un certo range di misurazione, ed una certa sensibilità al rumore, per cui non ha senso, come visto, considerare misurazioni sopra un massimo, sotto un minimo, e variazioni sotto una certa unità. Fissati i trasduttori, potremmo quindi già calcolare la massima quantità di informazione elaborabile dal sistema, e quindi dimensionare la capacità di calcolo necessaria, per il controllore che come vedremo sarà digitale.

La scelta dei trasduttori è stata fortemente ristretta al criterio della tecnologia più semplice in commercio, motivato dalla scarsa conoscenza dei dispositivi esistenti, dalla poca esperienza, e dalla necessità di avere un circuito stampato realizzabile con strumenti elementari ed un utilizzo non troppo laborioso del CAD e del CAM. Infatti la complessità del circuito stampato aumenta in maniera in genere non polinomiale per ogni componente aggiunto: bisogna portare sempre i fili di alimentazione, fili del segnale, e lo sbroglio deve avvenire sempre nei limiti di spazio consentiti, e senza troppi incroci, che comportano ponti, ossia problemi di saldature e di contatti. Inoltre l'aggiunta di un solo componente in più può rendere impossibile una soluzione planare di una certa area, se si crea un semplice poligono di Kuratowsky, che non è possibile rappresentare senza intersezioni con gli archi. D'altronde, si vuole una realizzazione a piste su un solo lato della scheda, a meno, come già accennato, di qualche ponticello sulle piste, dal lato componenti.

L'oggetto più semplice per rivelare una luce a varie intensità è una fotoresistenza, le cui caratteristiche, ottenute facendo prove sperimentali a varie distanze, sono mostrate alle pagine seguenti; le dimensioni della superficie sensibile sono dell'ordine del centimetro di diametro; la sua resistenza diminuisce all'avvicinarsi della luce, e va da un minimo di qualche decimo di $K\Omega$ a circa $2K\Omega$.

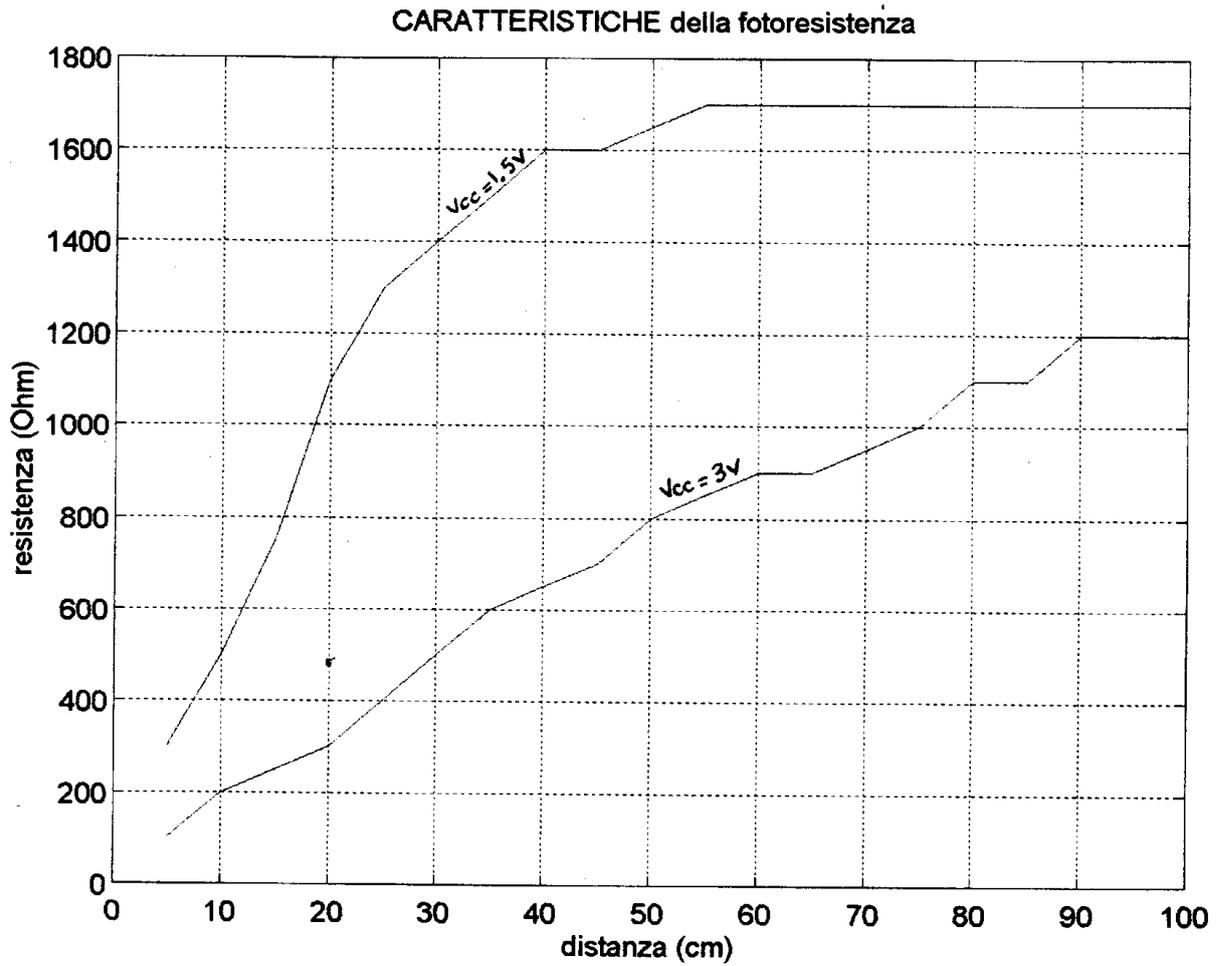
Per misurare l'intensità dei suoni, il sensore più elementare è la capsula microfonica preamplificata, che anch'essa ha una superficie sensibile di 1 cm di lato; produce una tensione analogica proporzionale all'intensità della pressione dell'onda sonora.

Invece, l'oggetto che misura la concentrazione di alcool è simile a quelli usati per la sicurezza da fughe di metano nelle abitazioni civili; funziona come un partitore resistivo in cui c'è una resistenza variabile con la concentrazione di particelle di gas nell'aria, e diminuisce all'aumentare di detta concentrazione. In tal caso però vogliamo ottenerne un segnale binario di presenza/assenza di gas; la continuità della variazione servirà però ad avere un circuito funzionante secondo varie tarature, perché la concentrazione 'critica' di segnalazione sorgente dipende dall'umidità dell'aria, dalla ventilazione, e da altre grandezze variabili di ora in ora.

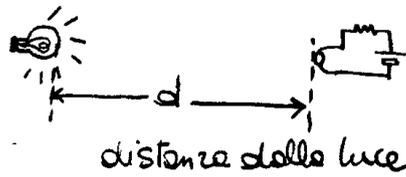
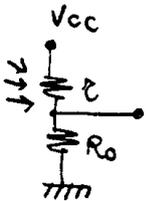
Infine, per rivelare la distanza da un muro bianco, si usa un sistema led emettitore e sensore ricevitore ad infrarossi (del tipo dei telecomandi) sincronizzati su un oscillatore opportuno, così che il led lampeggia ad una data frequenza, ed il ricevitore cerca in un intorno di quella frequenza, così da non confondersi con altri segnali infrarossi provenienti dall'ambiente. Un sistema siffatto è binario, e si accorge del muro a 5 cm circa, fornendo un segnale logico TTL, basso quando si è vicini al muro. Per avere una misura continua della distanza da un muro, bisognerebbe usare un sistema ad ultrasuoni, troppo complesso per i nostri scopi.

(fig. prove sperimentali effettuate sulla fotoresistenza)

d (cm)	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
r (Ω), Vcc=1.5v	300	500	750	1100	1300	1400	1500	1600	1600	1650	1700	1700	1700	1700	1700	1700	1700	1700	1700	1700
r (Ω), Vcc=3v	100	200	250	300	400	500	600	650	700	800	850	900	900	950	1000	1100	1100	1300	1200	1200



• circuito di prova



prendiamo $R_{min} = 100\Omega$
 $R_{max} = 2K\Omega$

• caratteristiche meccaniche



3.4 La scelta degli attuatori di movimento, e di segnalazione luminosa.

Come visto, i limiti nelle dimensioni del robot, e la direzionalità delle sorgenti, pongono necessariamente il problema del movimento, poiché non esiste una posizione da cui si potrebbe rilevare tutto. Va precisato inoltre che non è lecito scovare le sorgenti di suono (o di gas) attraverso misurazioni non di suono: sarebbe infatti computazionalmente molto più semplice (anche se circuitualmente più complesso) usare, ad esempio, una telecamera digitale, con cui esaminare il pavimento alla ricerca di buchi (sorgenti di gas) o gli ostacoli, alla ricerca di altoparlanti: nella descrizione non viene fornito alcun dettaglio a riguardo della forma delle sorgenti, quindi il sistema deve elaborare solo le informazioni caratteristiche di ciascuna sorgente.

Posto che il sistema dovrà muoversi, si pongono le seguenti scelte: con che mezzo muoversi? con quale strategia muoversi? La precisione dell'attuatore influenzerà il tipo di strategia, a prescindere da qualsiasi sistema di calcolo disponibile.

Il trasduttore più semplice di avanzamento è il motore in continua, la cui velocità è controllabile in tensione con la tecnica del PWM (Modulazione a Larghezza di Impulsi), che garantisce sempre una certa coppia motrice, consentendo di variare la velocità. Si potrebbe retroazionare l'attuazione di movimento mediante un encoder di posizione, ma l'assenza di esperienza sugli encoder, e l'obiettivo della semplicità, porta a scartare tale opzione; ad ogni modo si riuscirà a mantenere l'efficacia.

La scelta di eseguire movimenti in 'catena aperta', ha come conseguenza un errore percentuale di un 5% sui movimenti, dovuto a vari fattori: alimentazione della batteria decrescente nel tempo; necessità di transitori all'avvio ed alla frenata prima di raggiungere la velocità di regime, o soprattutto nelle piccole rotazioni a destra ed a sinistra; etc. Senza retroazione sulla posizione, è impossibile avere misure attendibili già dopo 10 metri percorsi, avendosi un errore di mezzo metro. E' possibile allora che il sistema sappia più o meno quanti metri ha percorso (da un conteggio temporale interno), ma non la sua posizione rispetto alla partenza.

Sulla scelta del tipo commerciale di motorino, ci affidiamo ai criteri su base 'storica' di gente con esperienza, che ci consiglia un tipo di motore della RS controllabile con l'integrato L298.

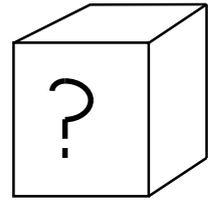
Nel data sheet viene riportata una velocità angolare $v=92$ giri/min= $1,53$ giri/sec. Si osservi innanzitutto che tale velocità è indipendente dal carico, trascurando le variazioni dell'attrito sull'asse, e quello dell'aria, ed è intesa come velocità di regime; se l'asse fosse perfettamente liscio (il pavimento deve essere rugoso, altrimenti il robot non parte! ma non conta come attrito passivo), il nostro motorino potrebbe trainare persino un camion, sino ad una velocità costante, dovuta alla forza di attrito con l'aria; solo che ci vorrebbero chilometri per fargli raggiungere tale velocità; nel vuoto poi la sua velocità aumenterebbe fino ai limiti relativistici. La tecnica del PWM fa sì che la coppia fornita sia sempre la stessa, così che si possa contrastare l'attrito sull'asse.

Possiamo invece decidere le dimensioni delle ruote, tra un minimo, dovuto alla velocità di avanzamento necessaria calcolata in precedenza ($v \geq 13$ cm/sec.), ed un massimo, dovuto alle dimensioni del robot, alla stabilità dinamica della struttura, alle dimensioni dei motorini; sarà l'argomento del prossimo paragrafo.

Come attuatori di segnalazione luminosa, scegliamo dei semplici led, di colore diverso per tipo di sorgente. Gli stati da segnalare sono quattro: nessuna sorgente, sorgente di luce, suono, gas; quindi basterebbero due fili, con una rete combinatoria opportuna.

3.5 Dimensionamento della struttura meccanica.

Dobbiamo decidere della forma di un oggetto contenuto in un cubo di 20x20x25 cm. In realtà, decideremo solo della forma della pianta, e dell'altezza, perché non è vantaggioso restringere le dimensioni man mano che si sale (c'è bisogno di spazio per le schede e di distanziamento dei vari trasduttori, soprattutto di suono); né tantomeno allargarle, per seri problemi di instabilità.



Quindi decidiamo se la base sarà quadrata, esagonale, circonferenza, etc. e quanto sarà alto.

Se fosse quadrata, o spigolosa in genere, dovremmo mettere dei trasduttori sugli spigoli, per evitare di urtare gli ostacoli mentre ruota; inoltre vogliamo mantenere separate le azioni di avanzamento e di rotazione, perché è necessario ruotare per rilevare i cambiamenti di intensità nelle onde, e decidere la direzione opportuna, prima di avanzare; tale rotazione dovrà mantenere poi invariata la distanza dei trasduttori dalla sorgente, per poter confrontare i valori; in pratica, poiché il robot non sa nulla sulla direzione delle sorgenti, deve avere una forma isotropa, per tutte le direzioni; in altre parole, se avesse una forma asimmetrica, ad esempio allungata, prediligerei la direzione in avanti, o peggio a destra quindi il robot andrebbe prevalentemente a destra, e per lunghi tratti spazzerebbe solo il lato destro del campo. La conclusione è una forma circolare (vedi fig. pagine seguenti).

Avremo bisogno di vari piani su cui sostenere le schede elettroniche ed i trasduttori; il materiale deve essere trasparente, per permettere un controllo visivo delle schede; elettricamente isolante, duttile, resistente; un materiale plastico, in particolare il plexiglas è ideale, e forse l'unico adatto. Di questi piani però non sappiamo ancora bene a che altezza sistemarli. Sicuramente ne servirà uno per le batterie, che sono gli oggetti più pesanti (160g quelle al piombo da 6v, 50g quelle non ricaricabili da 9v), e sarà situato in basso per stabilità maggiore; altri per le schede elettroniche. ed uno finale in alto di chiusura. Inoltre necessita una struttura a piani completamente smontabile, in caso di problemi elettrici; quindi le schede ai vari piani saranno collegate con cavi flessibili, a terminali a spina, ed i piani smontabili. Escludendo il sistema di collegamento ad incastri, che necessita di esperienza e strumenti di lavorazione, non rimane che usare dei bulloni. Nella fattispecie abbiamo una struttura a torre, per cui possiamo adoperare le colonnine filettate (almeno tre), che permettono anche di variare a piacere la distanza tra i piani; cosa molto importante, perché l'altezza cui si trovano i trasduttori è fissata, e viene influenzata dalle dimensioni delle ruote, degli oggetti posti sui vari piani (batterie, schede, integrati, dissipatori,...), in tal modo possiamo progettare tali dimensioni senza essere troppo vincolati.

Per realizzare la rotazione lungo l'asse, collocheremo le ruote motrici lungo un diametro, ed una rotellina di bilanciamento sul retro, posizionando opportunamente le batterie indietro in modo da avere il baricentro sul piano di appoggio; infatti sul davanti ci sarà il condotto di aspirazione, quindi non si potranno mettere altre ruote, al fine di evitare una caduta in avanti a seguito di frenate.

Quindi avremo dei piani circolari, del diametro di 18 cm; un po' meno del massimo, perché i led infrarossi dovranno un po' fuoriuscire dai bordi, quindi 1cm cadauno.

Parleremo ora del diametro delle ruote motrici.

Sapendo che $v \geq 13 \text{ cm/sec}$, e $v = 1,53 \text{ giri/sec}$; se d è il diametro delle ruote, in assenza di slittamento, $v\pi d = v$, dunque la precedente condizione si esprime come $d \geq v/v\pi \approx 2.7 \text{ cm}$.

Inoltre, all'aumentare del diametro della ruota, si solleva il baricentro del robot, e va anche più veloce; quindi c'è un limite oltre il quale il robot cade in avanti alla frenata, che dovrà avvenire in 2-3 cm, appena gli infrarossi segnaleranno un ostacolo davanti. Queste considerazioni sono espresse in fig. a pagina seguente. In base alla stabilità, si ottiene $d_{\text{max}} = 12 \text{ cm}$.

Un altro limite superiore infine, è dato dalla capienza con i motorini nei 18 cm, poiché le ruote ed i motorini andranno sul diametro; si ottiene $d_{\text{max}} = 12 \text{ cm}$. Questo valore è superiore a 10 cm, altezza a cui vanno i trasduttori di luce, e che deve essere sgombra almeno a 180° sul davanti; se le ruote fossero di 12cm, sui lati coprirebbero le fotoresistenze; quindi $d_{\text{max}} < 10 \text{ cm}$.

Concludendo, si decide un diametro di 6 cm, abbastanza lontano da 3 cm e da 10 cm.(valore medio). Come conseguenza, si avrà una velocità di 29 cm al secondo.

A questo punto possiamo realizzare la struttura meccanica di base, con il CAD; la cui stampa è mostrata alle pagine seguenti.

(fig.1. dimensionamento della struttura meccanica; stabilità alle frenate)

(fig.1. dimensionamento della struttura meccanica; stabilità alle frenate)

(fig.2. conseguenze della forma e del posizionamento dei trasduttori sulla traiettoria del robot)

(fig.3. numero di fotoresistenze)

(fig.4. composizione della struttura meccanica)

(fig.5. rendering della struttura)

• Dimensionamento delle ruote

a) in base allo spazio disponibile

• Considerando che motore + riduttore sono lunghi 5cm, che bisogna lasciare 1cm di spazio tra i motori per evitare contatti, che dall'esterno della ruota al riduttore ci sono circa 1cm, ho

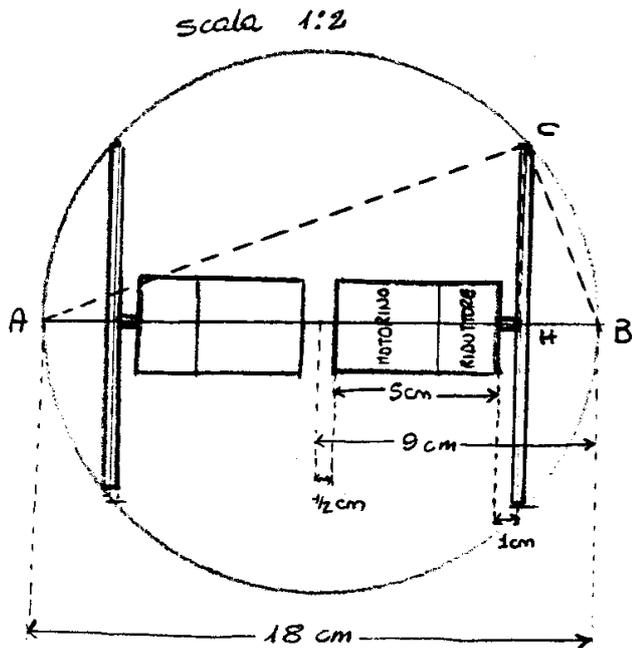
$$\overline{AH} = 9 + 0,5 + 5 + 1 \text{ cm} = 15,5 \text{ cm}$$

$$\overline{BH} = 18 - 15,5 \text{ cm} = 2,5 \text{ cm}$$

Dal teorema di Euclide

(ABC è inscritto nella circonferenza.)

quindi $d_{max1} = 12 \text{ cm}$



$$\frac{CH}{AH} = \frac{BH}{CH} \Rightarrow CH = \sqrt{AH \cdot BH} \approx 6 \text{ cm}$$

b) in base alla stabilità alle frenate

Voglio che si fermi in $s_f = 2 \text{ cm}$ - Supponendo la decelerazione costante, la forza apparente è

$$F = Ma = M \frac{v}{T} = M \frac{V^2}{S} = \frac{M \pi^2 v^2}{S} \cdot d^2$$

Dal teo. di Euclide

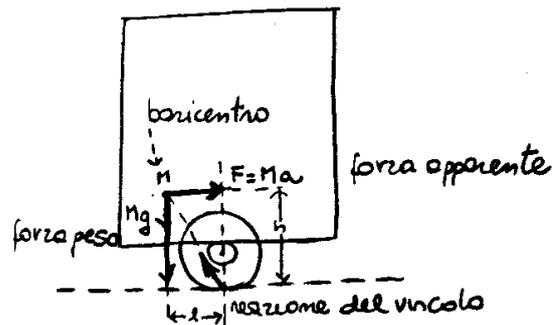
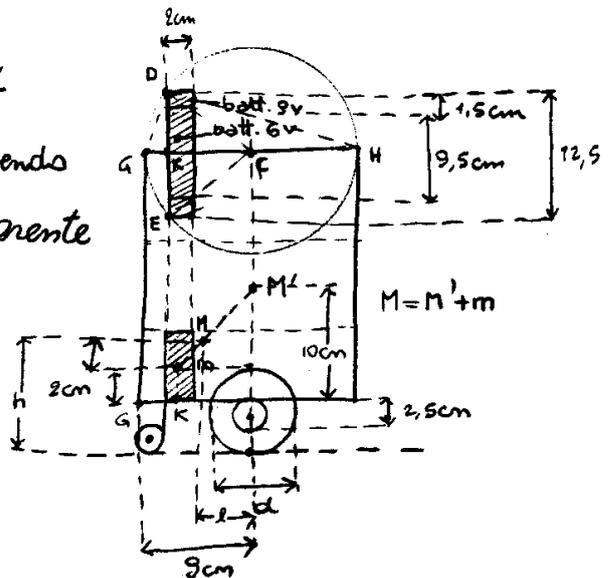
$$\frac{DK}{GK} = \frac{KH}{DK} \Rightarrow DK^2 = GK(d - GK)$$

$$GK^2 - dGK + DK^2 = 0 \rightarrow GK = 1,9 \text{ cm}$$

masse: batteria 6V 350g
batterie 9V 50g x 2 } m = 450g

plati plexiglas + schede 200g x 4
barre filettate 20g x 3 } M' ≈ 1000g
motorino ventola 40g
altro ≈ 100g

$M \approx 1450g$



Dobbiamo trovare le coordinate del baricentro:

$$\frac{SP}{SR} = \frac{M'}{M} = \frac{PR - SR}{SR} = \frac{M'}{M}; (PR - SR)M = M'SR$$

$$PRM = SR(M + M') \Rightarrow SR = PR \cdot \frac{M}{M + M'} \approx 3,3 \text{ cm}$$

$$SU = SR \cdot \frac{PQ}{PR} = \boxed{2,2 \text{ cm} = l}$$

$$ST = SP \cdot \frac{RQ}{PR} = (PR - SR) \cdot \frac{RQ}{PR} = 5,5 \text{ cm}$$

$$h = ST + 1,25 + d/2 = \boxed{6,8 + d/2 = h}$$

Il vincolo riesce ad impedire la rotazione finché la risultante di $\vec{F} = M\vec{a}$ ed $M\vec{g}$ cade prima del punto di contatto con la ruota. Oltre si crea una coppia

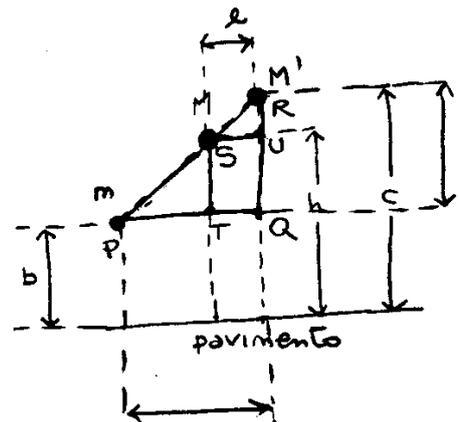
quindi $\beta \geq \alpha \rightarrow \tan \beta \geq \tan \alpha \Rightarrow$

$$\frac{Mg}{F} \geq \frac{h}{l} \Leftrightarrow \frac{Mg}{M \pi^2 v^2 / 9 \cdot d^2} \geq \frac{6,8 + d/2}{2,2}$$

$$\frac{2,2 \cdot 9 \cdot 5}{\pi^2 v^2} \geq 6,8 d^2 + d^3/2; \quad g = 9,8 \text{ m/s}^2 = 9800 \text{ cm/s}^2$$

$$d^3 + 13,6 d^2 - 3737 \leq 0; \Rightarrow \boxed{d \approx 12 \text{ cm}}_{\text{max 2}}$$

(casualmente è venuto come $d_{\text{max 1}}$)



$$PQ = 9 - 1,9 = 7,1 \text{ cm}$$

$$RQ = 10 - 2 = 8 \text{ cm}$$

$$PR = \sqrt{PQ^2 + RQ^2} = 10,7 \text{ cm}$$

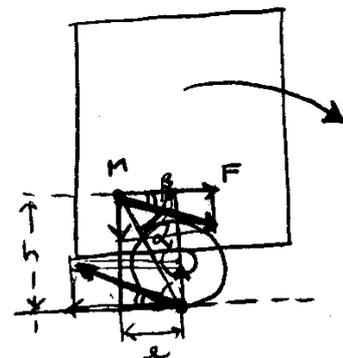
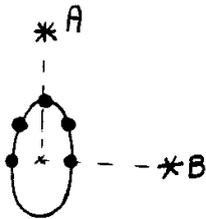


fig. coppia di rotazione

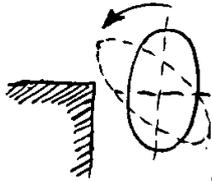
• Conseguenze della forma e del posizionamento dei sensori sulla traiettoria del robot

• forma allungata o con spigoli



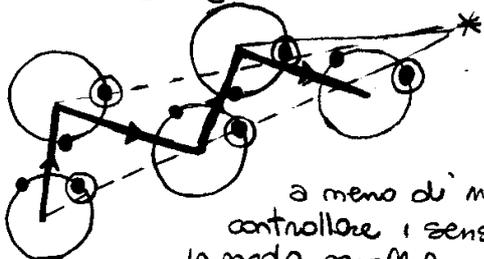
- la sorgente A sembrerebbe più vicina della B, ma non corrisponde al vero.

- nella notazione, servono sensori IR di notazione, oltre che di avanzamento



• l'unica forma che ruotando sul centro di simmetria non cambia lo spazio occupato, e mantiene inalterate le distanze dei sensori è la circonfenza

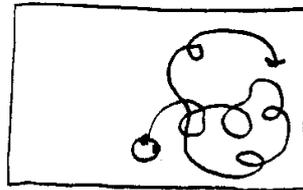
• senza sensore centrale, il robot va a zig-zag



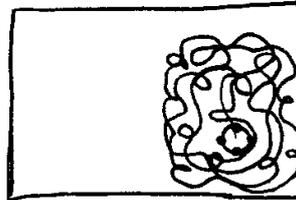
a meno di non controllare i sensori in modo parallelo

• posizionamento dei sensori

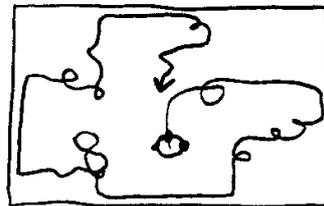
Ⓐ solo a destra (o sinistra): il robot girerà più spesso a destra, coprendo solo metà del campo



Ⓑ tutti intorno: alle lunghe il percorso sarà centrato intorno alla partenza, perché ritornerà spesso sui suoi passi

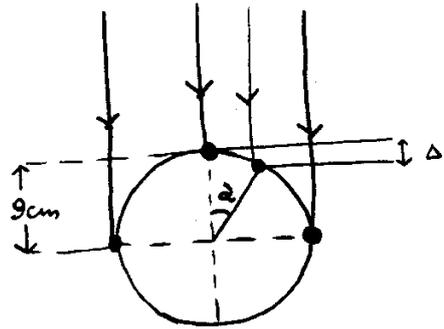


Ⓒ solo avanti, simmetrici: anche spesso sui bordi, e da lì in altre zone



• numero di fotoresistenze

la situazione più svantaggiata si ha quando la sorgente è molto lontana, e vogliamo sempre apprezzare la differenza di intensità, cioè avere almeno $1\text{ cm} = \Delta$.



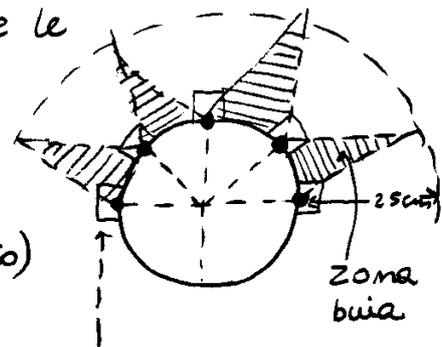
$$\Delta = r - r \cos \alpha = r(1 - \cos \alpha) = 1 \Rightarrow 1 - \cos \alpha = 1/9, \cos \alpha = 8/9;$$

$$\alpha = \arccos 8/9 \approx 0,48 \text{ rad} = 0,48 \cdot \frac{180}{\pi} \text{ }^\circ = 27^\circ \approx 30^\circ \Rightarrow 7 \text{ sensori di luce massimo.}$$

Se ne metto invece 5, ho $\alpha = \pi/4 \rightarrow \Delta = r(1 - \frac{\sqrt{2}}{2}) \approx \boxed{2\text{ cm}}$

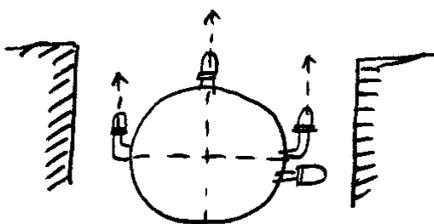
risparmiando molto in complessità e circuitale, e le prestazioni peggiorano di poco.

- Eventuali tubi non devono essere così lunghi da creare zone buie a lunghezza $> 25\text{ cm}$ (minimo concesso per il rilevamento)

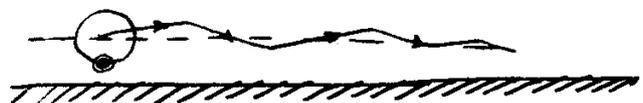


un sensore laterale è di "avvistamento"
il sensore centrale è di "inseguimento"

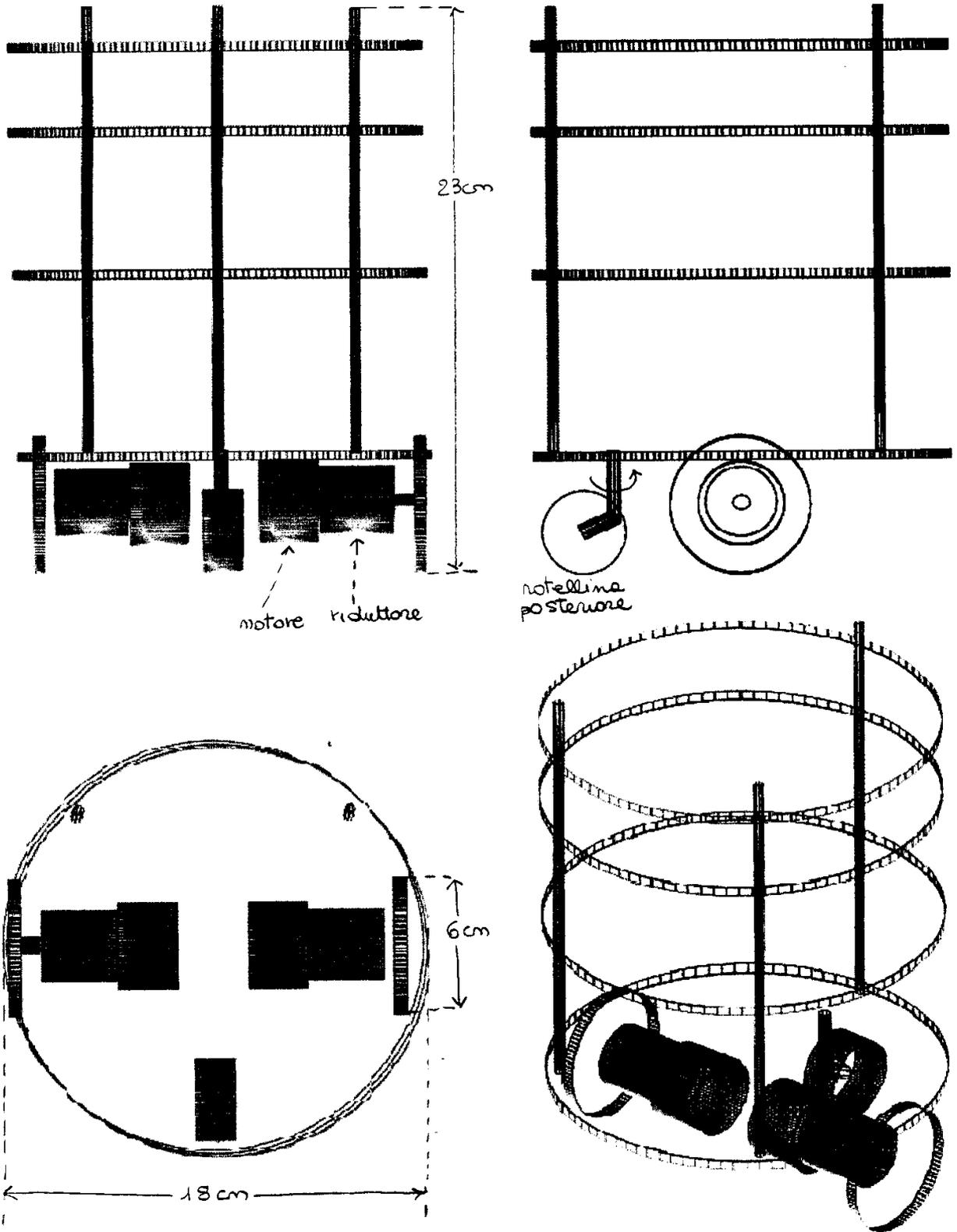
• numero di infrarossi



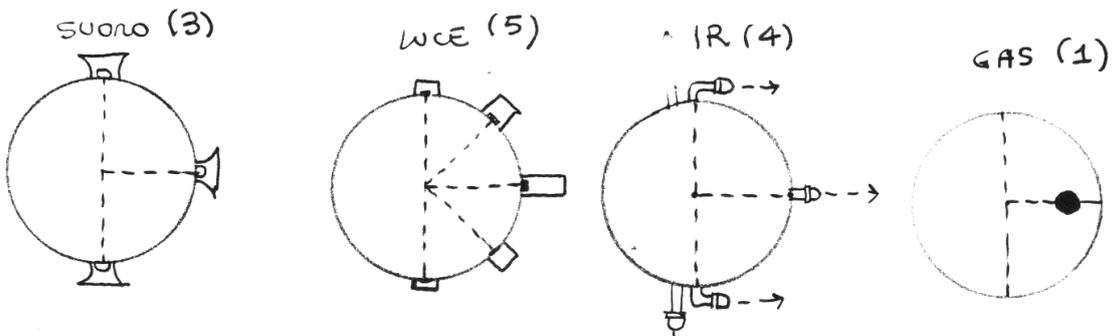
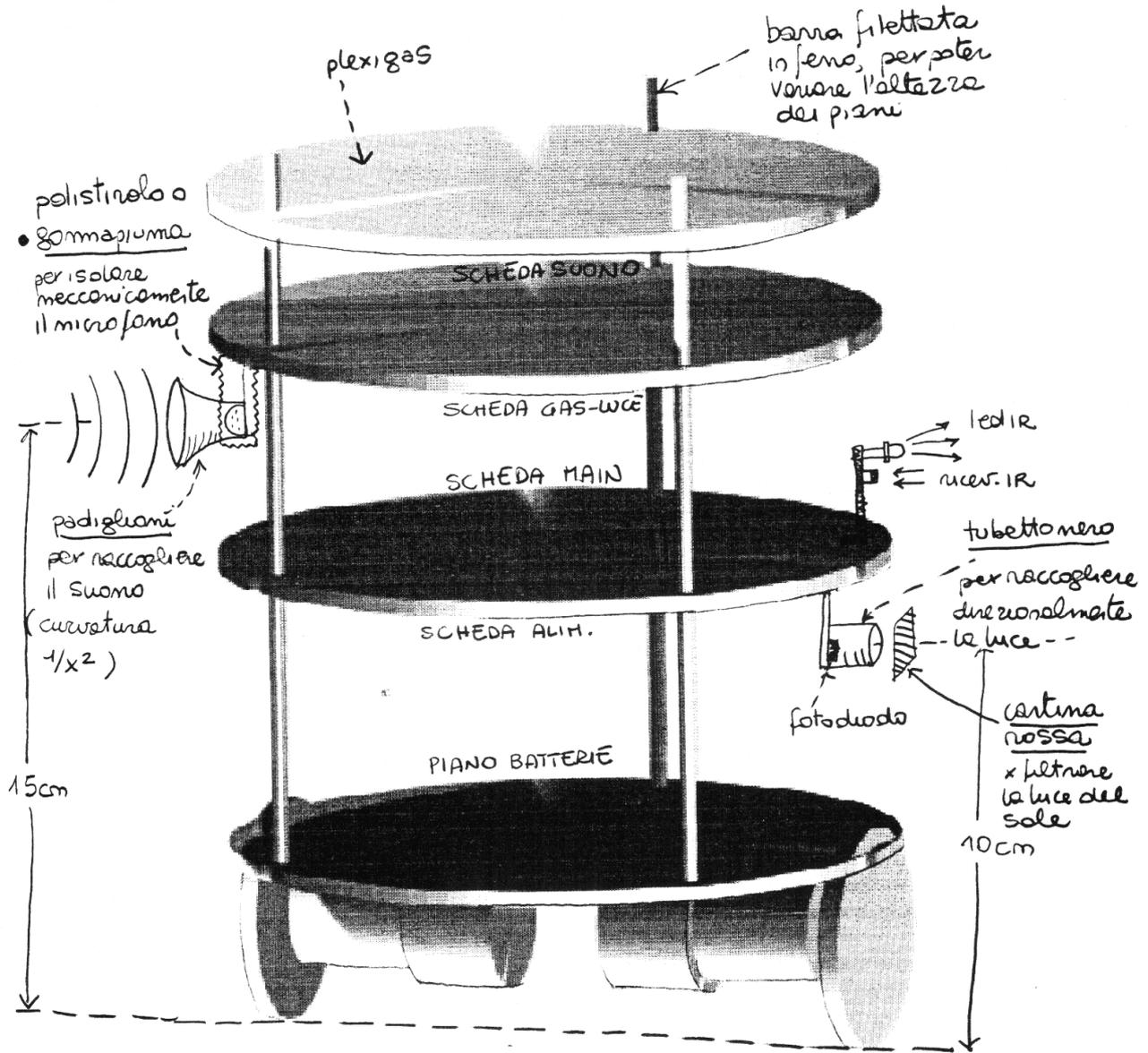
sono solo di avanzamento, perché durante la rotazione non può tornare. Uno laterale in più è strategico, perché permette di accostarsi al muro e seguirlo -
Questa tattica può servire ad aggirare ostacoli o seguire il bordo, per scovare altre sorgenti



• fig. Composizione della struttura meccanica



• fig: rendering della struttura, con l'aggiunta di soluzioni per filtrare e raccogliere le onde



3.6 Posizionamento dei trasduttori e strategia di ricerca.

Per rilevare tutte le sorgenti di luce e di suono bisogna spostarsi ed aggirare gli ostacoli; e per rilevare le sorgenti di gas bisogna fare in modo che il robot spazzi con il suo cerchio di azione, entro cui si accorge del gas, tutto il pavimento sgombro da ostacoli. Quindi bisogna garantire che la disposizione dei trasduttori sia tale da non privilegiare una direzione. All'interno di una piazzuola, il robot andrà diritto fino a rilevare qualche sorgente. In seguito, dovrà uscire dalle piazzuola, e visitare tutte le altre.

L'algoritmo di controllo è finalizzato localmente, ossia può utilizzare una qualsivoglia tattica, ad esempio una ad inseguimento nella direzione del sensore a livello massimo, quando ha qualche indizio delle sorgenti; ma serve una strategia globale che metta in condizione di perlustrare tutto il campo.

Se mettiamo i trasduttori a 360°, avremo un robot a grandi vedute, ma confuso, nel senso che la sua traiettoria sarà intorno al punto iniziale (vedi figure), ossia ritornerà spesso sui suoi passi; se li mettiamo solo da un lato, percorreremo solo un lato del campo; allora dobbiamo metterli in una direzione, e simmetrica; ossia sul semicerchio anteriore, avendo cura di mettere sempre due trasduttori agli estremi, di "percezione", che permettano di avere una differenza apprezzabile anche a grande distanza; e uno centrale, di "puntamento", senza il quale il robot andrebbe a zig-zag. Per il suono, abbiamo detto che devono distare almeno 10cm, quindi collochiamo tre microfoni: due estremi ed uno centrale; lo stesso per la luce, almeno tre. Quanti ne possiamo aggiungere lateralmente?

L'informazione trasportata dalla luce che giunge sul robot può essere analizzata in tre attributi, o grandezze variabili indipendentemente: la distribuzione spettrale dell'energia radiante abbiamo già fatto le considerazioni, ed è la stessa per tutte le sorgenti di luce; la direzione e l'intensità sono invece diverse per ogni sorgente, data una posizione.

Nel caso pessimo, la sorgente è molto distante dal robot, e viene colpita quindi da un fascio quasi parallelo; le fotoresistenze più vicine, nella direzione della luce, sono quelle frontali; dai calcoli si vede (fig. pagine precedenti) che per avere una distanza di almeno 1 cm, bastano 7 trasduttori di luce; mettendone 5, tale distanza passa a soli 2 cm; la nostra scelta sarà di 5 trasduttori di luce, poiché alla distanza massima di 80 cm, 2cm sono una precisione ottima; inoltre vogliamo utilizzare i tubicini per selezionare; essi saranno più lunghi avanti, e molto corti sui lati, perché mentre i laterali sono trasduttori di "avvistamento", cioè devono essere di ampie vedute per rilevare sorgenti lontane, quello davanti è il vero occhio che deve indicare la direzione da seguire, quindi un errore su quello davanti aumenta il tempo di ricerca, perché il robot si avvicina zig-zagando.

I led infrarossi sono infine solo di avanzamento, due di copertura degli urti di "spalla", ed uno centrale perché quelli laterali sono troppo indietro rispetto al fronte.

Un sensore laterale "di affiancamento" a destra, permette di avere una funzione di aggiramento ostacoli, o inseguimento del muro, utile per trovare altre sorgenti, tutte fissate sugli ostacoli.

Aver messo i trasduttori davanti non comporta una perdita delle sorgenti, perché se per assurdo ci fosse una sorgente alle spalle più vicina di una davanti allora il robot ci sarebbe già passato prima; se invece proveniva svoltando a destra, se ne sarebbe accorto il sensore di sinistra, e viceversa.

Un'altra soluzione poteva essere un'asta radiale che ruotava a 360° su un perno centrale, con un solo sensore in cima, che spazzava tutta l'area; il vantaggio è una ottima precisione; ma a patto di avere una meccanica molto raffinata, soprattutto con encoder di posizione sul motorino, sistema già considerato troppo complesso.

3.7 Controllore digitale o analogico ? Il PIC16C84.

Gli svantaggi principali di un sistema digitale (microcontrollore), sono dovuti alle proprietà del campionamento (condizione di Nyquist) che pongono un limite all'ampiezza di banda, alla quantizzazione che introduce errori e distorsioni, ed ai ritardi dovuti alla elaborazione a singolo flusso di istruzioni e dati nei sistemi uniprocessore. Nel nostro caso però, i segnali variano molto lentamente, per la velocità ridotta del robot (caso della luce e del suono), oppure variano a scatti (caso infrarossi e gas), ma in quest'ultimo caso interessa solo un segnale binario di presenza.

Ma è soprattutto sulla strategia di controllo che si impone il sistema digitale, dove le pesanti limitazioni di un sistema analogico rendono praticamente impossibile la realizzazione: complessità elettronica, non modificabilità, difficile leggibilità, scarsa integrabilità. Inoltre un sistema analogico richiede una tolleranza sull'alimentazione del 2%, mentre uno digitale sopporta anche 5-20% di disturbi; anzi, i sistemi digitali hanno la proprietà di rigenerare i segnali.

Entriamo ora in merito all'architettura del sistema digitale da adottare. Mentre un microprocessore, *general purpose*, viene progettato sostanzialmente per 'processare', ossia per eseguire istruzioni matematiche e logiche anche complesse, ad esempio per risolvere equazioni differenziali, od eseguire lunghe catene di inferenze, dove lo scopo è quello di ottenere dei risultati, ossia *comprimere l'informazione contenuta nei dati di partenza*, un microcontrollore è un dispositivo di elaborazione dedicato al controllo, ossia *a percepire ed agire*, quindi assumere delle variabili *fisiche* in ingresso, di natura diversa (come temperatura, pressione, luminosità) tramite trasduttori, e governare automaticamente delle grandezze di uscita, che producono delle azioni *fisiche* su un sistema da controllare, tramite attuatori. La disciplina del controllo si ispira alle azioni degli esseri viventi, e quindi tende a sostituirne l'intervento o a migliorarne l'efficienza. Un sistema di calcolo che opera su processi fisici deve essere prima di tutto veloce, ossia operare a *tempo reale*; deve avere istruzioni *non complesse* (nel senso computazionale, ossia composte da sub-problemi articolati nel tempo), ma di sufficiente *varietà* (nel senso di composte da sub-problemi articolati nello spazio, quindi da diverse unità, magari eseguite in parallelo); un sistema di controllo deve per prima cosa misurare e governare; poi, nei ritagli di tempo, può anche pensare (ossia eseguire procedure complesse). Si pensi ad esempio agli insetti (come una formica, che ha solo qualche migliaio di neuroni) o ai rettili (una lucertola), che hanno un ottimo sistema di controllo del moto.

L'architettura RISC, è quindi quella più adatta, perché riduce al minimo il set di istruzioni, per aumentare la velocità di clock, e per avere un certo livello di pipeline nelle fasi di elaborazione. In genere, i microcontrollori hanno già dei dispositivi di conversione A/D dei segnali, per poter leggere i segnali elettrici dei trasduttori; e dei driver di potenza, per poter pilotare degli attuatori di moto. Nel nostro caso, la scelta è limitata al PIC16C84, che permette, per la relativa semplicità, anche della scheda di programmazione, e la disponibilità dell'ambiente di sviluppo, di avere un corretto approccio (soprattutto didattico, che è lo scopo principale) al mondo dei microcontrollori.

Nelle pagine seguenti vengono mostrate: architettura interna, memoria di programma, e memoria SRAM dei registri. Le caratteristiche principali sono:

- architettura RISC, 35 istruzioni
- quasi tutte le istruzioni, eseguite in un clock
- 400ns per istruzione
- 1K di EEPROM di programma (quindi max 1024 righe di codice ASSEMBLER)
- 36x8 registri special function
- 64 registri 8-bit EEPROM per i dati (permette di memorizzare dati permanentemente, ad esempio per algoritmi di apprendimento)
- 8 livelli di stack
- indirizzamento diretto, indiretto, relativo

- 4 sorgenti di interrupt: INT esterno
 overflow di TIMER0
 cambio stato su RB4,RB5,RB6,RB7
 fine ciclo di scrittura EEPROM dati
- 1 milione di cancellazioni/scritture su EEPROM dati
- mantenimento dei dati su EEPROM per 40 anni
- 13 pin di I/O
- 20mA erogabili per pin
- 25 mA come sink
- contatore interno 8-bit + prescaler
- power-on reset
- Power-up timer
- Watchdog timer
- protezione del programma in lettura
- funzionamento in SLEEP
- oscillatore selezionabile tra 4
- programmazione seriale in-circuit
- tensione di alimentazione da 2 a 6 volt
- assorbimento a 5V e 4Mhz <2mA
- assorbimento in SLEEP <1µA .

La programmazione seriale ISP della EEPROM per dati e istruzioni, utilizza solo due pin. Alla pagina seguente è mostrato lo schema elettrico del programmatore utilizzato con il PROG84.

L'architettura Harvard, con memoria dati e memoria istruzioni separate, permette di avere tutte le istruzioni in un codice operativo di 14 bit, mentre la memoria rimane ad 8-bit; quindi ogni istruzione è contenuta in una sola word, ed eseguita in un singolo ciclo. Questo migliora la larghezza di banda rispetto alla tradizionale architettura di Von-Neumann, dove dati ed istruzioni sono nella stessa memoria; e permette di avere due livelli di pipeline, sovrapponendo fase di fetch ed esecuzione di due istruzioni consecutive. Quindi tutte le istruzioni sono eseguite in un singolo ciclo, tranne la chiamata di programma. Un ciclo di istruzione, durante la quale viene eseguita la Fetch INST(PC) e la Execute INST(PC+1), abbisogna di quattro colpi del clock esterno. Quindi se applichiamo un clock al quarzo da 4Mhz, avremo cicli istruzione da 1Mhz, ossia una istruzione ogni microsecondo.

La semplicità architetturale del PIC16C84, si paga con la necessità di introdurre un convertitore analogico digitale, multiingresso, ed un'interfaccia parallela di uscita per incrementare i driver disponibili.

3.8 Dimensionamento del clock del microcontrollore, e della risoluzione di conversione

L'informazione luce è composta da intensità (distanza) e direzione; abbiamo detto che posso apprezzare, nell'ottimo, distanze di 2cm; le distanze sono comprese tra 0 e 100, quindi ho 50 alternative possibili sulla distanza; cioè 6 bit; ciò vuol dire che se il convertitore A/D è alimentato da 0 a 5V, allora avrò una precisione di $5/50=0.1$ volt, più che sufficiente. Ma poiché tale segnale può variare la sua dinamica intorno ad una soglia variabile con le condizioni esterne, e si suppone che la taratura analogica presente a monte non possa compensare in tutto tali variazioni, c'è bisogno di una taratura 'digitale'; in pratica, il segnale varierà ad esempio, tra 2V e 4V, oppure tra 3V e 5V; quindi avviene una riduzione dei livelli, che diciamo non sarà maggiore di 5 volte (il che significherebbe es. un range di 1 solo volt); a fronte di ciò, prendiamo quindi $5*50=2500$ livelli, ossia 8 bit;

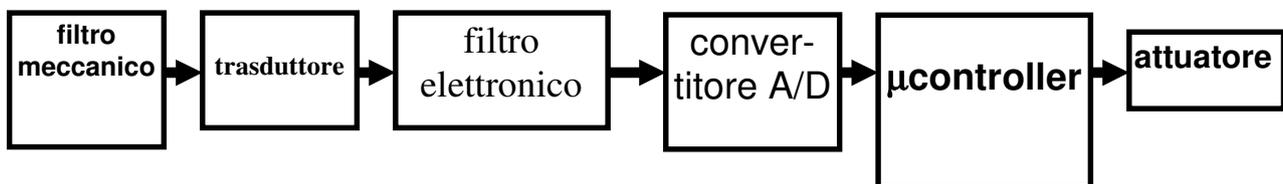
L'informazione sonora che arriva è simile, perché posso apprezzare differenze nell'intensità di 2cm, quindi 8 bit. Sul gas ho anche 8 bit, perché anche se l'informazione è solo di presenza/assenza, ha una variazione di soglia molto alta (in dipendenza di umidità dell'aria, temperatura, saturazione del sensore, etc.) quindi il bit d'informazione dopo la taratura è uno, ma prima della taratura ho molte alternative possibili. Sugli infrarossi ho un segnale binario (1 bit).

I segnali ad infrarossi, per la presenza di un ostacolo, sono eventi non prevedibili e sempre possibili; quindi il modo ideale di gestirli è tramite interrupt, a meno di non voler bloccare il microcontrollore con continui test sui piedini; il PIC16C84 può rilevare una transizione sui piedini RB4..7, che quindi saranno connessi direttamente alle uscite dei ricevitori IR. In realtà basterebbe una sola porta di ingresso, connettendo tutti gli IR ad una porta OR, a patto di avere tanti driver per accendere gli IR separatamente e quindi di sapere in anticipo quale IR può segnalare; nel nostro caso non ce n'è bisogno, ma vogliamo mantenere ugualmente l'accensione separata degli IR, nella circostanza in cui due di essi dovessero influenzarsi reciprocamente, essendo troppo vicini. E' possibile aumentare il campo di azione dei led IR, collegandone due in serie (non in parallelo, perché sono dei diodi, ed alla fine tutta la corrente andrebbe in uno solo) allo stesso ricevitore, diminuendo però un po' la corrente che vi passa.

Quindi abbiamo bisogno di un convertitore A/D con almeno 8 bit di risoluzione; I trasduttori sono 5 di luce, 3 di suono, 1 di gas, quindi serve una porta di trasmissione (multiplexer analogico) con 9 ingressi.

Affidandomi al solito esperto, mi viene consigliato un convertitore TLV1543C, ed un'interfaccia TPIC6B595. Il convertitore ha 10-bit di risoluzione, e 11 ingressi, quindi elimina il problema del multiplexer analogico; mentre l'interfaccia possiede 8 driver d'uscita, ed un driver per avere in cascata altre interfacce identiche. Inoltre i fili necessari al pilotaggio sono ridotti al minimo, essendoci una comunicazione di tipo seriale.

A questo punto abbiamo definito tutti i dispositivi; bisogna ora fare una stima della velocità di clock necessaria, per poi passare alla definizione del bus di interconnessione tra i moduli.



Sappiamo che il robot avanza a circa 30 cm al secondo; noi vogliamo che le elaborazioni più lunghe avvengano in tempi dei millisecondi, in cui il robot percorre una distanza trascurabile, in cui non varierà quasi per nulla le grandezze misurate.

La temporizzazione usata per il TLV1543C (una tra quelle del data sheet) impiega 10 colpi di clock in ognuno dei quali bisogna leggere il dato precedente, e mandare l'indirizzo prossimo (pipelining dei dati ed indirizzi); per ogni bit bisogna: preparare l'indirizzo (shift+set=4 istr.), settare il clock (1) prendere il dato (set+shift=4 istr.), abbassare il clock; in totale 10 istruzioni a clock; quindi 100 istruzioni per leggere 8 bit+il tempo di conversione (20 istruzioni). Per tutti i trasduttori, 1080 istruzioni circa.

Poi vogliamo paragonare tali letture; perlomeno trovare il massimo tra le luci (algoritmo a complessità lineare: 5 confronti, ossia prendi una lettura (2), prendine un'altra(1), confrontale(4), salva nel MAX la più grande(3); quindi $(1+1+4+3)*5= 50$ istruzioni circa;

Analogamente, vogliamo confrontare 3 letture del suono, quindi 30 istruzioni circa.

Infine, vogliamo accendere dei led con l'interfaccia parallela, che ha bisogno di scrivere su 8 registri, e per ognuno ci vogliono 3-5 istruzioni-> in totale 40 istruzioni circa.

Quindi, almeno vogliamo eseguire $1080+50+30+40=1200$ istruzioni circa in un millisecondo, quindi più o meno 1Mhz di clock. Riteniamo più che sufficiente il clock esterno da 4Mhz, anche perché il tempo di un millisecondo posto come minimo è esagerato.

(pagina 1 sul PIC)

(pagina 2 sul PIC)

(fig. schema funzionale)

Pin Diagram

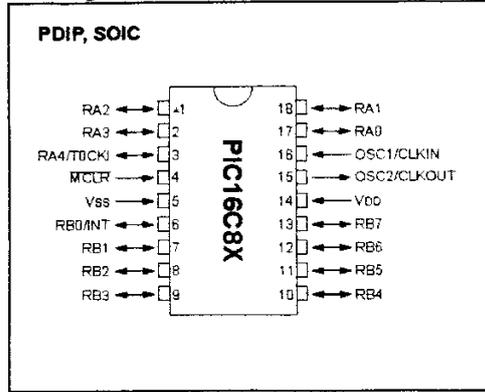
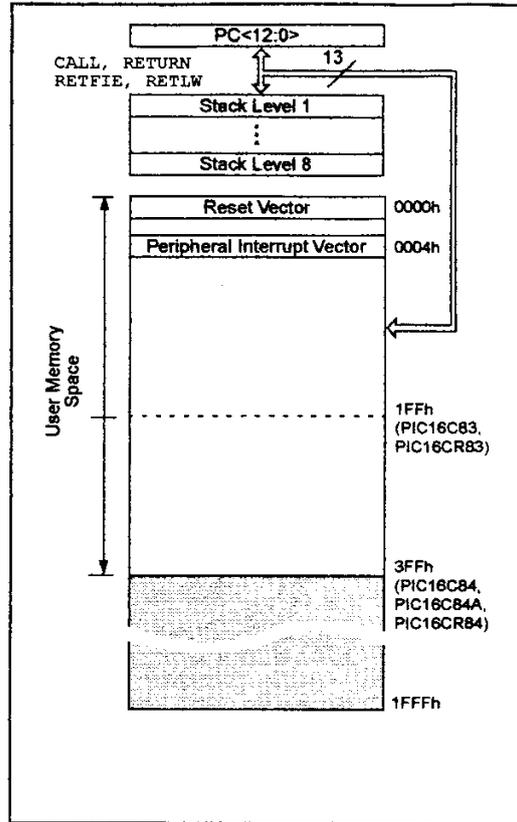


FIGURE 4-1: PROGRAM MEMORY MAP AND STACK



- In basso:
 programmatore della EEPROM
 del PIC16C84, da PC tramite
 porta seriale, usando il
 software PROG84.

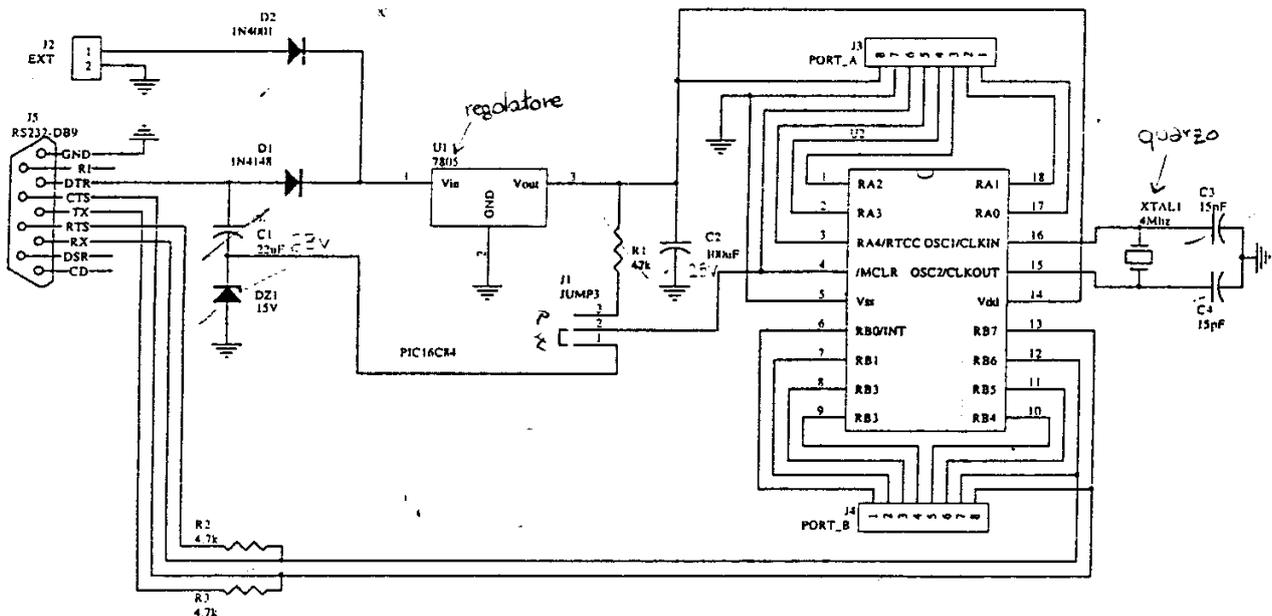


FIGURE 3-1: PIC16C8X BLOCK DIAGRAM

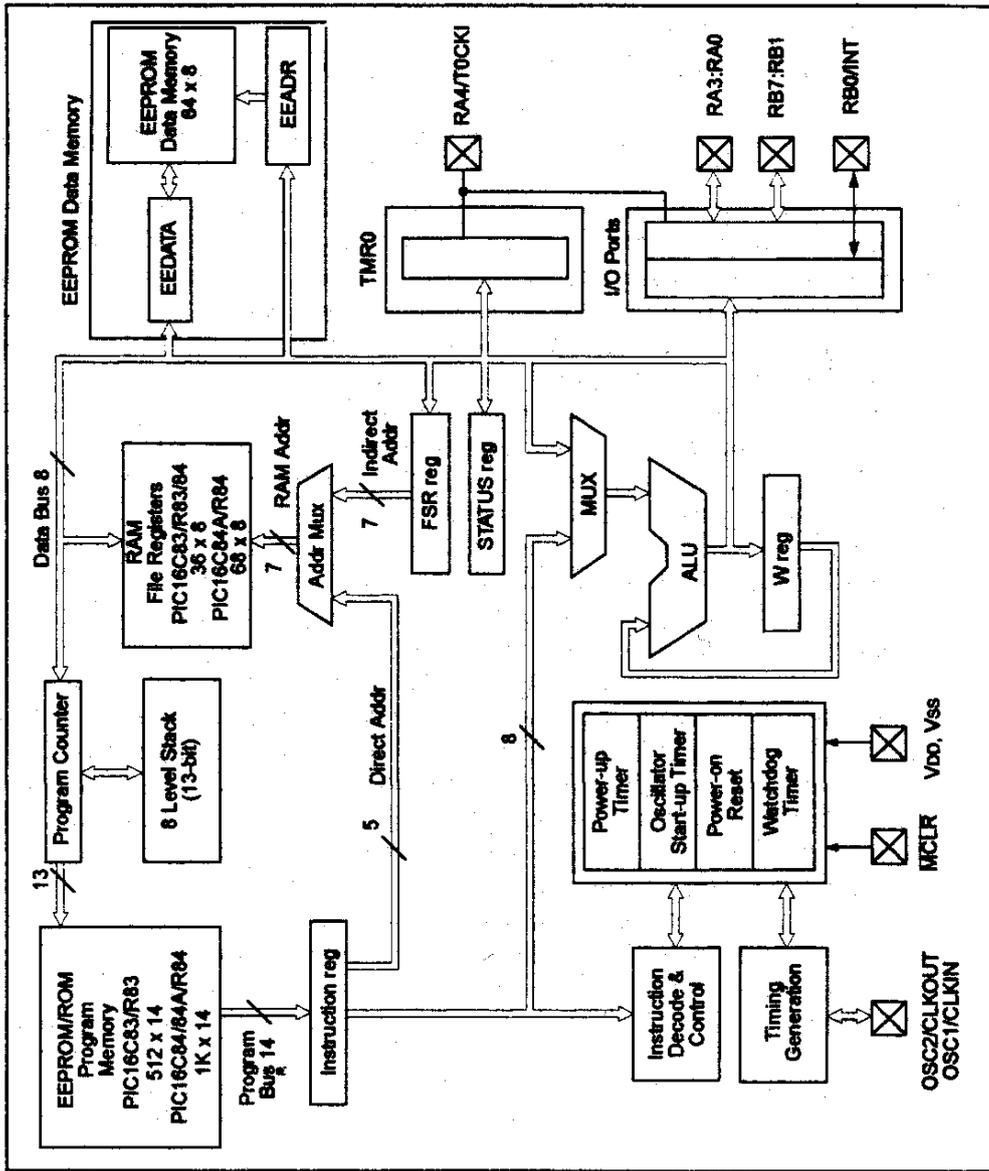


FIGURE 4-2: REGISTER FILE MAP

File Address	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾	File Address
00h	TMR0	OPTION	80h
01h	PCL	PCL	81h
02h	STATUS	STATUS	82h
03h	FSR	FSR	83h
04h	PORTA	TRISA	84h
05h	PORTB	TRISB	85h
06h			86h
07h	EEDATA	EECON1	87h
08h	EEADR	EECON2 ⁽¹⁾	88h
09h	PCLATH	PCLATH	89h
0Ah	INTCON	INTCON	8Ah
0Bh			8Bh
0Ch			8Ch
2Fh ⁽²⁾	36 / 68 General Purpose registers (SRAM)		AFh ⁽²⁾
30h ⁽²⁾			B0h ⁽²⁾
4Fh ⁽²⁾			CFh ⁽²⁾
50h ⁽²⁾			D0h ⁽²⁾
7Fh	Bank 0	Bank 1	FFh

Unimplemented data memory location; read as '0'.
 Note 1: Not a physical register.
 Note 2: The address depends on the device used.
 Devices with 36 bytes end at 2Fh, devices with 68 bytes end at 4Fh.

4. SCOMPOSIZIONE IN MODULI E LORO DEFINIZIONE.

4.1. Definizione del bus di interconnessione tra i moduli

Abbiamo detto che i piedini RB4...RB7 del PIC saranno dedicati agli infrarossi. Ci restano 9 piedini. Innanzitutto, vogliamo pilotare i motori in maniera immediata, senza mediazione dell'interfaccia parallela, per poter avere la migliore precisione; per governare i motori, l'integrato L298 abbisogna di tre piedini: due di direzione dei motori, ed uno di enable, che può essere anche un'onda quadra di PWM. Assegneremo i piedini RB1..RB3, con i pull-up abilitati.

Rimangono 6 piedini per gestire due integrati. Cercheremo di fare un BUS , poiché il convertitore A/D e l'interfaccia parallela si settano in momenti diversi.

Il convertitore ha bisogno di 5 fili di segnale: uno per essere settato (Chip Select); un altro per inviare il dato in seriale al micro (DOUT) , uno per ricevere in seriale l'indirizzo (ADDR), il segnale del clock (I/OCLK), e poi un'uscita per segnalare la fine conversione (EOC). E' possibile adottare varie temporizzazioni (vedere data sheet in appendice) a seconda che vogliamo un protocollo di comunicazione sincrono o asincrono, o usare il BUS durante il tempo di conversione; per motivi di semplicità e di leggerezza del codice, la scelta cade sulla modalità "fast 2" (fig. 10 del data sheet), in cui si utilizza un protocollo asincrono, con il chip sempre selezionato. Quindi, di tutti i fili del Chip, solo Chip Select non può essere condiviso, perché se venisse usato dall'interfaccia, selezionerebbe anche il convertitore, con problemi di conflitto. L'interfaccia è una rete sequenziale asincrona, e necessita di quattro fili: un filo per latchare i registri d'uscita (RCK), non condivisibile perché modifica i driver d'uscita, un filo per shiftare i dati nei registri interni (SRCK) condivisibile perché non modifica alcunché all'esterno; un filo di clear dei registri interni (SCLR) anch'esso condivisibile per lo stesso motivo, un filo per mandare i dati (SRIN) condivisibile, in quanto finché non sono latched fuori i registri esterni, non cambia nulla all'esterno; ed infine un filo di abilitazione globale (G) che può essere messo permanentemente a massa. Allora possiamo condividere i tre fili suddetti dell'interfaccia con altri tre del convertitore, come appare in figura a pagina seguente: in tutto necessitano 6 fili. Abbiamo così definito il BUS del sistema. E' il microprocessore che funge da unico MASTER, selezionando un dispositivo quando serve.

Osserviamo che il piedino EOC è di uscita, mentre tutti i piedini condivisibili dell'interfaccia sono di ingresso, allora sul micro avremo un piedino di ingresso-uscita.

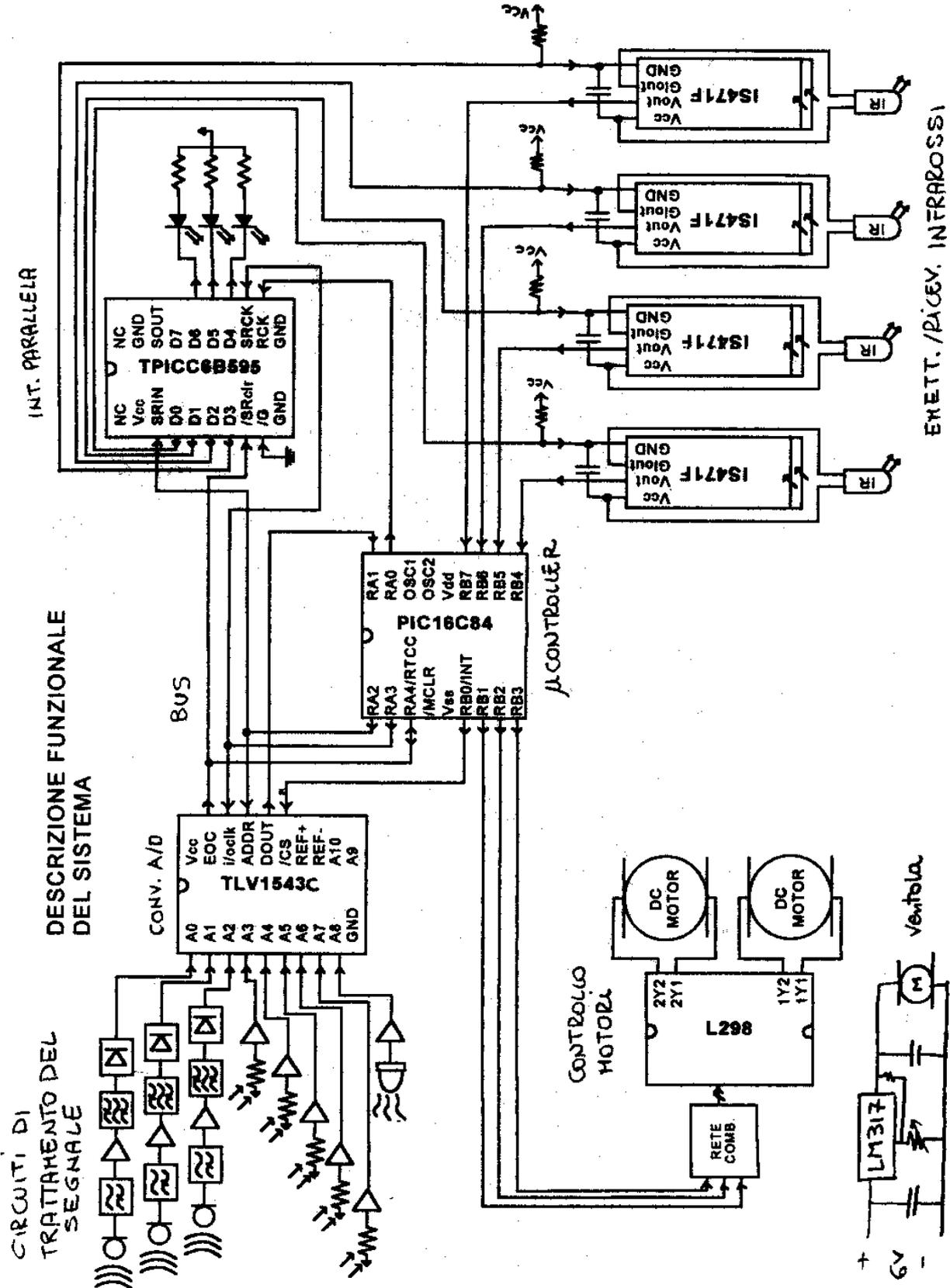
Si osservi infine, sullo stesso schema funzionale, la schematizzazione dei circuiti di trattamento del segnale: per ogni sensore c'è un amplificatore; nel caso del suono, c'è un filtro passa-basso, uno selettivo, ed un raddrizzatore, necessari, rispettivamente a filtrare i rumori, a selezionare i 4 Khz, ed a raddrizzarli per prendere l'ampiezza del segnale. Ciò sarà argomento del prossimo capitolo.

5. PROGETTAZIONE E SIMULAZIONE DEI SINGOLI MODULI.

Nelle pagine seguenti, verranno progettati i circuiti elettronici per elaborare i segnali rilevati dai trasduttori, ed in base ai data sheet, vengono dimensionati i circuiti per i vari integrati utilizzati; driver per i motori, ricevitore infrarossi, circuito del gas.

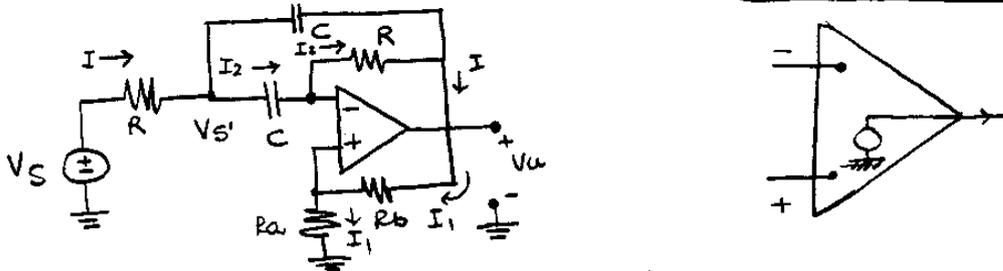
Vengono utilizzate alcune tecniche matematiche, come il Teorema di scomposizione, o strumenti grafici tipici dell'elettronica (diagrammi di Bode); e vengono poi simulati i circuiti al calcolatore, per verificare gli stessi.

In particolare, l'ultimo paragrafo mostra i valori commerciali dei componenti nei circuiti sintetizzati, leggermente diversi da quelli teorici previsti; ed infine viene calcolata approssimativamente la corrente assorbita da tutto il sistema, per una stima della durata delle batterie.



5.1 Dimensionamento del filtro di Delyannis

(A) Calcolo della f.d.t., nell'ipotesi di "corto circuito virtuale" (c.c.v.)



• Se l'amplificatore operazionale fosse ideale, avrei una impedenza d'ingresso infinita (cioè correnti all'ingresso praticamente piccolissime) ed un fattore di amplificazione infinito, con una resistenza d'uscita minima; quindi, per avere una tensione d'uscita finita, la tensione tra gli ingressi deve essere piccolissima. In base a tale ipotesi (c.c.v.) si ha

$$\begin{cases} V^+ = V^- \\ I^+ = I^- = 0 \end{cases} \Rightarrow I_1 = \frac{V^+}{R_a} = \frac{V_u}{R_a + R_b} \Rightarrow \boxed{V^+ = V_u \frac{R_a}{R_a + R_b} = \frac{V_u}{A_v}} \quad (1)$$

$$\Rightarrow I_2 = (V_s' - V^-) C_s = \frac{V^- - V_u}{R} \Rightarrow \boxed{V_s' = V^- \left(1 + \frac{1}{R C_s}\right) - \frac{V_u}{R C_s}} \quad (2)$$

$$\Rightarrow I = \frac{V_s - V_s'}{R} = \frac{V_s - V_u}{R \left(\frac{1}{C_s} \parallel (R + \frac{1}{C_s})\right)} \Rightarrow \boxed{V_s = V_s' \left[1 + \frac{(2 + R C_s) R C_s}{1 + R C_s}\right] - \frac{V_u (2 + R C_s) R C_s}{1 + R C_s}}$$

$$= \frac{1}{C_s} \cdot \frac{1 + R C_s}{C_s} \cdot \frac{C_s}{2 + R C_s} = \frac{1 + R C_s}{2 + R C_s} \cdot \frac{1}{C_s}$$

• facendo la sostituzione (1):

$$V_s' = V_u \left[\frac{1}{A_v} \left(1 + \frac{1}{R C_s}\right) - \frac{1}{R C_s} \right] = V_u \frac{1 - A_v + R C_s}{A_v R C_s}$$

• facendo la sostituzione (2):

$$V_s = V_u \left[\frac{1 - A_v + R C_s}{A_v R C_s} \cdot \left(1 + \frac{(2 + R C_s) R C_s}{1 + R C_s}\right) - \frac{(2 + R C_s) R C_s}{1 + R C_s} \right]$$

$$= V_u \frac{(1 - A_v)(1 + R C_s)}{A_v R C_s} \cdot \frac{(2 + R C_s) R C_s}{1 + R C_s} + \frac{1 - A_v + R C_s}{A_v R C_s} = V_u \frac{1 - A_v}{A_v} \cdot \frac{1}{R C_s} \left((R C_s)^2 + \left[2 + \frac{1}{1 - A_v}\right] R C_s + 1 \right)$$

• conclusione: $\frac{V_u}{V_s} = H(s) = \frac{R C_s \frac{A_v}{1 - A_v}}{(R C_s)^2 + \frac{2 + \frac{1}{1 - A_v}}{1 - A_v} R C_s + 1}$

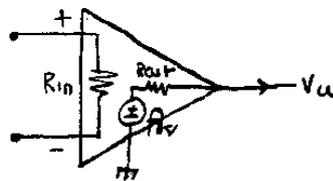
ⓑ) Limiti di solidità dell'ipotesi di corto circuito virtuale

- modello dell'amp. op. TL081 (ved. parametri di simulazione)

$$\begin{cases} A_o = 10^5 & (\text{Open-loop gain}) \\ R_{in} = 10^{12} \Omega \\ R_{out} = 75 \Omega \\ F_U = 4 \text{ MHz} & (\text{Unity-gain bandwidth}) \approx A_o f_p \Rightarrow f_p = 40 \text{ Hz} \end{cases}$$

quindi la risposta in frequenza sarà $A_{vol} \frac{A_o}{1 + j f/f_p}$

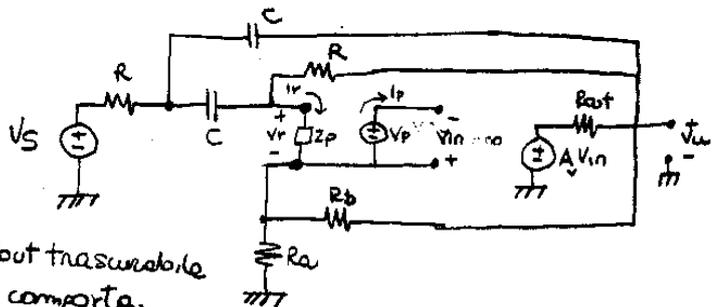
- la resistenza d'ingresso è così grande, che persino lo strato di polvere che si deposita sul componente tra i due ingressi, ha resistenza inferiore. Quindi la considereremo infinita.



- Applico il teo di scomposizione (teorema del Pella-grimi) facendo un taglio "differenziale", cioè all'ingresso dell'operazionale

$$\epsilon = \frac{I_p}{V_s} \Big|_{V_p=0} = 0 \Rightarrow Z_p = Z_{in} = \infty$$

$$\gamma = \frac{V_u}{V_s} \Big|_{V_p=0} = 0 \quad \text{facciamo l'ipotesi } R_{out} \text{ trascurabile poi vedremo cosa comporta}$$



quindi $A_F = \frac{V_u}{V_s} = \frac{dA}{1 - \beta A} = -\frac{\alpha}{\beta} \cdot \frac{-\beta A}{1 - \beta A}$; ora, se fosse $A_v \rightarrow \infty$ io devo ottenere

A_F ottenuta con l'ipotesi del c.c.v. (però vale il teo di scomposizione), e tale

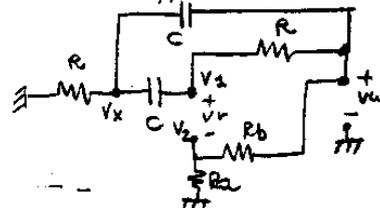
$$A_F \text{ è in modulo } \left| -\frac{\alpha}{\beta} \cdot \frac{-\beta A}{1 - \beta A} \right| \approx \left| -\frac{\alpha}{\beta} \right| \quad \text{però } |A_v| \rightarrow \infty \Rightarrow |A| \rightarrow \infty \Rightarrow |\beta A| \rightarrow \infty \Rightarrow |A_F| \approx 1$$

quindi utilizzare il c.c.v. equivale a supporre $|A_F| \approx 1$ oppure $\beta A < 0$ e $|\beta A| \gg 1$

$$A = \frac{V_u}{V_p} \Big|_{V_s=0} = \frac{A_v V_{in}}{-V_{in}} = -A_{vol}$$

$$\beta = \frac{V_r}{V_u} \Big|_{V_s=0} = \frac{V_1 - V_2}{V_u} = \frac{V_1}{V_u} \Big|_{V_s=0} - \frac{V_2}{V_u} \Big|_{V_s=0}$$

$$V_2/V_u \text{ è un partitore resistivo } \rightarrow \frac{R_a}{R_b + R_a} = \frac{1}{A_v}$$



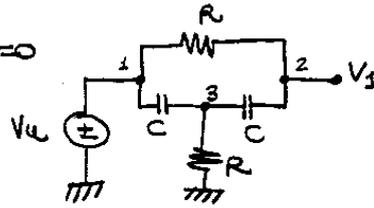
• Considero la rete per il calcolo di $V_2/V_u |_{V_S=0}$

per $\omega \rightarrow 0$ ho rami aperti $\rightarrow V_1 = V_u$

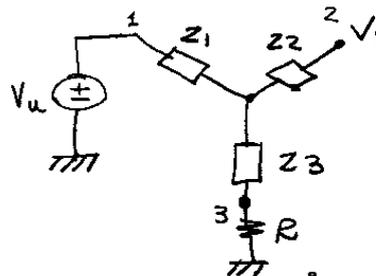
per $\omega \rightarrow \infty$ ho rami cortocircuitati $\rightarrow V_1 = V_u$

ci sono due el. reattivi ind. $\rightarrow 2$ poli

quindi 2 zeri



trasmf. triangolo stella



$$Z_1 = \frac{Z_{12} \cdot Z_{13}}{\sum Z} = \frac{R/Cs}{2/Cs + R} = \frac{R}{2 + RCs}$$

$$Z_3 = \frac{1/(Cs)^2}{2/Cs + R} = \frac{1}{Cs(2 + RCs)}$$

$$Z_2 = \frac{1}{Cs(2 + RCs)} + R = \frac{(1 + RCs)^2}{Cs(2 + RCs)}$$

$$\beta_1 = \frac{Z_3 + R}{Z_3 + Z_1 + R} = \frac{(1 + RCs)^2}{Cs(2 + RCs)} \cdot \frac{1}{\frac{(1 + RCs)^2}{Cs(2 + RCs)} + \frac{R}{2 + RCs}} = \frac{(1 + RCs)^2}{Cs(2 + RCs)} \cdot \frac{Cs(2 + RCs)}{1 + (RCs)^2 + 2RCs + RCs}$$

$$= \frac{(1 + RCs)^2}{1 + 3RCs + (RCs)^2}$$

gli zeri sono coincidenti, quindi i poli sono simmetrici rispetto ad essi, in scala logaritmica

$$s = \frac{-3 \pm \sqrt{5}}{2} \cdot \frac{1}{RC}$$

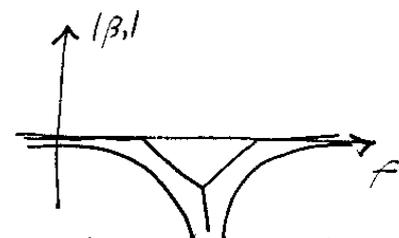
$$\sqrt{s_1 s_2} = s_0$$

$$\beta = \beta_1 - \frac{1}{A_v} = \frac{(1 + RCs)^2}{1 + 3RCs + (RCs)^2} - \frac{1}{A_v} =$$

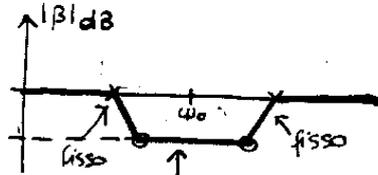
$$\frac{A_v + A_v(RCs)^2 + 2A_vRCs - 1 - 3RCs - (RCs)^2}{A_v(1 + 3RCs + (RCs)^2)} = \frac{(A_v - 1) + (2A_v - 3)RCs + (A_v - 1)(RCs)^2}{A_v[1 + 3RCs + (RCs)^2]}$$

$$\frac{A_v - 1}{A_v} \cdot \frac{1 + \frac{2A_v - 3}{A_v - 1}RCs + (RCs)^2}{1 + 3RCs + (RCs)^2}$$

l'effetto di A_v è di allontanare gli zeri



rete resistiva \rightarrow poli negativi e reali, gli zeri \rightarrow complessi



$$\beta_{CB} = \frac{1}{A_v}$$

$$\omega_0 = 1/RC$$

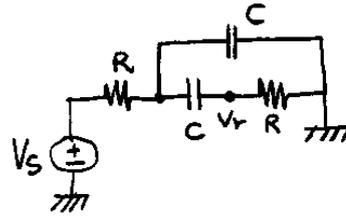
• Se $A_v \rightarrow \infty$ ho $\beta = \beta_1$ (filtro più selettivo)

Se $A_v \rightarrow 0$ ho poli e zeri coincidenti

$$|\beta|_{s=1/RC} = \frac{A_v - 1}{A_v} \cdot \left(1 - \frac{2A_v - 3}{A_v - 1} + 1 \right) = \frac{A_v - 1}{A_v} \cdot \frac{3 - 2}{A_v - 1} = \frac{1}{A_v}$$

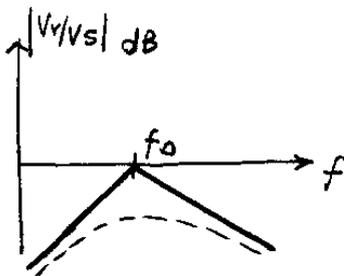
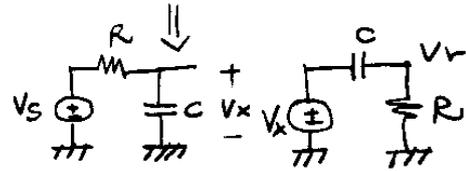
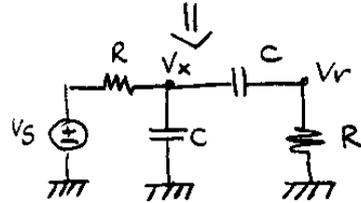
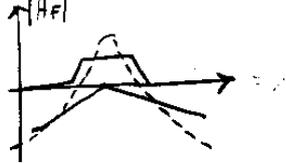
• Calcolo di d

$d = \frac{V_r}{V_s} \Big|_{V_p=0}$ è un passa basso seguito da un passa-alto

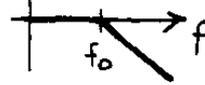


$$d = \frac{1}{1+RCS} \cdot \frac{RCS}{1+RCS}$$

$$|A(f)| = \left| \frac{d}{\beta} \right|$$



$$V_x/V_s$$



$$V_r/V_x$$



$$f_0 = \frac{1}{2\pi RC}$$

• limiti di 'velocità'

$$|\beta A| \gg 1;$$

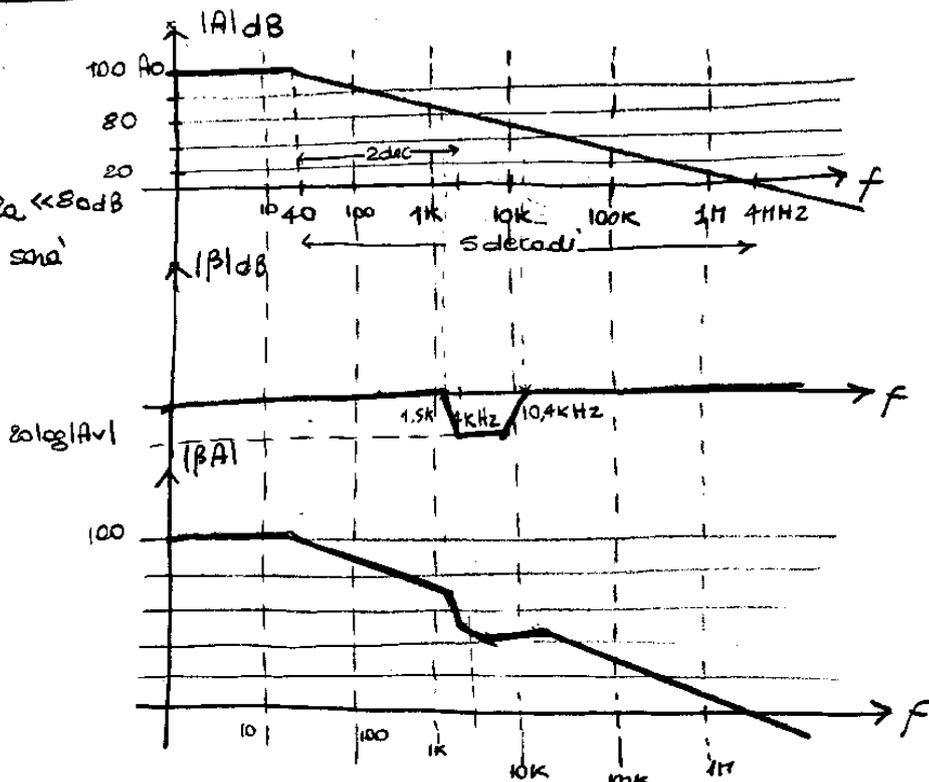
$$\text{Se } A_v = 1 + R_b/R_a \ll 80 \text{ dB}$$

allora il limite sarà

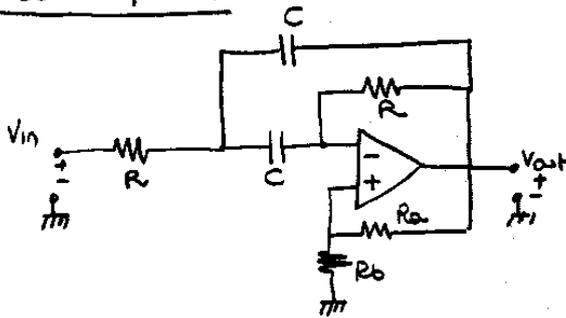
4 MHz

Cioè il filtro

non deve amplificare molto.



• Filtro di Delayamis



$$H(s) = \frac{A_v}{1-A_v} \cdot \frac{s/\omega_0}{(s/\omega_0)^2 + \frac{1}{Q}(s/\omega_0) + 1}$$

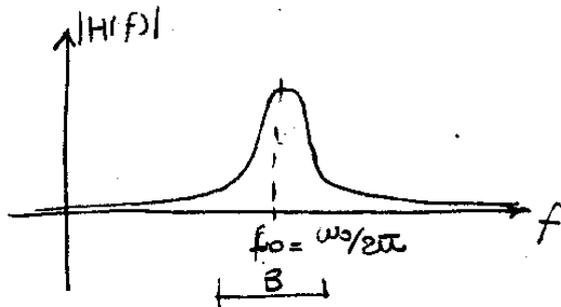
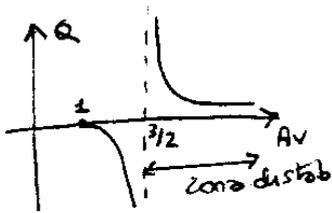
Limiti di validità:

$$f_H = \frac{1 \text{ MHz}}{A_v}$$

$$A_v = 1 + \frac{2R_a}{R_b}$$

$$\omega_0 = 1/RC$$

$$Q = \frac{1-A_v}{3-2A_v}$$



$$Q = \frac{f_0}{B} \text{ (selettività)}$$

← banda

• Valori commerciali di resistenza e capacità

$$1 \text{ k}\Omega < R < 1 \text{ M}\Omega \quad (\text{es. tipico } R = 33 \text{ k}\Omega)$$

$$1 \text{ mF} < C < 1 \mu\text{F}$$

dimensionamento

$f_0 = 4 \text{ kHz}$ fissato; si sceglie di norma $B = 10\% f_0 = \boxed{400 \text{ Hz}} \Rightarrow Q = 10$

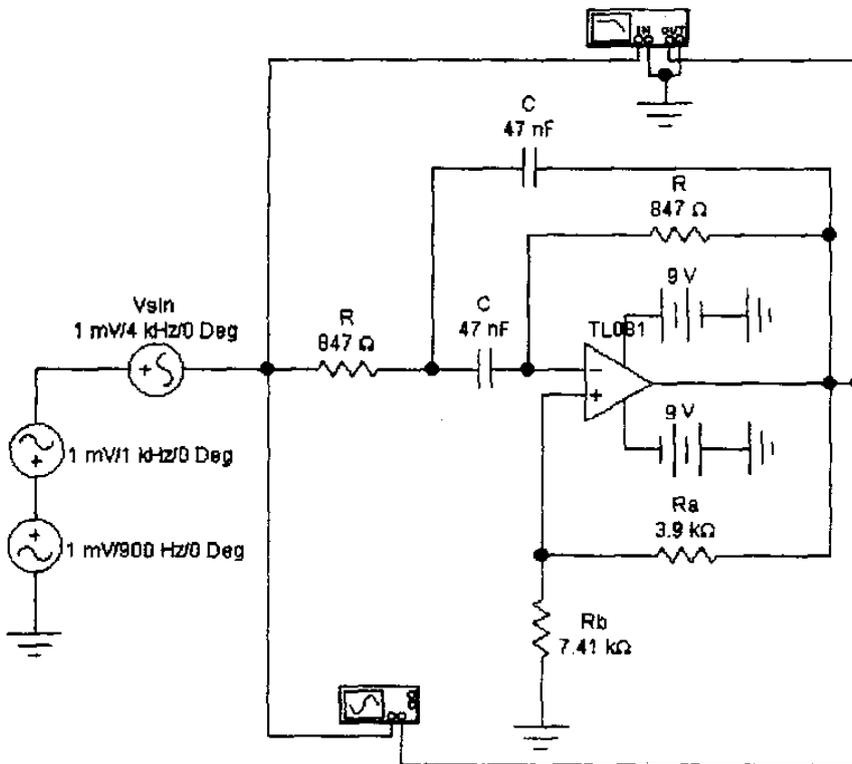
$$A_v = \frac{3Q-1}{2Q-1} = \frac{30-1}{20-1} = \frac{29}{19} \approx 1,526 \rightarrow \frac{R_a}{R_b} = 0,526 \quad \text{es. } \begin{cases} R_a = 3,9 \text{ k}\Omega \\ R_b = 7,41 \text{ k}\Omega \end{cases}$$

$$RC = \frac{1}{\omega_0} = \frac{1}{2\pi f_0} = 39,8 \cdot 10^{-6}; \text{ posto } R = 847 \Omega \rightarrow C = \boxed{47 \text{ nF}}$$

$f_H = 654 \text{ kHz}$ ok

Valutiamo il modulo per $f = f_0$

$$|H(f_0)| = \frac{A_v}{1-A_v} \cdot \frac{1/f_0}{\sqrt{1 + \frac{1}{Q^2} + 1}} = Q \cdot \frac{A_v}{1-A_v} \approx 29 = 29 \text{ dB}$$

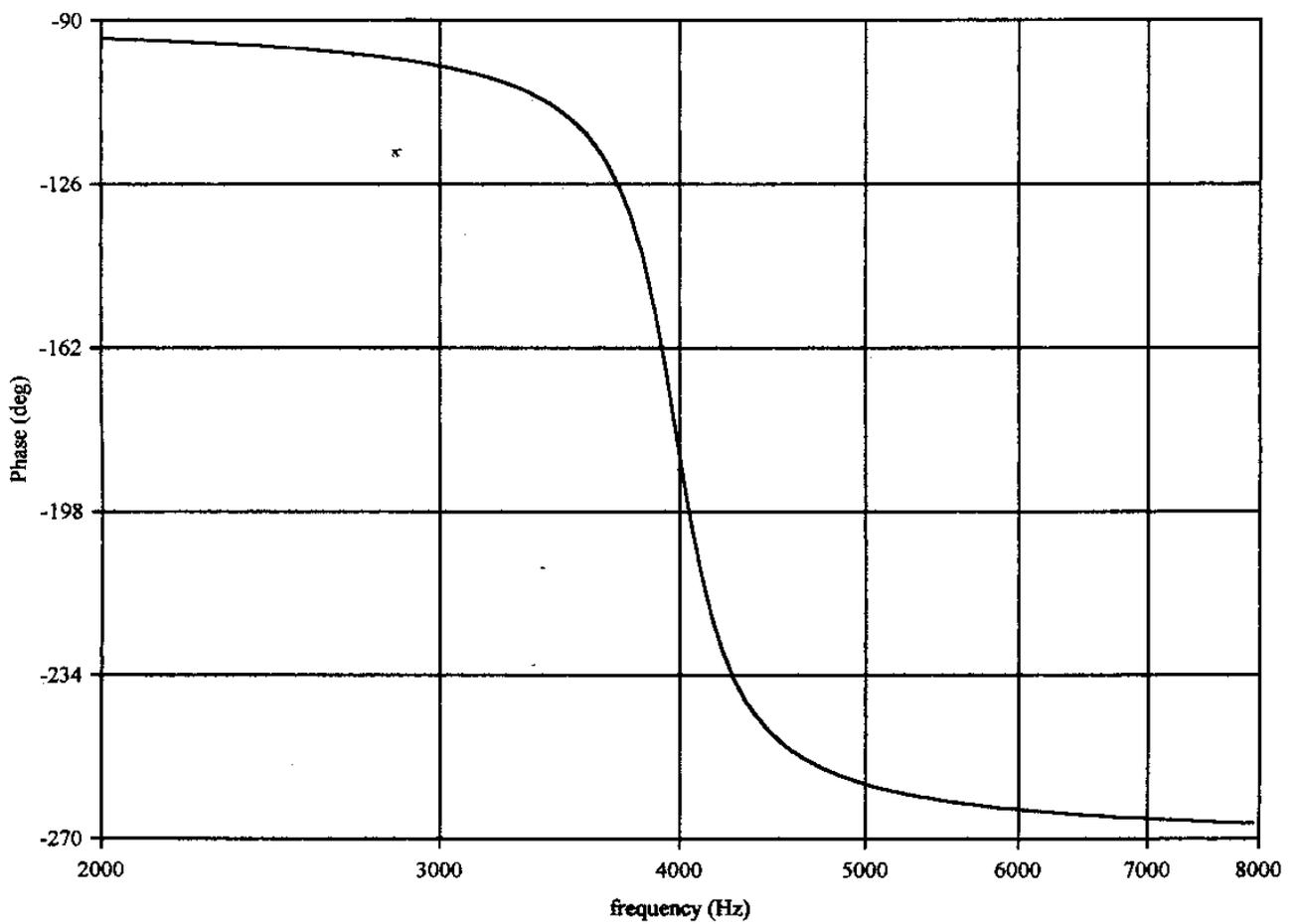
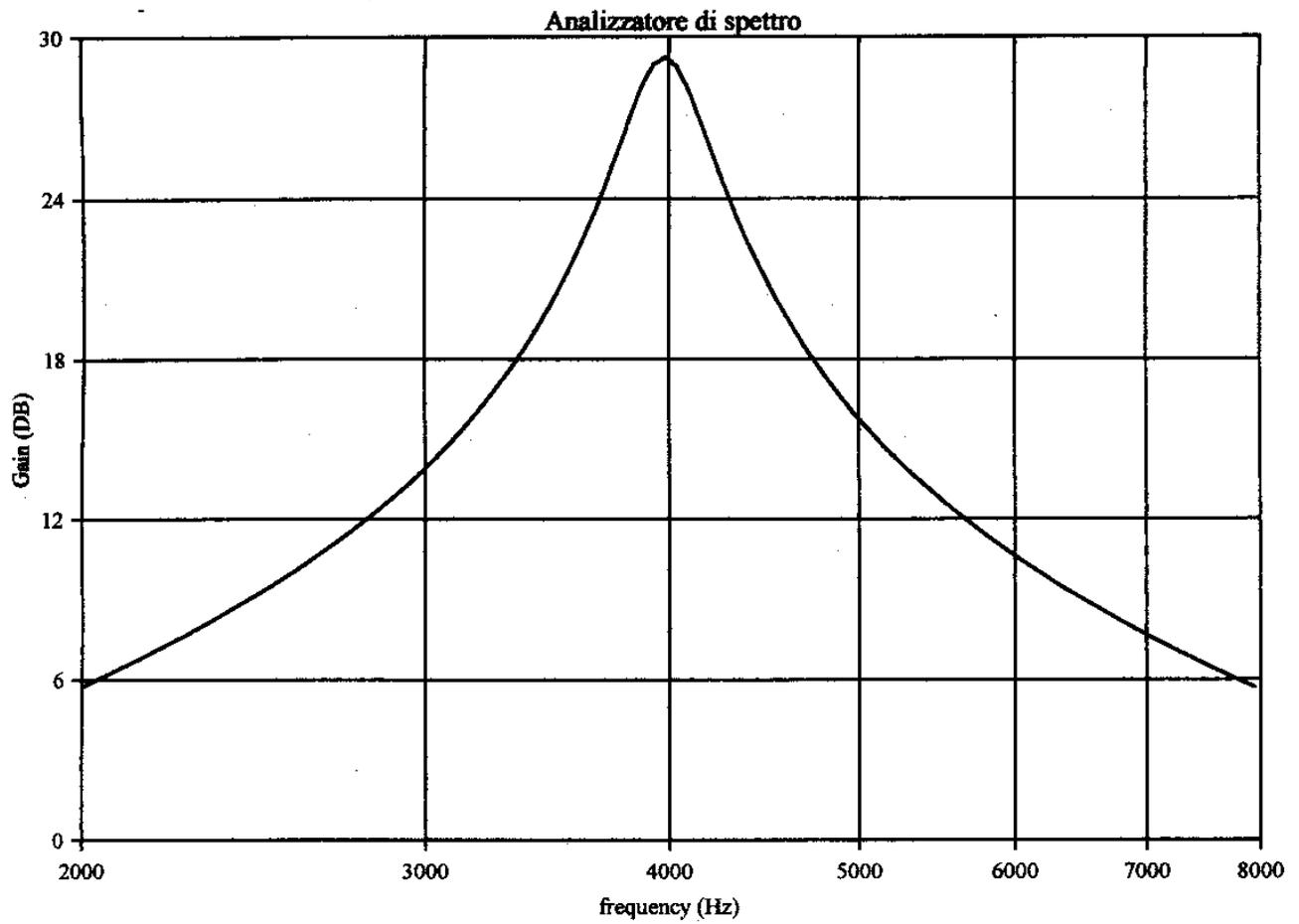


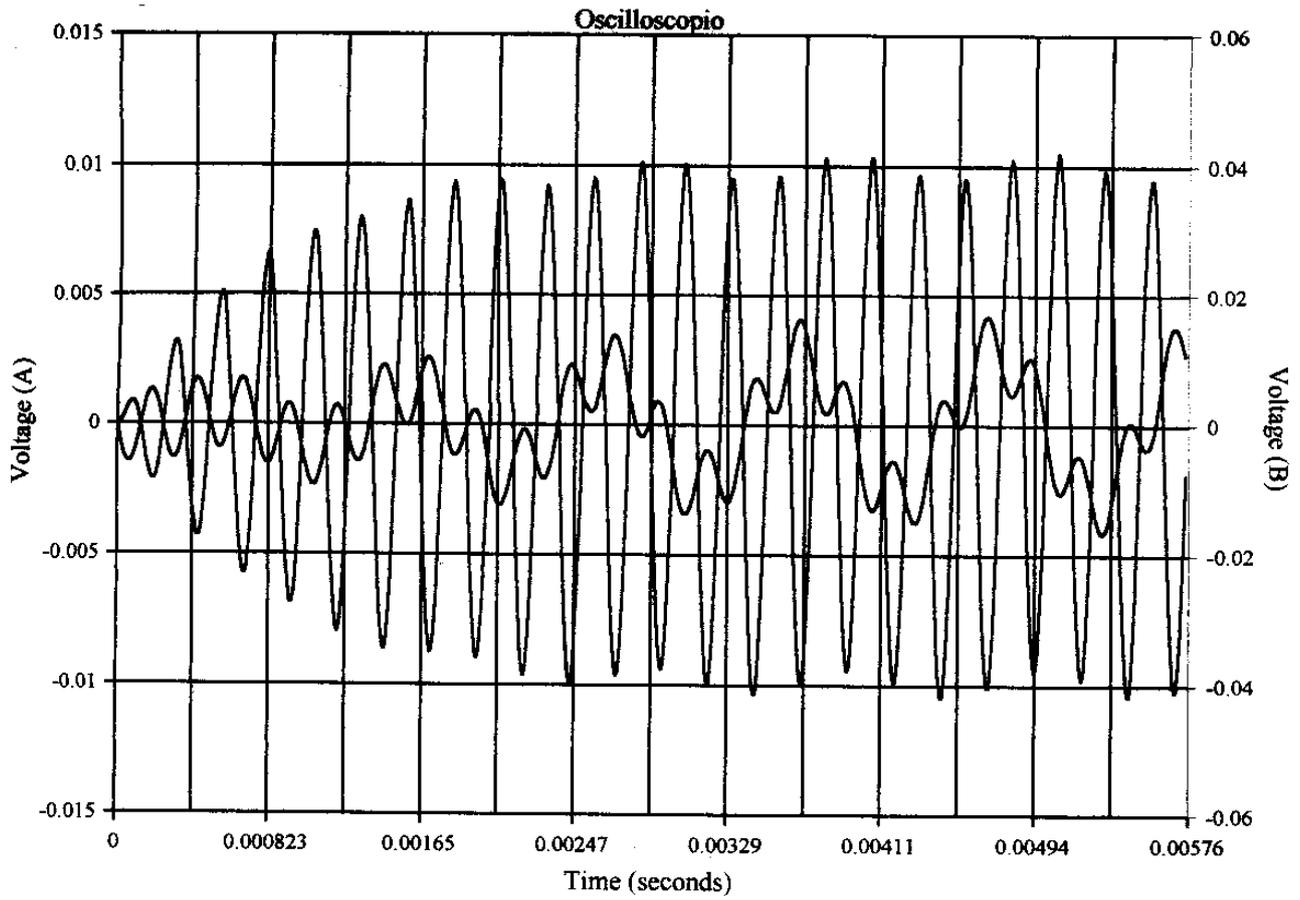
Parts list

Description	Quantity	Reference ID
5-Terminal Opamp, TL081	1	AR1
AC Voltage Source, 1mV, 900Hz, 0Deg	1	V7
AC Voltage Source, 1mV, 4kHz, 0Deg	1	V0
AC Voltage Source, 1mV, 1kHz, 0Deg	1	V6
Battery, 9V	2	V5,V1
Capacitor, 47nF	2	C1,C0
Connector	8	CONN10,CONN8,CONN7,CONN0,CONN2,CONN3,
Ground	5	0,0,0,0,0
Node	9	8,9,10,7,6,4,3,2,1
Resistor, 7.41kΩ, 0Ω/°C, 0Ω/°C²	1	R0
Resistor, 847Ω, 0Ω/°C, 0Ω/°C²	2	R3,R2
Resistor, 3.9kΩ, 0Ω/°C, 0Ω/°C²	1	R1

5-Terminal Opamp Model "TL081" Library "opa77-tl"

VCC	Positive power supply	18	V
VEE	Negative power supply	-18	V
CC	Compensation capacitance	3e-11	F
A	Open-loop gain	1e+05	
RI	Input resistance	1e+12	Ω
RO	Output resistance	75	Ω
VOS	Input offset voltage	0.005	V
IOS	Input offset current	2.5e-11	A
IBS	Input bias current	5e-11	A
VSW+	Positive voltage swing	13.5	V
VSW-	Negative voltage swing	-13.5	V
CMRR	Common mode rejection ratio	100	dB
ISC	Output short circuit current	0.03	A
SR	Slew rate	13	V/us
FU	Unity-gain bandwidth	4e+06	Hz
PM	Phase margin	50	Deg

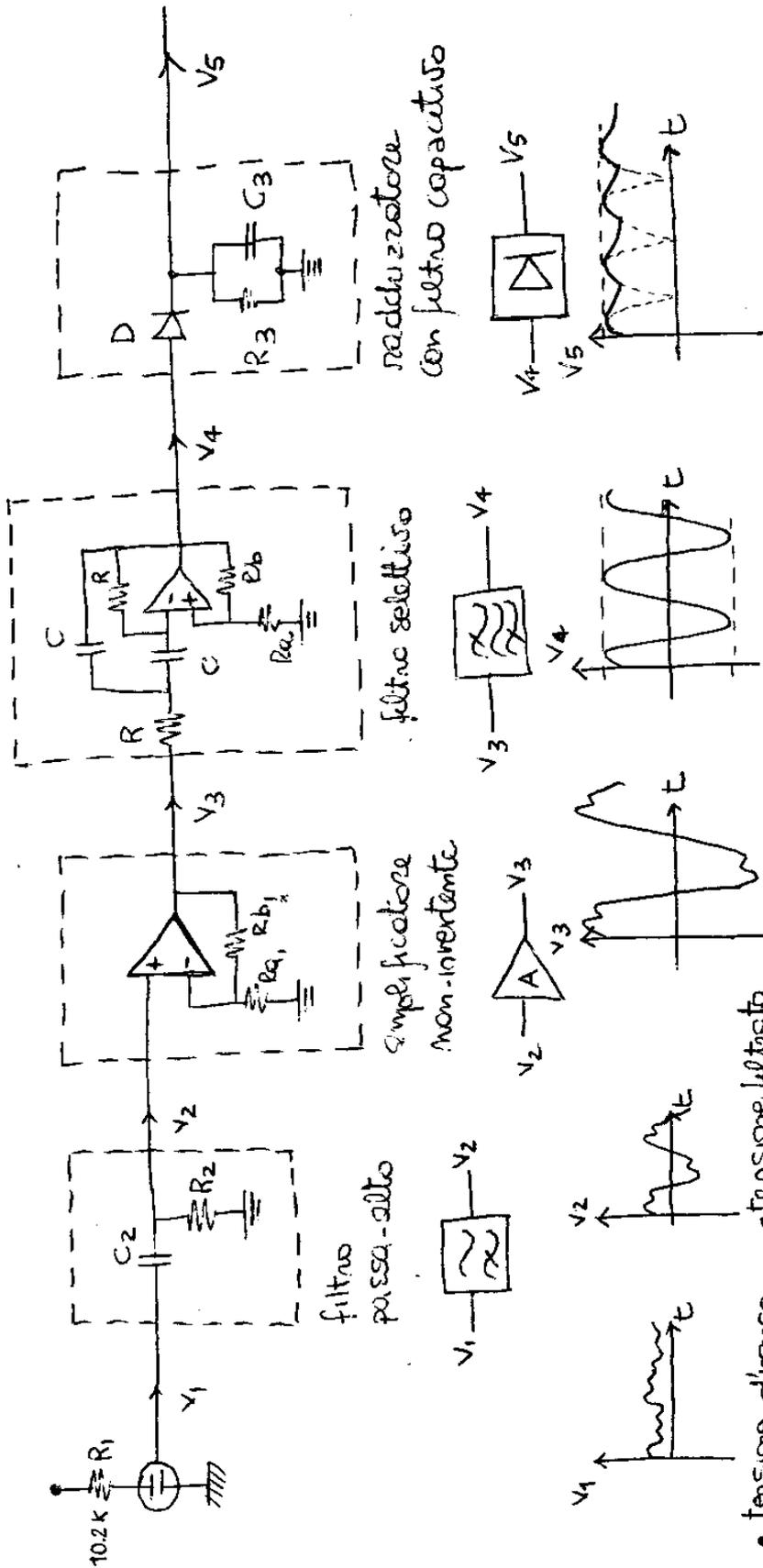




- fig. mandando in ingresso un'onda composta ($900\text{ Hz} + 1\text{ kHz} + 4\text{ kHz}$)
il sistema produce una sinusoida quasi costante a 4 kHz

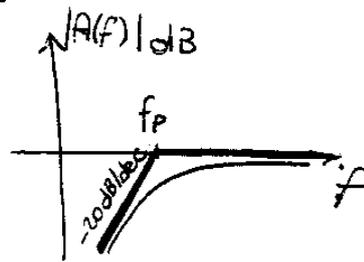
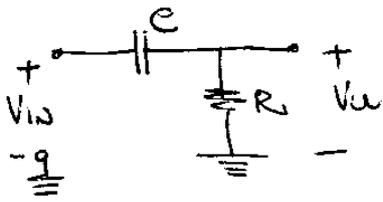
5.2 Dimensionamento e simulazione del circuito del suono

- Composizione del circuito del suono



- tensione d'ingresso
è dell'ordine dei mV ed è la sovrapposizione di rumore termico, (decime di Hz) vibrazione dei motori (100-200 Hz) voce del pubblico (centinaia di Hz) bisogna filtrarla con banda inferiore $f = \pm 1 \text{ kHz}$
- tensione filtrata
ora che abbiamo eliminato i numeri bassi possiamo amplificare il segnale fino a decimi di volt, quindi di un fattore 100 circa
- il filtro seleziona una banda vicino ai 4 kHz
amplificandola di qualche decina
- il segnale deve essere raddoppiato così da avere una tensione continua proporzionale all'ampiezza da convertire in digitali

• simulazione del filtro passa-alto



per basse freq. C è un ramo aperto $\rightarrow V_u/V_S = 0$ (zero nell'origine)

per alte freq. C è un corto circuito $\rightarrow V_u/V_S = 1$

il polo è $f_p = \frac{1}{2\pi RC}$ (metodo della res. vista)

$$A(f) = \frac{jf/f_p}{1 + jf/f_p}$$

• imponendo $f_p = 1 \text{ kHz} \Rightarrow RC = \frac{1}{2\pi f_p} = 1.59 \cdot 10^{-4}$

ora scegliamo la capacità (che è più difficile da trovare come valore commerciale) e si ottiene il valore di R

• le capacità commerciali sono dell'ordine dei nF. Non ha senso mettere capacità dei pF perché le capacità parassite sono dell'ordine dei f0pF. Le capacità dei μF sono elettrolitiche, cioè devono essere connesse solo in un verso

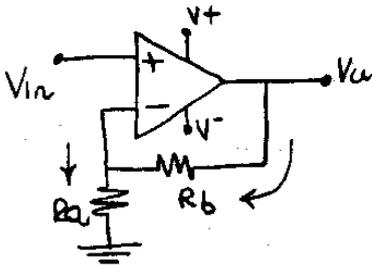
• le resistenze commerciali vanno da qualche Ohm al Mega Ohm al di sotto, ci sono i contatti con le piste (5Ω), al di sopra, la polvere sul componente che crea uno strato di $5 \text{ M}\Omega$.

• scelgo $C = 100 \text{ nF} \rightarrow R = 1.59 \text{ k}\Omega$

il valore commerciale più vicino è $R = 1.62 \text{ k}\Omega$ (all'1% di tolleranza)

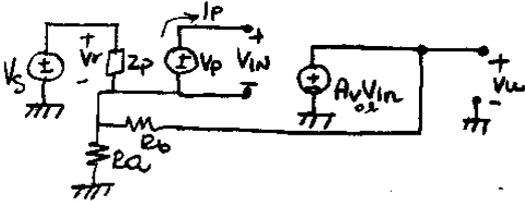
quindi il polo reale sarà $f_p = 983 \text{ Hz}$

• Dimensionamento dell'amplificatore



• dimensionazione deve essere più alta della dinamica del segnale d'uscita, altrimenti l'operazionale va in zone non lineari nelle vicinanze del max -
 Poiché i nostri segnali vanno da 0 a 5-6V, alimenteremo tutti gli operazionali con pile da 9V 100 Ah - ciò va bene, perché gli operazionali consumano μA di corrente -

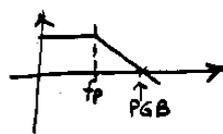
• limiti di validità del c.c.v.
 del teo di scomposizione (taglio differenziale)



$e = \frac{I_p}{V_s | V_p = 0} = 0; \rightarrow Z_p = Z_n = \infty$

$\delta = \frac{V_u}{V_s | V_p = 0} = 0; A = \frac{V_u}{V_p | V_s = 0} = A_{v_{ol}} = \frac{A_0}{1 + jf/f_p}; \beta = \frac{V_r}{V_u | V_s = 0} = -\frac{R_a}{R_b + R_a} = -\frac{1}{A_v}$

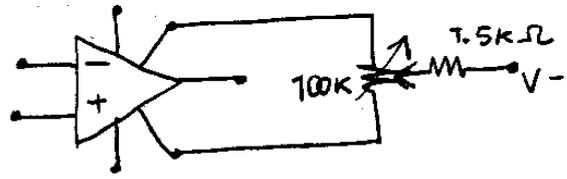
quindi $|\beta A| = \frac{A_0}{A_v} \cdot \frac{1}{jf/f_p + 1}$



$f_H = \frac{A_0}{A_v} \cdot f_p = \frac{10^5}{10^2} \cdot 40 = 40 \text{ KHz}$

Va bene, x di sono di max sui 16 KHz di segnale voce \rightarrow non distorce, aggiungendo nuove frequenze

• il circuito di regolazione offset è già fissato (v. simulazione, o data sheet TL081), e permette di variare il V_{min} e il V_{max} d'uscita senza cambiare la dinamica del segnale - In tal modo, se ho ad es. un uscita da 1.3V a 5.3V, posso portarla da 0 a 5V, regolando



il trimmer -

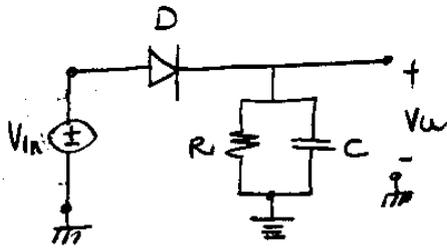
• Sotto l'ipotesi di corto circuito virtuale, si ha

$$\begin{cases} V^+ = V^- \\ I^+ = I^- = 0 \end{cases} \Rightarrow \frac{V_u - V^-}{R_b} = \frac{V^-}{R_a}$$

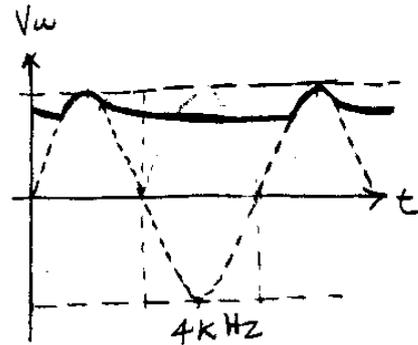
$\Rightarrow V_u = V^+ (1 + \frac{R_b}{R_a}) \cong 100 = A_v$

fissato $R_a = 1K\Omega$ come R_b metto un trimmer, così da poter variare l'amplificazione - Da prove sperimentali si vede che si può porre una $R_b' = 60.7K\Omega$ in serie ad un potenziometro da $22K\Omega$ così che R_b varia tra $60.7K\Omega$ e $82.7K\Omega$

• Dimensionamento del raddrizzatore con filtro capacitivo



- funzionamento: il diodo provvede ad aprirsi quando V_{in} è minore di V_f , così che sul filtro RC va solo la parte d'onda positiva (a meno di V_f). Quest'onda mentre sale carica il condensatore, poi comincia a scendere fino ad andare più veloce della scarica dello stesso. Così il Diodo è contro-polarizzato e si apre, e rimane la capacità che si scarica su R come $e^{-t/RC}$.
- il Diodo deve essere un diodo di segnale, come il 1N4148, o meglio ancora un diodo con $V_f \approx 0,7V$ (così che la caduta è ancora di meno, rispetto al caso ideale).

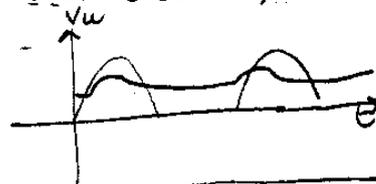


- $\tau = RC$ deve essere \gg del periodo, così che il segnale rimane rettilineo nel tratto esponenziale, ma deve essere max di μs , così se il segnale varia in ampiezza, V_u lo segue.

$$\frac{1}{4 \text{ kHz}} \ll RC \ll 5s$$

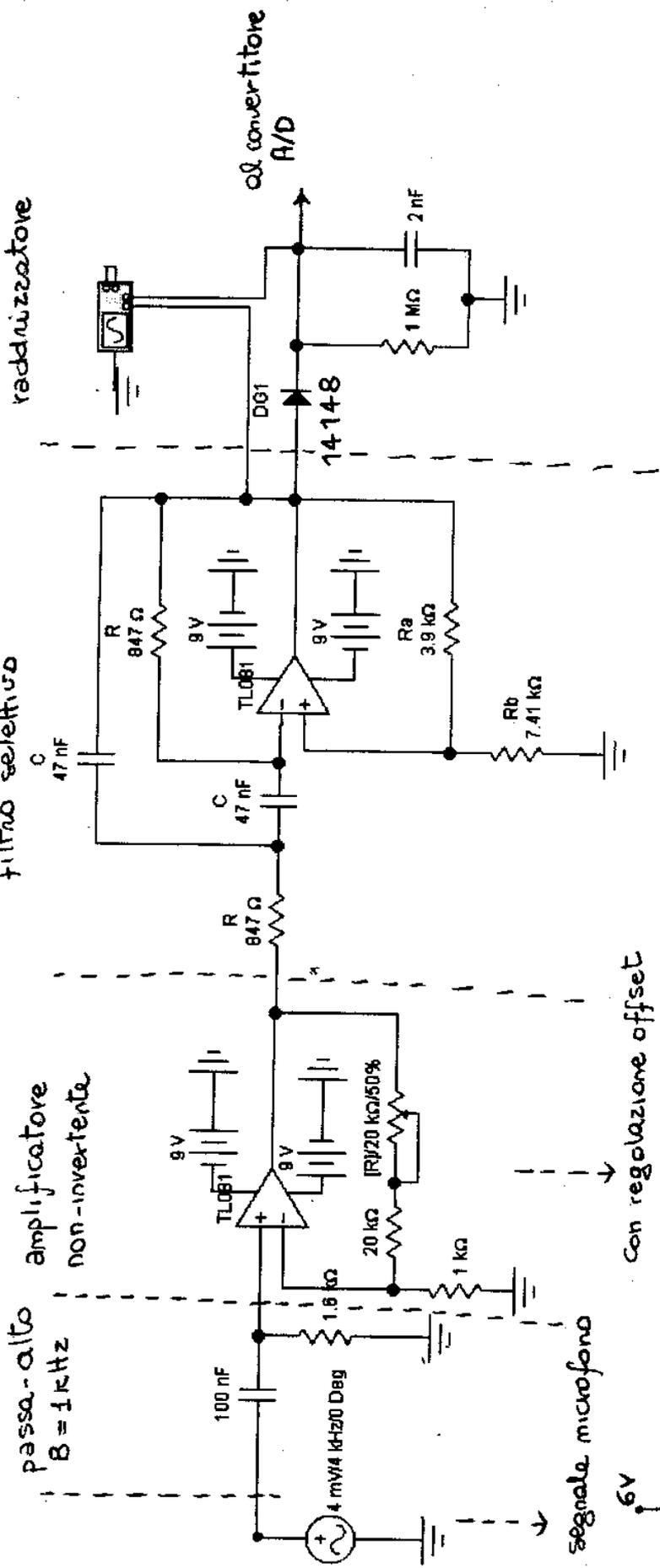
$$4 \text{ kHz} = 0,25 \mu s$$

- C deve essere piccola, così si carica subito, e ammonta al max è carico, altrimenti fa così!



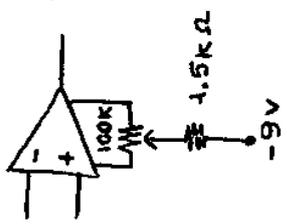
poniamo $C = 2.2 \text{ nF}$ valore commerciale

- R deve essere grossa, così che nella scarica passa poca corrente, e dura sufficientemente.
- prendo $\tau = 2,25 \Rightarrow R = 1M\Omega$
- oss i parametri R e C sono stati ottenuti prendendo dei tentativi e simulando il circuito elettrico.



• resistenze precise (tolleranza 1%)

Con regolazione offset



segnale microfono

capsula microfonica pre-amplificata

Parts list

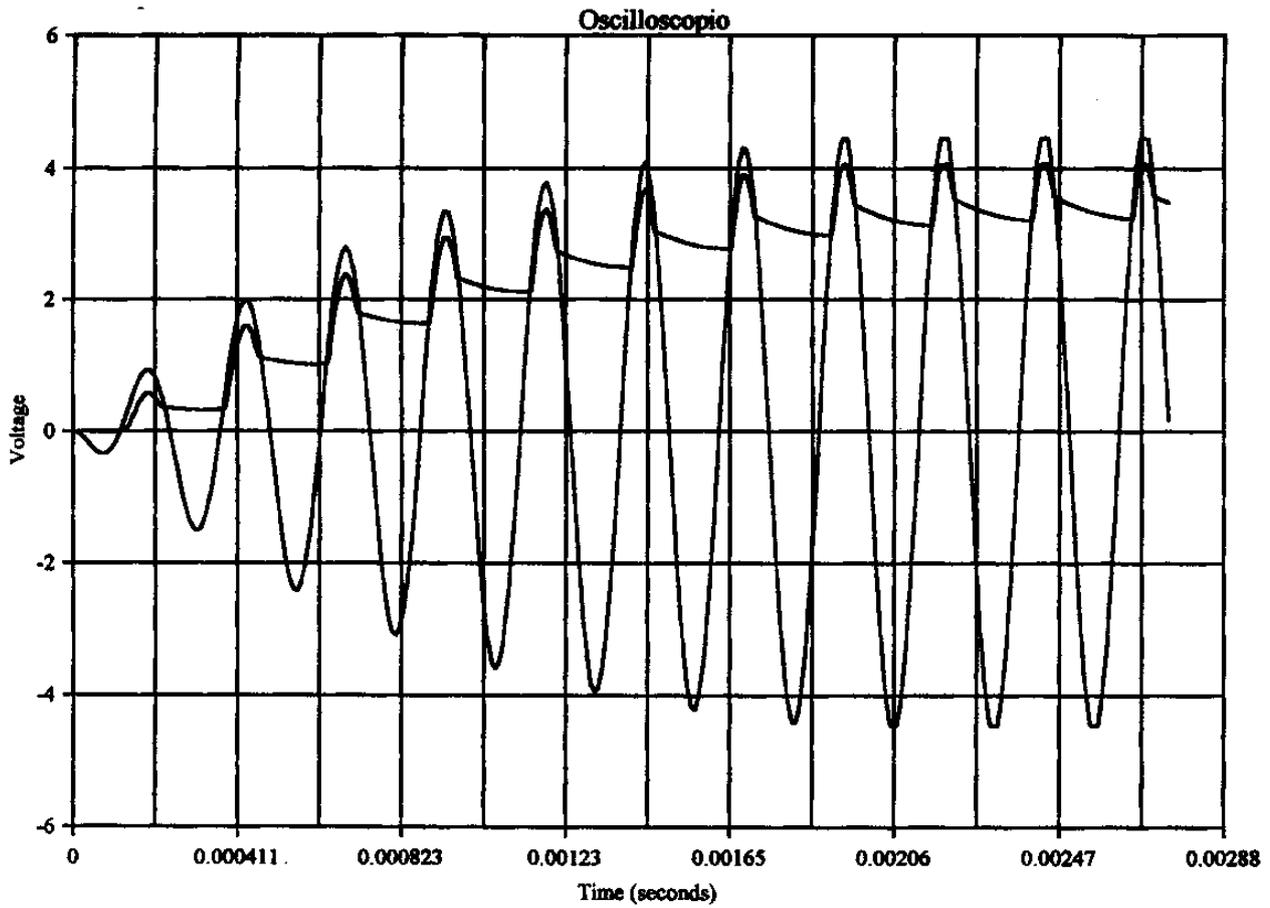
Description	Quantity	Reference ID
5-Terminal Opamp, TL081	2	AR1,AR4
AC Voltage Source, 4mV, 4kHz, 0Deg	1	V12
Battery, 9V	4	V9,V8,V5,V1
Capacitor, 47nF	2	C1,C0
Capacitor, 2nF	1	C3
Capacitor, 100nF	1	C2
Connector	13	CONN32,CONN22,CONN20,CONN18,CONN24,COI
Diode, DG1	1	D1
Ground	10	0,0,0,0,0,0,0,0,0
Node	14	18,6,9,14,20,19,12,11,10,7,4,3,2,1
Potentiometer, [R], 20k Ω , 50%, 1%	1	R13
Resistor, 20k Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	1	R9
Resistor, 847 Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	2	R2,R3
Resistor, 1M Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	1	R15
Resistor, 1.6k Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	1	R14
Resistor, 1k Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	1	R8
Resistor, 7.41k Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	1	R0
Resistor, 3.9k Ω , 0 Ω / $^{\circ}$ C, 0 Ω / $^{\circ}$ C ²	1	R1

5-Terminal Opamp Model "TL081" Library "opa77-tl"

VCC	Positive power supply	18	V
VEE	Negative power supply	-18	V
CC	Compensation capacitance	3e-11	F
A	Open-loop gain	1e+05	
RI	Input resistance	1e+12	Ω
RO	Output resistance	75	Ω
VOS	Input offset voltage	0.005	V
IOS	Input offset current	2.5e-11	A
IBS	Input bias current	5e-11	A
VSW+	Positive voltage swing	13.5	V
VSW-	Negative voltage swing	-13.5	V
CMRR	Common mode rejection ratio	100	dB
ISC	Output short circuit current	0.03	A
SR	Slew rate	13	V/us
FU	Unity-gain bandwidth	4e+06	Hz
\uparrow M	Phase margin	50	Deg

Diode Model "DG1" Library "general2"

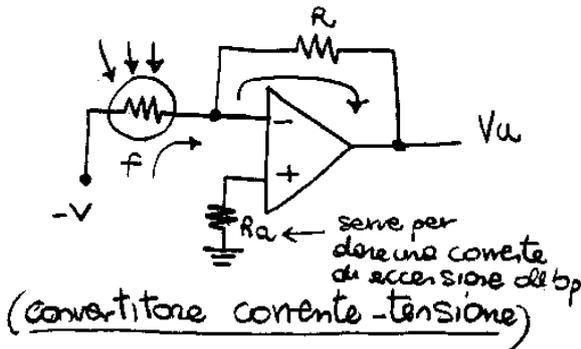
IS	Saturation current	1.43e-08	A
RS	Ohmic resistance	0.0741	Ω
CJO	Zero-bias junction capacitance	3.76e-11	F
VJ	Junction potential	0.391	V
TT	Transit time	4.93e-05	s
M	Grading coefficient	0.354	
BV	Reverse breakdown voltage	122	V
N	Emission coefficient	2	
EG	Activation energy	1.11	eV
XTI	Temperature exponent for effect on IS	3	
KF	Flicker noise coefficient	0	
AF	Flicker noise exponent	1	
FC	Coefficient for forward-bias depletion capacitance form	0.5	
IBV	Current at reverse breakdown voltage	0.00863	A
TNOM	Parameter measurement temperature	27	$^{\circ}$ C



• fig. Il segnale d'uscita è quasi continuo, tra 0 e 5V,
proporzionale all'ampiezza della sinusoide

5.3

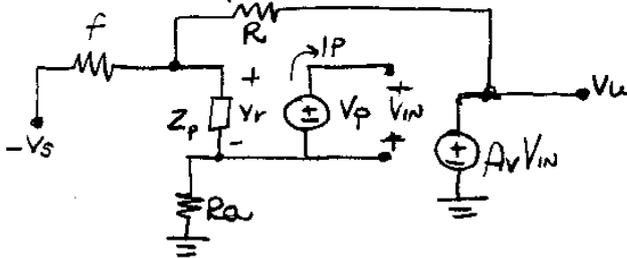
• Dimensionamento del circuito di luce



• sotto l'ipotesi di corto circuito virtuale:
 $V^+ = V^- = 0$
 $I^+ = I^- = 0 \Rightarrow -\frac{V}{f} = -\frac{V_u}{R}$

$V_u = V \frac{R}{f}$

dal teo di scomposizione



$e = \frac{I_P}{V_S} \Big|_{V_P=0} = 0; \rightarrow Z_p = Z_n = \infty$

$f = \frac{V_u}{V_S} \Big|_{V_P=0} = 0$

$A = \frac{V_u}{V_P} \Big|_{V_S=\infty} = -\frac{A_0}{1+jf/f_p}$

quindi $|BA| = \frac{A_0 f}{R+f} \cdot \frac{1}{1+jf/f_p}$

$\beta = \frac{V_r}{V_u} \Big|_{V_S=0} = \frac{f}{R+f}$

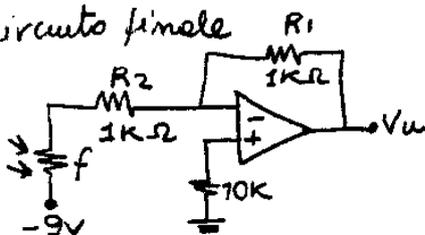
$f_H = \frac{A_0}{1 + R/f} \geq 40 \text{ kHz} \rightarrow R/f \leq 1,5$

• R non può essere + piccola di 1kΩ, altrimenti si dissipa troppa corrente (imped. d'ingresso piccola)

• f varia da 100Ω a 2kΩ allora basta mettere in altro res R2 in serie ad f

• di m. ±9V stilo

• circuito finale



$V_u = \frac{R_1}{R_2 + f} V_{cc}$; voglio una dinamica di 5V

$\Delta V_u = V_{cc} \cdot R_1 \left[\frac{1}{R_2 + f_{min}} - \frac{1}{R_2 + f_{max}} \right] = 5V$

$\rightarrow (R_2 + f_{min})(R_2 + f_{max}) = \frac{9 \cdot 1k}{5} \cdot f_{max} - f_{min}$

sost.

$V_{u, min} = \frac{R_1}{R_2 + f_{max}} \cdot V_{cc} \approx 3V$

$V_{u, max} = \frac{R_1}{R_2 + f_{min}} \cdot V_{cc} \approx 8V$

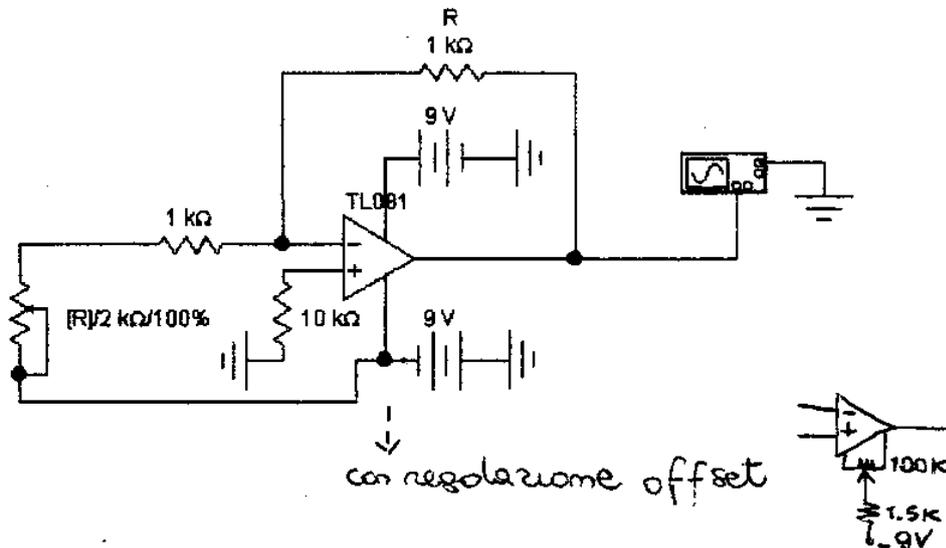
$R_2^2 + R_2(f_{max} + f_{min}) + f_{min}f_{max} - 3,42M$

$R_2 = 1.03k\Omega$

$f_{min} = 100\Omega$
 $f_{max} = 2000\Omega$

scolare offset di 3V

• SIMULAZIONE DEL CIRCUITO DI LOCE

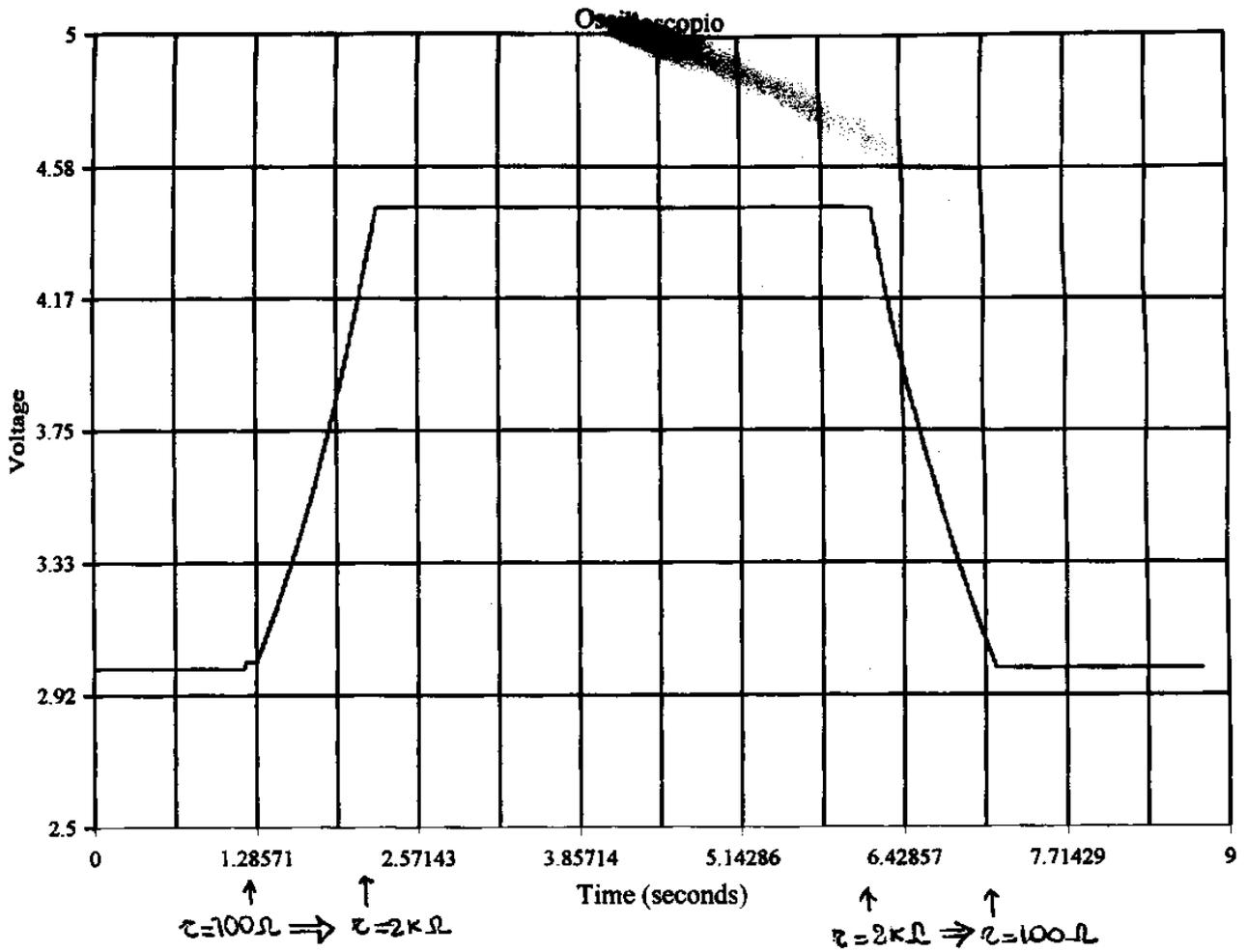


Parts list

Description	Quantity	Reference ID
5-Terminal Opamp, TL081	1	AR0
Battery, 9V	2	V2,V1
Connector	5	CONN9,CONN8,CONN5,CONN4,CONN6
Ground	5	0,0,0,0,0
Node	6	5,25,24,23,22,21
Potentiometer, [R], 2kΩ, 100%, 1%	1	R2
Resistor, 10kΩ, 0Ω/°C, 0Ω/°C ²	1	R1
Resistor, 1kΩ, 0Ω/°C, 0Ω/°C ²	2	R3,R0

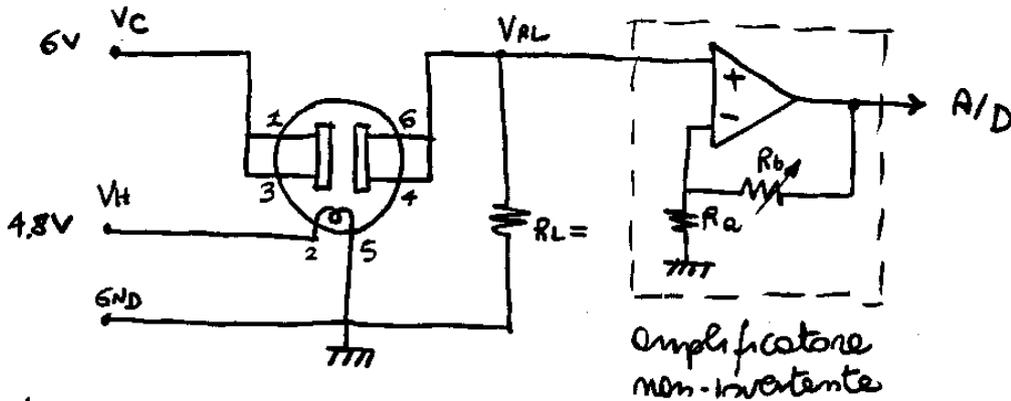
5-Terminal Opamp Model "TL081" Library "opa77-tl"

VCC	Positive power supply	18	V
VEE	Negative power supply	-18	V
CC	Compensation capacitance	3e-11	F
A	Open-loop gain	1e+05	
RI	Input resistance	1e+12	Ω
RO	Output resistance	75	Ω
VOS	Input offset voltage	0.005	V
IOS	Input offset current	2.5e-11	A
IBS	Input bias current	5e-11	A
VSW+	Positive voltage swing	13.5	V
VSW-	Negative voltage swing	-13.5	V
CMRR	Common mode rejection ratio	100	dB
ISC	Output short circuit current	0.03	A
SR	Slew rate	13	V/us
FU	Unity-gain bandwidth	4e+06	Hz
PM	Phase margin	50	Deg



• la differenza con i conti sta nelle R_{out}

Dimensionamento del circuito del GAS



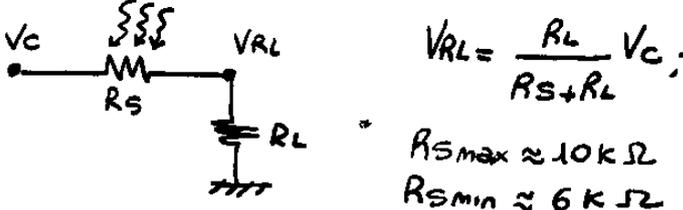
(ved. data sheet)

- V_H è la tensione di riscaldamento, che alimenta una resistenza dissipativa

$$I_H = \frac{P_H}{V_H} = \frac{925 \text{ mW}}{4.8 \text{ V}} = 193 \text{ mA} \approx 200 \text{ mA}$$

consuma come un motorino!
quindi va collegata alla pila da 6V, prendendo dalla V_{CC} dei motorini

- Modello è come una res. variabile con la concentrazione di alcool



$$V_{RL} = \frac{R_L}{R_S + R_L} V_C$$

$R_{Smax} \approx 10 \text{ k}\Omega$
 $R_{Smin} \approx 6 \text{ k}\Omega$

Vogliamo trovare il valore di R_L che ci dia la max dinamica

$$\Delta V_{RL} = V_C R_L \left(\frac{1}{R_L + R_{Smin}} - \frac{1}{R_L + R_{Smax}} \right)$$

$\approx R_L \rightarrow 0 \quad \Delta V_{RL} = 0$
 $\approx R_L \rightarrow \infty \quad \Delta V_{RL} = 0$

$$f(x) = \frac{x}{x+d} - \frac{x}{x+\beta} \Rightarrow f'(x) = \frac{x+d-x}{(x+d)^2} - \frac{x+\beta-x}{(x+\beta)^2} = \frac{(x+\beta)^2 d - (x+d)^2 \beta}{(x+d)^2 (x+\beta)^2} = 0$$

$$\text{se } dx^2 + d\beta x + d^2\beta - \beta x^2 - d^2\beta - 2d\beta x = 0$$

$$(d-\beta)x^2 + d\beta(d-\beta) = 0 \Leftrightarrow x^2 = d\beta \Rightarrow x = \sqrt{d\beta}$$

$$(d-\beta)(x^2 - d\beta) = 0$$

↑
negativo

quindi $R_L = \sqrt{R_{Smin} R_{Smax}} = \sqrt{60 \text{ k}\Omega} \approx 7.7 \text{ k}\Omega$

Val. commerciale
7.5 kΩ

$V_{RLmin} = 2.6 \text{ V}$
 $V_{RLmax} = 3.3 \text{ V}$
 $\Delta V = 0.7 \text{ V}$

se voglio $\Delta V = 5 \text{ V}$ devo
amplificatore di 6.6 $\rightarrow \frac{R_b}{R_a} = 5.6$

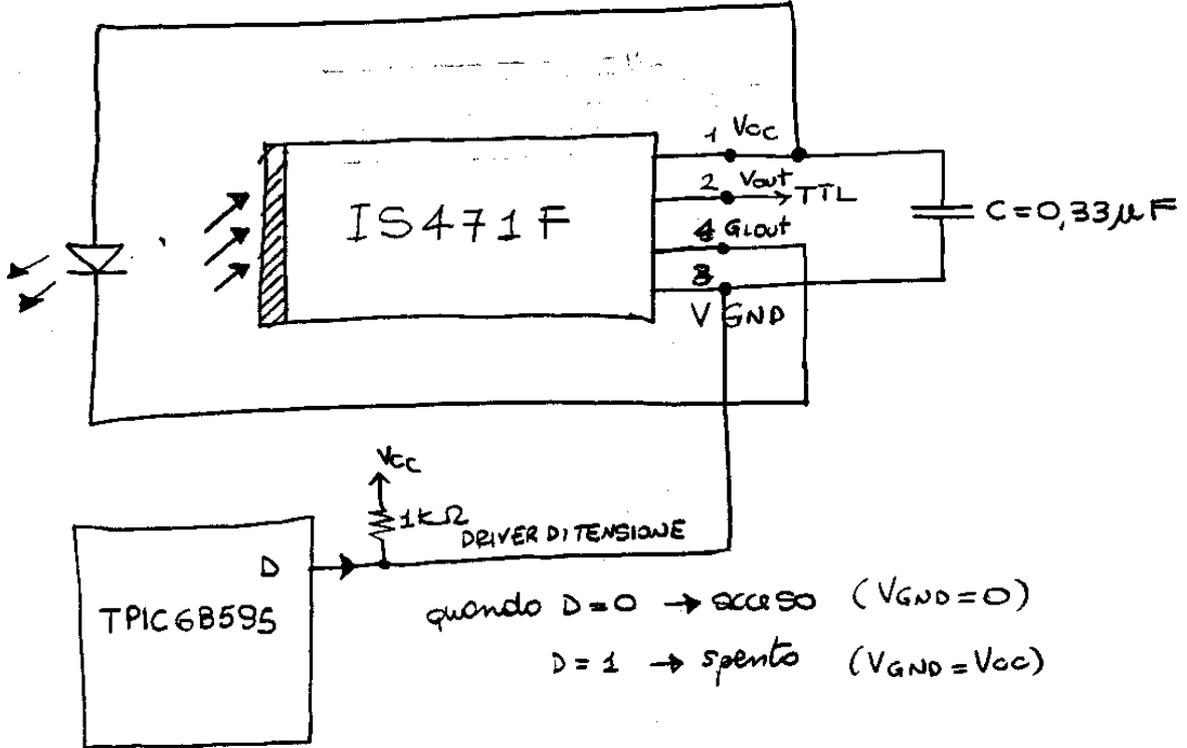
punto di max,
perché se $x < \sqrt{d\beta}$
 $f'(x) > 0$, se $x > \sqrt{d\beta}$
 $f'(x) < 0$



$\rightarrow R_a = 1 \text{ k}\Omega$
 $R_b = 2 \text{ k}\Omega + 3.6 \text{ k}\Omega$
 potenziometro

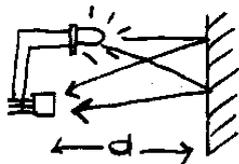
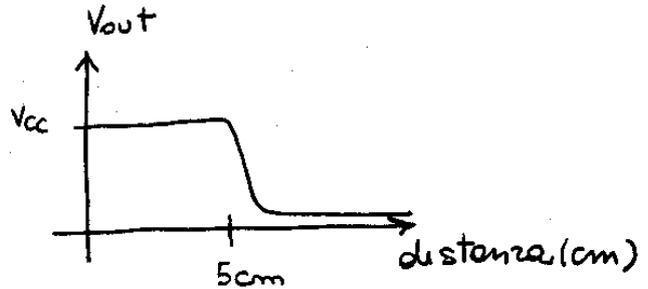
6.5 IL CIRCUITO DEGLI INFRAROSSI

A) piedinatura e montaggio



B) caratteristiche

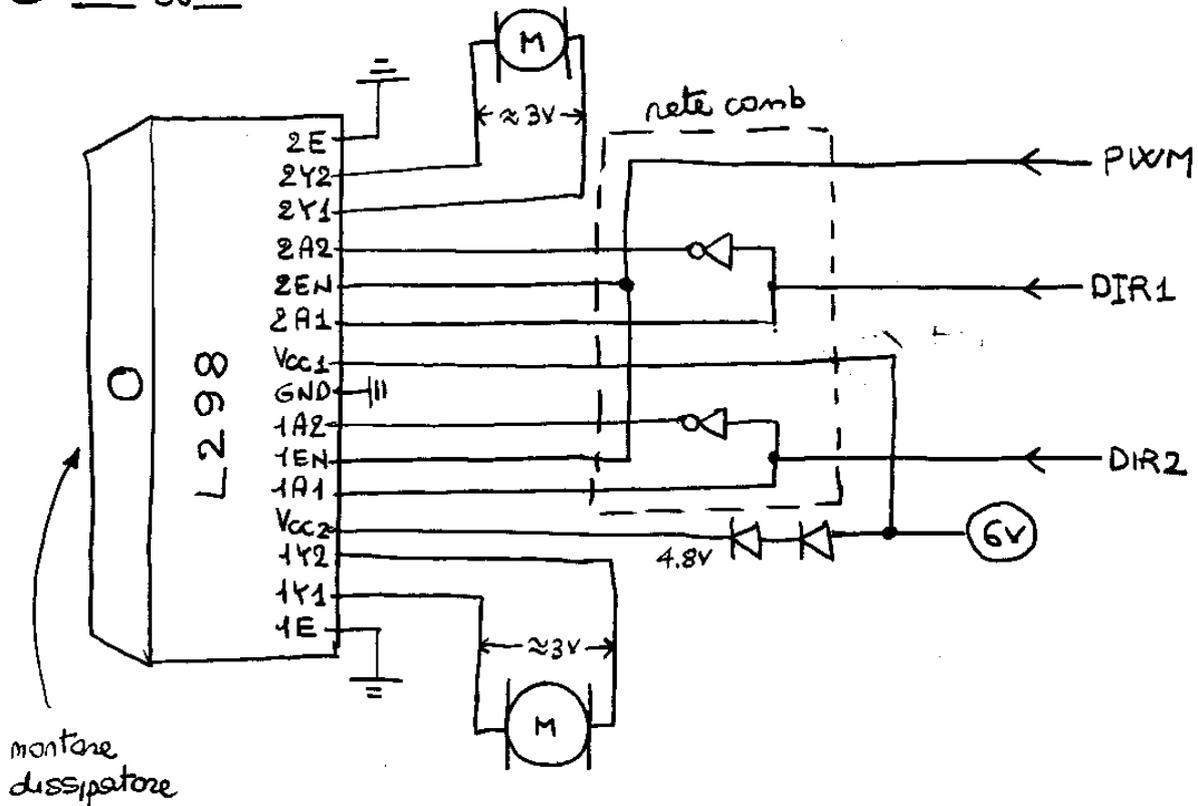
- $V_{cc} = 5V$ (da 4.5 a 16V)
- ondulamento dell'usata



- il fascio è molto direzionato (tipo telecomandi da TV)

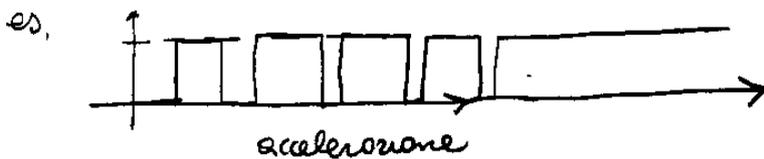
5.6 L'integrato per commutare i motori

(A) montaggio



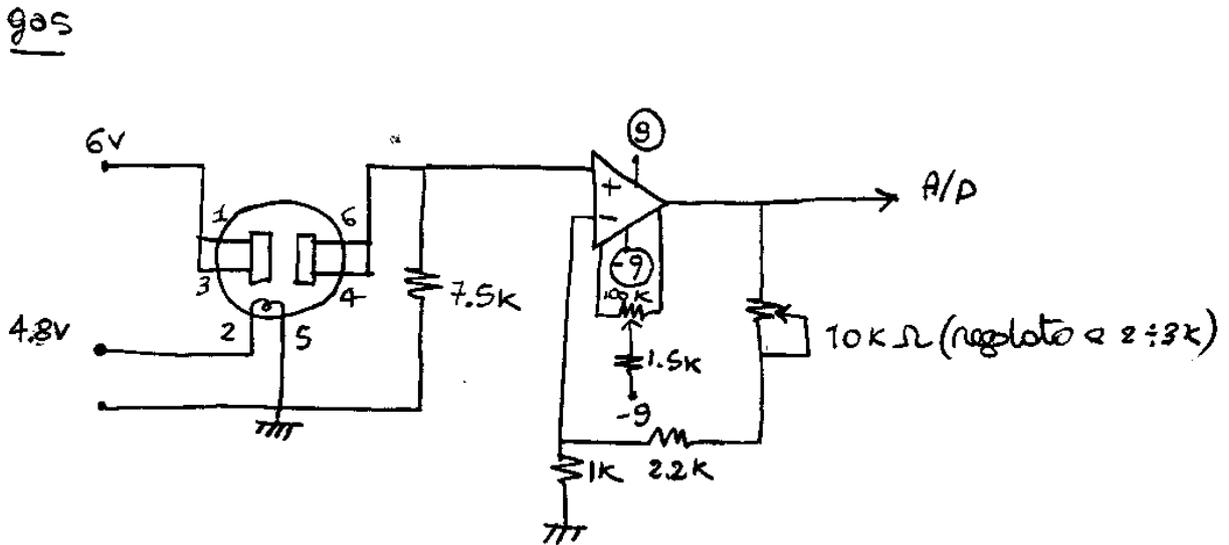
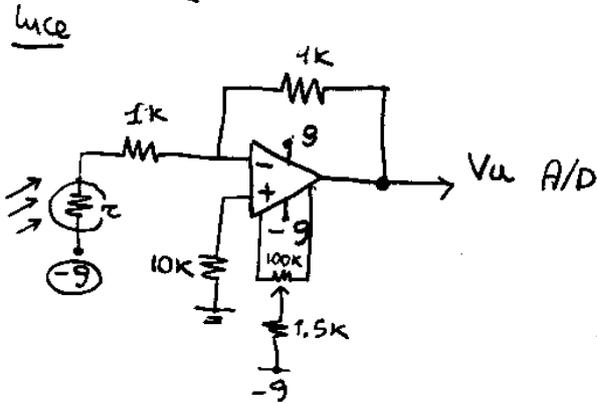
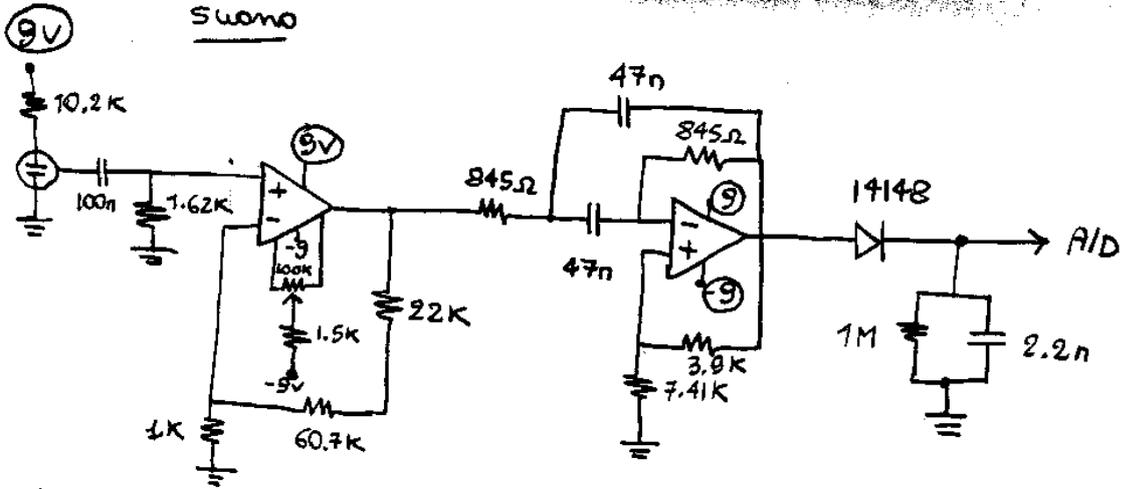
(B) caratteristiche

- DIR1 e DIR2 controllano il verso di rotazione di *ciascun* motore, \emptyset un verso, 1 un altro
- PWM è l'abilitazione, che può essere in PWM, cioè a larghezza d'impulso, per accelerare e decelerare.



- Vcc1 dim. di potenza 6V
- Vcc2 dim. di logica 5-4.8V.

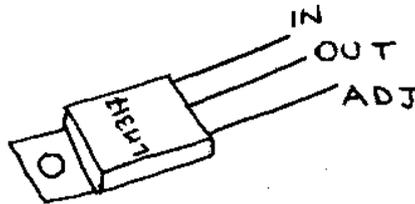
6.7 circuitistica dei sensori e dei commerciali dei componenti



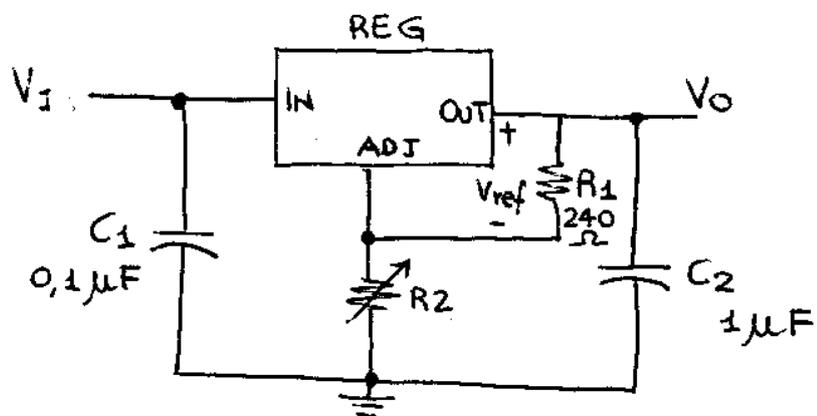
• REGOLATORE DI TENSIONE POSITIVA REGOLABILE

LM317

Ⓐ predefinatura



Ⓑ montaggio



Ⓒ Caratteristiche di funzionamento

$$V_O = V_{ref} \left[1 + \frac{R_2}{R_1} \right]$$

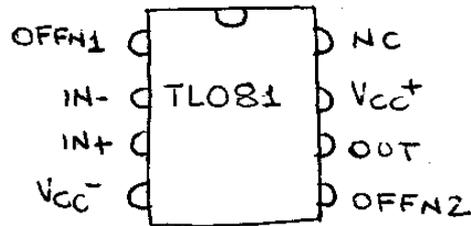
↑
tip. 1.25V

- variando il potenziometro R_2 , si può avere in uscita una tensione da 1,2V fino a 2-3 V al di sotto di V_I
- $V_{I\text{MIN}} = 4\text{V}$ $V_{I\text{MAX}} = 40\text{V}$
 $V_{O\text{MIN}} = 1,2\text{V}$ $V_{O\text{MAX}} = 37\text{V}$
- LM317T può erogare 1,5A max

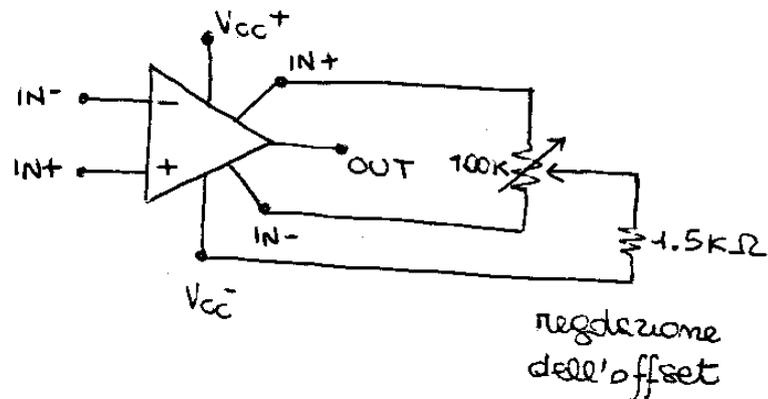
• AMPLIFICATORE OPERAZIONALE TLO81

con possibilità di azzeramento dell'offset

Ⓐ piedinatura



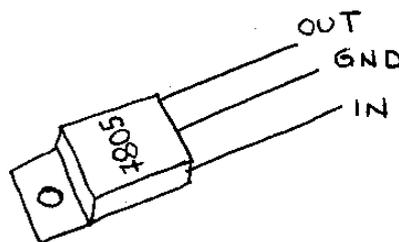
Ⓑ montaggio



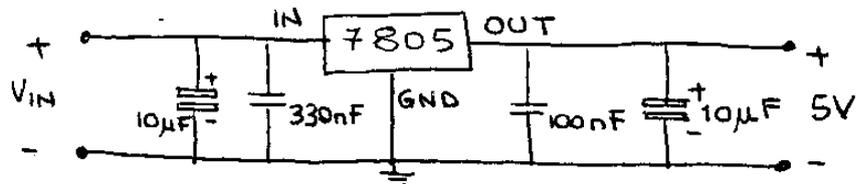
• REGOLATORE DI TENSIONE FISSA POSITIVA

TL 7805

Ⓐ piedinatura



Ⓑ montaggio

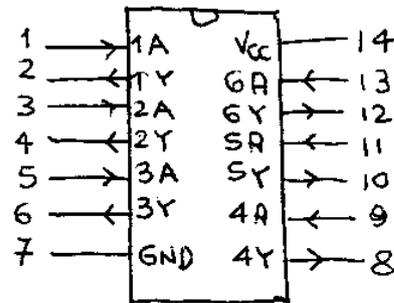


Ⓒ caratteristiche di funzionamento

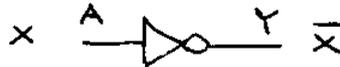
- presenta una tensione in uscita pari a 5V, con buon ripple, 1A.
- $V_{IN\ MAX} = 35V$, V_{IN} deve essere 2-3V superiore a 5V.

• GRUPPO LOGICO NOT 7404

(A) pedinatura



(B) montaggio

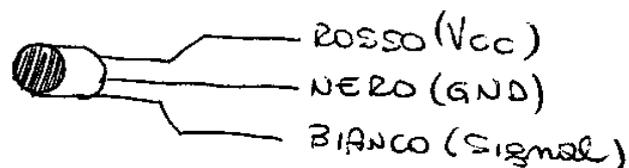


(C) caratteristiche di funzionamento

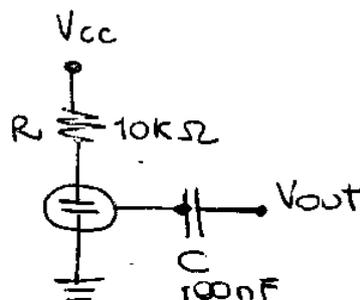
- alimentazione V_{cc} : 4.5-5V
- ogni coppia di piedini $nA-nY$ rappresenta un NOT

• CAPSULA MICROFONICA PREAMPLIFICATA

(A) pedinatura



(B) montaggio



(C) caratteristiche

- V_{cc} da 4.5V a 9V.
- C serve ad eliminare la componente continua.
- in uscita escono tensioni dei millivolt

5.8 Stima delle correnti totali assorbite, e durata delle batterie.

Gli operazionali devono essere alimentati con tensioni superiori a quelle dei segnali di ingresso, perché in prossimità degli estremi di alimentazione, i transistori vanno in zona di saturazione, per cui cessano di essere lineari. Inoltre devono essere alimentati con tensioni opposte, rispetto al riferimento, +Vcc e -Vcc, in quanto, soprattutto quelli del circuito di suono, elaborano delle tensioni prive di continua, quindi certamente negative in alcuni istanti (si pensi alla tensione sinusoidale a 4Khz). D'altra parte, gli operazionali di segnale consumano, con i loro carichi, decimi di mA di corrente, per cui non è necessaria una batteria di grosse capacità.

A differenza, abbiamo dei dispositivi di potenza, che dissipano centinaia di mA di corrente, per i quali è necessaria una batteria di qualche Amperora: motorini, resistenza di riscaldamento del gas.

In base a ciò, si utilizzano due batterie non ricaricabili, alcaline, da 9v per tutti gli operazionali, ed una batteria al piombo, ricaricabile, per tutto il resto, da 6v ; oppure solo una da 12v per tutto.

Le batterie sono però gli oggetti più ingombranti e pesanti di un sistema elettronico portatile (si pensi ad un cellulare, ad un laptop, ad una telecamera); mentre l'elettronica è riuscita a conseguire una miniaturizzazione sempre più spinta, le batterie non sono cambiate tanto in peso e dimensioni.

Una batteria al piombo da 6v pesa 350g ed è grande $9,5 \times 5 \times 2,5 \text{cm}^3$; una da 12v è più grande del 70% , quindi è impossibile da inserire sul robot. Una pila da 9v pesa 50g. Il robot, senza batterie pesa 1000g cioè il doppio del peso complessivo delle batterie (350+50+50g; si veda il calcolo del baricentro a pag. 14). Per regolare una tensione ad una soglia inferiore a quella d'alimentazione, ci sono tanti metodi. Le soluzioni che si adottano per stabilizzare un segnale, in genere non vanno bene per stabilizzare una tensione di alimentazione. Ad esempio, il partitore resistivo, ha il problema di consumare troppa corrente, anche a circuito aperto, e di non essere ben stabilizzato; anche un diodo Zener in parallelo, ha lo svantaggio di dissipare troppa corrente, e di non essere così stabile come negli zener di segnale, per la sua resistenza interna; oltre a consumare grossa potenza a circuito spento; infine, mettere tanti diodi in serie (ognuno dei quali abbassa la tensione di 0.7v) non dà la stabilità necessaria, per correnti di alimentazione. I dispositivi digitali possono ricevere tensioni in un grande range, ma devono essere ben stabili.

Esistono degli integrati (regolatori serie) che funzionano bene, anche se hanno un basso rendimento (50%); mentre la soluzione di regolatori switching è sofisticata per il nostro sistema.

I regolatori serie possono essere a tensione fissa (come il 7805) o regolabile tramite un potenziometro (LM317); però hanno il problema che la tensione che possono dare è al massimo 2-3volt inferiore a quella di alimentazione. Nei regolatori a tensione fissa, si riesce anche ad ottenere 1.5-2 volt inferiore. Esistono regolatori serie 'di precisione', che riescono a regolare una tensione ad un solo volt più in basso dell'alimentazione, e costano ovviamente di più.

Quindi, adoperando le 2 pile da 9v separatamente per gli operazionali, dobbiamo ottenere, dalla pila da 6v, 5v per tutti gli altri integrati (vedere datasheet, o circuiti relativi), 1.5v per il motorino della ventola, e 4.8v per l'integrato dei motori, come alimentazione di potenza; oltre ad avere, ovviamente 6v disponibili direttamente. Considerando che la pila da 6v un po' sovraccaricata, arriva a 7v (e li mantiene per 10 minuti di funzionamento) , possiamo usare un 7805 per avere 5v, ed un LM317 per avere 1.5v, mentre per 4.8v usiamo due diodi (sarà l'integrato a fornire la tensione ai motorini). Anche con 6v il 7805 produce almeno 4.5v, sufficiente ai circuiti digitali.

Le pile da 9v forniscono un centinaio di mAh ciascuno; i 12 operazionali con i loro circuiti consumeranno in media 1 mA a testa, quindi in totale una decina di mA, per cui le pile dureranno una ventina di ore. La pila da 6v, fornisce 1.2Ah; i consumi rilevanti sono dei motorini (200mA a testa), il motorino della ventola (100mA), la resistenza di riscaldamento del gas (200mA); gli integrati, in tecnologia C-MOS, consumano qualche mA: convertitore A/D (2.5 mA), interfaccia (0.3 mA), PIC16C84 (4.5mA); poi, i rivelatori IR(7 mA x 4), led IR(20mA x 4), i led colorati si accendono ogni tanto, perciò non dissipano; integrato motori (80mA); microfoni (2-3 mA); gas (2-3mA) Quindi, in tutto 900mA, ossia la pila durerà circa 1 ora; più che sufficiente.

6. VERIFICA DEI SINGOLI MODULI, REALIZZAZIONE DELLE SCHEDE ELETTRONICHE, MONTAGGIO DELLA STRUTTURA MECCANICA (di M.Cimino, E.Campanelli, A.Campana, P.Imbesi, L.Piccinini).

6.1 Test su basetta mille fori

Il processo di saldatura su piste di rame è la fase forse più delicata e con più varietà di tutte le altre. Tutto ciò che viene simulato o calcolato, richiede azioni complesse, quindi composte da sub-problemi semplici articolati nel tempo, ma si possono correggere e possono essere analizzate in più tempi. Mentre la saldatura richiede una certa varietà, ossia eseguire in contemporanea sub-problemi nello spazio (padronanza dei movimenti del saldatore, dello stagno, del tempo di contatto con le piste), e deve essere veloce; ma soprattutto è irreversibile, a meno di rompere i quarzi, i regolatori, o di aumentare le capacità e le resistenze di contatto.

Se volessimo confrontare tra di loro la complessità computazionale delle varie fasi, allora di certo la saldatura sarebbe quella più complessa; non a caso, è la cosa più difficile da automatizzare (un robot per le saldature costa centinaia di milioni); a noi sembra la più facile, perché possediamo già numerosi meccanismi automatici di controllo dei movimenti, che avvengono incoscientemente, senza che ce ne accorgiamo; ma se ammettiamo che una cosa è tanto più difficile, quanto più è difficile farla realizzare ad una macchina, allora saldare una resistenza è molto più difficile che progettare un filtro di suono.

D'altra parte, la qualità delle saldature è un fattore che limita fortemente le prestazioni di un circuito rispetto ad un altro, in quanto non si può correggere facilmente, ed è un processo non automatizzato (almeno nella creazione di prototipi); mentre tutti gli altri processi (simulazioni, fotoincisione, foratura, creazione dei circuiti integrati) vengono realizzate con strumenti (teorici o tecnologici) ad alto contenuto informativo, che facilitano enormemente la creazione e lo sviluppo di sistemi, perché l'operatore umano deve scegliere solo tra alternative limitate. Anche lo sbroglio dei circuiti, ad esempio, è il problema più complesso dopo quello delle saldature (i CAD non riescono a farlo bene); infatti ancora oggi, nelle aziende, fa parte di quel processo in cui il valore aggiunto del progettista conta molto. Ma, a differenza delle saldature, è possibile realizzarlo mediante CAM, quindi si può eseguirlo a piccoli passi, con tutto il tempo necessario, e poi realizzarlo.

I contatti con le piste sono la principale causa di malfunzionamento, dei circuiti, soprattutto alle alte frequenze, dove si fanno sentire le capacità e le resistenze parassite; poi vengono le ossidazioni, ridotte in parte da alcune sostanze spray.

Allora, prima di creare una pista stampata, è opportuno effettuare dei test su basette millefori, che sono costituite da una matrice di fori, con colonne tutte cortocircuitate, per cui è possibile montare direttamente i componenti acquistati.

In tal modo, è stato possibile provare il circuito del suono, mediante onde di vario genere, a varie frequenze, con un generatore software di suoni, e verificare che solo in un intorno dei 4Khz la tensione in uscita variava sensibilmente. Si è verificato inoltre che i microfoni sono molto più direzionali del previsto, e che già a 50-60 cm il suono cala fortemente.

Il circuito di luce è stato provato in varie condizioni di illuminamento ambiente, ed in tal senso è stato verificato che la luce ambientale da' troppo disturbo; è stato quindi indispensabile mettere i tubicini. Anche il circuito del gas risulta funzionare secondo i calcoli, e la ventola abbastanza efficace. Il circuito dei motori ha dato qualche problema, perché l'integrato è adatto per motori alimentati a tensioni di almeno 5V, ma sostanzialmente è utilizzabile. L'eccessiva temperatura raggiunta ha reso indispensabile un dissipatore. Anche gli infrarossi risultano funzionare bene, ma la loro estrema sensibilità a variazioni dell'alimentazione, renderà probabilmente difficile la loro accensione e spegnimento a vari istanti, a meno di non aspettare qualche millisecondo.

Durante i vari test, sono stati tarati i potenziometri di amplificazione e di offset.

6.2 Realizzazione del circuito stampato mediante il CAM e montaggio delle schede

Alcuni CAD di simulazione, come PROTEL, possiedono anche funzioni per creare la cosiddetta TRACCIA RAME del circuito, ossia il disegno dal lato rame della basetta. Nel nostro caso non è stato possibile trasferire lo schematico dal programma di simulazione (ELECTRONIC WORKBENCH E SPICE) a PROTEL, per cui si è dovuto ricostruire lo schematico.

Specificando per ogni componente le dimensioni, il diametro e la distanza dei fori, o per gli integrati le dimensioni fisiche dello zoccolino (in realtà c'è uno standard, per cui basta specificare il numero dei piedini), si passa automaticamente dalla descrizione schematica, alla creazione di una 'net list', che permette di avere il PCB, ossia il disegno meccanico dei componenti con le piste.

In questa seconda parte, si utilizza un CAM (Computer Aided Design), perché il file creato genererà un master che permetterà di incidere sulle basette di rame con fotoresist.

La procedura più complessa computazionalmente è quella dello sbroglio, in cui il computer tenta di mettere le piste in maniera tale da creare pochi incroci (che si risolvono poi in ponticelli dal lato componenti), e di non superare le dimensioni componenti.

Se facciamo partire lo sbroglio dell'intero circuito, probabilmente ci vorranno anni perché si fermi. Allora si utilizza, anche in ambienti professionali, un sistema misto. Si divide il circuito in parti logicamente distinte, e si eseguono le procedure automatiche su piccoli pezzi, poi montati 'a mano', con il mouse.

Nel nostro caso, abbiamo eseguito lo sbroglio di un solo circuito di suono, e poi abbiamo montato graficamente la scheda suono, costruendo con il mouse le piste di collegamento; idem per la luce, dove è stato possibile inserire anche il circuito del gas. Poi, in una terza scheda 'madre', sono stati collegati il circuito di programmazione (già sbrogliato), per permettere di programmare il micro direttamente sul robot, dalla porta seriale, e gli altri due integrati; il circuito di programmazione, richiede i piedini /MCLR, RB7, RB6 completamente scollegati da tutto il testo, per cui ci sono delle fascette per scollegare dal micro tali i piedini; per raggiungere la tensione di 15v necessaria a programmare, viene utilizzato un sistema a 'pompa di carica', mediante un condensatore ed uno zener; ciò non ha permesso di inserire un circuito di reset esterno con condensatore, ma un semplice tasto di cortocircuito tra /MCLR e massa, che va' pigiato 2-3 volte all'accensione, per dare un buon reset.

Infine è stata sbrogliata, in maniera semiautomatica, la scheda con l'integrato de motori e le varie alimentazioni.

Abbiamo quindi ottenuto (pag. seguenti) i MASTER, cioè la copia della traccia rame su acetato. Per quanto riguarda quest'ultimo aspetto, c'è da dire che una sola stampa non è abbastanza nera, per cui è necessario farne due; vengono realizzate specularmente, così da essere sovrapposte dal lato 'appiccicoso' dell'acetato (quello dove si attacca 'inchiostro), riducendo così il degrado, dovuto al calore della lampada ad ultravioletti, che alle lunghe scioglie l'inchiostro stesso. Le stampanti laser o le fotocopiatrici NON vanno bene, poiché scaldano il foglio di acetato, deformando un po' il disegno; ragion per cui le due copie non combaciano mai perfettamente con conseguente sfuocatura delle piste ramate. Sono ottime le stampanti a getto d'inchiostro, che stampano su acetato adatto, senza scaldare il foglio.

Una volta realizzato il master, si passa alla Fotoincisione. Abbiamo utilizzato basette di vetronite del tipo presensibilizzato (fotoresist) in formato 100x160. Togliendo la pellicola nera, abbiamo uno strato di materiale fotosensibile, resistente all'acido, poi uno strato di rame, ed infine la base di vetronite. Si appoggia il master sopra la basetta, e si mette un vetro per far aderire bene il master, ed evitare errori di parallasse. Le lampade ultravioletti (si possono usare anche le alogene da 150-300 watt) tolgono solo il fotoresist non coperto dalle tracce nere del master (in circa 5-6 minuti e ½ tenendo la lampada a ½ metro). Sulla basetta sensibilizzata non si vede ancora nulla. Successivamente si immerge la basetta in una soluzione di sviluppo composta da 13 grammi (tre cucchiaini da caffè) di soda caustica in un litro d'acqua fredda, e sciolta per bene senza scaglie.

Immersa la basetta, si fa oscillare la vaschetta per esercitare un'azione meccanica di erosione, fino a che non compare la traccia rame (5-10 minuti); dopodiché si lava la basetta. Si immerge poi in una soluzione di cloruro ferrico, che opera un'azione di erosione del rame intorno alle piste (il fotoresist protegge le piste). Il cloruro si vende già pronto liquido. Alla fine si pulisce la basetta con acqua, e poi con uno straccetto imbevuto di acetone o diluente al nitro, per togliere il fotoresist dalle piste.

Il cloruro ferrico e la soda caustica sono altamente tossiche.

Foratura; il normale diametro è 0.8 mm, mentre i trimmer, potenziometri, 1.2 mm. Sono necessari trapani a colonna, per avere un migliore controllo di precisione.

Dopo il montaggio dei componenti, è opportuno proteggere le piste dall'ossidazione mediante vernice spray, disponibile in commercio.

6.3 Composizione dei moduli, collegamenti elettrici tra schede, test di sistema finale

Le varie schede sono state collegate mediante fili con terminali a connettori, di vari colori, come indicato nelle prossime pagine; in tal modo sarà possibile smontare tutto e rimontare, in caso di problemi. I connettori a spina sono anche una parte delicata, che non di rado causa problemi di contatto, per cui è necessario eseguirli ad opera d'arte. Nelle prossime pagine saranno mostrate anche delle foto delle schede, e del montaggio finale.

Ad ogni problema elettrico, si procede ad esaminare, mediante un tester, il percorso del segnale attraverso i vari dispositivi, fino ad individuare il guasto, consultando le stampe dal lato componenti, mostrate alle pagine seguenti. Dopo aver testato tutte le piste ramate, ed i contatti a stagno, ed il funzionamento delle schede, queste in genere non danno più problemi, a meno che non subiscano continue sollecitazioni meccaniche (ad esempio, di connessione o disconnessione delle fascette). Dopodiché, la causa più frequente di guasto sono i contatti mobili (cavi).

Altro problema sono le batterie, che dopo circa quaranta minuti danno una carica di 5.5v; gli effetti si risentono soprattutto sui dispositivi di potenza 'a coppia': ad esempio, quando la corrente richiesta è superiore a quella disponibile, l'integrato dei motori non la distribuisce ad entrambi i motori, ma ad uno solo, mentre l'altro ruota a velocità inferiore, per cui il robot tende ad andare prevalentemente a destra. Tutti gli altri dispositivi, invece, continuano a funzionare bene, ma in maniera sempre più lenta. Ad ogni modo, nei 5 minuti di gara, le prestazioni non subiscono variazioni apprezzabili.

FIG. COLORAZIONI DEI FILI DI SEGNALE.

- **INFRAROSSI:**, (SIA SEGNALE TX, CHE RX)
 - SX GIALLO-BIANCO
 - DX BIANCO
 - CX VERDE
 - DD BLU-VERDE
- **MOTORI:**
 - SX BLU
 - DX GRIGIO
 - PWM GIALLO
- **LED:**
 - D4, LED ROSSO (GAS), GIALLO-GRIGIO
 - D5, LED VERDE (LUCE), BIANCO
 - D6, LED GIALLO (SUONO), VERDE-BIANCO
- **TRASDUTTORI:**
 - A0 MICROF. SX GIALLO
 - A1 MICROF. CX VERDE
 - A2 MICROF. DX BIANCO
 - A3 FOTOR. SX BLUVERDE
 - A4 FOTOR. SC MARRONE
 - A5 FOTOR. CX GRIGIO
 - A6 FOTOR. DC BLU-MARRONE
 - A7 FOTOR. DX GIALLO-MARRONE
 - A8 GAS VERDE-GIALLO

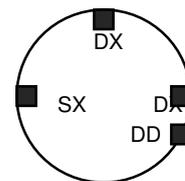
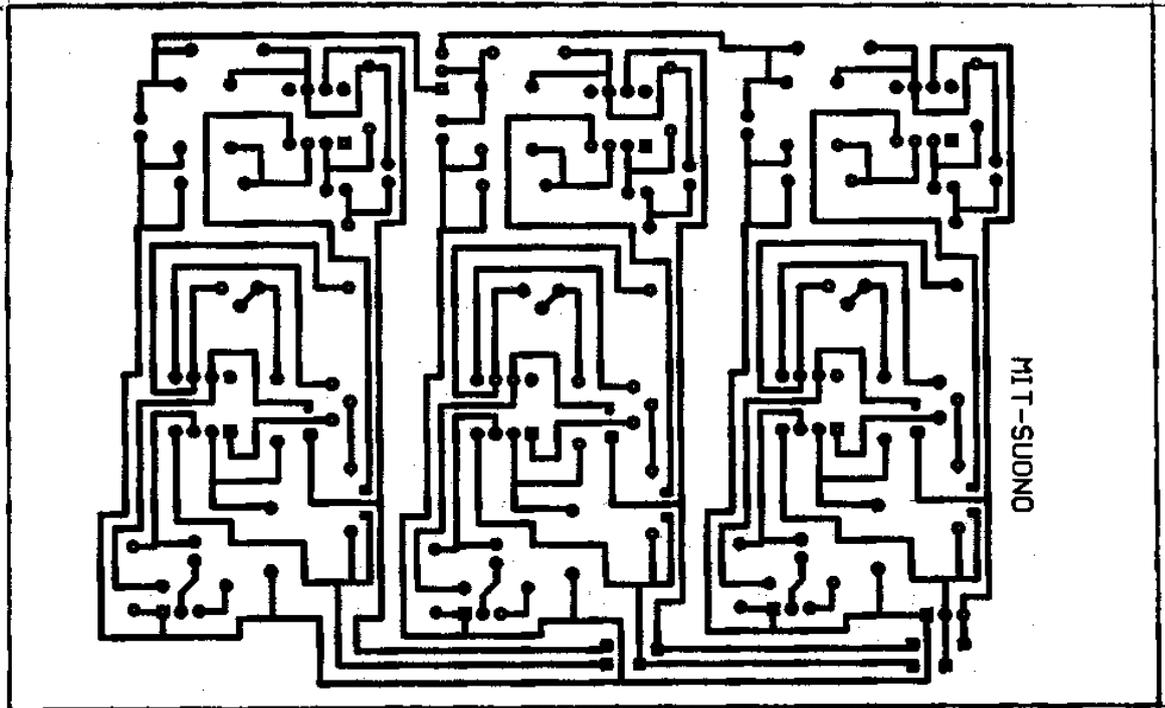


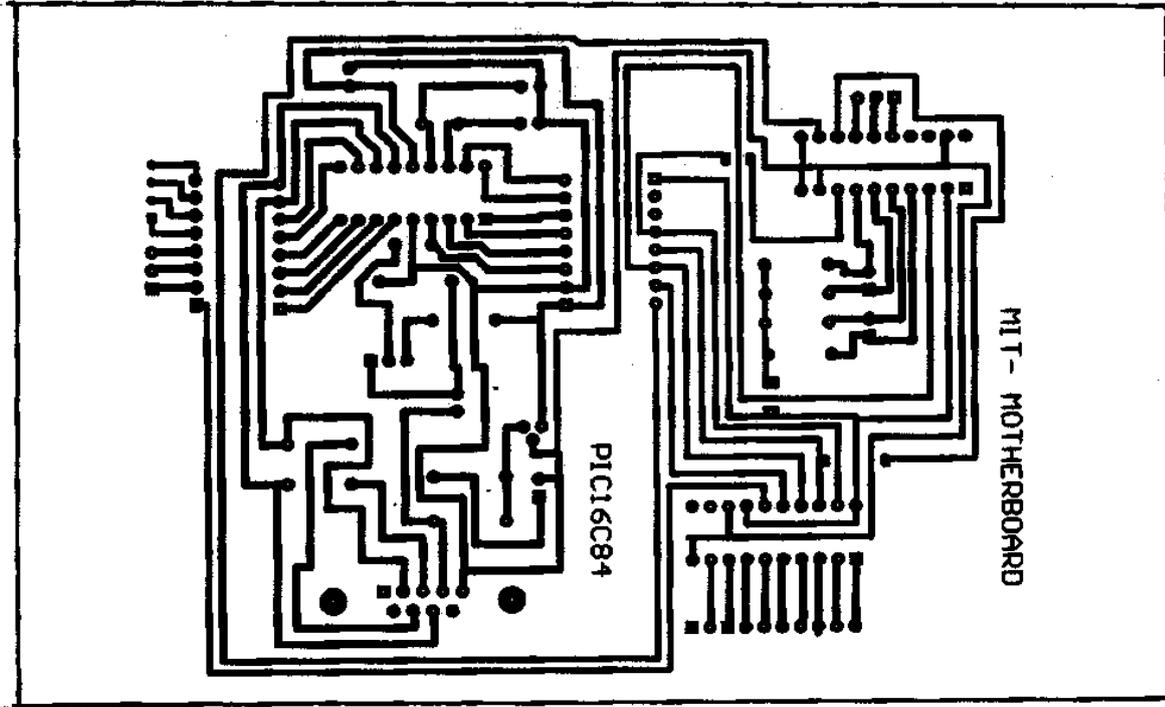
FIG. COLORAZIONI DEI FILI DI ALIMENTAZIONE.

- **GAS:** 6V BIANCO-BLU / 4.8V GIALLO / 0V BIANCO
- **MAINBOARD:** 6V GIALLO / 0V BIANCO
- **SCHEDA LUCE-GAS:** SEGNALE-GAS MARRONE-GIALLO / +9V BLU-VERDE / -9V MARRONE / 0V BIANCO
- **SCHEDA SUONO:** +9V BLU-VERDE / -9V MARRONE / 0V BIANCO
- **INFRAROSSI:** +5V GIALLO-BLU / 0V GIALLO-GRIGIO
- **LED:** 6V MARRONE-GRIGIO.

• MASTER PER CIRCUITO STRAPPATO ("TRACCIA RAME")

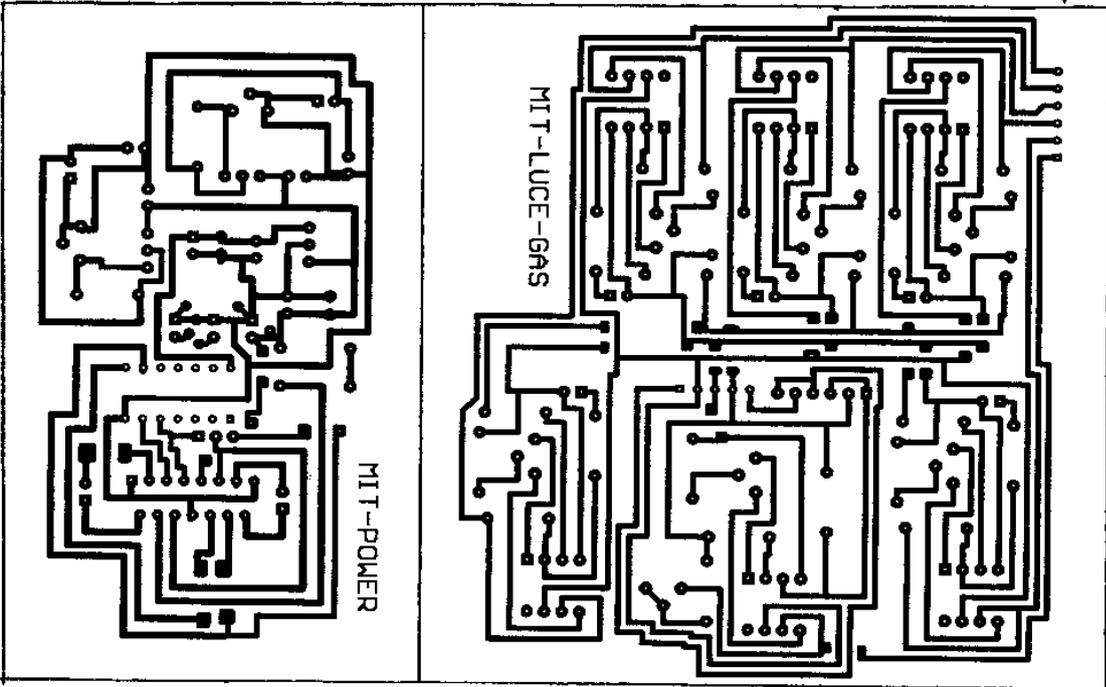


basette di
name con
forne sist
10x16 cm



• MASTER PER CIRCUITO STAMPATO ("TRACCIA RAME")

basetta di
rame con
fotore sist
10x16cm



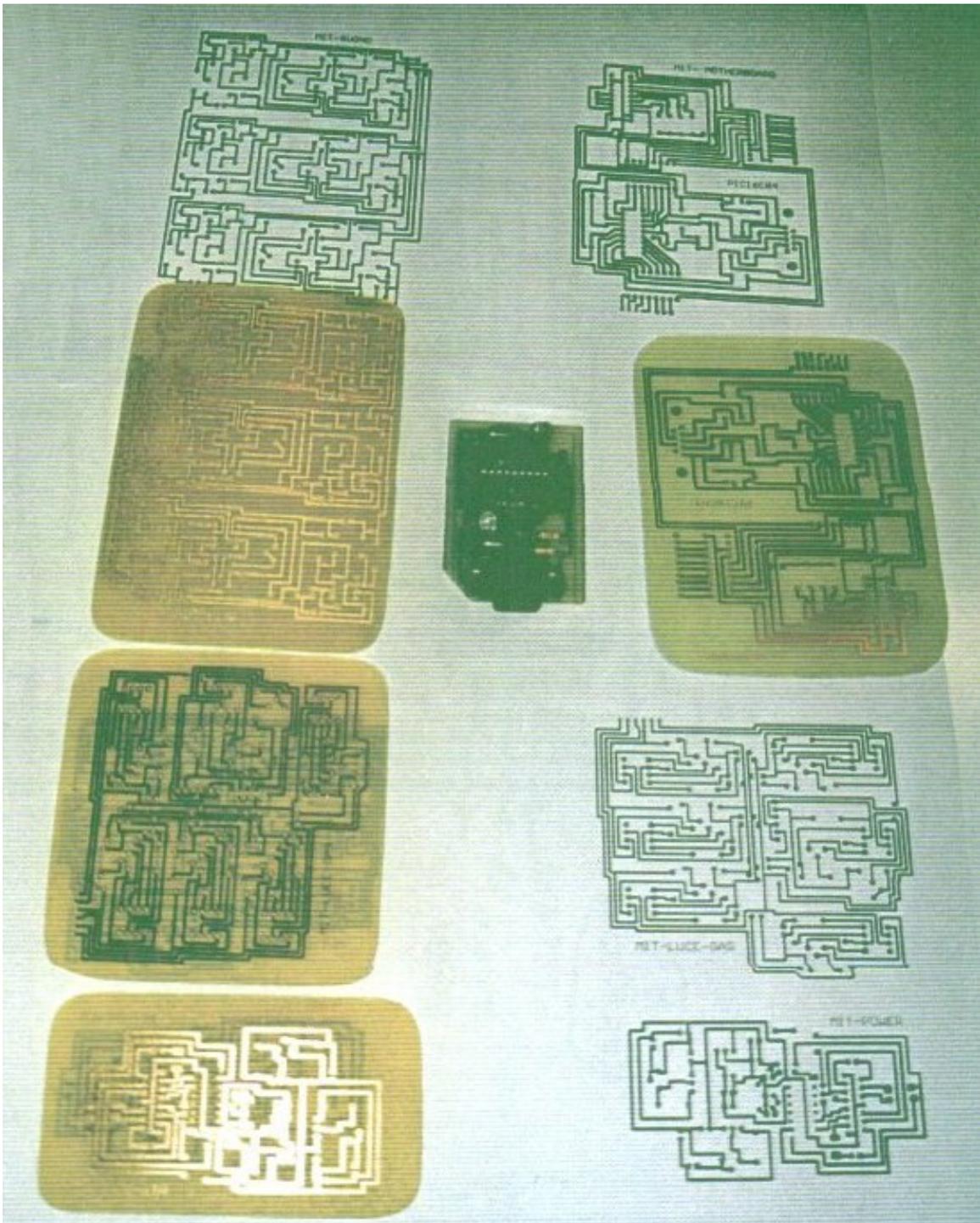
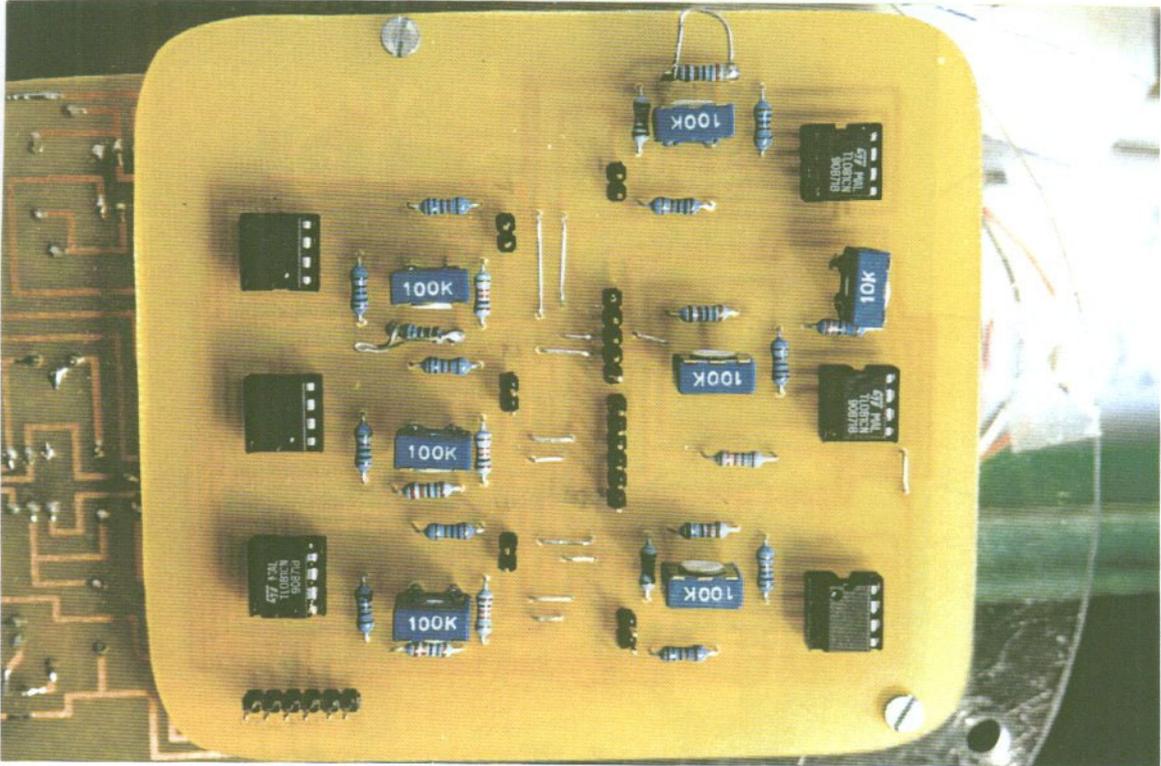
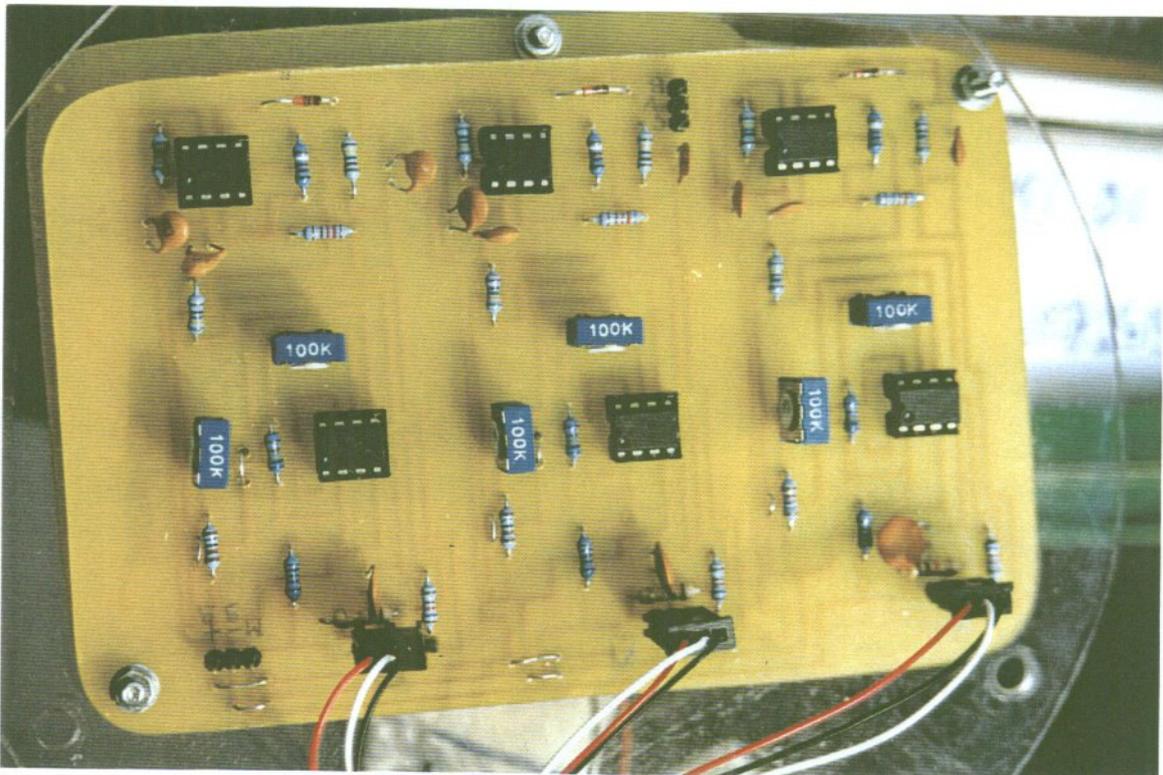


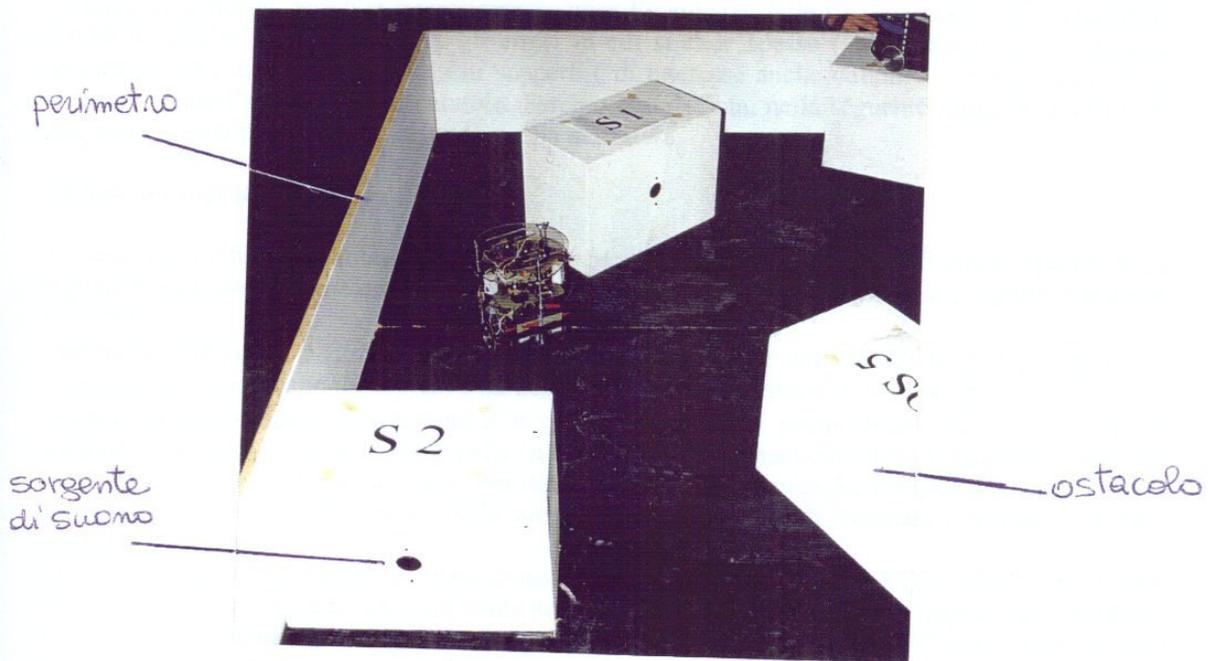
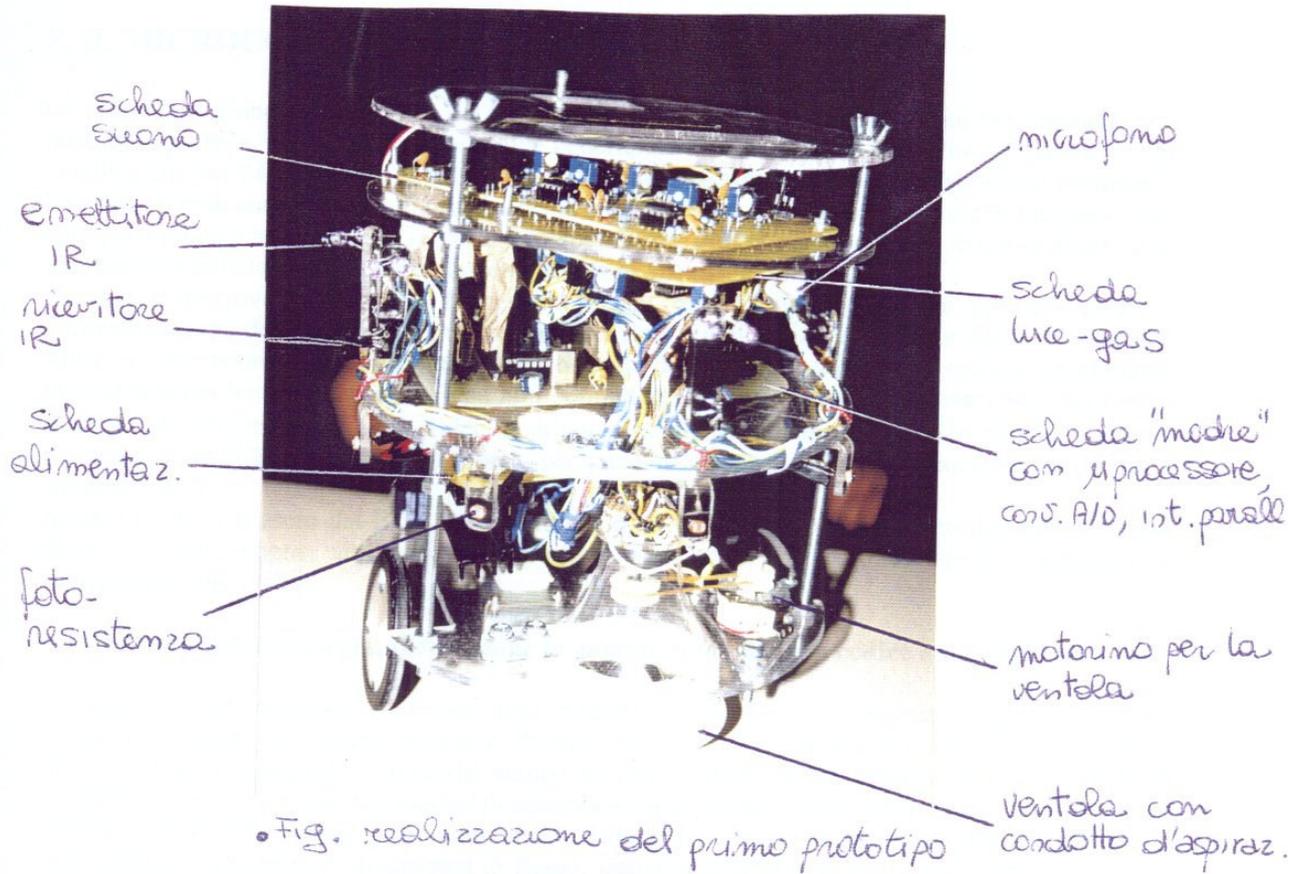
FIG. CREAZIONE DELLE PISTE IN RAME CON IL METODO DELLA FOTOINCISIONE



• fig. scheda della luce e del gas



• fig. scheda del suono



• Fig. campo di esplorazione e robot

7. IL MICROCONTROLLO E L'ASSEMBLER

La programmazione in assembler del microcontrollore, svolge essenzialmente tre funzioni, che possiamo porre a tre livelli, in base alla genericità: la strategia di controllo del sistema, la condivisione del BUS da parte dei dispositivi, la comunicazione con ciascun dispositivo connesso. La prima è indipendente dalla logica dei dispositivi, ma solo dalle informazioni che giungono nel circuito digitale e ne escono; la seconda è indipendente, grosso modo, dal tipo di convertitore, o di interfaccia parallela; la terza è direttamente legata ai dispositivi montati.

L'unico dispositivo che potrebbe fungere da Master, oltre al microprocessore, potrebbe essere il convertitore, che decide, dopo la trasmissione del dato e dell'indirizzo da parte del micro, quando finire la conversione, e quindi riprendere il controllo del BUS, per il prossimo ciclo. Considerando però la relativa lentezza del clock, che non permetterebbe di eseguire che 20 istruzioni nel tempo massimo di conversione (21 μ s), si decide di non sfruttare altrimenti tale periodo. Quindi abbiamo un unico MASTER (il microcontrollore), per cui non ci sono problemi di arbitraggio tra varie richieste del BUS.

In base a ciò, il rilascio del BUS al convertitore o all'interfaccia, viene attuato dal master, prima della sua utilizzazione, abilitando le opportune linee condivise, con le rispettive direzioni, e selezionando poi il chip.

7.1 Decisione della strategia, suddivisione in moduli, traduzione in codice e simulazione.

Gli algoritmi che presenterò tra breve, sono in realtà il risultato di un processo di approssimazione crescente, secondo uno schema ricorsivo. Prima si pensa ad una strategia, la quale viene simulata in un modellino (in scala) di cartone del campo di gara, e viene perciò corretta; poi la si divide in moduli, e si implementano tali moduli in assembler; ogni modulo viene compilato separatamente, e simulato passo dopo passo nell'ambiente di debugging, per cui subisce modifiche anche nella formulazione in termini di diagrammi di flusso; infine, si assembla tutto il programma, e vengono fatte delle prove sul robot reale, nell'ambiente reale, in base alle quali viene ancora modificata la strategia di partenza. Sarebbe interessante seguire l'evolversi delle varie versioni degli algoritmi, perché da essi si possono scoprire gli errori in cui si cade spesso (di interesse esclusivamente personale), ed i modi in cui sono stati scoperti (di interesse anche collettivo); ma disperderebbe eccessivamente l'argomento del capitolo. Faremo quindi finta, nella seguente trattazione, che ogni fase non sia stata poi ritoccata, essendo quella definitiva.

7.2 Subroutines per calcolare il tempo

Ci sono tre modi per calcolare il tempo; sono basati su conteggi del ciclo di clock, oppure di un segnale di sincronizzazione esterno. Ovviamente, il segnale di sincronizzazione è l'unità temporale di conteggio. Sappiamo che con il nostro quarzo da 4Mhz, il micro esegue una istruzione in 1 μ s; tranne le istruzioni che modificano il Program Counter (salti, chiamate e ritorni di sottoprogrammi).

Allora un primo modo, sicuramente il migliore per tempi di poche decine di μ s, è quello di eseguire n istruzioni che non fanno nulla (nop), o che fanno qualcosa che è vantaggioso fare, nel frattempo. Nell'esecuzione del ciclo, per determinare il valore iniziale, bisogna ovviamente tener conto di tutte le istruzioni eseguite nella subroutine, anche quelle di chiamata e ritorno (vedi pag. seguente).

Invece, per tempi fino ad un centinaio di ms, si può utilizzare il registro RTCC (Real Time Clock Counter), che viene incrementato automaticamente ogni n istruzioni eseguite, oppure ogni volta che avviene una transizione (che si può scegliere in salita o discesa) sul pin RTCC, a seconda di come si setta il bit RTS del registro OPTION (vedere data sheet del PIC16C84); il parametro n viene determinato in base ai bit PS<2,1,0> di OPTION; ad esempio, si può prendere n=128. Poiché RTCC è un registro ad 8 bit, dopo 255 va in overflow, e torna a zero. Tale evento può anche essere segnalato da interrupt, per cui il micro è libero di eseguire altri compiti, durante il conteggio.

Per tempi ancora più lunghi, poiché il contatore RTCC si può incrementare al massimo 255 volte, prima di un interrupt, si può mettere un altro contatore, che ad ogni interrupt di overflow di RTCC, si incrementa; e così via.

FIG. TEMPORIZZAZIONE A CONTEGGIO

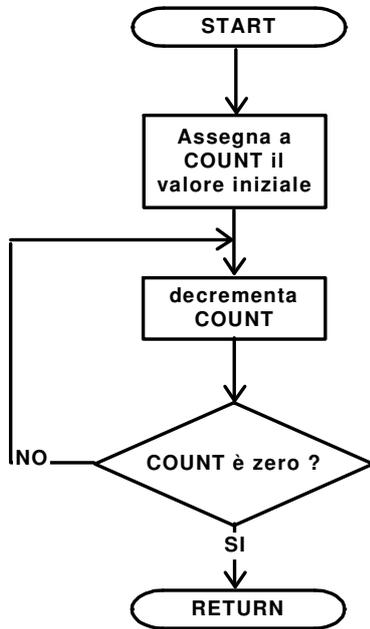
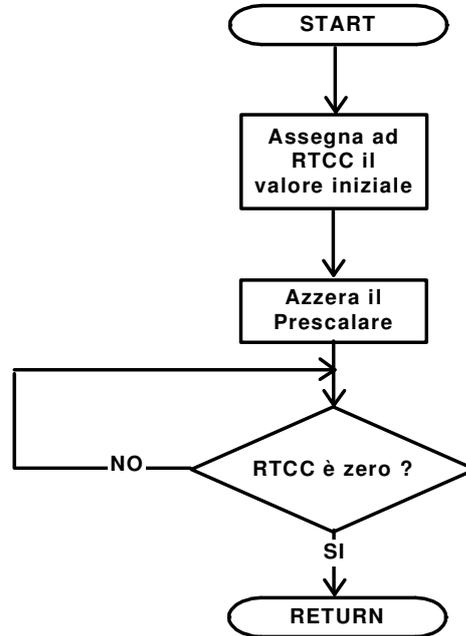


FIG. TEMPORIZZAZIONE AD EVENTO



```

;-----
; subroutine di attesa 10 microsecondi
;-----
DELAY10us
    movlw    .3                ;
    movwf   COUNT             ; carica 10->COUNT
decrem    decfsz  COUNT        ; decrementa e salta quando zero
    goto   decrem             ;
    return                    ;
;-----
; subroutine di attesa 5ms
;-----
    (nel vettore di reset)
    ...
    movlw  b'00000111'        ; pull up RB abilitati, interruzione ai fronti in...
    movwf  OPTIOM             ; ...discesa di RB, RTCC 1:256
    ...

DELAY5ms
    movlw  .100                ; ripristina RTCC
    movwf  RTCC                 ;
delay    clrwdt                ; azzerà watchdog
    movf   RTCC,0              ; RTCC viene automat. incrementato ogni 128 istr.
    skpz   ; RTCC->W
    goto   delay                ; salta se zero
    return                       ; (256-100)=5ms*4Mhz/(4*128)
;-----
; subroutine di attesa 3s
;-----
DELAY3s
    movlw  .60                ; compio 60 volte ritardi di 50ms
    movwf  COUNT              ;
DELAY50ms
    movlw  .61                ; ripristina RTCC
    movwf  RTCC                 ;
delay2   clrwdt                ; azzerà il prescalare
    movf   RTCC,0              ; RTCC->W
    skpz   ; salta se zero
    goto   delay2              ; (256-61)=50ms*4Mhz/(4*256)
    decfsz COUNT              ; decem. salta ze 0
    goto   DELAY50ms
return
  
```

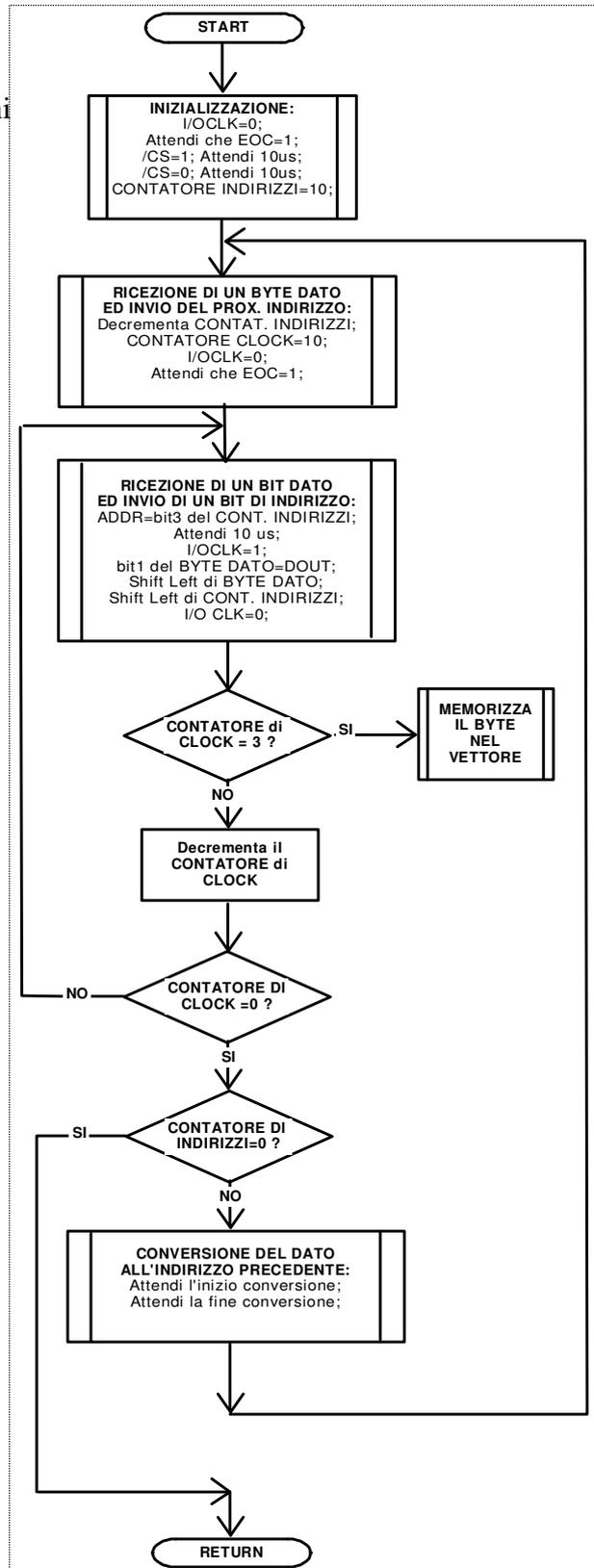
7.3 Il protocollo di comunicazione con il convertitore A/D; il salto a molte alternative.

Avevamo deciso, in fase di progettazione del BUS, di utilizzare un protocollo asincrono, con sfruttamento esclusivo dei fili durante la conversione.

Per ciò che riguarda i tempi, sappiamo che ogni istruzione del pic dura 1 μ s; quindi se abbiamo tempi superiori a questo, bisognerà inserire istruzioni nop, oppure dei cicli di attesa. Risultano particolarmente lunghi i tempi di transizione del /CS ed ADDR(10 μ s); inoltre il clock di I/O non può essere superiore ad 1.1 Mhz, il che significa che dopo aver settato il clock, bisogna aspettare almeno una istruzione prima di azzerarlo.

Quindi si ricava il seguente algoritmo, ed il corrispondente codice assembler. In particolare, la subroutine MEMORIZZA realizza un salto a più alternative, (del tipo case x of ...), perché a seconda del valore del contatore degli indirizzi, si deve memorizzare il dato nella variabile-sensore corrispondente. Ciò viene realizzato dividendo ogni subroutine in un numero K fisso di istruzioni (3 nella fattispecie), per cui basta sommare al Program Counter il fattore K*contatore_indirizzi, per saltare alla subroutine opportuna.

fig. diagramma di flusso del protocollo per il conv. A/D



;fig. protocollo di comunicazione con il conv. A/D
TLV1543M

bsf RP0

; seleziona banco 1

```

        bsf    TRIS_A,4           ; set RA4 come ingresso per EOC
        bcf    RP0                ; seleziona banco 0
        bcf    IOCLK              ; clock a zero, all'initialize
initEOC btfss  EOC                ; attendi EOC=1 all'initialize
        goto  initEOC            ;
        bsf    _CS                ; _CS=1 prima dell'initialize
        call  DELAY10us          ; aspetta 10us per stabilizzare CS
        bcf    _CS                ; seleziono il chip
        call  DELAY10us          ; aspetta 10us per stabilizzare CS
        movlw .10                ; primo indirizzo 10 (sarà decrementato:il primo è 8)
        movwf ADD_CNT            ;
nextbyte
        decf  ADD_CNT            ; decrementa indirizzo
        movlw .10                ; copia 10 nel clock counter
        movwf CLK_CNT            ; 10° clock da effettuare
        bcf    IOCLK              ; il clock parte con zero (indirizzi campionati in salita)
waitEOC btfss  EOC                ; aspetto che eoc=1
        goto  waitEOC            ;
        decf  ADD_CNT,0          ; ADD_CNT-1->W
        movwf ADDRESS            ; w->ADDRESS
        swapf ADDRESS            ; mi serve la parte bassa da mandare dalla + significativa
nextbit  btfsc  ADDRESS,7        ; copia il bit7 di ADDRESS in ADDR
        bsf    ADDR              ;
        btfss ADDRESS,7         ;
        bcf    ADDR              ;
        call  DELAY10us          ; aspetta 10us per stabilizzare ADDR
        bsf    IOCLK              ; fronte di salita del clock
        btfsc DOUT              ; copia il bit di DOUT in DATAS
        bsf    C                  ; copio il bit nel carry
        btfss DOUT              ;
        bcf    C                  ;
        rlf   DATAS              ; shft left e digerisci il carry
        rlf   ADDRESS            ; preparo il successivo bit di addr
        bcf    IOCLK              ; fronte in discesa del clock
        movlw .3                 ; se siamo al 8° clock, salva il byte
        xorwf CLK_CNT,0         ;
        btfsc Z                  ; (se non è 3 salta la prox)
        call  MEMORIZZA          ; (altrim. memorizza DATAS)
        decfsz CLK_CNT          ; decrementa il contatore di clock
        goto  nextbit            ; ritorna se non zero
        movf  ADD_CNT            ; se è zero ADD_CNT finisci
        btfsc Z                  ;
        return                    ;
clrEOC  btfsc  EOC                ; aspetta che inizi la conversione
        goto  clrEOC            ;
setEOC  btfss  EOC                ; aspetta che finisca la conversione
        goto  setEOC            ;
        goto  nextbyte          ; ritorna al prox. ciclo di lettura del byte
;-----
MEMORIZZA
        bcf    Z                  ; flag Z reset
        rlf   ADD_CNT,0          ; ADD_CNT*2->W
        addwf ADD_CNT,0          ; ADD_CNT*(2+1)->W
        addwf PCL,1              ; (ADD_CNT*3)+PCL->PCL : prox. istruz. 1+ADD_CNT*3.
        movf  DATAS,0            ; (ADD_CNT=0) DATAS->W
        movwf SND_SX             ; W->SND_DX
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=1) DATAS->W
        movwf SND_CX             ; W->SND_CX
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=2) DATAS->W
        movwf SND_DX             ; W->SND_SX
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=3) DATAS->W
        movwf LUX_SX             ; W->LUX_DX
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=4) DATAS->W
        movwf LUX_SC             ; W->LUX_DC
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=5) DATAS->W
        movwf LUX_CX             ; W->LUX_CX
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=6) DATAS->W
        movwf LUX_DC             ; W->LUX_SC
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=7) DATAS->W
        movwf LUX_DX             ; W->LUX_SX
        return                    ;
        movf  DATAS,0            ; (ADD_CNT=8) DATAS->W
        movwf GAS_CX             ; W->GAS_CX
        return                    ;
        return                    ; (ADD_CNT=9) prima lettura del conv.A/D non scrive dati

```

(diagr. di flusso e temporizzazioni)

Temporizzazione del conv. A/D

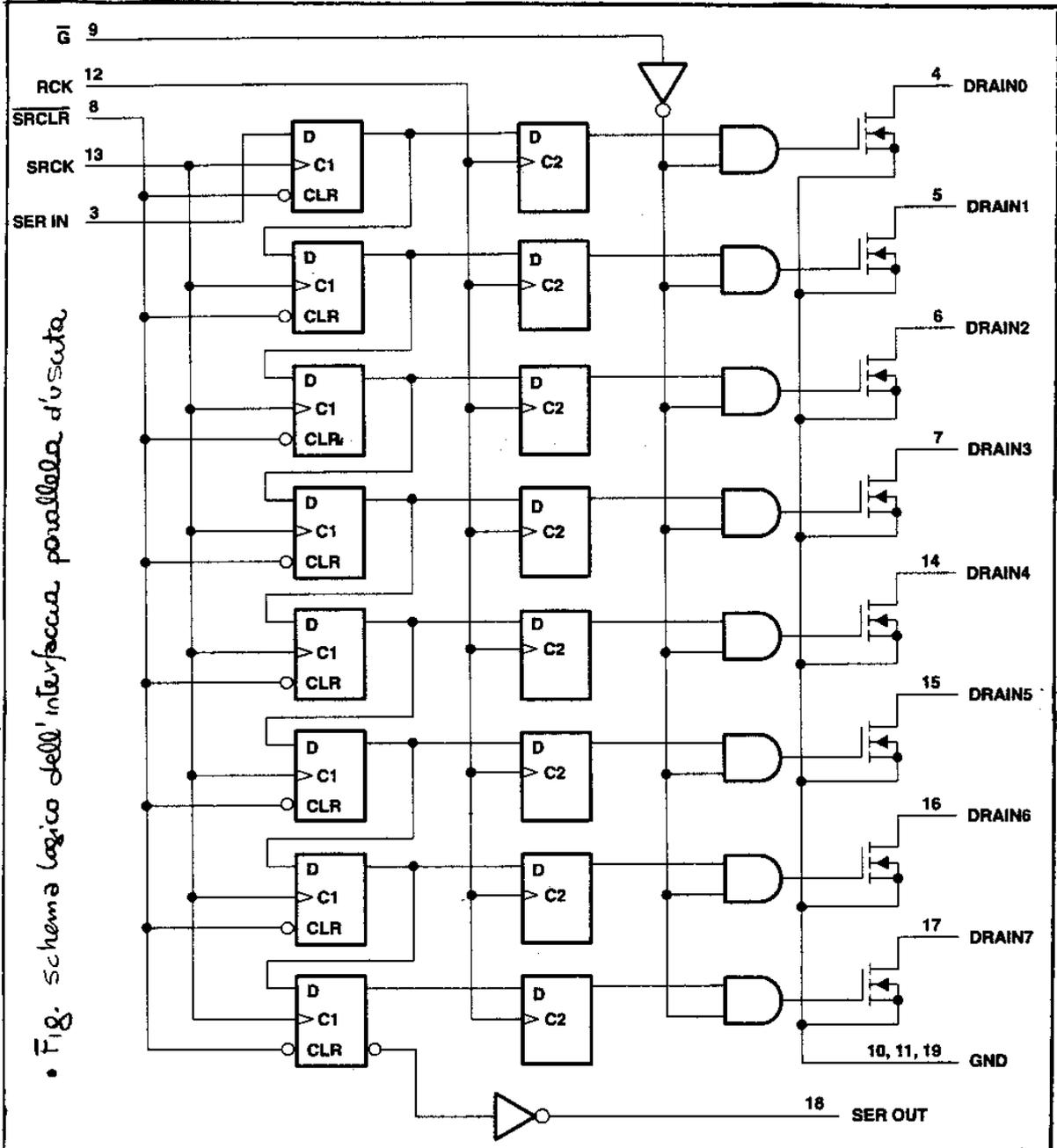
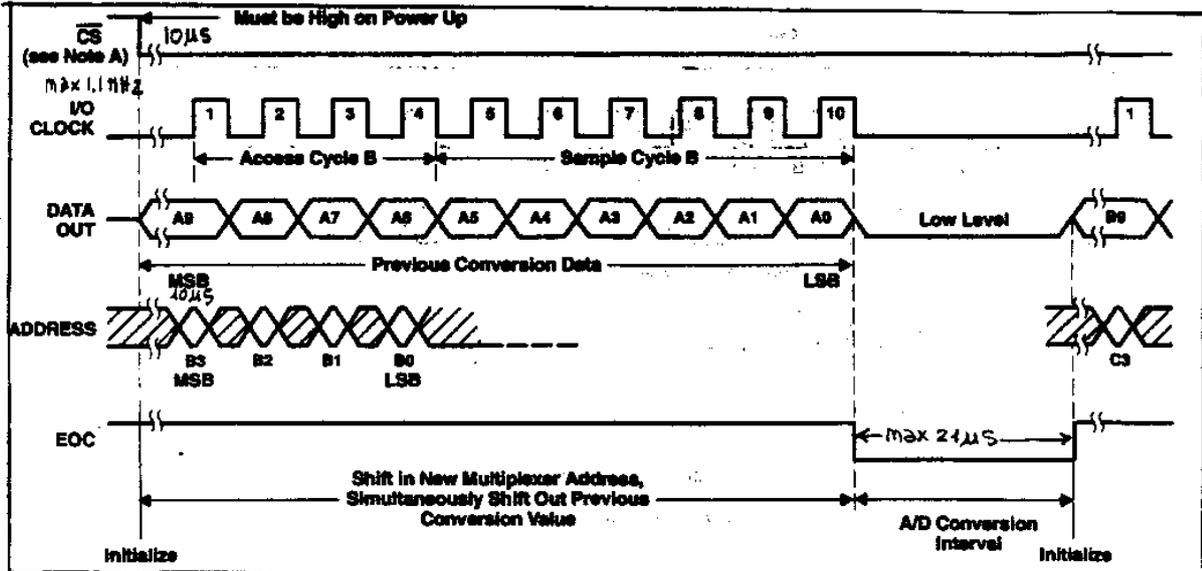


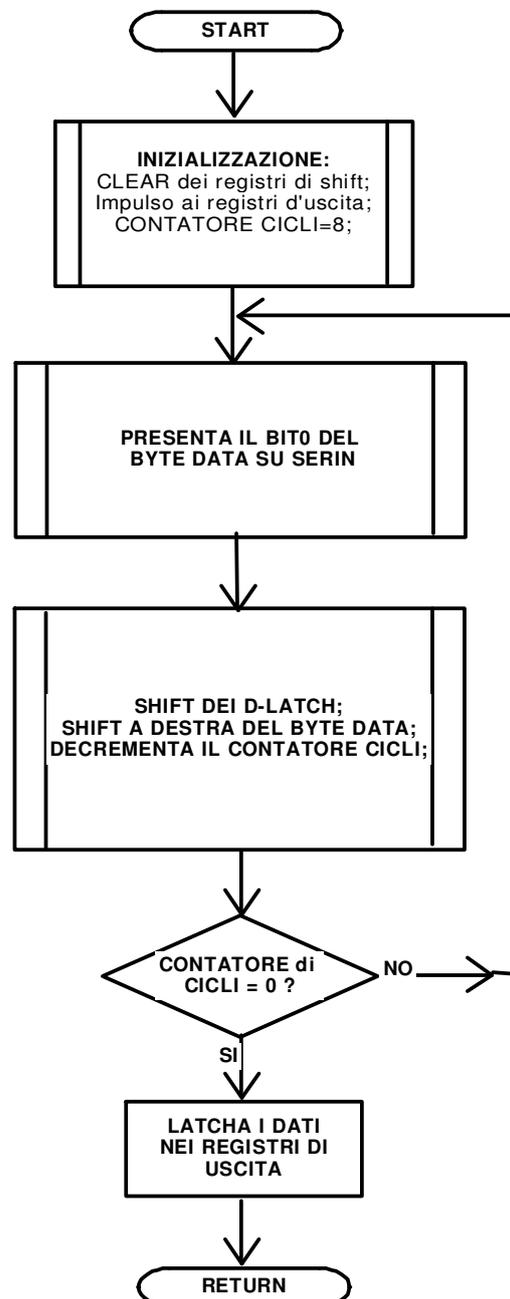
Fig. Schema logico dell'interfaccia parallela d'uscita

7.4 La gestione dell'interfaccia parallela d'uscita

L'interfaccia parallela di uscita, è una rete sequenziale asincrona, costituita da un primo strato di flip-flop D-latch in serie, ognuno dei quali manda la sua uscita anche ad un altro D-latch, l'insieme dei quali costituisce un secondo strato, le cui uscite pilotano delle porte open-collector, integrabili con resistenze di pull-up, per avere l'uno logico.

Ad ogni colpo di clock (seriale), i dati presenti nel primo strato di flip-flop vengono shiftati, così che dopo otto colpi, il primo strato presenta al secondo strato l'intera sequenza di bit, che può quindi essere digerita dagli altri D-latch, e portata in uscita, sotto la barriera del Global Select. L'ultimo D-latch dà la sua uscita ad un serial-out, che potrebbe essere il ser-in di un altro dispositivo identico (modularità). I tempi di transizione sono tutti dell'ordine delle centinaia di ns, inferiori al ciclo di una istruzione; si è preferito però mantenere ogni transizione almeno un ciclo di clock, per coprire eventuali disturbi dovuti alle capacità della linea (si veda il problema delle saldature). In basso viene mostrato il diagramma di flusso ed il codice.

; FIG. DIAGRAMMA DI FLUSSO PER L'INTERFACCIA PARALLELA



```

;-----
; protocollo di trasmissione dati per l'interfaccia parallela
; porta il byte contenuto di W su tutti i pin d'uscita
; il bit0->D7, bit1->D6, ... bit7->D0
;-----
TPIC6B595
    bsf    RP0                ; seleziona banco 1
    bcf    TRIS_A, 4          ; set RA4 come uscita per /SRLCLR
    bcf    RP0                ; seleziona banco 0
    bcf    _SRCLR             ; inizio reset dei registri D-latch
    movwf  DATAS              ; carico il vettore dati
    movlw  .8                 ;
    bsf    RCK                ; apro i latch per buttare i vecchi dati registrati
    movwf  COUNT              ; contatore cicli =8
    bcf    RCK                ; chiudo i latch
    bsf    _SRCLR             ; fine reset dei registri D-latch
invia1  btfss  DATAS,0        ; se il bit0 è 1 salta
        goto  invia0          ; se il bit0 è 0 invia 0
        bsf    SRIN           ; invia 1
        goto  shift           ; vai a preparare il prox. bit
invia0  bcf    SRIN           ; invia 0
        nop                ; attendi almeno 1 clock
shift   bsf    SRCK           ; inizia lo shift dei registri d-latch
        rrf    DATAS          ; ruota il vettore dati
        bcf    SRCK           ; fine dello shift dei registri d-latch
        decfsz COUNT,1       ; salta se hai mandato 8 bit
        goto  invia1         ;
        bsf    RCK            ; apro i latch per buttare i dati registrati
        nop                ; attendo
        bcf    RCK            ; chiudo i latch
        return

```

7.5 La subroutine per l'aggiramento degli ostacoli

Sin dall'inizio del programma, è sempre attivo l'interrupt per la presenza degli ostacoli. Qualsiasi cosa faccia il robot, sia rilevare una sorgente, o cercarne altre, l'interrupt agisce sempre ad una distanza di pochi cm dagli ostacoli, minore di quella di segnalazione sorgente. Quando parte, la routine di interrupt, cerca di liberare il robot davanti e da entrambi i lati, concludendosi quando tutti i lati sono liberi. Particolare attenzione meritano due soluzioni, all'interno dell'algoritmo. La prima è sul tempo minimo di spostamento e/o rotazione da effettuare. La seconda, sulla direzione di svolta in caso di ostacoli davanti. Quando almeno un ricevitore viene attivato (basso), avviene un interrupt, che interrompe i motori come prima cosa; se il robot non aveva molta inerzia (ad esempio perché stava girandosi verso una sorgente), si ferma istantaneamente, rimanendo su questa zona di transizione, in cui il ricevitore infrarosso non vede nettamente il muro, ma oscilla un po', per cui l'interrupt cesserà quasi subito, senza aver prodotto alcun movimento nel robot, per poi riprendere subito dopo; visto che il robot è nella stessa posizione, il ricevitore infrarosso oscillerà sempre, e il risultato è che si entra in un loop, perché pochi microsecondi di azione sui motorini non bastano a farli partire, quindi a spostarsi almeno un po'. Quindi bisogna, ad ogni azione "gira" o "avanza", far partire i motorini per almeno 10ms, così che escano da un eventuale stallo. Ciò significa fare un filtro digitale che dall'oscillazione del ricevitore, prende un segnale piatto ad uno per almeno 10ms. D'altronde questo tempo non deve essere troppo grande, altrimenti i "passi" saranno troppo ampi, con minore precisione, e quindi la traiettoria sarà a zig-zag, dovendo continuamente correggere gli spostamenti eccessivi. E' stato adoperato un tempo di 5ms, contenuto nei limiti suddetti. Per quanto riguarda la questione dell'ostacolo esclusivamente davanti; se il robot si comporta ad essi sempre girando dallo stesso lato (es. destra), andrà a spazzare solo una metà di campo (la destra). La soluzione adottata è girare una volta a destra ed un'altra a sinistra; questo si implementa togliendo un bit ogni volta che si svolta, e decidere in base a tale bit. Alle pagine seguenti, si trova un esempio di simulazione dell'algoritmo di aggiramento ostacoli, su un tipico campo di gara; è stato ottenuto utilizzando le regole del diagramma di flusso, direttamente su carta. Nella realtà, sono state effettuate simulazioni su un modellino di cartone del campo (eseguendo passo per passo gli algoritmi del diagramma di flusso), e poi direttamente con il robot sul campo reale. Si osservi che appena usciti dall'interrupt, il robot è libero a destra, a sinistra, e davanti.

fig. algoritmo per aggirare gli ostacoli

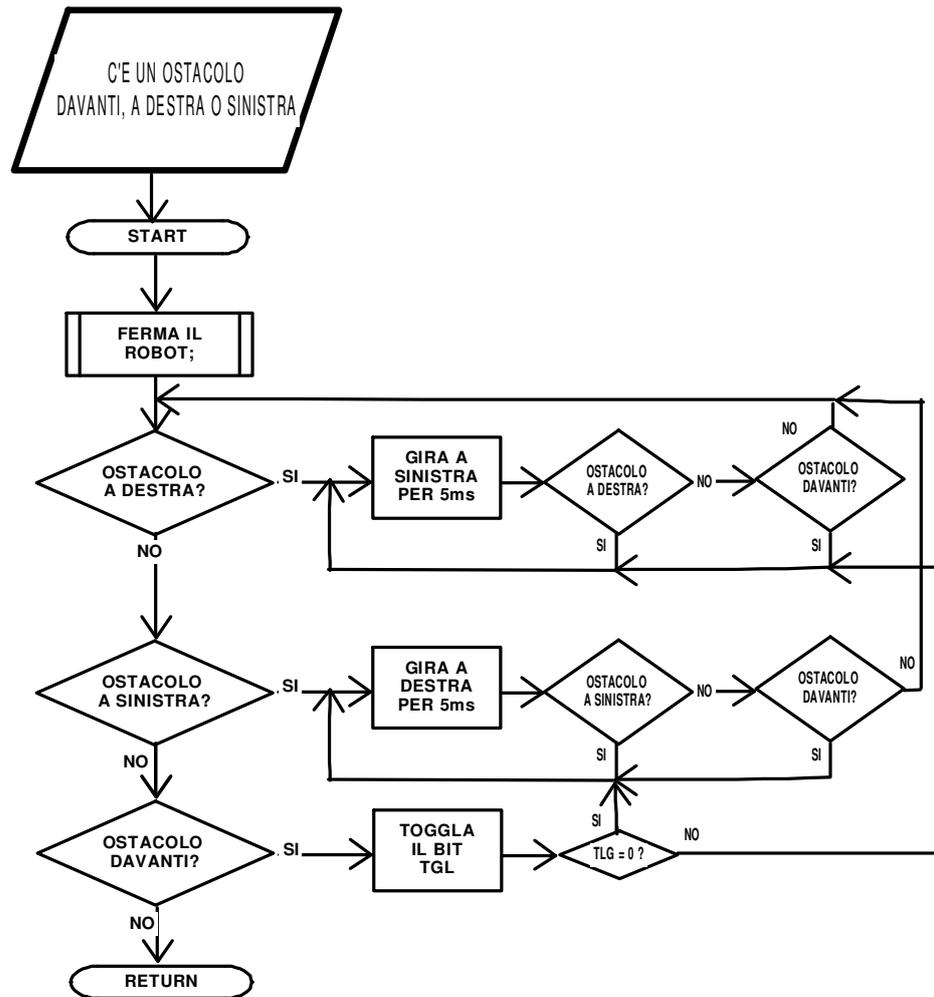


fig. subroutine assembler per aggirare gli ostacoli

```

;-----
; subroutine di interruzione
;-----

ORG 0 ; direttiva: prossima istruzione all'indirizzo zero
goto START ; Reset vector

ORG 4 ; Interrupt vector
movwf WTEMP ; memorizzo il registro W in WTEMP
swapf STATUS,0 ; non uso movf perchè cambia il flag Z
bcf RP0 ; selez. banco 0 (potrebbe essere nel banco 0 o 1)
movwf STEMP ; memorizzo il reg. STATUS ('swappato') in STEMP

;
btfss RBIF ; interrupt da RB ? (potrebbe essere stato RTCC)
goto ENDINT ; no! allora esci
bcf RBIF ; reset RBIF

;
call STOP ; motori spenti
tasta btfss INFR_DX ; se c'è un ostacolo a destra -> vai a sinistra...
goto GIRA_SX ; ...fino a che sei libero a destra
btfss INFR_SX ; se c'è un ostacolo a sinistra -> vai a destra...
goto GIRA_DX ; ... fino a che sei libero a sinistra
btfss INFR_CX ; se c'è un ostacolo davanti -> vai a GIRA
goto GIRA ;
goto ENDINT ; sei libero da ostacoli ,esci dall'interrupt

;-----
; gira a destra, se prima avevi girato a sinistra (strategia globale)
;-----
GIRA movlw b'00000001' ; toglia il bit TGL
xorwf CTRL ;
btfss TGL ;
goto GIRA_DX ; se il bit è 0 vai a destra
goto GIRA_SX ; se il bit è 1 vai a sinistra

;-----
; gira a destra fino a quando non ti liberi a sinistra (tattica locale)
;-----

```

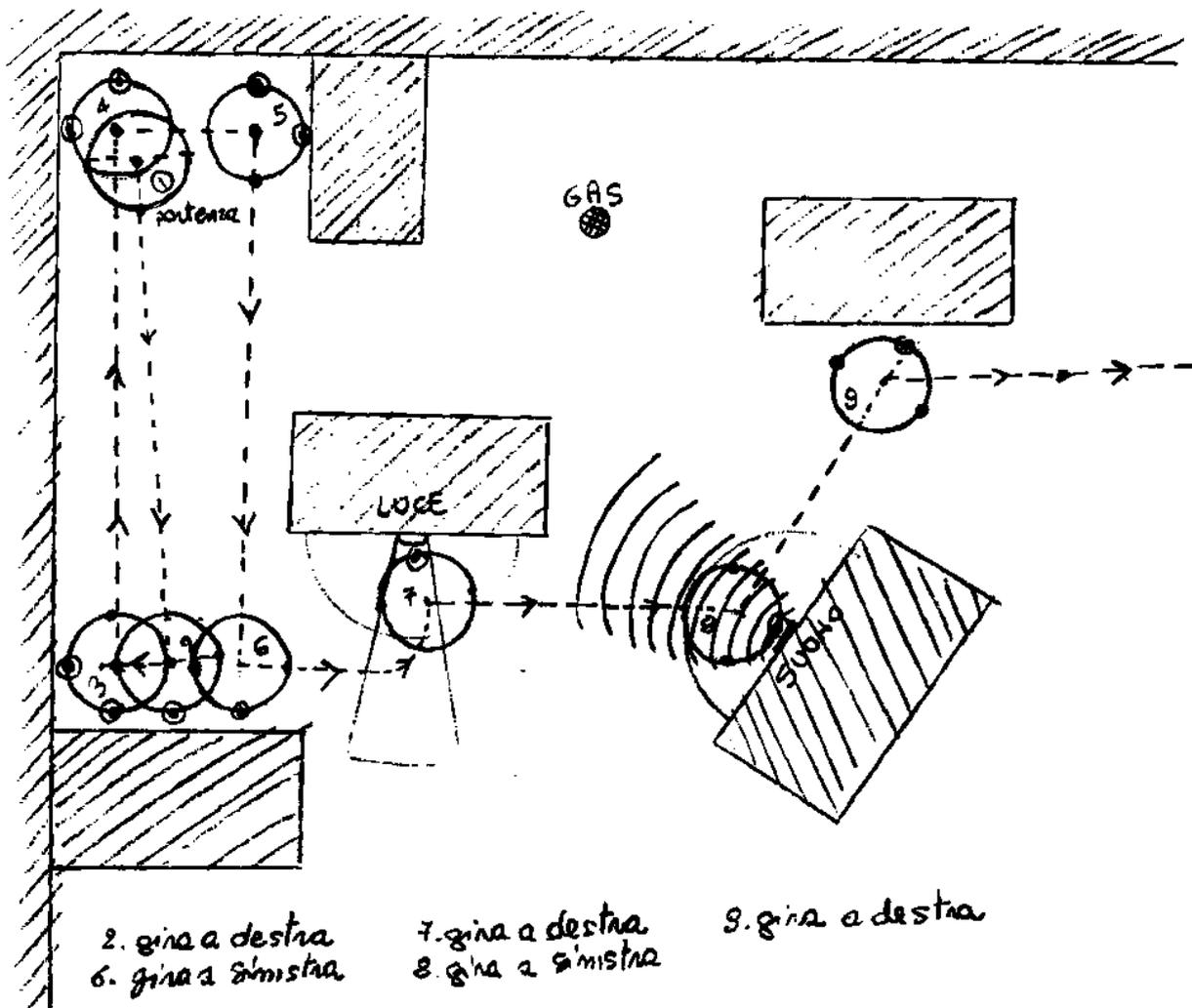
```

GIRA_DX bcf    SX                ; motore SX avanti
        bcf    DX                ; motore DX indietro
        bcf    PWM               ; accendi motori (per un tempo minimo di regime)
        call   DELAY5ms          ;
        btfss  INFR_SX           ; se non sei libero a SX gira ancora
        goto   GIRA_DX           ;
        btfss  INFR_CX           ; se non sei libero a CX gira ancora
        goto   GIRA_DX           ;
        goto   GIRA_DX           ; ora che sei libero a SX, esci
;-----
; gira a sinistra fino a quando non ti liberi a destra (tattica locale)
;-----
GIRA_SX bcf    SX                ; motore SX indietro
        bcf    DX                ; motore DX avanti
        bcf    PWM               ; accendi motori (per un tempo minimo di regime)
        call   DELAY5ms          ;
        btfss  INFR_DX           ; se non sei libero a DX gira ancora
        goto   GIRA_SX           ;
        btfss  INFR_CX           ; se non sei libero a CX gira ancora
        goto   GIRA_SX           ;
        goto   GIRA_SX           ; ora che sei libero a DX, esci

ENDINT  call   STOP              ;
        bcf    RBIE              ; abilita interrupt RB
        bcf    GIE               ; abilita interrupt globale
        bcf    RPO               ; page 0
        swapf  STMP,0            ; ripristino STATUS (due swapf si annullano)
        movwf  STATUS            ; ripristina W senza sporcare STATUS (con le swap)
        swapf  WTEMP,1           ; swap WTEMP->WTWEMP
        swapf  WTEMP,0           ; swap WTEMP->W
        retfie

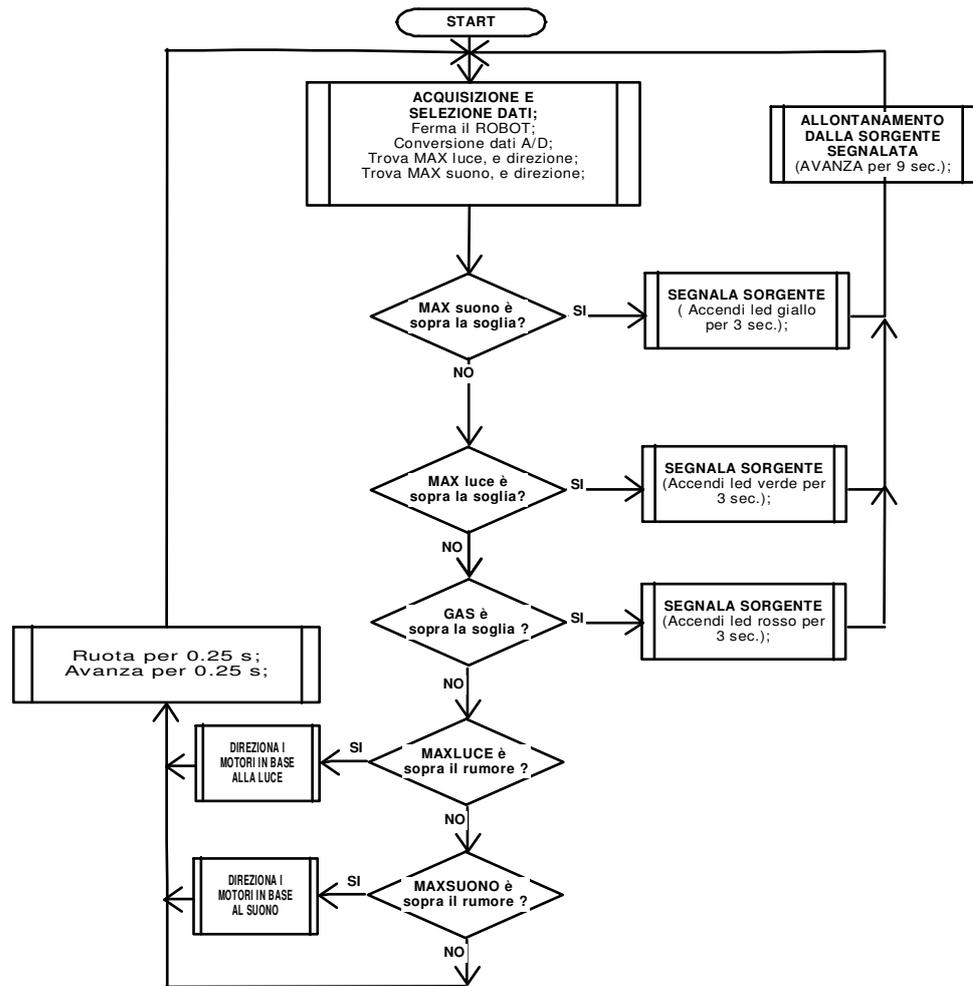
```

fig. simulazione dell' algoritmo su un campione



7.6 La strategia di ricerca e segnalazione; subroutine assembler per trovare il massimo.

Sostanzialmente, la strategia di ricerca confronta tutti i sensori di luce, e ne trova il massimo; poi quelli di suono, trovandone anche il massimo; dopodiché, se il massimo della luce o del suono risultano essere superiori al livello di rumore, avviene una piccola rotazione nel verso di tale massimo, poi un piccolo avanzamento, e ricomincia tutto daccapo. I test di segnalazione delle sorgenti avvengono prima della strategia, così che se si è vicini ad una sorgente, viene segnalato immediatamente.



Dopo la segnalazione, il robot parte e si allontana per 9 secondi e ricomincia tutto d'accapo.

Viene fatta l'ipotesi che, laddove la luce è sopra il rumore, non arriva suono sopra rumore (mentre il viceversa può avvenire, per la caratteristica del suono di essere distribuito nell'ambiente), ma in tal caso viene data importanza prima al suono.

In particolare, per trovare il massimo tra tutte le misurazioni di suono, o di luce, si opera con l'algoritmo, a complessità lineare seguente: si assegna come massimo il 1° elemento, e poi si confronta tale massimo con il 2°, 3°,... elemento, assumendo ogni volta il valore dell'elemento più grande tra i due.

Inoltre, ad ogni assegnamento di massimo, viene memorizzata la direzione in cui ruotare, attraverso i valori di DX ed SX, che sono le direzioni dei due motori. La subroutine ORIENTA, non dovrà far altro che prendere tali direzioni ed accendere i motori.

Nel caso del suono (simile quello della luce,) l'algoritmo precedente è così tradotto:

```

MAX_SUONO
    bsf     DXs                ; suppongo di dover andare avanti (per cambiare dir basta 1
bit)
    bsf     SXs                ;
    movf   SND_CX, 0           ; SND_CX->W
    movwf  SND_MAX             ; W->SND_MAX
    movf   SND_DX, 0           ; SND_DX->W
    subwf  SND_MAX, 0          ; W-SND_MAX->W
    btfsc  C                   ; se SND_DX>SND_MAX è il nuovo massimo

```

```

        goto    snd1                ; altrim. prosegui
        movf   SND_DX,0             ; SND_DX->W
        movwf  SND_MAX              ; W->SND_MAX
        bcf   DXs                   ; direz. destra
        bsf   SXs                   ;
;
snd1    movf   SND_SX,0             ; SND_SX->W
        subwf  SND_MAX,0           ; W-SND_MAX->W
        btfsc C                     ; se SND_SX>SND_MAX è il nuovo massimo
        return                       ;
        movf   SND_SX,0             ; SND_SX->W
        movwf  SND_MAX              ; W->SND_MAX
        bcf   SXs                   ; direz. sinistra
        bsf   DXs                   ;
return

```

Segue ora, il codice assembler per la ricerca delle sorgenti.

```

;-----
; subroutine di controllo principale
;-----
MAIN   clrwdt                       ; azzera prescalare
        call   STOP                  ; ferma il robot
        call   TLV1543M              ; misure dal conv. A/D
        bsf   DX                     ; ipotesi: in mancanza d'altro vai avanti
        bsf   SX                     ;
;
        call   MAX_LUCE               ; max luce -> LUX_MAX; direz da seguire->DX1 SX1
        call   MAX_SUONO              ; max suono-> SND_MAX; direz da seguire->DXs SXs
;
        movlw  SND_THR                ; SND_THR->W
        subwf  SND_MAX,0             ; SND_MAX-SND_THR->W
        btfsc C                       ; se sndmax<sndsoglia salta
        goto   STOP_SND               ; subroutine per segnalare i suono
;
        movlw  LUX_THR                ; LUX_THR->W
        subwf  LUX_MAX,0             ; LUX_MAX-LUX_THR->W ;
        btfsc C                       ; se luxmax<luxsoglia salta
        goto   STOP_LUX               ; subroutine per segnalare la luce
;
        movlw  GAS_THR                ; GAS_THR->W
        subwf  GAS_CX,0              ; GAS_CX-GAS_THR->W
        btfsc C                       ; se gas<sndsoglia salta
        goto   STOP_GAS               ; subroutine per segnalare il gas
;
; decisione della sorgente da seguire
; ipotesi: una sola è sopra il background
; LUX_BKG->W
        movlw  LUX_BKG                ; LUX_MAX-LUX_BKG->W
        subwf  LUX_MAX,0             ; se luxmax>luxBKG continua
        btfsc C                       ; controlla indizi sul suono
        goto   cfrsuon                ; DX1->DX orientati alla luce
        btfss DX1                     ;
        bcf   DX                      ;
        btfsc DX1                     ;
        bsf   DX                      ;
        btfss SX1                     ; SX1->SX
        bcf   SX                      ;
        btfsc SX1                     ;
        bsf   SX                      ;
        movlw  LUX_LED                 ; accendo led luce
        call   TPIC6B595              ; (penso)
        movlw  .2                      ;
        movwf  COUNT                  ;
        call   DELAY50ms              ; accendo per 100ms
        goto   ORIENTA                ;
;
cfrsuon movf   SND_BKG,0              ; SND_BKG->W
        subwf  SND_MAX,0             ; SND_MAX-SND_BKG->W
        btfsc C                       ; se sndmax>sndBKG continua
        goto   ORIENTA                ; vai a spostare il robot
        btfss DXs                     ; DX1->DX
        bcf   DX                      ;
        btfsc DXs                     ;
        bsf   DX                      ;
        btfss SXs                     ; SX1->SX
        bcf   SX                      ;
        btfsc SXs                     ;
        bsf   SX                      ;
;
ORIENTA movlw  IR_ALL                  ; spengo i led
        call   TPIC6B595              ;
        bsf   PWM                      ; ruota verso la sorgente

```

```

movlw .5 ;
movwf COUNT ; ruota per 0.25s (a meno di interrupt)
movlw .1
movwf COUNT
call DELAY50ms ;
btfss DX ; se il max è centrale, avanza un po'
goto MAIN ;
btfss SX ;
goto MAIN ;
call AVANTI ; vai avanti per 0.25 s
movlw .5 ;
movwf COUNT ; avanti per 0.25 ms (a meno di interrupt)
call DELAY50ms ;
goto MAIN ;

STOP_LUX
call STOP ; segnalo la luce
movlw LUX_LED ;
goto SEGNALA ;

STOP_GAS
call STOP ; segnalo il gas
movlw GAS_LED ;
goto SEGNALA ;

STOP_SND
call STOP ; segnalo il suono
movlw SND_LED ;
goto SEGNALA ;

SEGNALA call TPIC6B595 ; accende i led per 3 sec
call DELAY3s ;
movlw IR_ALL ; spegni led
call TPIC6B595 ;
call AVANTI ; per 9 sec devo allontanarmi dalla sorg
call DELAY3s ;
call DELAY3s ;
call DELAY3s ;
goto MAIN ; ritorna a misurare
;-----
END ; direttiva di fine programma

```

7.7 Listato completo del codice assembler, e memoria di programma del microcontrollore

Nelle pagine seguenti viene mostrato il listato completo del programma assembler, e la sua compilazione in memoria di programma del micro, che mostra le dimensioni effettive (352 byte) e l'espansione di eventuali istruzioni mnemoniche complesse in linguaggio macchina.

Ci sono ancora 672 istruzioni per riempire tutta la memoria; nel prossimo paragrafo vedremo cosa è possibile fare ancora.

In particolare, sono stati inseriti nelle label mnemoniche, anche i registri speciali per la gestione della EEPROM dati (64 registri ad 8-bit), dove i dati permangono anche in assenza di alimentazione, come il codice di programma, a differenza della SRAM (36 registri ad 8-bit), di cui ne abbiamo sfruttati 19.

Fig. Codice Assembler

```

TITLE 'PIC3' ; controllo del ROBOT
list F=INHX8M, P=16C84 ; direttive: files output .HEX, e pic di tipo 16C84
;__config b'00000000010001' ; configuraz. fusibili CP=off,WDT=off,PWRT=on,OSC=xt
;-----
; assegnazione di label mnemoniche ai registri generali
;-----
RTCC EQU 01H ; Real Time Clock Counter
PCL EQU 02H ; Program Counter Low
STATUS EQU 03H ; Registro di stato
PORT_A EQU 05H ; pin RA<0..4>
PORT_B EQU 06H ; pin RB<0..7>
EEDATA EQU 08H ; EEPROM Data register
EEADR EQU 09H ; EEPROM Address register
EECON1 EQU 88H ; EEPROM Control register
EECON2 EQU 89H ; Registro non fisico
INTCON EQU 0BH ; INTerrupt Control register
TRIS_A EQU 85H ; TRIsate port A (set I/O)
TRIS_B EQU 86H ; TRIsate port B (set I/O)
OPTIOM EQU 81H ; Configuration register ("OPTION" è reserved word)
;-----
;assegnazione di label mnemoniche ad altre celle della SRAM
;-----
CTRL EQU 0CH ; registro dei bit di controllo
COUNT EQU 0DH ; contatore di cicli trasmissione dati int. parallela
DATAS EQU 0EH ; byte dati da inviare serialmente all'int. parallela
SND_DX EQU 0FH ; tensione (digitale) del suono a destra
SND_CX EQU 10H ; tensione (digitale) del suono a centro
SND_SX EQU 11H ; tensione (digitale) del suono a sinistra
LUX_DX EQU 12H ; tensione (digitale) della luce a destra
LUX_DC EQU 13H ; tensione (digitale) della luce in avanti-destra
LUX_CX EQU 14H ; tensione (digitale) della luce in avanti
LUX_SC EQU 15H ; tensione (digitale) della luce in avanti-sinistra
LUX_SX EQU 16H ; tensione (digitale) della luce a sinistra
GAS_CX EQU 17H ; tensione (digitale) del gas
ADD_CNT EQU 18H ; address counter per il conv.A/D
ADDRESS EQU 19H ; address indirizzo preso col conv. A/D
CLK_CNT EQU 1AH ; clock counter per il conv. A/D
LUX_MAX EQU 1BH ; massimo della luce
SND_MAX EQU 1CH ; massimo del suono
STEMP EQU 02EH ; memorizzazione stato
WTEMP EQU 02FH ; memorizzazione W
;-----
;assegnazione di label mnemoniche per alcuni bit di cotrollo
;-----
#define RP0 STATUS,5 ; flag di selezione banco RAM
#define Z STATUS,2 ; flag zero
#define C STATUS,0 ; flag del carry
#define RBIF INTCON,0 ; Int sul fronte di RB<4..7>
#define RBIE INTCON,3 ; abilita RBIF
#define RTIF INTCON,2 ; RTCC Interrupt Flag
#define RTIE INTCON,5 ; RTCC Interrupt Enable
#define GIE INTCON,7 ; Global interrupt Enable
;-----
;assegnazione di label mnemoniche per i pin di I/O
;-----
#define _CS PORT_B,0 ; (out) Chip Select del conv. A/D
#define SX PORT_B,1 ; (out) Direzione sinistra per motori
#define PWM PORT_B,2 ; (out) PWM (Enable ad impulsi) per motori
#define DX PORT_B,3 ; (out) Direzione destra per motori
#define INFR_SX PORT_B,4 ; (in) Infra Red di sinistra
#define INFR_CX PORT_B,5 ; (in) Infra Red davanti
#define INFR_DX PORT_B,6 ; (in) lnfra Red destra
#define INFR_DD PORT_B,7 ; (in) Infra Red destra per strategia del cieco
#define RCK PORT_A,0 ; (out) Register Clock int. parallela
#define DOUT PORT_A,1 ; (in) Data Out del conv. A/D
#define ADDR PORT_A,2 ; (out) Address del conv. A/D
#define SRIN PORT_A,2 ; (out) Serial Input int. parallela
#define IOCLK PORT_A,3 ; (out) I/O clock conv. A/D
#define SRCK PORT_A,3 ; (out) Shift Register Clock int. par
#define EOC PORT_A,4 ; (out) End Of Conversion conv. A/D
#define _SRCLR PORT_A,4 ; (in) Schift Register Clear int. parallela
;-----
;assegnazione di label mnemoniche per i pin di controllo
;-----
#define TGL CTRL,0 ; Toggle: bit che si setta e resetta alternativamente
#define DXs CTRL,1 ; direzione motore DX secondo il suono
#define SXs CTRL,2 ; direzione motore SX secondo il suono
#define DXl CTRL,3 ; direzione motore DX secondo la luce
#define SXl CTRL,4 ; direzione motore SX secondo la luce
;-----

```

```

;assegnazione di label mnemoniche per l'interfaccia parallela TPIC6b595
;-----
IR_ALL EQU    b'11110000'      ;metto a zero i D<0,1,2,3> alta imped. gli altri
LED_ALL EQU   b'11111110'      ;accendo tutti i led
GAS_LED EQU   b'11111000'      ;metto a zero il D4 (led rosso)
LUX_LED EQU   b'11110100'      ;metto a zero il D5 (led verde)
SND_LED EQU   b'11110010'      ;metto a zero il D6 (led giallo)
;-----
;assegnazione di label mnemoniche per i valori di soglia e di fondo
; formula per convertire le tensioni in bit: Lbit=Lvolt/6volt*256bit
;-----
SND_THR EQU   b'01000000'      ; sound threshold, soglia di livello suono (3V->128)
LUX_THR EQU   b'11101010'      ; light threshold, soglia di livello luce (5.5V->234)
GAS_THR EQU   b'11101010'      ; gas threshold, soglia di livello gas (5.5V->234)
LUX_BKG EQU   b'10101010'      ; light background, luce di fondo (4V->170)
SND_BKG EQU   b'00010101'      ; sound background, suono di fondo(0.5V->21)
;-----
; subroutine di interruzione
;-----
                ORG    0          ; direttiva: prossima istruzione all'indirizzo zero
                goto   START      ; Reset vector
                ;
                ORG    4          ; Interrupt vector
                movwf  WTEMP       ; memorizzo il registro W in WTEMP
                swapf  STATUS,0    ; non uso movf perchè cambia il flag Z
                bcf    RP0         ; selez. banco 0 (potrebbe essere nel banco 0 o 1)
                movwf  STEMP       ; memorizzo il reg. STATUS ('swappato') in STEMP
                ;
                btfss  RBIF        ; interrupt da RB ? (potrebbe essere stato RTCC)
                goto   ENDINT      ; no! allora esci
                bcf    RBIF        ; reset RBIF
                ;
                call   STOP        ; motori spenti
tasta          btfss  INFR_DX      ; se c'è un ostacolo a destra -> vai a sinistra...
                goto   GIRA_SX     ; ...fino a che sei libero a destra
                btfss  INFR_SX     ; se c'è un ostacolo a sinistra -> vai a destra...
                goto   GIRA_DX     ; ... fino a che sei libero a sinistra
                btfss  INFR_CX     ; se c'è un ostacolo davanti -> vai a GIRA
                goto   GIRA        ;
                goto   ENDINT      ; sei libero da ostacoli ,esci dall'interrupt
;-----
; gira a destra, se prima avevi girato a sinistra (strategia globale)
;-----
GIRA          movlw   b'00000001'  ; toglia il bit TGL
                xorwf  CTRL        ;
                btfss  TGL        ;
                goto   GIRA_DX     ; se il bit è 0 vai a destra
                goto   GIRA_SX     ; se il bit è 1 vai a sinistra
;-----
; gira a destra fino a quando non ti liberi a sinistra (tattica locale)
;-----
GIRA_DX       bsf    SX           ; motore SX avanti
                bcf    DX         ; motore DX indietro
                bsf    PWM        ; accendi motori (per un tempo minimo di regime)
                call   DELAY5ms    ;
                btfss  INFR_SX     ; se non sei libero a SX gira ancora
                goto   GIRA_DX     ;
                btfss  INFR_CX     ; se non sei libero a CX gira ancora
                goto   GIRA_DX     ;
                goto   tasta       ; ora che sei libero a SX, esci
;-----
; gira a sinistra fino a quando non ti liberi a destra (tattica locale)
;-----
GIRA_SX       bcf    SX           ; motore SX indietro
                bsf    DX         ; motore DX avanti
                bsf    PWM        ; accendi motori (per un tempo minimo di regime)
                call   DELAY5ms    ;
                btfss  INFR_DX     ; se non sei libero a DX gira ancora
                goto   GIRA_SX     ;
                btfss  INFR_CX     ; se non sei libero a CX gira ancora
                goto   GIRA_SX     ;
                goto   tasta       ; ora che sei libero a DX, esci

ENDINT        call   STOP        ;
                bsf    RBIE       ; abilita interrupt RB
                bsf    GIE        ; abilita interrupt globale
                bcf    RP0        ; page 0
                swapf  STEMP,0     ; ripristino STATUS (due swapf si annullano)
                movwf  STATUS      ; ripristina W senza sporcare STATUS (con le swap)
                swapf  WTEMP,1     ; swap WTEMP->WTWEMP
                swapf  WTEMP,0     ; swap WTEMP->W
                retfie

```

```

;-----
; subroutine di reset
;-----
START  bsf    RP0                ; seleziona SRAM banco 1
        movlw b'00000010'        ; setta RA<0,2,3,4> come out, RA1 come in
        movwf TRIS_A            ;
        movlw b'11110000'        ; setta RB<0..3> come out, RB<4..7> come in
        movwf TRIS_B            ;
        movlw b'00000111'        ; pull up RB abilitati, interruzione ai fronti in...
        movwf OPTIOM            ; ...discesa di RB, RTCC 1:256
        bcf    RP0                ; seleziona SRAM banco 0
        clrf  PORT_A            ; reset RA ed RB
        clrf  PORT_B            ;
        clrwdt                ; azzerata prescalare
        movlw IR_ALL            ; accendo gli IR (D0..D3), saranno sempre accesi
        call  TPIC6B595         ;
        call  DELAY5ms          ; aspetto che vadano a regime i ricevitori
        clrf  INTCON            ; disabilita interrupt
        bsf   RBIE              ; abilita interrupt RB
        bsf   GIE               ; abilita interrupt globale
        movlw LED_ALL          ; accendo tutti i led (D4..D6)
        call  TPIC6B595         ; (penso)
        call  DELAY3s           ; *****
        goto  MAIN              ; salto alla procedura di controllo principale
;-----
; subroutines di movimento del robot
;-----
AVANTI  bsf    DX                ; motore destro avanti
        bsf    SX                ; motore sinistro avanti
        bsf    PWM              ; abilitazione motori
        return

STOP    bcf    PWM              ; disabilitazione motori
        bcf    DX                ; motore destro retromarcia
        bcf    SX                ; motore sinistro retromarcia
        bsf    PWM              ; abilita la "frenata"
        ; eventuale ritardo
        bcf    PWM              ; disabilita motori
        return
;-----
; protocollo di trasmissione dati per l'interfaccia parallela
; porta il byte contenuto di W su tutti i pin d'uscita
; il bit0->D7,bit1->D6,...bit7->D0
;-----
TPIC6B595
        bsf    RP0                ; seleziona banco 1
        bcf    TRIS_A,4         ; set RA4 come uscita per /SRLCLR
        bcf    RP0                ; seleziona banco 0
        bcf    _SRCLR           ; inizio reset dei registri D-latch
        movwf DATAS             ; carico il vettore dati
        movlw .8                ;
        bsf    RCK              ; apro i latch per buttare i vecchi dati registrati
        movwf COUNT            ; contatore cicli =8
        bcf    RCK              ; chiudo i latch
        bsf    _SRCLR           ; fine reset dei registri D-latch
invia1  btfss  DATAS,0          ; se il bit0 è 1 salta
        goto  invia0            ; se il bit0 è 0 invia 0
        bsf    SRIN             ; invia 1
        goto  shift            ; vai a preparare il prox. bit
invia0  bcf    SRIN             ; invia 0
        nop                    ; attendi almeno 1 clock
shift   bsf    SRCK             ; inizia lo shift dei registri d-latch
        rrf    DATAS            ; ruota il vettore dati
        bcf    SRCK             ; fine dello shift dei registri d-latch
        decfsz COUNT,1         ; salta se hai mandato 8 bit
        goto  invia1            ;
        bsf    RCK              ; apro i latch per buttare i dati registrati
        nop                    ; attendo
        bcf    RCK              ; chiudo i latch
        return
;-----
; subroutine che legge tutti i sensori e mette i valori in un vettore
;-----
TLV1543M
        bsf    RP0                ; seleziona banco 1
        bsf    TRIS_A,4         ; set RA4 come ingresso per EOC
        bcf    RP0                ; seleziona banco 0
        bcf    IOCLK            ; clock a zero, all'initialize
initEOC btfss  EOC              ; attendi EOC=1 all'initialize
        goto  initEOC          ;
        bsf    _CS              ; _CS=1 prima dell'initialize
        call  DELAY10us        ; aspetta 10us per stabilizzare CS
        bcf    _CS              ; seleziono il chip

```

```

        call    DELAY10us      ; aspetta 10us per stabilizzare CS
        movlw  .10             ; primo indirizzo 10 (sarà decrementato:il primo è 8)
        movwf  ADD_CNT        ;
nextbyte
        decf   ADD_CNT         ; decrementa indirizzo
        movlw  .10             ; copia 10 nel clock counter
        movwf  CLK_CNT        ; 10° clock da effettuare
        bcf   IOCLK           ; il clock parte con zero (indirizzi campionati in salita)
waitEOC btfss   EOC            ; aspetto che eoc=1
        goto  waitEOC         ;
        decf   ADD_CNT,0      ; ADD_CNT-1->W
        movwf  ADDRESS        ; w->ADDRESS
        swapf  ADDRESS        ; mi serve la parte bassa da mandare dalla + significativa
nextbit  btfsc  ADDRESS,7     ; copia il bit7 di ADDRESS in ADDR
        bsf   ADDR           ;
        btfss ADDRESS,7     ;
        bcf   ADDR           ;
        call  DELAY10us      ; aspetta 10us per stabilizzare ADDR
        bsf   IOCLK         ; fronte di salita del clock
        btfsc DOUT          ; copia il bit di DOUT in DATAS
        bsf   C              ; copio il bit nel carry
        btfss DOUT          ;
        bcf   C              ;
        rlf   DATAS          ; shft left e digerisci il carry
        rlf   ADDRESS        ; preparo il successivo bit di addr
        bcf   IOCLK         ; fronte in discesa del clock
        movlw .3              ; se siamo al 8° clock, salva il byte
        xorwf CLK_CNT,0      ;
        btfsc Z              ; (se non è 3 salta la prox)
        call  MEMORIZZA     ; (altrim. memorizza DATAS)
        decfsz CLK_CNT      ; decrementa il contatore di clock
        goto  nextbit        ; ritorna se non zero
        movf  ADD_CNT        ; se è zero ADD_CNT finisci
        btfsc Z              ;
        return              ;
clrEOC  btfsc  EOC           ; aspetta che inizi la conversione
        goto  clrEOC        ;
setEOC  btfss  EOC           ; aspetta che finisca la conversione
        goto  setEOC        ;
        goto  nextbyte      ; ritorna al prossimo ciclo
;-----
; copia DATAS all'indirizzo (SND_CX+ADD_CNT)
;-----
MEMORIZZA
        bcf   Z              ; flag Z reset
        rlf   ADD_CNT,0      ; ADD_CNT*2->W
        addwf  ADD_CNT,0     ; ADD_CNT*(2+1)->W
        addwf  PCL,1         ; (ADD_CNT*3)+PCL->PCL : prox. istruz. 1+ADD_CNT*3.
        movf  DATAS,0        ; (ADD_CNT=0) DATAS->W
        movwf SND_SX         ; W->SND_DX
        return              ;
        movf  DATAS,0        ; (ADD_CNT=1) DATAS->W
        movwf SND_CX         ; W->SND_CX
        return              ;
        movf  DATAS,0        ; (ADD_CNT=2) DATAS->W
        movwf SND_DX         ; W->SND_SX
        return              ;
        movf  DATAS,0        ; (ADD_CNT=3) DATAS->W
        movwf LUX_SX         ; W->LUX_DX
        return              ;
        movf  DATAS,0        ; (ADD_CNT=4) DATAS->W
        movwf LUX_SC         ; W->LUX_DC
        return              ;
        movf  DATAS,0        ; (ADD_CNT=5) DATAS->W
        movwf LUX_CX         ; W->LUX_CX
        return              ;
        movf  DATAS,0        ; (ADD_CNT=6) DATAS->W
        movwf LUX_DC         ; W->LUX_SC
        return              ;
        movf  DATAS,0        ; (ADD_CNT=7) DATAS->W
        movwf LUX_DX         ; W->LUX_SX
        return              ;
        movf  DATAS,0        ; (ADD_CNT=8) DATAS->W
        movwf GAS_CX         ; W->GAS_CX
        return              ;
        return              ; (ADD_CNT=9) prima lettura del conv.A/D
                                non scrive dati
;-----
; subroutine di attesa 10 microsecondi
;-----
DELAY10us
        movlw .3              ;
        movwf COUNT          ; carica 10->COUNT

```

```

decrem decfsz COUNT          ; decrementa e salta quando zero
goto decrem                  ;
return                        ;
;-----
; subroutine di attesa 5ms
;-----
DELAY5ms
movlw .100                    ; ripristina RTCC
movwf RTCC                    ;
delay clrwdt                  ; azzera watchdog
                                ; RTCC viene automat. incrementato ogni 128 istr.
movf RTCC,0                   ; RTCC->W
skpz                          ; salta se zero
goto delay                    ; (256-100)=5ms*4Mhz/(4*128)
return

;-----
; subroutine di controllo principale
;-----
MAIN clrwdt                    ; azzera prescalare
call STOP                     ; ferma il robot
call TLV1543M                 ; misure dal conv. A/D
bsf DX                         ; ipotesi: in mancanza d'altro vai avanti
bsf SX                         ;

call MAX_LUCE                  ; max luce -> LUX_MAX; direz da seguire->DX1 SX1
call MAX_SUONO                 ; max suono-> SND_MAX; direz da seguire->DXs SXs

movlw SND_THR                  ; SND_THR->W
subwf SND_MAX,0               ; SND_MAX-SND_THR->W
btfsc C                        ; se sndmax<sndsoglia salta
goto STOP_SND                 ; subroutine per segnalare i suono

movlw LUX_THR                  ; LUX_THR->W
subwf LUX_MAX,0               ; LUX_MAX-LUX_THR->W ;
btfsc C                        ; se luxmax<luxsoglia salta
goto STOP_LUX                 ; subroutine per segnalare la luce

movlw GAS_THR                  ; GAS_THR->W
subwf GAS_CX,0                ; GAS_CX-GAS_THR->W
btfsc C                        ; se gas<sndsoglia salta
goto STOP_GAS                 ; subroutine per segnalare il gas

                                ; decisione della sorgente da seguire
                                ; ipotesi: una sola è sopra il background
movlw LUX_BKG                  ; LUX_BKG->W
subwf LUX_MAX,0               ; LUX_MAX-LUX_BKG->W
btfsc C                        ; se luxmax>luxBKG continua
goto cfrsuon                  ; controlla indizi sul suono
btfss DX1                      ; DX1->DX orientati alla luce
bcf DX                          ;
btfsc DX1                      ;
bsf DX                          ;
btfss SX1                      ; SX1->SX
bcf SX                          ;
btfsc SX1                      ;
bsf SX                          ;
movlw LUX_LED                  ; accendo led luce
call TPIC6B595                 ; (penso)
movlw .2
movwf COUNT
call DELAY50ms                 ; accendo per 100ms
goto ORIENTA                   ;

cfrsuon movf SND_BKG,0         ; SND_BKG->W
subwf SND_MAX,0               ; SND_MAX-SND_BKG->W
btfsc C                        ; se sndmax>sndBKG continua
goto ORIENTA                   ; vai a spostare il robot
btfss DXs                      ; DX1->DX
bcf DX                          ;
btfsc DXs                      ;
bsf DX                          ;
btfss SXs                      ; SX1->SX
bcf SX                          ;
btfsc SXs                      ;
bsf SX                          ;

ORIENTA
movlw IR_ALL                   ; spengo i led
call TPIC6B595                 ;
bsf PWM                         ; ruota verso la sorgente
movlw .5
movwf COUNT                     ; ruota per 0.25s (a meno di interrupt)
movlw .1

```

```

movwf COUNT
call DELAY50ms
btfss DX ; se il max è centrale, avanza un po'
goto MAIN
btfss SX
goto MAIN
call AVANTI ; vai avanti per 0.25 s
movlw .5
movwf COUNT ; avanti per 0.25 ms (a meno di interrupt)
call DELAY50ms
goto MAIN

MAX_LUCE
bit) bsf DX1 ; suppongo di dover andare avanti (per cambiare dir basta 1
      bsf SX1 ;
      movf LUX_CX,0 ; LUX_CX->W
      movwf LUX_MAX ; W->LUX_MAX
      movf LUX_DC,0 ; LUX_DC->W
      subwf LUX_MAX,0 ; W-LUX_MAX->W
      btfsc C ; se LUX_DC>LUX_MAX è il nuovo massimo
      goto lux1 ;
      movf LUX_DC,0 ; LUX_DC->W
      movwf LUX_MAX ; W->LUX_MAX
      bcf DX1 ; direz. destra
      bsf SX1 ;

lux1 movf LUX_DX,0 ; LUX_DX->W
     subwf LUX_MAX,0 ; W-LUX_MAX->W
     btfsc C ; se LUX_DX>LUX_MAX è il nuovo massimo
     goto lux2 ; altrim. prosegui
     movf LUX_DX,0 ; LUX_DX->W
     movwf LUX_MAX ; W->LUX_MAX
     bcf DX1 ; direz. destra
     bsf SX1 ;

lux2 movf LUX_SC,0 ; LUX_SC->W
     subwf LUX_MAX,0 ; W-LUX_MAX->W
     btfsc C ; se LUX_SC>LUX_MAX è il nuovo massimo
     goto lux3 ; altrim. prosegui
     movf LUX_SC,0 ; LUX_SC->W
     movwf LUX_MAX ; W->LUX_MAX
     bcf SX1 ; direz. sinistra
     bsf DX1 ;

lux3 movf LUX_SX,0 ; LUX_SX->W
     subwf LUX_MAX,0 ; W-LUX_MAX->W
     btfsc C ; se LUX_SX>LUX_MAX è il nuovo massimo
     return ; altrim. esci
     movf LUX_SX,0 ; LUX_SX->W
     movwf LUX_MAX ; W->LUX_MAX
     bcf SX1 ; direz. sinistra
     bsf DX1 ;

return

MAX_SUONO
bit) bsf DXs ; suppongo di dover andare avanti (per cambiare dir basta 1
      bsf SXs ;
      movf SND_CX,0 ; SND_CX->W
      movwf SND_MAX ; W->SND_MAX
      movf SND_DX,0 ; SND_DX->W
      subwf SND_MAX,0 ; W-SND_MAX->W
      btfsc C ; se SND_DX>SND_MAX è il nuovo massimo
      goto snd1 ; altrim. prosegui
      movf SND_DX,0 ; SND_DX->W
      movwf SND_MAX ; W->SND_MAX
      bcf DXs ; direz. destra
      bsf SXs ;

snd1 movf SND_SX,0 ; SND_SX->W
     subwf SND_MAX,0 ; W-SND_MAX->W
     btfsc C ; se SND_SX>SND_MAX è il nuovo massimo
     return ;
     movf SND_SX,0 ; SND_SX->W
     movwf SND_MAX ; W->SND_MAX
     bcf SXs ; direz. sinistra
     bsf DXs ;

return

STOP_LUX
call STOP ; segnalo la luce

```

```

        movlw LUX_LED           ;
        goto  SEGNALA         ;

STOP_GAS
        call  STOP             ; segnale il gas
        movlw GAS_LED         ;
        goto  SEGNALA         ;

STOP_SND
        call  STOP             ; segnale il suono
        movlw SND_LED         ;
        goto  SEGNALA         ;

SEGNALA call  TPIC6B595        ; accende i led per 3 sec
        call  DELAY3s         ;
        movlw IR_ALL          ; spegni led
        call  TPIC6B595        ;
        call  AVANTI          ; per 9 sec devo allontanarmi dalla sorg
        call  DELAY3s         ;
        call  DELAY3s         ;
        call  DELAY3s         ;
        goto  MAIN            ; ritorna a misurare

DELAY3s
        movlw .60             ; compio 60 volte ritardi di 50ms
        movwf COUNT          ;

DELAY50ms
        movlw .61             ; ripristina RTCC
        movwf RTCC           ;
delay2  clrwdt                ; azzera il prescalare
        movf  RTCC,0          ; RTCC->W
        skpz                ; salta se zero
        goto delay2          ; (256-61)=50ms*4Mhz/(4*256)
        decfsz COUNT         ; decrem. salta ze 0
        goto  DELAY50ms

return

;-----
END                                     ; direttiva di fine programma

```

(program memory)

• fig. PROGRAM MEMORY

Address	Hex	Label	Instruction	Hex	Label	Instruction
1	0000		goto START	79	004E	1086 bcf 0x6,0x1
2	0001		addlw 0xff	80	004F	1506 bsf 0x6,0x2
3	0002		addlw 0xff	81	0050	1106 bcf 0x6,0x2
4	0003		addlw 0xff	82	0051	0008 return
5	0004		movwf 0x2F	83	0052	1683 TPIC6B bsf 0x3,0x5
6	0005		swapf 0x3,W	84	0053	1205 bcf 0x5,0x4
7	0006		bcf 0x3,0x5	85	0054	1283 bcf 0x3,0x5
8	0007		movwf 0x2E	86	0055	1205 bcf 0x5,0x4
9	0008		btfs 0xB,0x0	87	0056	008E movwf 0xE
10	0009		goto ENDINT	88	0057	3008 movlw 0x8
11	000A		bcf 0xB,0x0	89	0058	1405 bsf 0x5,0x0
12	000B		call STOP	90	0059	008D movwf 0xD
13	000C	tasta	btfs 0x6,0x6	91	005A	1005 bcf 0x5,0x0
14	000D		goto GIRA_SX	92	005B	1605 bsf 0x5,0x4
15	000E		btfs 0x6,0x4	93	005C	1C0E invial btfs 0xE,0x0
16	000F		goto GIRA_DX	94	005D	2860 goto invia0
17	0010		btfs 0x6,0x5	95	005E	1505 bsf 0x5,0x2
18	0011		goto GIRA	96	005F	2862 goto shift
19	0012		goto ENDINT	97	0060	1105 invia0 bcf 0x5,0x2
20	0013	GIRA	movlw 0x1	98	0061	0000 nop
21	0014		xorwf 0xC	99	0062	1585 shift bsf 0x5,0x3
22	0015		btfs 0xC,0x0	100	0063	0C8E rrf 0xE
23	0016		goto GIRA_DX	101	0064	1185 bcf 0x5,0x3
24	0017		goto GIRA_SX	102	0065	0B8D decfsz 0xD
25	0018	GIRA_D	bsf 0x6,0x1	103	0066	285C goto invial
26	0019		bcf 0x6,0x3	104	0067	1405 bsf 0x5,0x0
27	001A		bsf 0x6,0x2	105	0068	0000 nop
28	001B		call DELAY5ms	106	0069	1005 bcf 0x5,0x0
29	001C		btfs 0x6,0x4	107	006A	0008 return
30	001D		goto GIRA_DX	108	006B	1683 TLV154 bsf 0x3,0x5
31	001E		btfs 0x6,0x5	109	006C	1605 bsf 0x5,0x4
32	001F		goto GIRA_DX	110	006D	1283 bcf 0x3,0x5
33	0020		goto tasta	111	006E	1185 bcf 0x5,0x3
34	0021	GIRA_S	bcf 0x6,0x1	112	006F	1E05 initEO btfs 0x5,0x4
35	0022		bsf 0x6,0x3	113	0070	286F goto initEOC
36	0023		bsf 0x6,0x2	114	0071	1406 bsf 0x6,0x0
37	0024		call DELAY5ms	115	0072	20BB call DELAY10us
38	0025		btfs 0x6,0x6	116	0073	1006 bcf 0x6,0x0
39	0026		goto GIRA_SX	117	0074	20BB call DELAY10us
40	0027		btfs 0x6,0x5	118	0075	300A movlw 0xA
41	0028		goto GIRA_SX	119	0076	0098 movwf 0x18
42	0029		goto tasta	120	0077	0398 nextby decf 0x18
43	002A	ENDINT	call STOP	121	0078	300A movlw 0xA
44	002B		bsf 0xB,0x3	122	0079	009A movwf 0x1A
45	002C		bsf 0xB,0x7	123	007A	1185 bcf 0x5,0x3
46	002D		bcf 0x3,0x5	124	007B	1E05 waitEO btfs 0x5,0x4
47	002E		swapf 0x2E,W	125	007C	287B goto waitEOC
48	002F		movwf 0x3	126	007D	0318 decf 0x18,W
49	0030		swapf 0x2F	127	007E	0099 movwf 0x19
50	0031		swapf 0x2F,W	128	007F	0E99 swapf 0x19
51	0032		retfie	129	0080	1B99 nextbi btfs 0x19,0x7
52	0033	START	bsf 0x3,0x5	130	0081	1505 bsf 0x5,0x2
53	0034		movlw 0x2	131	0082	1F99 btfs 0x19,0x7
54	0035		movwf 0x5	132	0083	1105 bcf 0x5,0x2
55	0036		movlw 0xF0	133	0084	20BB call DELAY10us
56	0037		movwf 0x6	134	0085	1585 bsf 0x5,0x3
57	0038		movlw 0x7	135	0086	1885 btfs 0x5,0x1
58	0039		movwf 0x1	136	0087	1403 bsf 0x3,0x0
59	003A		bcf 0x3,0x5	137	0088	1C85 btfs 0x5,0x1
60	003B		clrf 0x5	138	0089	1003 bcf 0x3,0x0
61	003C		clrf 0x6	139	008A	0D8E rlf 0xE
62	003D		clrwdt	140	008B	0D99 rlf 0x19
63	003E		movlw 0xF0	141	008C	1185 bcf 0x5,0x3
64	003F		call TPIC6B595	142	008D	3003 movlw 0x3
65	0040		call DELAY5ms	143	008E	061A xorwf 0x1A,W
66	0041		clrf 0xB	144	008F	1903 btfs 0x3,0x2
67	0042		bsf 0xB,0x3	145	0090	209B call MEMORIZZA
68	0043		bsf 0xB,0x7	146	0091	0B9A decfsz 0x1A
69	0044		movlw 0xFE	147	0092	2880 goto nextbit
70	0045		call TPIC6B595	148	0093	0898 movf 0x18
71	0046		call DELAY3s	149	0094	1903 btfs 0x3,0x2
72	0047		goto MAIN	150	0095	0008 return
73	0048	AVANTI	bsf 0x6,0x3	151	0096	1A05 clrEOC btfs 0x5,0x4
74	0049		bsf 0x6,0x1	152	0097	2896 goto clrEOC
75	004A		bsf 0x6,0x2	153	0098	1E05 setEOC btfs 0x5,0x4
76	004B		return	154	0099	2898 goto setEOC
77	004C	STOP	bcf 0x6,0x2	155	009A	2877 goto nextbyte
78	004D		bcf 0x6,0x3	156	009B	1103 MEMORI bcf 0x3,0x2

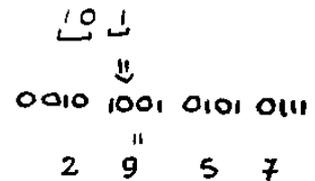

```

313 0138 108C      bcf    0xC,0x1
314 0139 150C      bsf    0xC,0x2
315 013A 0811      snd1   movf   0x11,W
316 013B 021C      subwf  0x1C,W
317 013C 1803      btfsz  0x3,0x0
318 013D 0008      return
319 013E 0811      movf   0x11,W
320 013F 009C      movwf  0x1C
321 0140 110C      bcf    0xC,0x2
322 0141 148C      bsf    0xC,0x1
323 0142 0008      return
324 0143 204C      STOP_I call   STOP
325 0144 30F4      movlw  0xF4
326 0145 294C      STOP_G goto   SEGNALA
327 0146 204C      call   STOP
328 0147 30F8      movlw  0xF8
329 0148 294C      goto   SEGNALA
330 0149 204C      STOP_S call   STOP
331 014A 30F2      movlw  0xF2
332 014B 294C      goto   SEGNALA
333 014C 2052      SEGNAL call  TPIC6B595
334 014D 2155      call   DELAY3s
335 014E 30F0      movlw  0xF0
336 014F 2052      call  TPIC6B595
337 0150 2048      call   AVANTI
338 0151 2155      call   DELAY3s
339 0152 2155      call   DELAY3s
340 0153 2155      call   DELAY3s
341 0154 28C7      goto   MAIN
342 0155 303C      DELAY3 movlw  0x3C
343 0156 008D      movwf  0xD
344 0157 303D      DELAY5 movlw  0x3D ←
345 0158 0081      movwf  0x1
346 0159 0064      delay2 clrwdt
347 015A 0801      movf   0x1,W
348 015B 1D03      btfsz  0x3,0x2
349 015C 2959      goto   delay2
350 015D 0B8D      decfsz 0xD
351 015E 2957      goto   DELAY50ms ←
352 015F 0008      return
    
```

esempio: codice operativo dell'istruzione

goto k: 10 1KKK KKKK KKKK

nel nostro caso, DELAY50ms => 0157h = 0000 0001 0101 0111 :



7.8 Ulteriori migliorie possibili, con lo stesso hardware.

Oltre a rendere più complessa la ricerca delle sorgenti, è possibile introdurre alcune procedure, di carattere 'nuovo', che permettono di creare algoritmi con funzioni tipicamente considerate "intelligenti", come la supervisione del proprio operato, l'apprendimento delle soglie, e l'acquisizione di esperienza per ottimizzare le strategie.

La supervisione dell'operato di altre procedure è un'operazione tipica nei sistemi ad alto grado di complessità, sia per evitare situazioni di stallo, sia per misurare le performances.

Per l'aggiramento degli ostacoli, ad esempio, possiamo mettere un contatore di rotazioni, che si azzerava ogni volta che parte l'interrupt; se il robot non riuscisse a fuoriuscire da una nicchia, tale contatore andrebbe in overflow, e ciò potrebbe essere rivelato dalla subroutine 'supervisore' che provvederebbe a compiere delle azioni drastiche, come ad esempio bloccare i motori, spegnere tutti i led, e mandare il processore in sleep, per evitare di esaurire l'energia.

Per misurare le performances, si può avere un contatore di sorgenti rilevate dopo il primo minuto, così da sapere se si è in ritardo o meno, ed attivare eventuali strategie d'emergenza.

Le soglie di segnalazione e di fondo possono essere rilevate dal robot in un giro di perlustrazione, in cui effettua varie misurazioni, e memorizza in EEPROM i minimi livelli ed i massimi.

Nel giro di perlustrazione, di durata superiore ai cinque minuti, o anche durante la gara, il robot potrebbe registrare dei parametri, come ad esempio il numero di ostacoli incontrati ogni minuto, le sorgenti incontrate, così da avere una stima delle sorgenti da ricercare; oppure effettuare una mappatura grossolana del campo (ad esempio il numero di piazzuole) e memorizzare tutto in EEPROM; si capisce bene, che queste sono forme di elaborazione ad un livello superiore rispetto alla strategia di ricerca, perché il robot, in tal modo, assumerà uno "stato" che si evolve col tempo, e modifica il suo comportamento.

Sembrerebbe, quindi, che al comportamento intelligente non servano linguaggi ad alto livello, in quanto ottenuto dalla concomitanza di tanti piccoli agenti di calcolo, ciascuno dei quali è abbastanza semplice, e non intelligente.

BIBLIOGRAFIA

TESTI:

- P.Giusti, G.Pellacani, *Fondamenti di Chimica*, pagg.260 e segg, III Ed. ETS -Pisa 1991
- S.Rosati, *Fisica Generale*, vol.I, pagg.394-409;596-597, Ed. Ambrosiana - Milano 1991.
- L. Lovitch, S. Rosati, *Fisica Generale*, vol. II, pagg.452-464, Ed. Ambrosiana - Milano 1991.
- A.Landi, *Lezioni di Motion Control*, Edizioni ETS - Pisa 1996.
- G. Marro, *Controlli Automatici*, IV ed., Zanichelli -Bologna 1997.
- G. Marro, *Complementi di Controlli Automatici*, Zanichelli - Bologna 1994.
- L. Rizzo, *Sistemi di Elaborazione* , Ed. SEU - Pisa 1997.
- J. Millman, A.Grabel, *Microelectronics*, II Ed. MCGRAW-HILL 1987.
- B.Pellegrini, *Note sulla reazione*, Ed. SEU - Pisa 1993.
- *Elettronica dei Sistemi Digitali*, appunti del corso, prof. R. Saletti, a.a. 95-96.
- *Controlli Automatici I e II*, appunti del corso, proff. A. Balestrino, A. Landi, a.a.96-97.
- *Sistemi di Elaborazione 1*, appunti del corso, prof. M.Avvenuti, a.a. 96-97.

RIVISTE SPECIALIZZATE:

- A. Sbrana, *La programmazione in assembler del PIC16C84*, Progetto Elektor, numm.: 1-2-3-4-5-6-7-8-9-10-11-12 anno 1996, Gruppo Editoriale JCE.
- M. Zamana, *Come realizzare i circuiti stampati*, Radiokit Elettronica, n.10 - ott.98, CMP Bologna.

MANUALI (SOFTWARE):

- *MPASM Assembler*, Quick reference guide, Microchip Technology 1994.
- *MPSIM Simulator*, User's guide, Microchip Technology 1995.
- E. Forsans, *Come usare 3DSTUDIO v.3.0*, McGraw-Hill Libri Italia s.r.l. - Milano 1994.

PRODOTTI SOFTWARE:

- *Autodesk 3D Studio*, release 3, Yost Group Inc. 1990-1993.
(CAD di simulazione della struttura meccanica e dell'ambiente)
- *Prog84*, software for programming PIC16C84, freeware.
(Software per programmare il PIC16C84)
- *Protel Electronics Design System (Advanced PCB)*, version 2.8, Protel Technology Inc., 1991-94.
(CAM di creazione del master per circuito stampato)
- *Advanced Schematic Capture*, version 2.04, Protel Technology Inc., 1992-1994.
(CAD di creazione dello schematic entry)
- *Electronics Workbench EDA*, version 5.0a, Interactive Image Technologies Ltd, 1992-1996.
(CAD di simulazione elettrica, dotato dello stesso nucleo di calcolo di SPICE, ma più interattivo)
- *CorelFLOW (TM)*, version 2.00.D2, Corel Corporation, 1992-1994.
(Editor grafico, per la creazione di descrizioni logiche, elettriche, diagrammi di flusso)
- *MPLab*, for Windows/16, 3.31.00, Microchip Technology Inc., 1997.
(Ambiente di sviluppo del software per il Microcontrollore)
- *MPASM*, v. 2.01, Microchip Technology Inc., 1996.
(Compilatore Assembler del PIC16C84)
- *MPSIM*, v. 5.0, Microchip Technology Inc., 1995.
(Simulatore Assembler del PIC16C84)
- *Wave Editor*, version 1.03 Keith W. Boone, 1992.
(Generatore di forme d'onda, per i test di verifica sul filtro audio)

DATA SHEET (HARDWARE):

- Catalogo di componenti industriali, *RS Components*, feb/ago 97.
- *TGS813* - for the detection of Combustible Gases, product information, Figaro USA inc., oct 1996.
- *TLV1543C/M* - 3.3-V 10-bit Analog-to-Digital converters with serial control and 11 analog inputs,
Texas Instruments Incorporated, march 1995.
- *TPIC6B595* - Power Logic 8-bit Shift Register, Texas Instruments Incorporated, july 1995.
- *PIC16C84* - 8-bit CMOS EEPROM Microcontroller, Microchip Technology Inc., 1993.
- *TPIC0298* (SGS L298) - Dual Full-H Driver, Texas Instruments Incorporated, 1990
- *238-9658* - Motoriduttore con ingranaggi in c.c., RS Components, 1998.
- *IS471F* - OPIC Light Detector, Sharp, 1998.

INDICE

1. INTRODUZIONE E FINALITA'	3
2. ANALISI DEL PROBLEMA	4
3. SCHEMATIZZAZIONE DEL PROBLEMA E DEFINIZIONE DELL' ARCHITETTURA PER RISOLVERLO.....	5
3.1. Le sorgenti di segnale.....	6
3.2. Il campo di esplorazione.....	8
3.3. La scelta dei trasduttori di misura.....	10
3.4. La scelta degli attuatori di movimento, e di segnalazione luminosa.....	12
3.5. Dimensionamento della struttura meccanica.....	13
3.6. Posizionamento dei trasduttori e strategia di ricerca.....	20
3.7. Controllore digitale o analogico ? Il PIC16C84.....	21
3.8. Dimensionamento del clock del microcontrollore, e della risoluzione di conversione.....	22
4. SCOMPOSIZIONE IN MODULI E LORO DEFINIZIONE.....	26
4.1 Definizione del bus di interconnessione tra i moduli.....	26
5. PROGETTAZIONE E SIMULAZIONE DEI SINGOLI MODULI.....	26
5.1 Dimensionamento e simulazione del filtro di Delyannis.....	28
5.2 Dimensionamento e simulazione del circuito di trattamento del suono.....	36
5.3 Dimensionamento e simulazione del circuito di luce.....	43
5.4 Il circuito del sensore di gas.....	46
5.5 Il circuito degli infrarossi.....	47
5.6 L'integrato per controllare i motori.....	48
5.7 Circuitistica dei trasduttori, con dati di alcuni componenti commerciali.....	49
5.8 Stima delle correnti totali assorbite, e durata delle batterie.....	53
6. VERIFICA DEI SINGOLI MODULI, REALIZZAZIONE DELLE SCHEDE ELETTRONICHE, MONTAGGIO DELLA STRUTTURA MECCANICA (di M.Cimino, E.Campanelli, A.Campana, P.Imbesi, L.Piccinini).....	54
6.1 Test su basetta mille fori.....	54
6.2 Realizzazione del circuito stampato mediante il CAM e montaggio delle schede.....	55
6.3 Composizione dei moduli, collegamenti elettrici tra schede, test di sistema finale.....	56
7. IL MICROCONTROLLO E L'ASSEMBLER.....	64
7.1 Decisione della strategia, suddivisione in moduli, traduzione in codice e simulazione.....	64
7.2 Subroutines per calcolare il tempo.....	64
7.3 Il protocollo di comunicazione con il convertitore A/D; il salto a molte alternative.....	66
7.4 La gestione dell'interfaccia parallela d'uscita.....	69
7.5 La subroutine per l'aggiramento degli ostacoli.....	70
7.6 La strategia di ricerca e segnalazione; subroutine assembler per trovare il massimo.....	73
7.7 Listato completo del codice assembler, e memoria di programma del microcontrollore....	75
7.8 Ulteriori migliorie possibili, con lo stesso hardware.....	86
BIBLIOGRAFIA.....	87
APPENDICE: Data Sheets.....	a0
A.1 Microcontrollore, PIC16C84.....	a1
A.2 Convertitore A/D, TLV1543C.....	a22
A.3 Interfaccia parallela d'uscita, TPIC6B595.....	a39
A.4 Driver per i motori, L298N.....	a48
A.5 Motori in c.c. con riduttore, RS238-9658.....	a55
A.6 Sensore di gas, TGS813.....	a62
A.7 Ricevitore di infrarossi, IS471F.....	a63