Consiglio Nazionale delle Ricerche

Nota Interna

Progetto e realizzazione del controllo dell'interfaccia gestuale PalmDriver basato su microcontrollore

S. Lunardi, M. Magrini, L. Tarabella, G. Bertini

B4-13 maggio 2006

ISTI-CNR

Consiglio Nazionale delle Ricerche

Nota Interna

Progetto e realizzazione del controllo dell'interfaccia gestuale PalmDriver basato su microcontrollore

S. Lunardi*, M. Magrini, L.Tarabella & G. Bertini

*Lavoro di stage per laurea I° livello in Ing. Elettronica , Univ. di Pisa

B4 - maggio 2006

ISTI-CNR

Progetto e realizzazione del controllo dell'interfaccia gestuale PalmDriver basato su microcontrollore

INDICE

PRESENTAZIONE

In questa nota sono descritti la realizzazione e il funzionamento dell'interfaccia gestuale PalmDriver, sviluppata presso il Laboratorio Segnali e Immagini dell'ISTI – CNR di Pisa, e il cui impiego riferito al settore della musica elettronica è quello di permettere nuove modalità di operare nelle performance dal vivo.

Più in dettaglio, il PalmDriver è un'interfaccia uomo – macchina preposta alla gestione di apparecchiature di sintesi audio, controllata mediante il movimento e la posizione assunta dalle mani dell'operatore; l'acquisizione delle movenze dell'utente, ad opera di *transceivers* all'infrarosso, permette di conferire nuova carica interpretativa ed emotiva a composizioni di musica elettronica più vicine a quelle della musica della tradizione.

Inoltre, per la peculiarità dell'interazione, il dispositivo si differenzia profondamente da altri dispositivi come tastiere o batterie elettroniche: il controllo, infatti, è effettuato con continuità semplicemente muovendo le mani, traslandole o ruotandole in modo indipendente, creando nello spettatore l'impressione che l'artista stia suonando qualche tipo di strumento invisibile.

L'effetto finale si deve principalmente all'efficace interpretazione (mapping) dei movimenti del *performer,* demandata ad un calcolatore su cui è in esecuzione un *software* specifico, mentre l'interfaccia in questione rappresenta l'anello intermedio tra i sensori e il sistema di sintesi del suono.

INTRODUZIONE

La musica è forse la forma d'arte più diffusa e apprezzata, quella che più facilmente riesce ad entrare in contatto con la sfera emotiva degli individui; per questa sua proprietà essa è insieme potente veicolo di contenuti e messaggi, capace di influenzare culture e tendenze fin dall'antichità. Nelle migliaia d'anni di storia della musica, innumerevoli artisti si sono avvicendati, esplorando le infinite combinazioni e sfumature consentite dai propri strumenti, e sviluppandone di nuovi qualora il messaggio (in senso lato) da trasmettere avesse richiesto diverse sonorità.

In tempi più recenti, con l'avvento dell'elettronica, nuove tipologie di strumenti musicali hanno rivoluzionato il modo di far musica: il segnale originale, ora trasformato in segnale elettrico, può essere alterato elettronicamente per restituire suoni profondamente differenti, con grandi potenzialità in termini espressivi; o ancora, ad un circuito elettronico sono demandate assieme la generazione e il condizionamento (in breve la sintesi) del suono. Acustica e meccanica cedono il posto all'elettronica e questa è anche in grado di generare suoni originali e modificarli a piacimento: la musica intesa in senso tradizionale diventa così musica elettronica.

Il binomio artista – strumento tipico della musica tradizionale cambia forma, e in particolare quest'ultimo si tramuta in uno strumento *esploso (bibliog. Tarabella)*, composto da un calcolatore e da un sintetizzatore di suoni, sui quali l'artista interviene per definire gli eventi del brano musicale. Tuttavia, lasciare ad un calcolatore l'intero processo d'esecuzione del brano musicale snatura in gran parte l'atto creativo, ridotto così alla sola composizione dello stesso: dopodiché, l'artista semplicemente cede il posto all'elettronica che si occupa di eseguire per lui la sequenza di eventi che si traduce in musica. E' palese da questa posizione capire che due calcolatori diversi eseguiranno la medesima sequenza indifferentemente, senza apportare estro o spessore interpretativo che solo un *performer umano* è capace di conferire.

Per recuperare allora la gestualità propria di chi suona strumenti convenzionali, numerosi metodi sono stati proposti, sia per replicare le movenze (e quindi l'emotività, le dinamiche, le espressioni) di un artista alle prese con un equivalente strumento tradizionale, sia per sviluppare nuovi modi d'interazione tra esecutore e brano musicale.

E' nell'ambito di quest'attività che è stato sviluppato alcuni anni fa il sistema *Twin Towers* (in bibliografia sono citate le realizzazioni precedenti) da cui il PalmDriver discende. Lo sviluppo della versione oggetto di questa nota ne rappresenta infatti l'ultima evoluzione, ottimizzata sia nei componenti hardware che nelle sue funzionalità: è basata su microcontrollore e su elettronica integrata è facilmente configurabile e dai consumi ridotti.

Vediamo allora di seguito alcune delle più conosciute realizzazioni di interfacce gestuali, in modo da comprenderne le principali problematiche comuni.

PRECEDENTI REALIZZAZIONI DI INTERFACCE GESTUALI

Nel novero delle interfacce uomo - macchina sviluppate nell'ambito della musica elettronica cui si accennava in precedenza, rientra a pieno titolo la creazione di Lev S. Termen (poi anglicizzato col nome di Leon Thermin), forse il primo esempio di questo genere: il *Theremin*. Realizzato in Russia nel 1919, sfruttava il principio del generatore di battimento, amplificando il tono differenza (udibile, ovviamente) tra due oscillatori ad alta frequenza, uno stabile ed uno comandabile muovendo le mani in prossimità di due antenne: con una si regolava l'altezza della nota, mentre con l'altra il volume. L'interazione era di tipo proporzionale, l'effetto finale variava uno ad uno con la posizione assunta dalle mani, e non si poteva configurare *on the fly*. Il suono ottenuto era ancestrale e monocorde; ciononostante non mancò di figurare in diverse produzioni cinematografiche[1] e di contare numerosi interpreti, dei quali la più celebre oggigiorno è sicuramente Lydia Kavina, discendente dello stesso Termen.

Un esempio più convenzionale è il *Radio Drum*[2] di Max Mathews: tramite una superficie di fili intrecciati che fungono da antenna, si ricava la posizione spaziale, nonché la velocità istantanea, di una coppia di bacchette (sulla cui estremità è installato un trasmettitore radio) del tutto simili a quelle usate per suonare una batteria o delle percussioni generiche. Chiaramente le informazioni provenienti dai sensori non sono sufficienti da sole a supportare l'artista nella sua interpretazione: a differenza dell'esempio precedente, allora, il *Radio Drum* necessita di un calcolatore per tradurre i segnali in comandi e parametri di esecuzione. Per garantire la massima efficacia del sistema è stato sviluppato

---

[1] Ricordiamo il film "*Spellbound*" di Alfred Hitchcock (USA,1945), noto in Italia col titolo "*Io ti salverò*".

[2] Il sito web di riferimento è: http://www.mistic.ece.uvic.ca/research/controllers/radiodrum/

anche un linguaggio software *ad hoc*, il MAX, con cui sfruttarne tutte le potenzialità espressive.

Un altro strumento, per certi versi simile alla realizzazione di Mathews, è il *Pianoforte Immaginario*, di cui un prototipo è stato realizzato nel 1997 al CNUCE - IEI di Pisa. Come suggerisce il nome, l'interprete sostanzialmente esegue dei brani musicali alla tastiera di un pianoforte fittizio, immaginario per l'appunto: il segreto consiste in una telecamera che, insieme ad un *personal computer*, si occupa di acquisire le informazioni circa la velocità e la posizione delle dita rispetto ad un piano di riferimento e di generare i suoni appropriati, mediante software dedicato. L'effetto scenografico è sorprendente[3], e lo spettatore non può che domandarsi dove sia nascosto lo strumento: come negli altri casi, l'efficacia dell'illusione è dovuta alla veloce risposta del sistema, che consente di evidenziare la relazione causa – effetto agli occhi del pubblico e quindi di dare realmente l'impressione che l'utente sia alle prese con un pianoforte vero.

Citiamo infine l'interfaccia *Twin Towers*[4], i cui prototipi sono stati sviluppati e realizzati presso il CNUCE – IEI di Pisa nei primi anni '90. Con l'ausilio del PC, il sistema in questione acquisisce informazioni relative a posizione e velocità delle mani dell'utilizzatore e le usa per la sintesi e la gestione dei suoni: tutto questo è reso possibile dall'uso di trasduttori IR, che sfruttando l'alta riflettività della pelle umana nei confronti dell'infrarosso vicino, consentono di ricavare la collocazione spaziale delle mani rispetto ad un riferimento. Il sistema si è poi evoluto, cambiando nome in PalmDriver (oggetto di questa nota) a seguito della strage dell'11 settembre 2001 e a cambiamenti dell'architettura che ne hanno migliorato le funzionalità rispetto a quelle iniziali.

---

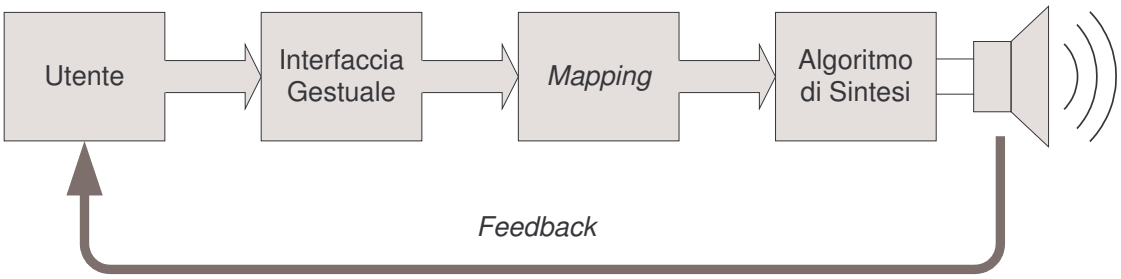
[3] Una breve clip video è disponibile all'indirizzo: http://cnuce.isti.cnr.it/tarabella/piano.mov

[4] Una demo è reperibile all'indirizzo: http://cnuce.isti.cnr.it/tarabella/TwinTowers.mov

PROBLEMATICHE COMUNI DI BASE

Dalle casistiche esaminate si può desumere l'evoluzione della ricerca nell'ambito della musica elettronica: mentre il *Theremin* produce un suono dipendente dalla posizione assoluta delle mani, introducendo con questa relazione un legame diretto tra la postura dell'artista e il risultato sonoro, gli altri esempi riportati (più recenti) sostanzialmente si occupano di carpire tutte le informazioni possibili sulla configurazione posizionale dell'utente e lasciano ad un calcolatore il compito di interpretarle e di usarle per la generazione o il controllo dei suoni.

Questo nuovo approccio, certamente più gravoso in termini di risorse impiegate, risulta molto più flessibile, in quanto la relazione causa – effetto non è più lineare (o almeno, non necessariamente) e può essere cambiata in ogni momento (paradigma detto *mapping*). Lo scotto da pagare, come accennato, è dato dalla necessità di affidarsi a soluzioni *hardware* e *software* più complesse e costose: da una parte sensori e controllori sempre più evoluti e sofisticati, dall'altra linguaggi dedicati, algoritmi che richiedono notevole potenza di calcolo e uso di protocolli specifici.



I sensori, elementi chiave che, di fatto, determinano l'architettura del sistema, assolvono il compito di ottenere le informazioni su posizione e attitudine di specifiche parti del corpo dell'interprete: essi generalmente forniscono segnali analogici, ossia variabili con continuità in analogia con le grandezze che sono chiamati a rilevare. Ne esistono, e ne sono stati impiegati, di diverse tipologie, quali sensori magnetici, capacitivi, resistivi, di trazione, di accelerazione, di inclinazione o all'infrarosso, per citarne alcuni; è però difficile pensare, per

queste applicazioni, a dei sensori puramente digitali, in quanto inadatti a cogliere, con i loro soli due stati possibili, le infinite sfumature che l'utente può riprodurre.

Nella pratica, comunque, si preferisce rinunciare alla possibilità di rappresentare tutte queste *infinite sfumature* (che non tutti sono poi in grado di cogliere), sia per le difficoltà intrinseche, sia perché, come abbiamo già avuto modo di spiegare, spetta tuttavia ad un calcolatore il compito di interpretarle e tradurle in comandi: poiché quest'ultimo non è in grado di manipolare direttamente segnali analogici, serve allora una traduzione del segnale da analogico a digitale, ovvero il passaggio da un segnale continuo, rappresentativo delle movenze del *performer*, ad un segnale digitale corrispondente che può essere elaborato tramite *computer*. Questa operazione, intuitivamente chiamata conversione analogico - digitale, non avviene a costo nullo e comporta necessariamente una perdita d'informazione per via dei finiti valori ottenibili; inoltre, il tempo richiesto per effettuare la conversione limita di fatto la massima frequenza con cui il segnale può essere campionato: pur trattandosi, nella fattispecie, di un segnale variabile molto lentamente (poche decine di Hertz al massimo), qualora i sensori impiegati siano parecchi la velocità di conversione può divenire critica. Il principale vincolo da rispettare è infatti dettato dal teorema di Shannon, che impone di campionare ad una frequenza doppia rispetto a quella massima del segnale, pena la perdita delle informazioni necessarie alla ricostruzione dello stesso. In altri termini, si ottengono distorsioni inaccettabili che compromettono il risultato, ovvero un controllo fallace, una risposta imprevedibile, un *feedback* non conforme.

La mole di dati generata dai sensori deve poi essere gestita da un calcolatore: nonostante la potenza di calcolo dei moderni PC sia più che sufficiente per i flussi considerati, raramente si usa affidare tutti i calcoli e il condizionamento dei segnali al *computer*. Poiché quest'ultimo deve anche eseguire il codice preposto alla traduzione e interpretazione dei dati (che può essere anche complesso), è preferibile, quando non inevitabile, impiegare dei microcontrollori che realizzino in *hardware* una prima parte del lavoro. Ciò può comprendere la

gestione dei sensori, il filtraggio digitale, la linearizzazione e compressione dei dati, l'indirizzamento di memorie esterne, la comunicazione seriale o parallela o anche l'implementazione di semplici interfacce utente (ad esempio, su *display* LCD). Questo sgrava il calcolatore centrale da una moltitudine di routine, riducendo i tempi d'esecuzione e comunicazione, conseguendo una maggiore efficacia (ed efficienza) dell'interazione e, nei sistemi più evoluti, una minima, o nulla, dipendenza dal PC.

La scelta del microcontrollore deve focalizzarsi su delle caratteristiche precise:

- una frequenza di lavoro adeguata: un microcontrollore veloce è opportuno qualora il numero di operazioni che deve svolgere sia elevato, oppure la temporizzazione degli eventi vada scandita con precisione;

- un numero di porte sufficienti: per porta, in questo contesto, si intende un gruppo di terminali (generalmente in numero multiplo di 2, spesso 4 o 8) facenti capo ad un unico registro, che rende possibile la scrittura o la lettura di parole binarie in parallelo. Nel caso di vincoli imposti sul costo del sistema o sulle sue dimensioni si preferisce ricorrere a tecniche di *multiplexing* o ad appositi *I/O expanders* e risparmiare sul microcontrollore;

- sottosistemi integrati: per evitare di complicare eccessivamente la circuiteria, è comodo disporre di microcontrollori che integrano periferiche importanti, come interfacce seriali, convertitori A/D, comparatori analogici, *controller* PWM, *timer* e altre più specifiche. Alcune sono dotazioni pressoché standard, di altre invece va tenuto conto per evitare di impiegare componenti le cui potenzialità sono esuberanti (come quindi il costo) rispetto alle richieste;

- reperibilità, costo e supporto software: chiaramente la differenziazione tra microcontrollori non consiste solamente nelle caratteristiche *hardware*, ma anche nella disponibilità nel mercato dei componenti, nel supporto della casa costruttrice in termini di assistenza e ambienti di sviluppo e, non da ultimo, nel costo unitario, che può essere sostenibile nel caso di pochi prototipi ma non esserlo per una eventuale produzione di serie.

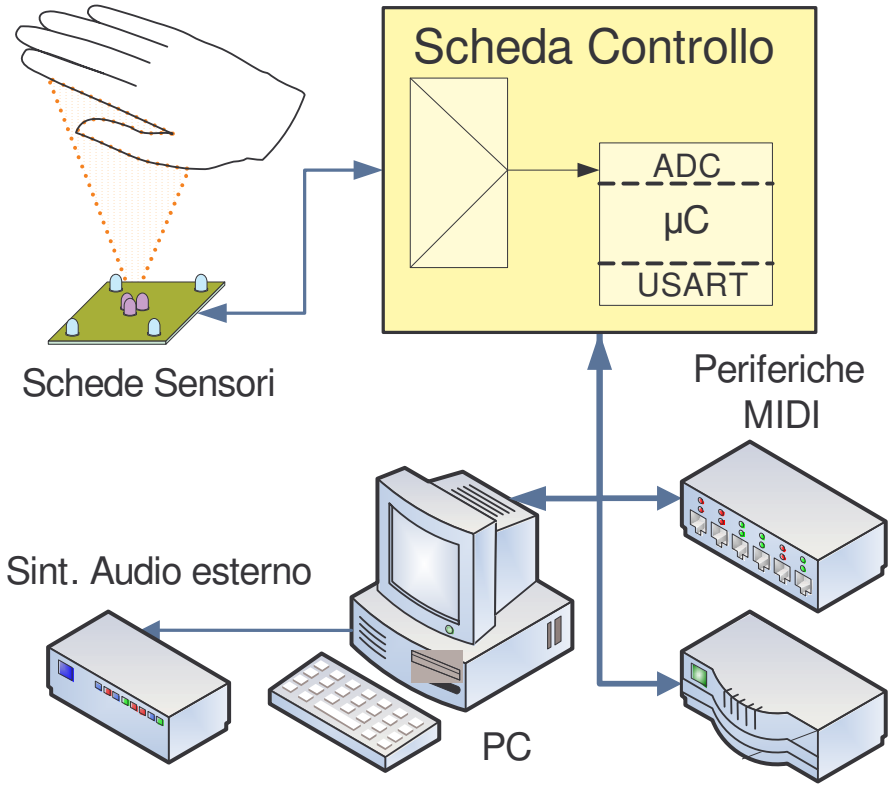
Fortunatamente lo sviluppo dei sistemi VLSI[5] ha reso disponibile una gran varietà di microcontrollori per ogni fascia di prezzo e prestazioni: per trovare quello più adatto alle specifiche è sufficiente una ricerca nei numerosi canali d'informazione disponibili nel mercato.

Il programma principale, quello eseguito dal PC ospite (in inglese *host*), è la chiave del *feedback psicoacustico*: è su di esso che l'artista deve intervenire per associare gli eventi ai comandi, e così tradurre le movenze in espressioni musicali. Riguardo ai protocolli per lo scambio di informazioni tra sistemi di computer music si può affermare che lo standard *de facto* è rappresentato sicuramente dal MIDI (*Musical Instruments Digital Interface*): si tratta di un formato ormai universale, con il quale periferiche musicali di ogni tipo possono dialogare tra loro o anche con un personal computer: la sua invenzione si è resa necessaria quando i PC hanno cominciato ad entrare nelle case, e i musicisti più innovativi hanno compreso la potenzialità del sistema in ambito musicale. Il protocollo MIDI definisce sia le specifiche dal punto di vista dell'elettronica d'interfaccia che quelle di comunicazione, stabilendo i numerosi parametri e i comandi da impiegare. Lo scambio d'informazioni MIDI si basa su circuiti accoppiati con optoisolatori, comandati in corrente, sui quali sono inviati due o tre byte tra dati e comandi di sistema, secondo le esigenze; la trasmissione è seriale e avviene ad una velocità di 31250 bit per secondo. Si deve evidenziare però che le informazioni MIDI non hanno niente a che vedere con la forma d'onda del brano musicale: esse riguardano invece gli *eventi* che compongono il brano stesso, come l'istante di attacco l'attacco o di decadimento di una nota, il tipo di strumento, il numero di canale e *controller*, volume, *velocity*, *panning* e tanti altri; in altre parole, il MIDI descrive la musica similmente ad uno spartito, ma non contiene (in genere) dati audio veri e propri, risultando così molto leggero in termini di scambio d'informazioni.

[5] *Very Large Scale of Integration*: si riferisce al totale di porte logiche equivalenti integrate su un unico chip. Per i sistemi VLSI il numero varia da 100000 a 1000000.

DESCRIZIONE DEL SISTEMA

Il PalmDriver si compone di diverse parti che dialogano tra loro per ottenere il risultato sonoro finale (come visibile in figura): per questo motivo il sistema completo viene indicato con la dicitura di strumento *esploso* (da una citazione di Tarabella*)*, per differenziarlo dagli strumenti "classici" che riuniscono in un unico insieme controllore e generatore del suono, e le cui caratteristiche timbriche sono imposte dalle combinazioni dei materiali costituenti, dalle geometrie e dalle meccaniche.



Nel PalmDriver distinguiamo almeno tre elementi principali:

- l'*array* dei sensori IR, col quale l'utente interagisce; si compongono di 32 fotosensori disposti a gruppi (moduli) di 4, in maniera tale da rilevare altezza media dal piano e inclinazione delle mani sugli assi. I moduli sensori sono poi assemblati a coppie, sfruttando lo stesso connettore a 15 poli e costituendo così delle unità facilmente posizionabili e gestibili;

- la scheda di controllo, basata su microcontrollore, col compito di gestire i segnali provenienti dai moduli; in particolare, detti segnali vengono convertiti in dati digitali tramite ADC integrato e trasmessi secondo lo standard MIDI all'*host* PC (il formato utilizzato per la trasmissione dei dati, pur utilizzando la convenzione *status byte* + 2 *data byte*, è stato deciso *ad hoc* e viene discusso più avanti nella nota);

- un personal computer, sul quale è in esecuzione un software di interpretazione dedicato: il suo compito è ricevere i dati provenienti dalla scheda controllo e, in base alla mappatura dei comandi impostata e ai movimenti effettuati dall'artista, generare comandi MIDI opportuni per la sintesi o la manipolazione del suono.

Il suono può essere generato dal PC stesso (tramite la scheda audio) oppure mediante sintetizzatori MIDI esterni. E' evidente che, tramite il software di controllo, il sistema acquisisce estrema flessibilità d'utilizzo, nel tipo di mappatura (impostabile anche *on the fly* dall'utente), nel numero e nella natura dei comandi MIDI generabili, rendendosi quindi adatto a molteplici esigenze.

Specifiche di progetto

Il sistema oggetto della presente nota rappresenta, come già anticipato, l'ultima evoluzione dell'interfaccia *Twin Towers*, di cui si è provveduto a ridisegnare completamente la scheda di controllo, avvalendosi ora dell'uso di un PIC ad alte prestazioni che consentisse di semplificare considerevolmente la circuiteria elettronica di contorno, soppiantando il precedente DSP con la relativa scheda DSK[6]. Questo implica mantenere le stesse specifiche del sistema precedente, e migliorarle ulteriormente laddove si richiedano prestazioni più elevate.

[6] DSK: *Developer Starter Kit*, acronimo che indica un *set* di *hardware* e *software* atti ad offrire un primo ambiente di sviluppo per il componente in oggetto; sono acquistabili, quando disponibili, presso il produttore.

Le specifiche, oltre ai vincoli di cui sopra, impongono in particolare:

- la possibilità di utilizzare fino a 8 moduli sensori, che implica l'acquisizione periodica dei valori di tensione da 32 fototransistori e l'emissione impulsiva su 8 LED all'infrarosso per l'illuminazione delle mani del *performer*;

- la facile trasportabilità del sistema, quindi peso e dimensioni contenuti;

- la possibilità di essere alimentato con un normale alimentatore *general purpose* disponibile in commercio, implicando così un modesto consumo di potenza e utilizzo di tensioni unipolari di facile impostazione;

- l'utilizzo del protocollo MIDI per la comunicazione con altre periferiche musicali o con *PC host*, sia per quanto riguarda il formato dell'informazione che l'elettronica di interfaccia;

- la velocità di scansione deve essere più che sufficiente a cogliere i movimenti, anche i più repentini, dell'interprete: in questo senso si può stimare la velocità massima del movimento delle mani in circa 15 Hertz ; per il teorema di Shannon già citato nel primo capitolo, si impone una frequenza di scansione minima del sistema pari ad almeno 30 Hz;

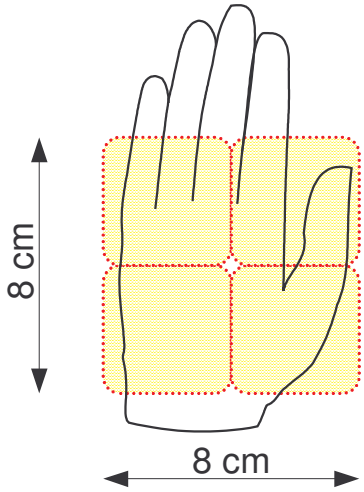
- una sensibilità adeguata alla misura da effettuare: imposto un range per la distanza delle mani dal piano dai 10 ai 50 centimetri, si ritiene accettabile un errore massimo di 2 millimetri.

Per quest'ultimo punto va osservato come l'errore massimo ammesso non discenda, in realtà, da un calcolo preciso, quanto da un effettivo riscontro empirico ottenuto dall'uso dell'interfaccia. Un esame rigoroso dovrebbe considerare le geometrie dei fasci luminosi e le intensità dei lobi di emissione e ricezione, e prevedere l'influenza di diverse variabili più o meno aleatorie: tutto questo è però relativamente superfluo, in quanto l'interfaccia gestuale, come già illustrato, è solo un anello della catena che coinvolge il *feedback* psicoacustico; in altre parole, si può affermare che è la precisione differenziale, e non quella assoluta, quella che conta, perché è quella che si può realmente percepire e che risulta determinante nell'efficacia del sistema.

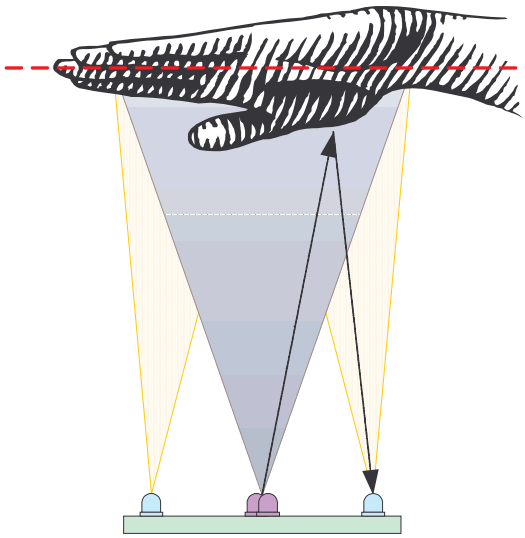
I moduli sensori

Il sistema utilizza un insieme di sensori che rilevano l'intensità della radiazione infrarossa riflessa dalle mani dell'utente, previa "illuminazione" delle stesse tramite lampi di luce IR generati da LED appositi. La pre – illuminazione si è resa necessaria per innalzare il rapporto segnale – rumore del sistema, in quanto la sola emissione infrarossa da parte delle mani è insufficiente a fornire un adeguato segnale di ritorno. L'intensità della radiazione riflessa, allora, aumenta al diminuire della distanza delle mani dal piano dei sensori, secondo una relazione certamente non lineare e difficilmente prevedibile in maniera esatta per via delle numerose variabili più o meno aleatorie che incidono sull'entità del segnale di ritorno.

Mano, vista superiore



Mano, vista laterale



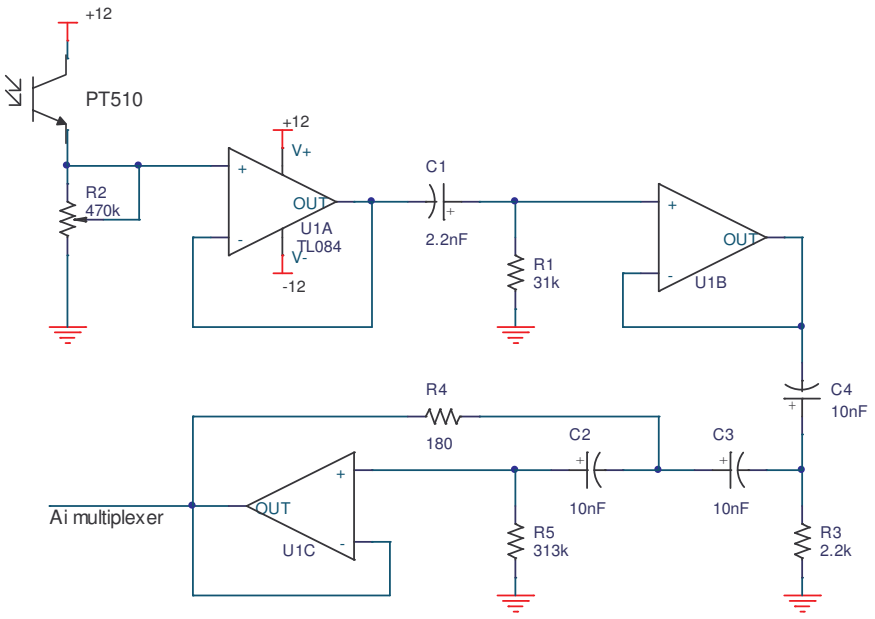
Si osservi che, mentre la non linearità della relazione altezza – intensità può essere corretta con ottima approssimazione tramite opportuni circuiti o algoritmi *software*, correggere completamente l'effetto delle altre variabili in gioco è molto più arduo. Queste variabili infatti, come la non costante riflettività della pelle, la luminosità ambiente, la dispersione delle caratteristiche dei sensori, non sono esattamente quantificabili, variano nel tempo e da persona a persona: al fine

allora di restringere il campo di imprevedibilità dello strumento si rende necessaria una procedura di taratura.

La taratura dello strumento consente di ritrovare ogni volta la prevista relazione causa – effetto, ma poiché diversi fattori variano nel tempo, neanche questa rappresenta una procedura esatta. Ma del resto si tratta di uno strumento musicale, e come tale soffre come i suoi simili di certe limitazioni che l'artista impara a conoscere e a sfruttare a proprio vantaggio; come inoltre si accennava al precedente paragrafo, data la natura del controllo, non è rilevante la precisione assoluta ma quella differenziale, ovvero la capacità dello strumento di rispondere ai movimenti delle mani in maniera plausibile piuttosto che riconoscere determinate posture con gran precisione. L'utente, infatti, non interagisce con lo strumento mediante sequenze di posture rigide diverse, ma piuttosto con una serie di movimenti coordinati e continui che richiamano anche visivamente il controllo voluto, come se si manipolasse fisicamente il suono: non la sensibilità sulla posizione assoluta allora, ma la sensibilità allo spostamento, da cui l'espressione "*touching the sound*".

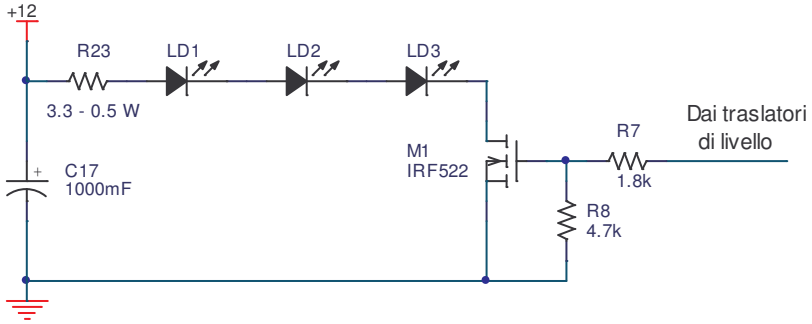
I moduli sensori ssono composti da insiemi di fotosensori e da LED emettitori all'infrarosso, i primi dei quali con il compito di rilevare l'intensità della radiazione IR emessa dai secondi e riflessa dalle mani dell'utente. Un modulo, in particolare, si compone di:

- quattro fototransistori PT510, disposti ai vertici di un quadrato di lato 8 cm, in modo da rilevare, oltre all'altezza media, anche l'inclinazione delle mani su due assi e quindi *mappare* altri due gradi di libertà; il segnale proveniente da ciascun sensore viene quindi filtrato tramite un filtro passa alto di Chebyshev del 4° ordine (realizzato per mezzo di un amplificatore operazionale quadruplo a FET TL084) per reiettare la componente continua relativa alla luminosità ambientale ed alla polarizzazione del fototransistore (è possibile intervenire manualmente su quest'ultima agendo sul potenziometro R2, in modo da correggere asimmetrie circuitali imputabili in genere alla dispersione delle caratteristiche nei dispositivi a semiconduttore);nella figura seguente è visibile il circuito relativo ad un sensore:



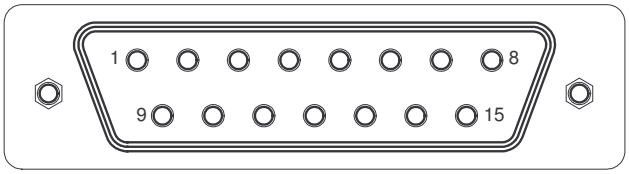
- tre LED IR SFH485 in serie, collocati al centro del suddetto quadrato, pilotati impulsivamente tramite un MOSFET di potenza IRF522. Non è stato possibile fornire un'illuminazione costante alle mani dell'utente in quanto la bassa potenza dissipabile dai LED avrebbe imposto un livello d'irradiazione troppo basso per il pratico utilizzo: tramite pilotaggio impulsivo allora è possibile innalzare la potenza radiante istantanea scongiurando il pericolo di bruciare i diodi.

Le soluzioni fin qui esposte sono state asviluppate nelle precedenti realizzazioni l'unica variante introdotta riguarda l'uso tre elementi radianti connessi in serie (quindi pilotati dalla medesima corrente) invece di uno solo. Il valore della corrente circolante nei LED (e quindi la luminosità del *flash*) è imposto dal valore della tensione di *gate* del MOSFET, ottenuta tramite partizione resistiva degli impulsi di comando, a 12 volt. La coppia LED – fototransistore, infine, è



stata scelta per l'ottima corrispondenza tra i diagrammi di irradiazione e sensibilità. Chiaramente si è ottenuto una migliore resa nel segnale riflesso consentendo una maggiore escursione in altezza della posizione delle mani rispetto alle precedenti realizzazioni semplicemente riducendo opportunamente il valore della resistenza in serie ai tre diodi.

I moduli così costituiti sono assemblati in gruppi di due, condividendo chiaramente le tensioni di alimentazione e il riferimento di massa, e connessi alla scheda di controllo tramite connettore Canon a 15 poli:



1: sensore 1, modulo 1 (rosso)	5: massa (nero)	9: sensore 1, modulo 2 (rosso)	13: massa (nero)
2: sensore 2, modulo 1 (giallo)	6: massa (nero)	10: sensore 2, modulo 2 (giallo)	14: – V_{DD} (verde)
3: sensore 3, modulo 1 (verde)	7: V_{DD} (rosso)	11: sensore 3, modulo 2 (verde)	15: comando flash, modulo 2 (rosso)
4: sensore 4, modulo 1 (bianco)	8: comando flash, modulo 1 (rosso)	12: sensore 4, modulo 2 (blu)	

La scheda controllo

La scheda di controllo gestisce il flusso dei segnali provenienti dai sensori, li converte in dati digitali a 10 bit e, previa formattazione, li invia tramite porta MIDI ad un *personal computer* sul quale è in esecuzione un programma di interpretazione dedicato.

Tutto il design è incentrato su un moderno microcontrollore ad alte prestazioni, un PIC 16F877A della Microchip, Inc., il quale incorpora un SAR ADC a 10 bit per la conversione analogico – digitale dei segnali, riducendo così la componentistica esterna e i problemi di *setup*. L'ADC integrato dispone di fino a 8 canali di ingresso commutabili: tuttavia la multiplazione dei segnali provenienti dai sensori è affidata a due multiplexer analogici esterni (MC14067) a 16 ingressi, indirizzati dallo stesso PIC in maniera mutuamente esclusiva. Questa soluzione comporta due importanti vantaggi:

- in primo luogo, aumenta il numero di segnali convogliabili all'ADC, che arrivano così a 32 (per un totale di 8 moduli sensori);

- inoltre, è possibile in questo modo sfruttare i piedini RA2 e RA3 del microcontrollore sui quali è possibile impostare la dinamica di conversione dell'ADC, e sfruttare in tal modo tutta la risoluzione (ossia la sensibilità) possibile.

Gli altri componenti della scheda comprendono un integrato del tipo 7805 per ricavare la tensione di alimentazione a 5 volt per il microcontrollore e la logica integrata, due traslatori di livello (MC14504) per il pilotaggio dei LED IR, un *inverter* (74HC14) per l'implementazione della porta MIDI e il pilotaggio dei multiplexer e infine un convertitore DC/DC TEN5 TRACO per ricavare la tensione di -12 volt (necessaria ai moduli sensori) dalla tensione unica di alimentazione (a 12 volt).

Inoltre, sono inclusi alcuni componenti passivi, come i potenziometri per fissare la dinamica dell'ADC, il cristallo di quarzo a 20 MHz, un fusibile da 500mA, un

pulsante di *reset* e un commutatore la cui funzione verrà discussa a proposito del *firmware* scritto per il microcontrollore.

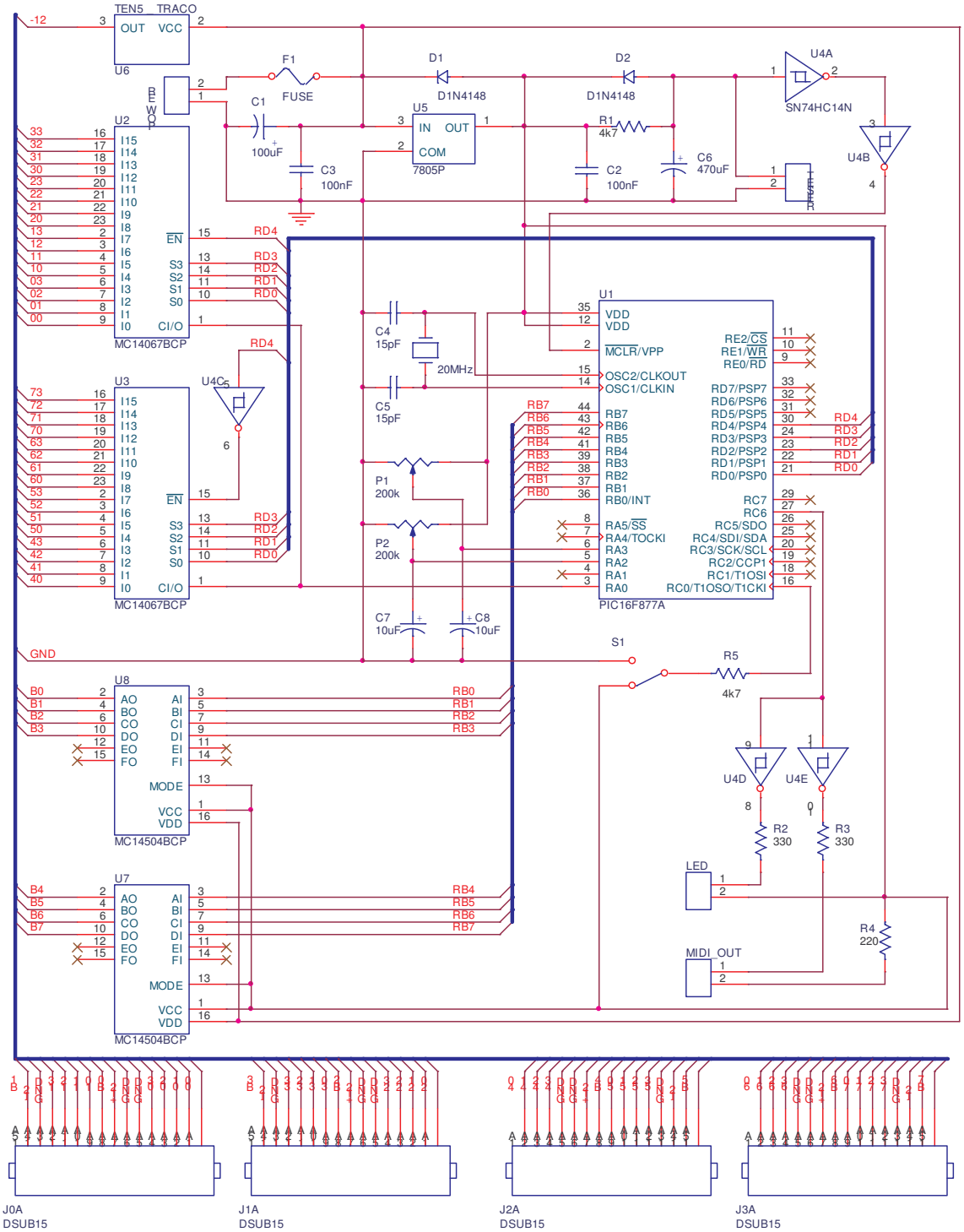
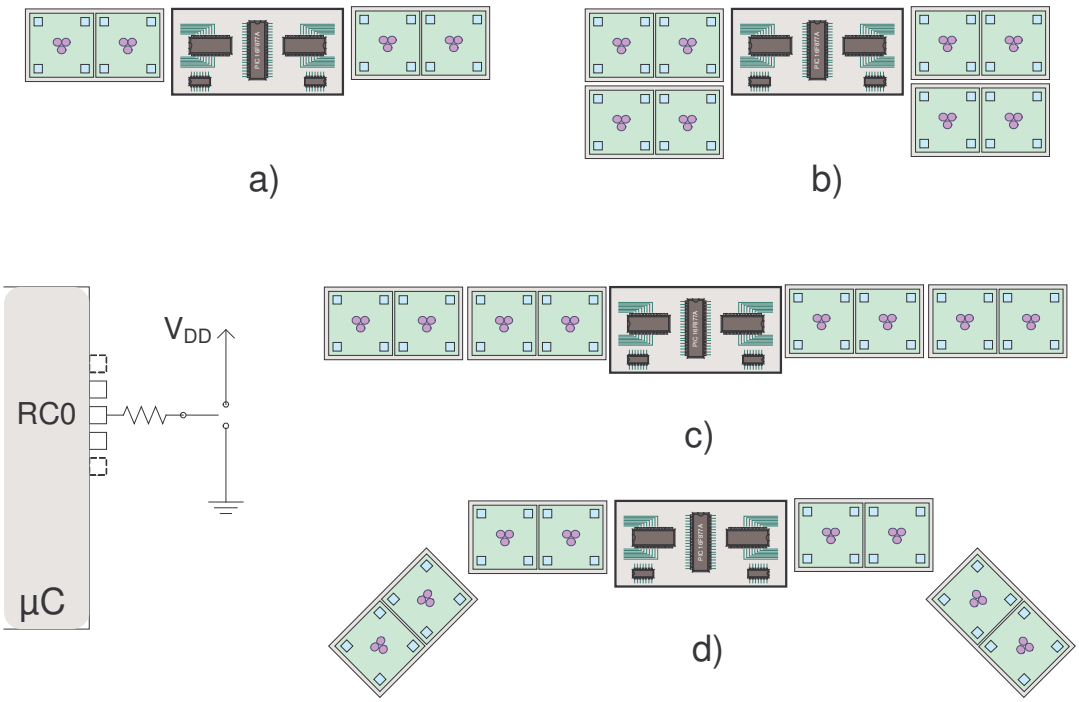


Figura 1: schema circuitale della scheda di controllo.

Utilizzo dell'interfaccia

Vengono discusse ora le modalità di utilizzo e di *setup* del sistema: anzitutto si presuppone che si disponga di un generico alimentatore capace di fornire una tensione continua di 12 volt e almeno mezzo àmpere, e che il sistema sia connesso ai moduli sensori e al PC tramite cavo MIDI adeguato.

Il programma principale prevede la possibilità di configurare la scheda controllo per la scansione di quattro oppure otto moduli: la specifica discende dal fatto che per alcune esibizioni più semplici, oppure semplicemente laddove non siano necessari più moduli, è possibile risparmiare in ingombro e peso, e guadagnare inoltre in velocità di scansione. La scelta della configurazione (vedi figura) avviene per mezzo di un semplice deviatore, disponibile esternamente per essere operato dall'utente, collegato al *pin* RC0 del microcontrollore: quest'ultimo, in base al valore letto, imposta la procedura di scansione desiderata; per un corretto utilizzo la configurazione va eseguita a sistema spento oppure in seguito a *reset* dello stesso.



Accenniamo qui alle diverse disposizioni verosimili dei moduli, che l'utente avrà modo di impostare secondo il tipo di *performance* o di controllo che vorrà

attuare; ogni configurazione chiaramente porta con sé pregi e difetti e l'artista sceglierà quella più opportuna in base anche a preferenze personali e alla propria sensibilità: per fare un paragone, allo stesso modo in cui un batterista decide il posizionamento di fusti e piatti del proprio strumento. Le configurazioni già visibili in figura sono essenzialmente tre:

- la disposizione LINE (a, c) è forse la più ovvia, e permette una netta separazione dei parametri controllati dalla mano destra da quelli della mano sinistra. Nella versione "*full*" con 8 moduli è possibile un'accurata rilevazione di scorrimenti delle mani da un capo all'altro oppure dall'esterno verso l'interno e viceversa;

- la disposizione MTX (b) può apparire scomoda, e in effetti è più difficile discriminare tra i sensori anteriori e quelli posteriori durante l'esibizione: tuttavia permette di esercitare il controllo agendo su due direzioni contemporaneamente, in altre parole l'utente guadagna un ulteriore grado di libertà disponendo di un intero piano orizzontale in cui muoversi;

- la disposizione MULTI (d) permette un ottimo controllo della sintesi in quanto all'utente si presenta una consolle centrale e due satelliti laterali (una configurazione per certi versi simile ad un piccolo *set* di percussioni), che può suonare molto comodamente e gestire come meglio crede.

L'elenco non è comunque esaustivo: dal momento che i collegamenti coi moduli si possono effettuare con delle comuni 'piattine', l'utente può liberamente posizionare gli elementi sensibili in configurazioni originali (purché funzionali) che più rispondono alle sue esigenze.

Per il *reset* del microcontrollore (e quindi dell'intero sistema) si è realizzata una piccola interfaccia con un pulsante che l'utente può premere in qualsiasi momento, credibilmente per cambiare la modalità di scansione del programma.

Un LED fornisce inoltre un *echo* visivo del traffico MIDI, con cui l'utente può verificare qualitativamente il dialogo con il PC *host*.

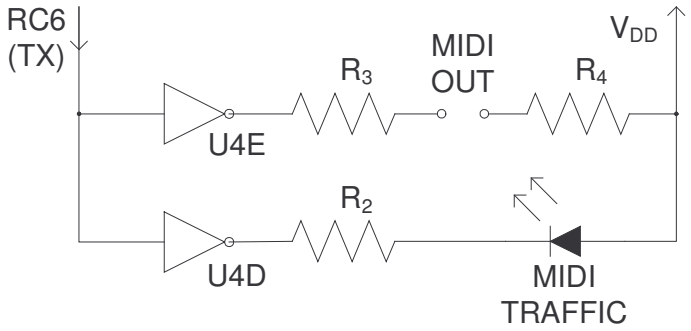
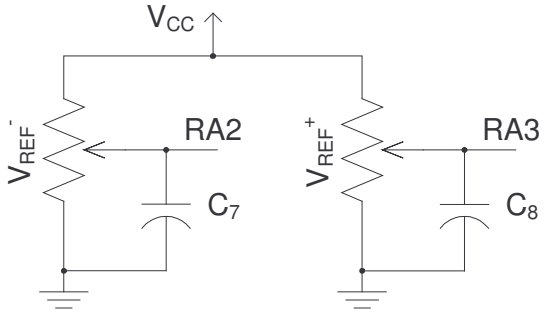


Figura 3: circuito di *output* MIDI e segnalazione luminosa.

Infine, nella scheda controllo sono stati inseriti due potenziometri (*trimmer* multigiro) da 200 kΩ tra la linea a +5 volt e la massa, con lo scopo di fornire al convertitore A/D le tensioni di riferimento superiore e inferiore e quindi stabilire la dinamica di ingresso in base ai segnali da convertire; per far questo il cursore dei potenziometri è stato connesso rispettivamente ai terminali RA3 e RA2, con un condensatore ciascuno per filtrare disturbi provenienti dall'alimentazione. Data la crucialità della funzione, sono stati collocati in maniera da non essere direttamente accessibili dall'esterno, in modo da evitare lo sconvolgimento della taratura per via di operazioni involontarie o improprie; lo schema è visibile nella figura seguente:



Descrizione delle routines software

Il *software* scritto per il microcontrollore, poiché non è direttamente accessibile dall'utente, prende il nome di *firmware*, ed è quello che permette al PIC di gestire il flusso delle informazioni in entrata ed in uscita dalla scheda controllo.

Fondamentalmente, il *firmware* svolge quattro operazioni di base:

- seleziona la coppia sensore/modulo di interesse generando un opportuno indirizzo per i multiplexer analogici esterni;

- pilota i LED IR del modulo corrente in modo da emettere un lampo della durata di 20 µs (durata dettata dal compromesso già discusso in precedenza a proposito della potenza massima dei LED IR);

- a questo punto l'ADC acquisisce e converte il valore di picco del segnale in dato digitale;

- successivamente vengono svolte alcune operazioni sul dato:

    - viene confrontato con la media degli ultimi due campioni per verificare se ci sono state variazioni (movimento); intuitivamente, è inutile trasmettere dati invarianti nel tempo e saturare così il canale MIDI, soprattutto in virtù della natura differenziale del controllo desiderato;

    - se sussiste variazione del dato, viene formattato nei tre byte previsti dalla trasmissione MIDI e spedito al PC per l'interpretazione;

  la sequenza viene così conclusa e il ciclo riprende da capo con un nuovo indirizzo per i multiplexer, finché tutti i sensori non sono stati interrogati. Non è possibile interrompere la sequenza delle operazioni se non con lo spegnimento della macchina o, temporaneamente, tramite il reset della stessa: gli unici input forniti dall'utente sono infatti i dati che la macchina acquisisce sulla posizione delle mani, tuttalpiù nulli, ma niente che porti il sistema in uno stato indefinito non previsto per il quale la sequenza si interrompa.

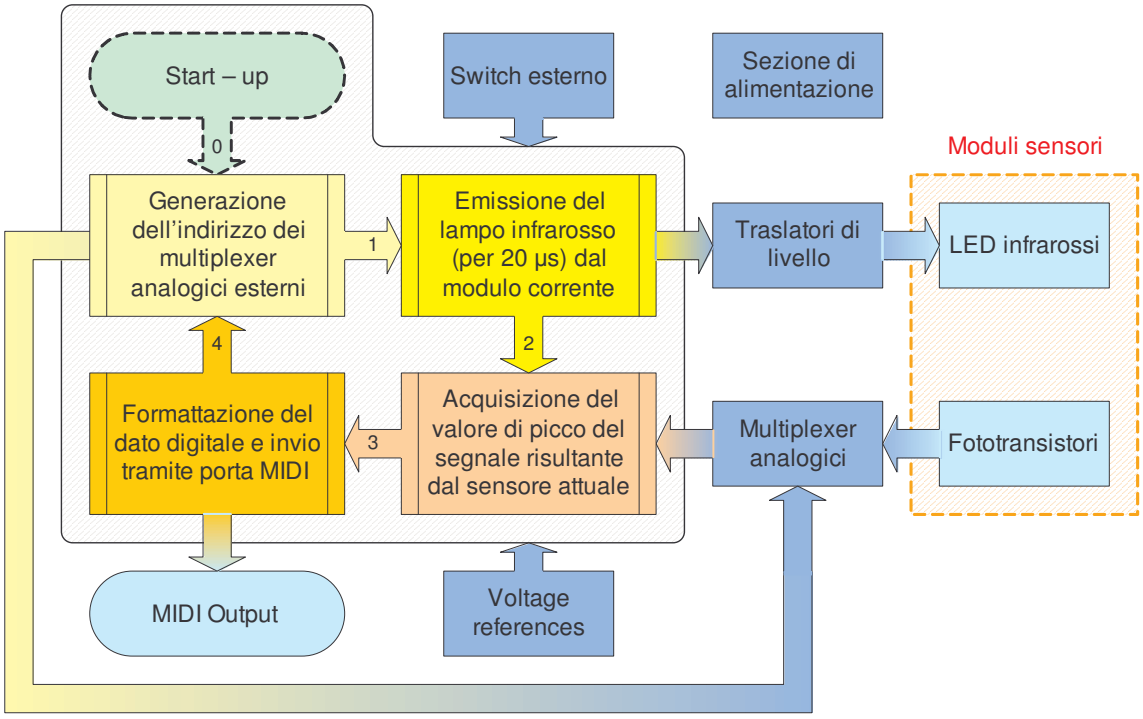


Figura 4: diagramma di flusso delle operazioni base costituenti il *loop* principale del *firmware*, evidenziante le relazioni coi componenti *hardware* esterni.

Un deviatore connesso al *pin* RC0 del PIC consente di presentarvi in ingresso un livello logico 0 oppure 1, a seconda che venga commutato a massa oppure alla linea di alimentazione a 5 volt. In questo modo è possibile scegliere la modalità di scansione dei sensori: mentre un uno logico consente la scansione di tutti gli 8 moduli, uno zero logico sul *pin* RC0 configura il *software* per l'interrogazione di quattro moduli, guadagnando in velocità di campionamento.

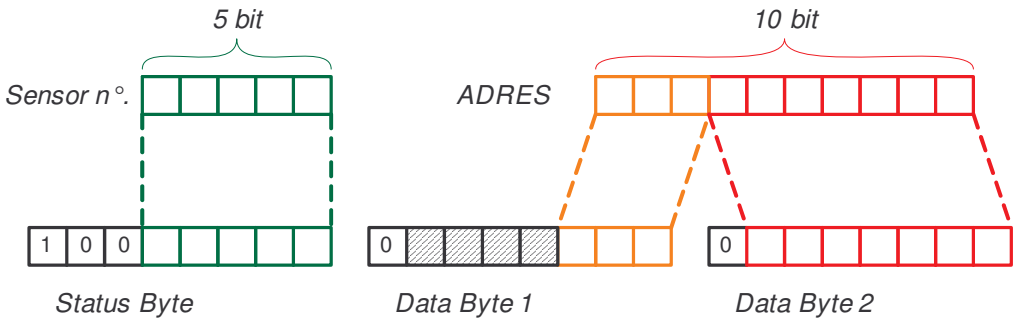
Il formato con cui il dato digitale (a 10 bit) è suddiviso nel pacchetto in uscita, formato da tre byte, non è standard ma è stato stabilito in sede di scrittura del *firmware* per il microcontrollore: del protocollo MIDI è stato rispettato il formato *status byte* + 2 *data bytes*, ma il significato del contenuto di questi è arbitrario. I dati sono infatti rivolti all'interpretazione da parte del *software* proprietario in esecuzione sul *personal computer*, che ovviamente è programmato per estrarre le informazioni utili dal pacchetto in maniera coerente.

Il formato segue due regole generali:

- il primo byte, essendo uno *status byte*, presenta il MSB pari a uno, mentre i seguenti due *data bytes* hanno il MSB pari a zero, come da specifiche MIDI;

- nello *status byte* viene inserito il numero di sensore (0..31) allineato a destra (cinque bit meno significativi), e nei due *data bytes* viene inserito il dato digitale (allineato a destra).

Nella figura che segue è visibile uno schema della formattazione:



Si noti la variabile ADRES che contiene il risultato della conversione A/D (che viene implementata internamente al microcontrollore tramite due registri, ADRESL e ADRESH; il dato può essere formattato a destra o sinistra tramite il bit ADFM); i tre MSB dello *status byte*, pari a 100, indicherebbero un comando del tipo NOTE_ON oppure NOTE_OFF, ma questo è irrilevante dal momento che il *software* di interpretazione cui il pacchetto è destinato è programmato in maniera coerente per estrarne le informazioni utili.

APPENDICE 1: IL MICROCONTROLLORE PIC16F877A

Il cuore del sistema, come discusso nel corpo della nota, è il PIC 16F877A prodotto dall'americana Microchip, un microcontrollore RISC evoluto realizzato con tecnologia CMOS che integra diverse periferiche, tra cui:

- tre *timer*, di cui uno a 16 bit, dotati di *prescaler*;
- un convertitore A/D a 10 bit con *sample & hold* e multiplexer analogico interno a 8 canali, oggetto della successiva appendice;
- un comparatore analogico;
- due moduli PWM a 10 bit per il controllo di dispositivi di potenza;
- porte USART (*Universal Synchronous Asynchronous Receiver Transmitter*) e SSP (*Synchronous Serial Port*) con SPI e I^2C per la comunicazione seriale e PSP (*Parallel Slave Port*) per la comunicazione parallela con altri processori o periferiche.

Tra le caratteristiche principali si segnalano:

- 368 byte di RAM, 256 byte di E^2PROM per i dati persistenti e 8K x 14 bit di memoria FLASH per il codice;
- 5 porte di *input*/*output*, di cui una configurabile come ingresso analogico, che possono erogare o assorbire fino a 25 mA di corrente da un qualsiasi piedino e fino a 200 mA complessivamente;
- un *set* ridotto di 35 istruzioni (per lo più a singolo ciclo di *clock*), 14 sorgenti di *interrupt*, uno *stack hardware* a 8 livelli e fino a 20 MHz di velocità operativa;
- *Power On Reset*, *Brown Out Reset*, *Oscillator Start Up Timer*, *Watchdog Timer* (con oscillatore RC indipendente) e fusibili di programmazione assicurano condizioni operative ottimali evitando corruzione dei dati e del *software*;
- la possibilità di programmare il microcontrollore direttamente nel circuito di destinazione tramite pochi componenti aggiuntivi (ICSP);

- alimentazione a 5 volt, SLEEP *mode* per consumi più ridotti, potenza massima dissipabile di 1 watt e disponibilità in package PDIP, PLCC e QFP ne fanno un integrato relativamente poco esigente in termini energetici e di dimensioni.

Un ulteriore valore aggiunto è costituito dall'ampia *knowledge base* disponibile su tutto ciò che riguarda i PIC, ossia la notevole mole di materiale didattico, *tutorials*, ambienti di sviluppo *software* e *hardware*, nonché il supporto *online* della casa costruttrice: il *know – how*, l'esperienza acquisita e resa liberamente disponibile è impagabile nello sviluppo di nuovi sistemi che sfruttano la medesima tecnologia.

E' evidente, insomma, la notevole dotazione *hardware* del PIC; ciò nonostante, per gli scopi di questo progetto solo alcune periferiche sono state sfruttate, direttamente o indirettamente, e verranno di seguito descritte le tecniche di impiego e le problematiche affrontate.

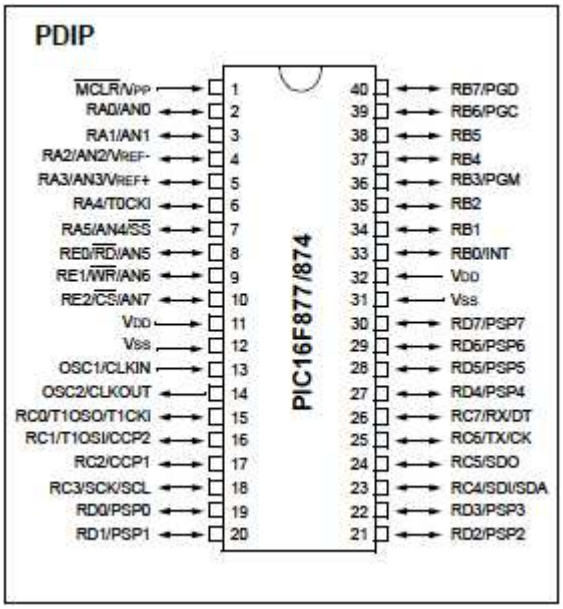


Figura 5: *Pinout* del PIC 16F877A in formato PDIP.

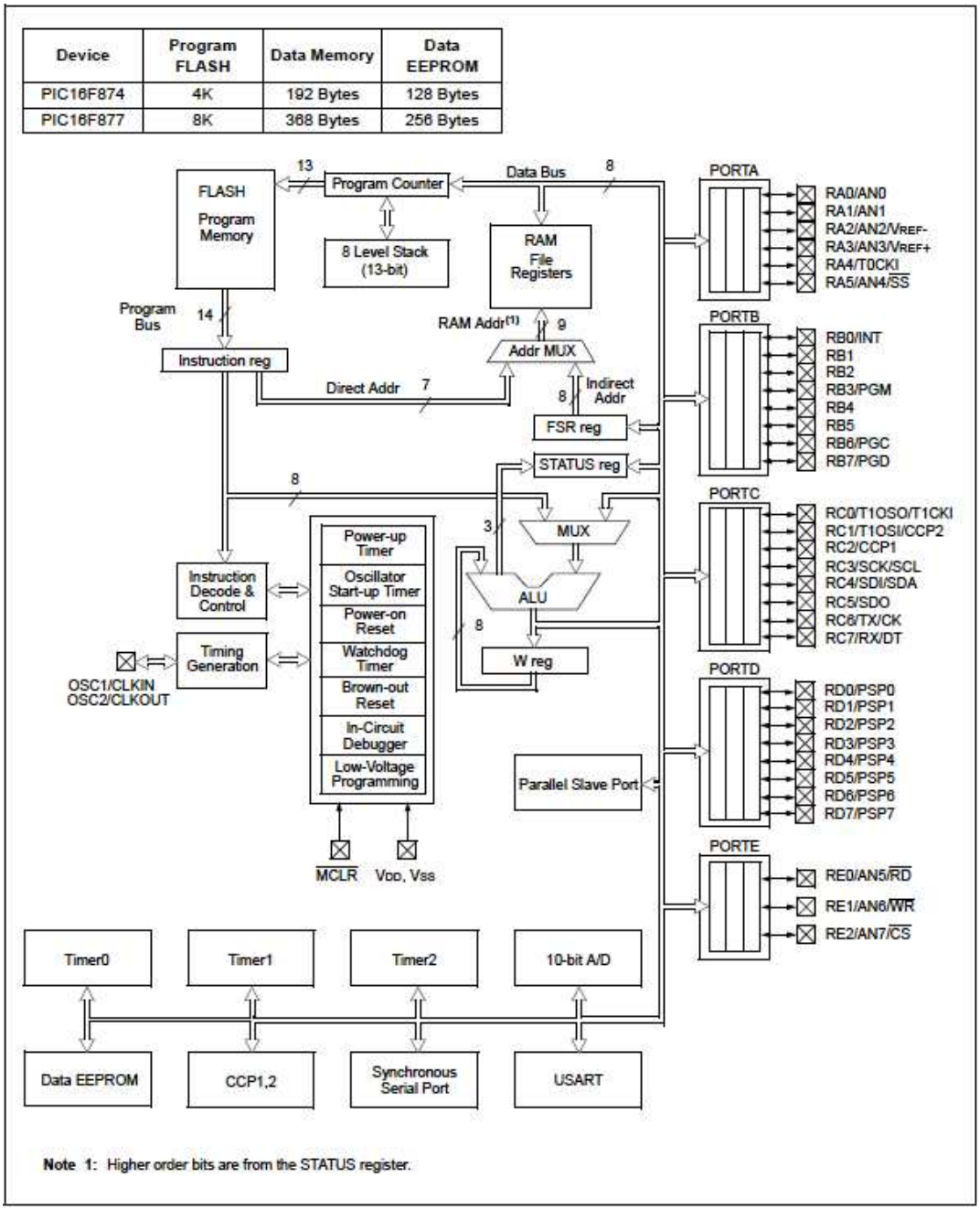


Figura 6: schema a blocchi del microcontrollore. E' evidente l'organizzazione interna tipica dell'architettura Harvard, con memoria programma e memoria dati separate e con bus dedicati; in basso sono visibili le periferiche integrate. Si notino inoltre il *program bus* a 14 bit e l'insolita suddivisione di terminali tra la porta A e la porta E. Particolare attenzione va posta, in fase di progetto, alle duplici, quando non triplici, funzioni della gran maggioranza dei terminali di I/O, per evitare di impegnare terminali facenti capo a periferiche altrimenti utili per altri scopi.

Il convertitore A/D integrato nel microcontrollore è trattato nell'appendice specifica; l'altra periferica di cui si è fatto uso è l'**USART** (*Universal Synchronous Asynchronous Receiver Transmitter*), ossia la porta di trasmissione seriale integrata conosciuta anche

come SCI (*Serial Communications Interface*). La porta è accessibile esternamente dai piedini RC6 e RC7, e si può configurare come asincrona (il trasferimento dati avviene in modalità *full duplex*), o come sincrona, sia *master* sia *slave* (i dati sono trasferiti in modalità *half duplex*); la velocità di trasmissione è configurabile tramite i registri di controllo fino a 57600 baud al secondo. L'USART è stata usata per trasmettere i dati elaborati alle periferiche o al PC *host* a valle del sistema: la comunicazione MIDI, infatti, avviene serialmente alla velocità di 31250 baud al secondo. Poiché al momento non è prevista una comunicazione bidirezionale tramite porta MIDI, ma soltanto una trasmissione in uscita, si è adoperato il solo *pin* RC6 (TX), configurando i registri di controllo solamente per quanto riguarda l'*output*; la comunicazione può avvenire tramite una sola connessione in quanto asincrona: il protocollo MIDI, infatti, non prevede segnalazioni di *handshaking*. In figura 5 è rappresentato lo schema *hardware* del modulo di trasmissione USART: si evidenziano in particolare il registro SPBRG (che stabilisce la velocità dell'informazione) e il registro TXREG che contiene il nuovo dato da inviare.

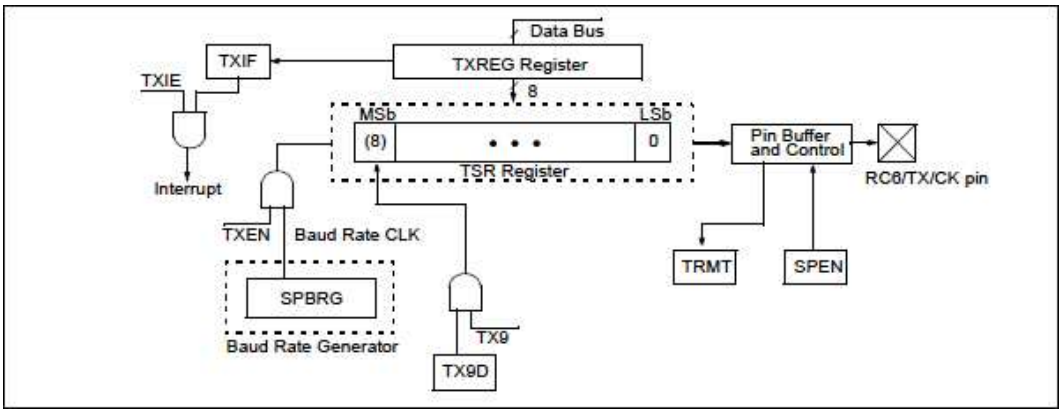


Figura 7: modulo di trasmissione della porta USART.

La configurazione della porta e la trasmissione si ottengono intervenendo sui valori contenuti in 4 registri:

- RCSTA (*Receive Status And Control Register*, 18h): questo registro controlla i parametri relativi alla ricezione seriale di dati da altre periferiche e/o da altri controllori; nonostante non sia previsto traffico dati verso la scheda controllo, è comunque necessario operare sul bit più significativo (SPEN, *Serial Port Enable*) di questo registro per abilitare o disabilitare la porta seriale;

- TXSTA (*Transmit Status And Control Register*, 98h): controlla le impostazioni riguardanti la trasmissione dei dati. In particolare, il bit 5 (TXEN, *Transmit Enable*) abilita la trasmissione, il bit 4 (SYNC, USART *Mode Select Bit*) stabilisce se la comunicazione deve essere sincrona o asincrona, mentre il bit 2 (BRGH, *High Baud Rate Select Bit*) abilita/disabilita la trasmissione ad alta velocità;

- SPBRG (*Baud Rate Generator Register*, 99h): viene caricato manualmente con un valore ricavato dal *datasheet* (vedi tabella 2), che determina la velocità di trasmissione;

- TXREG (USART *Transmit Register*, 19h): è il registro che contiene di volta in volta i dati da trasmettere; la trasmissione inizia appena i dati sono caricati, ma va fatta attenzione a non caricare il dato successivo mentre si sta ancora inviando il dato attuale, pena la sovrascrittura e quindi la corruzione dell'*output*: per effettuare un *check* sul contenuto di TXREG si può ricorrere al bit 4 (TXIF, USART *Transmit Interrupt Flag Bit*) del registro PIR1 (*Peripheral Interrupt Register*, 0Ch). Il microcontrollore, infatti, pone ad uno il valore di TXIF (USART *Transmit Interrupt Flag Bit*) fintanto che il registro TXREG contiene un dato valido, e lo azzera al termine della trasmissione: in questo modo si può impostare una procedura che controlli il valore di TXIF prima di caricare il nuovo dato; in maniera analoga si può interrogare il microcontrollore sul contenuto del bit 1 di TXSTA (TRMT, *Transmit Shift Register Status Bit*). Una volta caricato il nuovo dato in TXREG, il suo contenuto viene passato nel registro TSR (*Transmit Shift Register*, non accessibile direttamente) e inviato alla velocità stabilita.

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

Tabella 1: *baud rates* in modalità asincrona (BRGH=0) per diverse frequenze di clock del PIC; nella colonna di destra sono indicati i valori da caricare nel registro SPBRG per impostare la velocità di trasmissione desiderata.

APPENDICE 2: IL CONVERTITORE A/D INTEGRATO

Riguardo al presente progetto, dal momento che spetta ad un calcolatore il compito di interpretare e tradurre in comandi i segnali provenienti dai fotosensori, serve una traduzione del segnale da analogico a digitale, ovvero il passaggio da un segnale continuo, rappresentativo delle movenze del *performer*, ad un segnale digitale corrispondente che può essere elaborato tramite *computer*. La conversione, come anticipato, viene effettuata dall'ADC integrato nel microcontrollore, che è di tipo SAR (*Successive Approximation Register*). Un ADC SAR si basa su un principio semplice, quello della ricerca per bisezione dell'intervallo: un comparatore confronta ad ogni passo la tensione in ingresso con una tensione (generata internamente da un DAC) pari alla media del semi-intervallo di riferimento, che viene dimezzato ad ogni *step*. In figura è riportato uno schema di principio e un grafico che esemplifica la modalità di acquisizione del dato analogico:

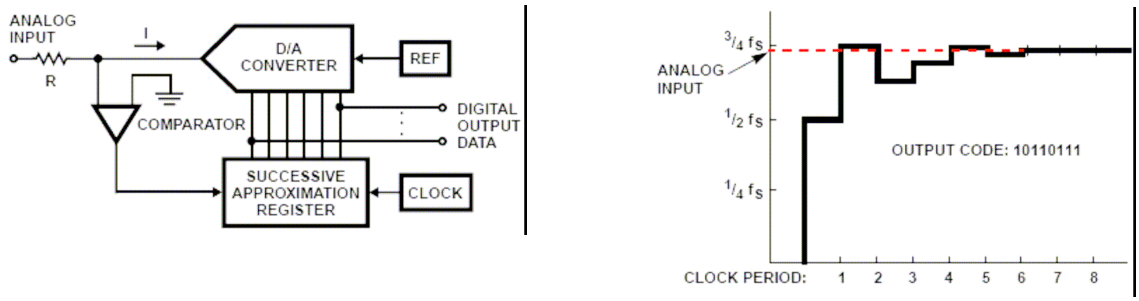


Figura 8: schema di principio del convertitore A/D di tipo SAR (a sinistra); il segnale in ingresso viene approssimato per bisezione ciclica dell'intervallo di tensione in cui ricade. I bit del risultato sono ricavati uno alla volta, serialmente, per cui il tempo di conversione può diventare relativamente lungo per i convertitori SAR ad elevata risoluzione.

L'*output* di un convertitore SAR è quindi intrinsecamente seriale, anche se in genere un segnale di fine conversione rende disponibile il risultato parallelamente su un registro; la velocità di conversione colloca questo tipo di convertitori nella parte medio – alta della scala: non sono infatti veloci come i convertitori *flash* ma sicuramente molto più dei convertitori a integrazione. Si riportano di seguito le tabelle dei principali parametri riguardanti il convertitore integrato, di cui si evidenzia il tempo di acquisizione T_{ACQ} pari a 19,72 µs e il tempo di conversione T_C (alla frequenza di 20 MHz di clock, $T_{AD} = 32 \cdot T_{OSC}$) di 19,2 µs:

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
A01	NR	Resolution		—	—	10-bits	bit	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A03	EIL	Integral linearity error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A04	EDL	Differential linearity error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A06	EOFF	Offset error		—	—	< ± 2	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A07	EGN	Gain error		—	—	< ± 1	LSb	VREF = VDD = 5.12V, VSS ≤ VAIN ≤ VREF
A10	—	Monotonicity(3)		—	guaranteed	—	—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference voltage (VREF+ - VREF-)		2.0	—	VDD + 0.3	V	Absolute minimum electrical spec. To ensure 10-bit accuracy.
A21	VREF+	Reference voltage High		AVDD - 2.5V		AVDD + 0.3V	V	
A22	VREF-	Reference voltage low		AVSS - 0.3V		VREF+ - 2.0V	V	
A25	VAIN	Analog input voltage		VSS - 0.3 V	—	VREF + 0.3 V	V	
A30	ZAIN	Recommended impedance of analog voltage source		—	—	10.0	kΩ	
A40	IAD	A/D conversion current (VDD)	Standard	—	220	—	µA	Average current consumption when A/D is on (Note 1)
			Extended	—	90	—	µA	
A50	IREF	VREF input current (Note 2)		10	—	1000	µA	During VAIN acquisition. Based on differential of VHOLD to VAIN to charge CHOLD, see Section 11.1.
				—	—	10	µA	During A/D Conversion cycle

Tabella 2: dichiarazione delle prestazioni e delle caratteristiche principali del convertitore A/D integrato nel microcontrollore PIC 16F877A (*datasheet* ufficiale Microchip). Le non linearità integrale e differenziale, così come l'errore di guadagno, si attestano su 1 LSB, mentre l'errore di *offset* può riguardare ben due dei bit meno significativi. Il costruttore inoltre garantisce la monotonicità della caratteristica di trasferimento e l'assenza di *missing codes*.

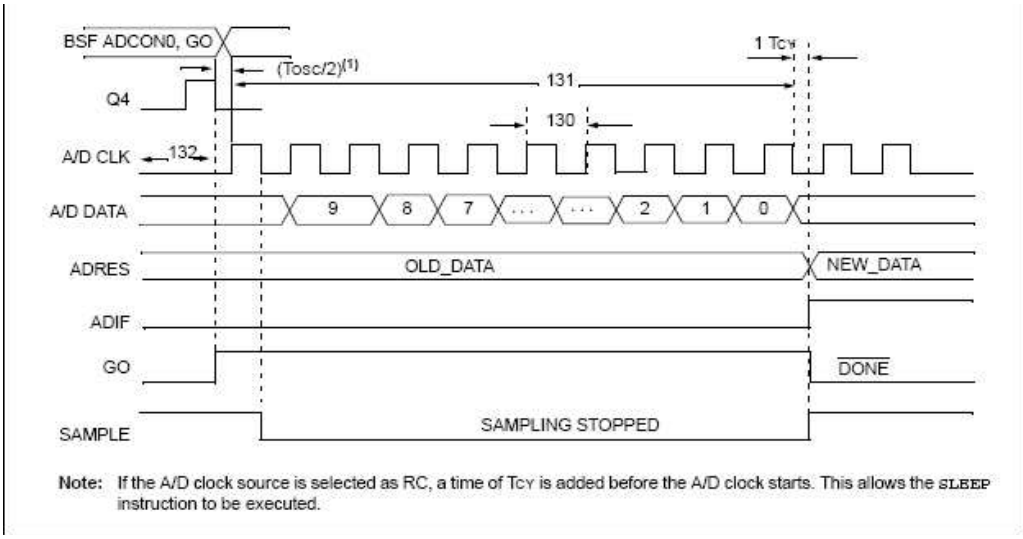


Figura 9: diagramma delle temporizzazioni relative alla conversione A/D; i numeri fanno riferimento ai parametri riportati in tabella 3.

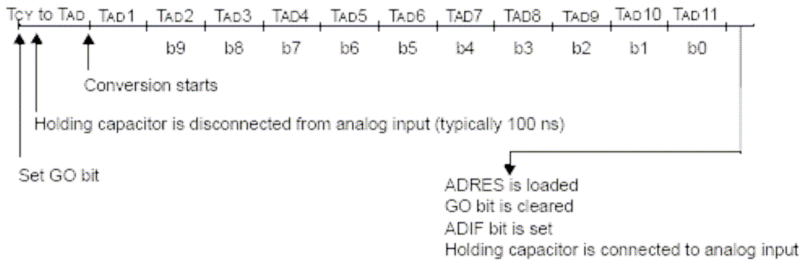


Figura 10: ciclo di conversione; il condensatore di tenuta non viene scaricato.

Param No.	Sym	Characteristic		Min	Typ†	Max	Units	Conditions
130	TAD	A/D clock period	Standard(F)	1.6	—	—	µs	Tosc based, VREF ≥ 3.0V
			Extended(LF)	3.0	—	—	µs	Tosc based, VREF ≥ 2.0V
			Standard(F)	2.0	4.0	6.0	µs	A/D RC mode
			Extended(LF)	3.0	6.0	9.0	µs	A/D RC mode
131	TCNV	Conversion time (not including S/H time) (Note 1)			—	12	TAD	
132	TACQ	Acquisition time		(Note 2)	40	—	µs	
				10*	—	—	µs	The minimum time is the amplifier settling time. This may be used if the "new" input voltage has not changed by more than 1 LSb (i.e., 20.0 mV @ 5.12V) from the last sampled voltage (as stated on CHOLD).
134	TGO	Q4 to A/D clock start		—	Tosc/2 §	—	—	If the A/D clock source is selected as RC, a time of TCY is added before the A/D clock starts. This allows the SLEEP instruction to be executed.

Tabella 3: definizione dei parametri di cui alla figura 9.

Il convertitore analogico/digitale integrato nel PIC 16F877A è corredato di circuito di *sample* & *hold* e può acquisire fino a 8 canali da un multiplexer analogico interno, i cui ingressi sono costituiti dai terminali della porta A e della porta E. Può inoltre operare in SLEEP *mode* (purché il suo segnale di *clock* sia derivato dall'oscillatore RC interno) e può disporre di due terminali (i *pin* RA2 e RA3) con cui impostare le tensioni di riferimento superiore e inferiore, per sfruttare al meglio la dinamica di ingresso. E' gestito internamente mediante quattro registri:

- ADCON0 (A/D *control register* 0, 1Fh): controlla il modulo A/D, stabilendo il canale di ingresso e la frequenza di *clock* per la conversione; il bit ADCON0.2 (GO/$\overline{\text{DONE}}$) in particolare ha una duplice funzione: viene posto manualmente a 1 per iniziare la conversione, e quando questa è terminata il PIC lo reimposta a 0, fungendo così sia da bit di *start* che da bit di stato;

- ADCON1 (A/D *control register* 1, 9Fh): tramite questo registro si stabilisce come deve essere formattato (allineato) il risultato della conversione e la configurazione

dei *pin* di ingresso, ovvero quali sono assunti come ingressi analogici, quali come I/O digitali e quali come *voltage references* (vedi tabella 1);

- ADRESL e ADRESH (A/D *Result Low*/*High Register*, 9Eh/1Eh): in questi due registri viene memorizzato il risultato dell'ultima conversione effettuata, allineato a destra (i 6 MSBs di ADRESH sono posti a 0) o a sinistra (i 6 LSBs di ADRESL sono posti a 0) secondo l'impostazione del bit ADCON1.7 (ADFM, A/D *Result Format Select Bit*).

Data la presenza di un multiplexer interno a 8 canali, si rendono possibili diverse soluzioni per convogliare i 32 segnali al convertitore, tramite l'utilizzo di multiplexer esterni a 4, 8 o 16 ingressi. La soluzione migliore, che esamineremo più avanti in maggior dettaglio, è in questo caso quella che prevede due integrati 4067 pilotati in mutua esclusione: benché in questo modo non si sfrutti il multiplexer interno, il numero di circuiti integrati addizionali è così ridotto al minimo, e il pilotaggio è semplice e senza problemi. Diversamente, si sarebbe dovuto ricorrere a 4 integrati 4052 (doppi multiplexer analogici a 4 ingressi), rinunciando peraltro alla possibilità di regolare la dinamica del convertitore, oppure a 4 integrati 4051 (multiplexer analogici a 8 ingressi), con una inutile complicazione circuitale e ingombro e consumi sicuramente superiori. Per chiarire meglio il quadro delle possibilità si riportano in tabella le configurazioni di ingresso impostabili, desunte dal *datasheet* del microcontrollore:

PCFG3: PCFG0	AN7(1) RE2	AN6(1) RE1	AN5(1) RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-
0000	A	A	A	A	A	A	A	A	VDD	VSS
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS
0010	D	D	D	A	A	A	A	A	VDD	VSS
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS
0100	D	D	D	D	A	D	A	A	VDD	VSS
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS
011x	D	D	D	D	D	D	D	D	VDD	VSS
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2
1001	D	D	A	A	A	A	A	A	VDD	VSS
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2
1110	D	D	D	D	D	D	D	A	VDD	VSS
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2

Tabella 4: tramite l'impostazione dei bit meno significativi del registro ADCON1 (chiamati PCFG3:PCFG0) si configurano i terminali delle porte A ed E come analogici, digitali oppure V_{REF}.

La scelta effettuata, oltre a non impegnare i terminali RA2 e RA3 che determinano la dinamica dell'ADC, utilizza un solo ingresso analogico della porta A, evitando quindi di cambiare canale (e quindi risparmiando tempo) ad ogni conversione e lasciando liberi diversi piedini per altri eventuali scopi.

Nell'interfacciamento con le sorgenti di segnale esterne si deve tener conto dell'impedenza che queste presentano in uscita: diverse non idealità riguardanti il circuito di ingresso del convertitore pongono infatti un limite superiore alla resistenza serie del generatore di segnale. In figura è raffigurato lo schema equivalente dei terminali di ingresso all'ADC:

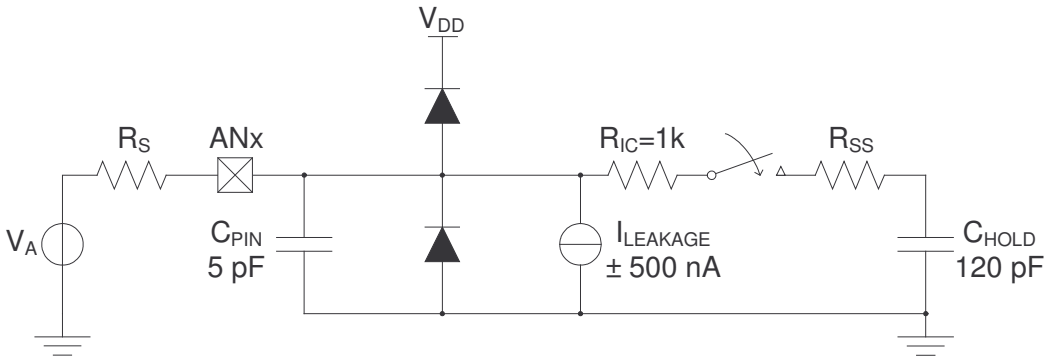


Figura 11: circuito equivalente di ingresso al convertitore A/D.

In figura sono evidenziati la capacità d'ingresso C_{PIN}, la resistenza di interconnessione $R_{IC,}$ i diodi di protezione, la corrente di perdita degli amplificatori interni; il condensatore C_{HOLD} schematizza la capacità di mantenimento del circuito di *sample* & *hold*, mentre R_{SS} rappresenta la resistenza differenziale del *sampling switch* (realizzato con una *pass gate*): il suo valore varia con la tensione di alimentazione e vale circa 7 kΩ per V_{DD} = 5 V (a *switch* chiuso). Come anticipato, particolare importanza assume il valore della resistenza serie della sorgente di segnale, raffigurata come R_S, che secondo i dati del costruttore non deve superare il valore di 10 kΩ; infatti, assieme alla resistenza differenziale R_{SS}, influenza direttamente la carica del condensatore di mantenimento, prolungando il tempo di acquisizione del convertitore per valori superiori. Nel nostro caso non si è dovuto adoperare nessuna circuiteria aggiuntiva per rispettare le specifiche, in quanto la resistenza interna della sorgente di segnale si può ricondurre alla resistenza serie dei multiplexer analogici (che dal relativo *datasheet* risulta pari al massimo a 1050 Ω) e dello stadio di uscita dei moduli sensori, veramente molto bassa: in totale, ben al di sotto dei 10 kΩ limite indicati.

APPENDICE 3: IL PROTOCOLLO MIDI

Le specifiche hardware

Il MIDI (*Musical Instrument Digital Interface*) è uno standard di comunicazione adottato ormai universalmente dai costruttori di strumenti musicali e apparecchiature audio elettroniche in generale, per permettere a dispositivi di natura e marca diverse di dialogare mediante un semplice cavo coassiale. Questo consente ad uno strumento musicale o altra apparecchiatura elettronica (*Master*) di controllarne altre (*Slave*), in modo da eseguire in sincronia tutti gli eventi che compongono il brano.

Dal punto di vista *hardware* i dispositivi MIDI – compatibili dispongono solitamente di 3 porte di comunicazione: MIDI IN, MIDI OUT e MIDI THRU, che si presentano all'utente come normali connettori DIN femmina a 5 poli. La porta MIDI OUT, durante l'esecuzione, trasmette i dati alle periferiche MIDI ad essa collegate, fungendo da *master*; viceversa, le periferiche che ricevono informazioni sulla porta MIDI IN sono conseguentemente configurate come *slave*; la porta MIDI THRU infine serve a replicare (in uscita) i messaggi che riceve da MIDI IN, rendendo così possibile il collegamento in cascata di altri dispositivi, anch'essi *slave*.

La trasmissione dei dati, secondo lo standard, deve essere seriale, asincrona, unidirezionale, e deve avvenire alla velocità di 31250 bit al secondo, con un bit di start e un bit di stop, senza *parity check* e senza segnalazioni di *handshaking*. La linea, pilotata in corrente, richiede un optoisolatore sul circuito ricevente, in modo da disaccoppiare elettricamente le apparecchiature MIDI connesse: in questo modo gli strumenti non hanno la massa in comune, con conseguenti riduzioni dei disturbi legati agli anelli di massa, particolarmente indesiderati in campo audio. Nello stato di riposo (*spacing*), la linea è mantenuta al livello logico alto (corrente nulla) dal master; quando la linea è portata al livello logico basso (*mark*, cui corrisponde una corrente di 5 mA) si ha il bit di *start* e inizia la trasmissione del primo byte di comando (detto *status byte*, a partire dall'LSB, con i dati complementati), cui segue un bit di *stop* (livello logico alto) e altri uno o due byte di dati (*data bytes*). Nella tabella a pagina seguente sono riassunte le principali caratteristiche *hardware* e *software* della trasmissione MIDI:

Formato del *frame*	1 bit di start 1 bit di stop 8 bit di dati Nessuna parità
Baud rate	31250 ± 1%
Pilotaggio	Corrente 0 logico: 5 mA 1 logico: 0 mA
Carico del ricevitore	Optoisolatore
Handshake	Nessuno
Connettore	DIN 5

Tabella 5: principali caratteristiche hardware dettate dal protocollo MIDI.

Dalla tabella si evince anche come sia impossibile usare una porta seriale RS-232 come porta MIDI: la RS-232 è infatti pilotata in tensione, non supporta la velocità di 31250 bit/s e richiede un carico dai 3 ai 7 kΩ al ricevitore.

Le specifiche software

Le comunicazioni MIDI si compongono di pacchetti di due o tre byte, secondo il tipo di messaggio. Di questi, il primo byte (*Status Byte*) codifica il tipo di messaggio, indicando come interpretare i byte successivi; gli altri invece prendono il nome di *Data Byte* e rappresentano informazioni quantitative per l'istruzione definita nello *Status Byte*. Per distinguere i due tipi di byte si fa riferimento al bit più significativo: se l'MSB è 1, il byte rappresenta uno *Status Byte* (128..255), se invece è 0, si ha un *Data Byte* (0..127). In figura è visibile un esempio di trasmissione di uno *Status Byte*:

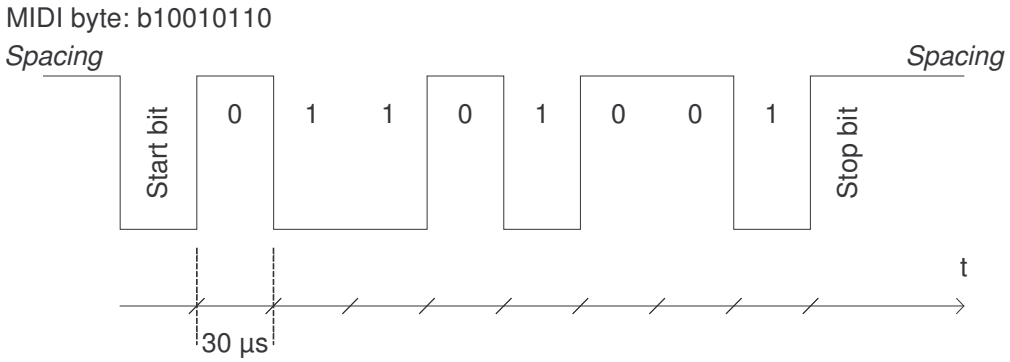


Figura 12: trasmissione di uno *Status Byte* in formato MIDI; come già detto, i bit vengono inviati dal meno significativo al più significativo, complementati.

Il tempo richiesto per la trasmissione di tre byte in formato MIDI richiede 960 µs, composti dai 30 µs per ognuno dei 10 bit e per tre byte, più altri 30 µs "separatori" tra il primo ed il secondo byte e 30 µs tra il secondo e il terzo.

I comandi MIDI sono divisi in due macrocategorie, i messaggi di canale e i messaggi di sistema, ulteriormente suddivisi in cinque categorie (messaggi di voce e di modo tra quelli di canale, e messaggi esclusivi, comuni e *real time* tra quelli di sistema), come raffigurato in figura:

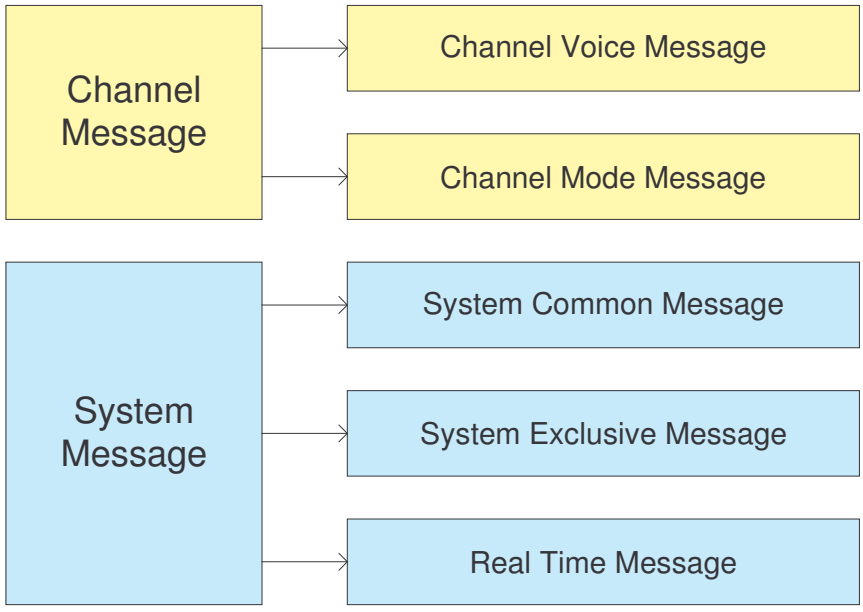


Figura 13: suddivisione dei comandi MIDI nelle sottocategorie, illustrate di seguito.

I messaggi di canale sono indirizzati indifferentemente su uno qualsiasi dei sedici canali disponibili, e sono eseguiti solo dagli *slave* configurati per ricevere su quello specifico canale; i primi quattro MSB dello *Status Byte* codificano il comando e i

restanti indicano il canale MIDI. I *Channel Voice Message* controllano le voci, ossia le modalità con cui sono eseguite le note e le polifonie: essi comprendono i messaggi *Note On*, *Note Off*, *Polyphonic Key Pressure*, *Control Change*, *Program Change*, *Channel Pressure* e *Pitch Bender*. I *Channel Mode Message* sono semplicemente un sottoinsieme dei messaggi di *Control Change*, e riguardano principalmente il controllo generale di un'apparecchiatura, non già di un singolo parametro.

Per quanto concerne i messaggi di sistema, essi non hanno riferimento ai canali e vengono ricevuti da tutte le periferiche MIDI connesse. I messaggi esclusivi riguardano comandi proprietari definiti dalle case costruttrici per i propri strumenti; i messaggi comuni riguardano il brano in esecuzione, per il corretto posizionamento nella sinfonia, e i messaggi *real time* fungono da metronomo di sistema mantenendo il sincronismo tra i vari moduli.

APPENDICE 4: IL *FIRMWARE*

Si riporta di seguito il listato del *firmware*, scritto in PICBasic:

```
'                    --------------
' ****************** Multiplex v8.1 ******************
'                    --------------

'-------------------------------------------------------
'                     Inizializzazione
'-------------------------------------------------------

' La sezione che segue include direttive al compilatore,
' in particolare la dichiarazione del tipo di PIC, della
' frequenza operativa e della risoluzione dell'ADC.

DEVICE 16F877A
XTAL 20
DECLARE SHOW_SYSTEM_VARIABLES = ON
DECLARE WARNINGS = ON
DECLARE REMINDERS = ON
DECLARE ADIN_RES 10

' Segue la dichiarazione delle porte e dei piedini
' usati come ingressi e uscite

OUTPUT PORTB                                    ' comandi LED
OUTPUT PORTC                                ' output seriale
OUTPUT PORTD                  ' indirizzi multiplexer esterni
INPUT PORTA                               ' ingressi analogici
INPUT PORTC.0                                 ' switch esterno

' Definizione degli alias

SYMBOL GO = ADCON0.2
SYMBOL SWITCH = PORTC.0

' Impostazione dei parametri relativi all'ADC: RA0 come
' unico ingresso analogico, FAD=Fosc/32, accensione del
' convertitore e formattazione a destra

ADCON0 = %10000001
ADCON1 = %10001111

' Dichiarazione delle variabili utilizzate

DIM INDIRIZZO,MIDI,INDICE AS BYTE
DIM MODULO,SENSORE AS BYTE
DIM MEDIA,OLD[32] AS WORD
DIM LETTURA AS ADRES.WORD
```

```
'-----------------------------------------------------
'                    Start-up
'-----------------------------------------------------

DELAYMS 500                  ' aspettiamo lo start-up del PIC

' Inizializzazione della comunicazione seriale

GOSUB SETUART

' Inizializzazione delle porte

LOW PORTB
LOW PORTC
LOW PORTD

' Inizializzazione del vettore delle medie precedenti

FOR INDIRIZZO=0 TO 31
   OLD[INDIRIZZO]=0
NEXT

' Inizializzazione delle variabili

CLEAR INDIRIZZO        ' indirizzo per i multiplexer esterni
CLEAR INDICE                          ' variabile d'appoggio
CLEAR SENSORE                    ' numero di sensore (0..3)
CLEAR MODULO                      ' numero di modulo (0..7)
CLEAR MEDIA               ' media degli ultimi due campioni
CLEAR MIDI            ' variabile di trasmissione temporanea

DELAYMS 50

'-----------------------------------------------------
'                     Main Program
'-----------------------------------------------------

' Il ciclo più esterno (LOOP) scandisce in sequenza i 4
' sensori del modulo corrente, selezionato tramite il
' ciclo più interno (ACQUIRE)

LOOP:    FOR SENSORE=0 TO 3
             INDIRIZZO.3 = SENSORE.0
             INDIRIZZO.4 = SENSORE.1

' Il ciclo più interno interroga in successione gli otto
' moduli, saltandone quattro se il valore letto su RC0
' (collegato al deviatore esterno) è pari a uno.

ACQUIRE:     INDIRIZZO.0 = MODULO.0
             INDIRIZZO.1 = MODULO.1

             IF SWITCH=1 THEN
                  INDIRIZZO.2 = MODULO.2
```

```
            ELSE INDIRIZZO.2 = MODULO.1
            ENDIF

' L'indirizzo per i multiplexer così ottenuto (composto
' dal numero di sensore e di modulo, eventualmente
' corretti) è quindi settato sulla porta D alla quale
' sono connessi i MC14067

            PORTD = INDIRIZZO

' A questo punto il programma estrae il numero di modulo
' interessato mascherando i primi tre bit della variabile
' INDIRIZZO e pone a uno il bit corrispondente della porta
' B (connessa ai traslatori di livello MC14504): questo
' comporta l'emissione di un lampo infrarosso dai moduli
' che dopo 20 µs viene spento resettando il valore della
' porta.

            INDICE = INDIRIZZO & 7
            SETBIT PORTB,INDICE
            DELAYUS 20
            LOW PORTB

' Ponendo il bit ADCON0.2 (alias GO) pari a uno si dà
' inizio alla conversione; al termine di questa il bit
' viene resettato dando così la possibilità di sapere
' se il procedimento è andato a termine (ciclo WHILE)

            GO = 1
            WHILE GO = 1 : WEND

' Il programma salva il valore acquisito nella variabile
' LETTURA (a 16 bit), che usa per calcolare la media
' sugli ultimi due campioni (i valori precedenti sono
' memorizzati nella variabile OLD)

            LETTURA = ADRES
            MEDIA = LETTURA + OLD[INDIRIZZO]
            MEDIA = MEDIA>>1

' Se il valore appena acquisito differisce dalla media
' vuol dire che c'è stata una variazione significativa
' che vale la pena spedire; allora il programma aggiorna
' la media e spedisce i tre byte formattando il dato come
' già visto, invocando ogni volta la procedura SEND

            IF MEDIA<>OLD[INDIRIZZO] THEN
            OLD[INDIRIZZO] = MEDIA
            MIDI = INDIRIZZO + 128
            GOSUB SEND
            CLEAR MIDI
            MIDI = MEDIA>>7
            MIDI.0 = MEDIA.7
            GOSUB SEND
```

```
            MIDI = MEDIA.LOWBYTE
            CLEARBIT MIDI,7
            GOSUB SEND
            ENDIF

' Infine serve controllare se la scansione è completa:
' se abbiamo scandito tutti i moduli, azzeriamo il valore
' di MODULO e ripartiamo daccapo, altrimenti incrementiamo
' il valore di MODULO e torniamo ad ACQUIRE

            IF INDICE=7 THEN
               MODULO = 0
            ELSE
            INC MODULO
            GOTO ACQUIRE
            ENDIF

         NEXT
         GOTO LOOP

'-------------------------------------------------------
'           Sub-routines di trasmissione seriale
'-------------------------------------------------------

' La procedura seguente, invocata all'avvio del programma,
' consente di configurare il microcontrollore per la
' trasmissione seriale a 31250 b/s

SETUART:    BSF 3,5                  ' pagina 1 della memoria
            SPBRG = 9               ' settiamo la velocità di
                                                trasmissione
            TXSTA = 160
            BCF PIE1,4             ' azzeriamo l'interrupt di
                                                trasmissione
            BCF 3,5                  ' pagina 0 della memoria
            RCSTA = 144
            RETURN

' La subroutine SEND si occupa di riempire il registro
' di trasmissione ogni volta che è invocata, attendendo
' prima che lo stesso sia vuoto

SEND: IF PIR1.4 = 0 THEN SEND
            TXREG = MIDI
            RETURN
```

BIBLIOGRAFIA

Tarabella L., Bertini G. - *Original gesture interfaces for live interactive multimedia performances*. JIM'97. Quatriemes Journees d'Informatique Musicale (Lyon, 1997).

Tarabella L., Bertini G., Sabbatini T. - *The twin towers: a remote sensing device for controlling live-interactive computer music*. M.C.P.A.'97 2nd Int. Workshop on Mechatronical Computer Systems for Perception and Action. Proceedings (Pisa, 1997), 41-46. Edited by G. Buttazzo and E. Ricciardi.

Tarabella L., Bertini G. - *Giving expression to multimedia performance*. ACM2000 - Bridging the gap workshop, Los Angeles, CA, USA.

Tarabella L., Bertini G., Boschi G. - *A data streaming based controller for real-time computer generated music*. International Symposium on Musical Acoustics. ISMA 2001 (Perugia, Italy, 10-14 September, 2001)

Tarabella L., Boschi G., Bertini G. - *Gesture, mapping and audience in live computer music*. Technical report IEI-CNR , TR 55 – 12 – 2001, Dicembre 2001.

Tarabella L., Bertini G. - *Wireless technology in gesture controlled computer generated music*. Workshop on Current Research Directions in Computer Music. Proceedings (Barcelona, Spain, November 15-17, 2001), 102-109. Claudia Lomeli and Ramon Loureiro (eds.).

Intersil Corp., *Principles of data acquisition and conversion*. Application note, October 1986 (http://www.intersil.com/data/an/an002.pdf)

Microchip Technology Inc., PIC16F87X *Data Sheet* - 28/40-Pin 8-Bit CMOS FLASH *Microcontrollers*. Datasheet, February 2001 (http://ww1.microchip.com/downloads/en/DeviceDoc/30292c.pdf).

Microchip Technology Inc., *FLASH memory programming specifications*. Application note, October 2002 (http://ww1.microchip.com/downloads/en/DeviceDoc/39589b.pdf).

Microchip Technology Inc., *10 – bit A/D Converter*. User Manual, 1997 (http://ww1.microchip.com/downloads/en/DeviceDoc/31023a.pdf).