

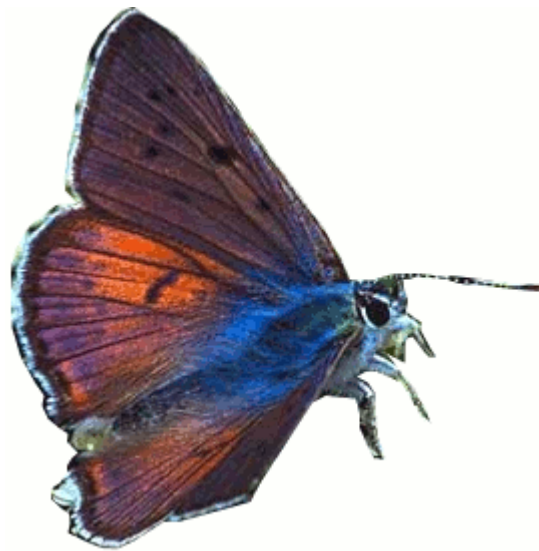
Vincenzo Grossi 2007-2008

# **BUTTERFLY** - *system*

---

Luminosi suoni volan via dal buio

**Un sistema hardware e software  
per il controllo in real-time  
delle sintesi audio e grafica correlate**



BUTTERFLY-*system*

## Ringrazio:

*Michelangelo Lupone,*

per essere un esempio etico, artistico e didattico, nonché per avermi guidato nella realizzazione del mio sogno in tutti questi anni ravvivando continuamente la mia passione per la musica con precise critiche e fertili dubbi.

*Lorenzo Seno,*

per il risolutivo apporto scientifico nella progettazione e tutta la puntuale assistenza nella realizzazione.

*Marco Giordano,*

per tutta la parte che concerne la trasmissione dati e per la decisiva scelta del Controller.

*Mariaclara Cervelli,*

per la messa a punto del sistema di connessione, compresa l'algoritmica fondamentale in Max

*Luigino Pizzaleo,*

per i suggerimenti della sezione Flosc e la collaborazione alla prima performance del sistema

inoltre, per ragioni molteplici ed importanti, tutte legate alla mia formazione musicale, elettronica ed informatica in ordine di conoscenza cronologica:

*Marco Fumo, Giampaolo Chiti, Riccardo Bianchini, Teodoro Pace, Luigi Di Giampietro, Armando Gentilucci, Pietro Grossi, Andrea Camilleri, Pino Monopoli, Antonio Iafigliola, Walter Ciancusi, Pablo Montagne, Laura Bianchini, Beatrice Lasio, Silvia Lanzalone*

qualche altro musicista:

*Skryabin, Debussy, Bartok, Webern, Ligeti, Stockhausen, Nono, Taylor, Hendrix*

qualche regista:

*Stroheim, Griffith, Ray, Ruttmann, Bunuel, Beckett, Bene, Cavallone, Kubrik, Iodorowsky, Greenaway*

qualche artista visivo:

*Mondrian, Malevich, Kandinsky, Dalì, Magritte, Ernst, Vasarely*

*Un libro a parte richiederebbe il ringraziamento ai miei familiari,  
... ma non c'è bisogno di scriverlo.*

# INDICE

<b>1 – IDEAZIONE</b> .....	<b>09</b>
1.1 – Quale strumento e per fare quale musica.....	10
1.2 – La musica elettronica in real-time .....	11
1.2.1 – Precedenti generali .....	11
1.2.2 – Precedenti personali: realizzazioni attinenti le problematiche della performance elettronica e il rapporto suono-luce .....	14
1.2.2.1 – “A” (1979) per flauto, clarinetto, pianoforte e percussioni .....	15
1.2.2.2 – “Mariti ciechi” “Femmine folli” “Giglio infranto” “Rapacità”(1980-86).16	
1.2.2.3 – “Cuscinetti a sfera celeste” (1984) .....	16
1.2.2.4 – “Inutile voltarsi, impossibile vederli” (2002) .....	16
1.2.2.5 – “Trilateri” (marzo 2003) .....	17
1.2.2.6 – “Mediateca” (aprile 2003) .....	18
1.2.2.7 – “Unreal-time” (maggio 2003) .....	18
1.2.2.8 – “Out of control” (marzo 2004) .....	19
1.2.2.9 – “Le metamorfosi” (aprile 2004) .....	19
1.2.2.10 – “Motus nervi et cordis” (giugno 2004) .....	19
1.2.2.11 – “Film” (novembre 2004) .....	20
1.2.2.12 – “Piet” [progetto] (dicembre 2004) .....	20
1.2.2.13 – “Isteresi” (dicembre 2005) .....	21
1.2.2.14 – “Laria” (giugno 2006) .....	22
1.2.2.15 – “Progetto educativo di approccio formale alla musica elettronica, un sistema plurimediale integrato” (marzo 2007).....	22
1.2.3 – Una possibile sintesi .....	23
1.3 – Considerazioni sul gesto, lo strumento e la visione della musica.....	23
1.4 – Relazioni possibili audio-grafiche ed implicazioni psicologiche.....	25
1.4.1 – SINESTESIA: metafora, fenomeno percettivo o patologia? .....	25
1.4.1.1 – Come nasce .....	25
1.4.1.2 – In cosa consiste .....	26
1.4.1.3 – L’ascolto colorato e i fotismi .....	28
1.4.1.4 – Sinestesia ed informatica .....	29
1.4.2 – Considerazioni sulla fisiologia della vista e dell’udito .....	31
1.4.3 – Effetti di fusione: differenze incollabili .....	32
1.4.4 – Analogie strutturali .....	33
1.4.5 – Passeggiata storica .....	33
1.4.5.1 – Fisica .....	33
1.4.5.2 – Musica, pittura e teatro .....	34
1.4.5.3 – Cinema e Video .....	35
1.5 – Delle opere d’Autore e delle opere di più autori .....	37
1.6 – Dubbi residui: meglio un libro o un ologramma dinamico con audio 24.1? .....	38
<b>2 – PROGETTAZIONE</b> .....	<b>39</b>
2.1 – Obiettivi .....	40
2.2 – Generalità.....	40
2.2.1 – Sottosistema hardware .....	40
2.2.2 – Sottosistema software .....	40
2.2.3 – Rilevazione dati ed interfacciamento .....	41
2.3 – Gerarchia procedurale .....	41
2.3.1 – Interfaccia fisica: obiettivi e problematiche .....	41
2.3.1.1 – Quantità di controlli <i>versus</i> qualità del controllo.....	42
2.3.1.2 – Informazione di stato .....	43

2.3.1.3 – Feedback fisico e devianza .....	43
2.3.1.4 – Hardware e Software per la rilevazione dei dati .....	43
2.3.2 – Connessioni e trasmissione dati: ipotesi di lavoro.....	44
2.3.3 – Tipologie dei software di sintesi audio-grafiche .....	44
2.3.4 – La composizione, ultimo e principale obiettivo .....	45
2.4 – Interfaccia fisica: ipotesi di lavoro .....	46
2.5 – La conversione A/D ed il trasferimento dati .....	49
<b>3 – REALIZZAZIONE .....</b>	<b>50</b>
3.1 – Scelta dei componenti hardware .....	51
3.1.1 – La sensoristica .....	51
3.1.1.1 – SLIDE V.2.1 .....	52
3.1.1.2 – TOUCH v.1.4 .....	53
3.1.1.3 – BEND SENSOR (wired) .....	54
3.1.1.4 – POTENZIOMETER 10 k .....	54
3.1.2 – La conversione A/D e la comunicazione: “ <u>Make Controller Kit</u> ”.....	55
3.1.2.1 – Make Application Board .....	56
3.1.2.1.1 – Input analogici .....	56
3.1.2.1.2 – Alimentazione degli input analogici .....	57
3.1.2.1.3 – Voltaggi di ingresso.....	58
3.1.2.1.4 – Alimentazione.....	58
3.1.2.1.5 – Componenti per l’utente: Status LEDs.....	59
3.1.2.2 – Make Controller.....	60
3.1.2.2.1 – Panoramica del Controller.....	60
3.1.2.2.2 – SAM7X.....	60
3.1.2.2.3 – LED’s.....	62
3.1.2.2.4 – Alimentazione.....	62
3.1.2.2.5 – Ethernet PHY.....	62
3.1.2.2.6 – CAN.....	63
3.1.2.2.7 – EEPROM.....	63
3.1.2.2.8 – USB.....	63
3.1.2.2.9 – JTAG.....	64
3.1.2.2.10 – Input / Output .....	64
3.1.2.2.11 – Firmware .....	64
3.1.2.2.12 – <u>McHelper</u> : programmazione della scheda e upload del Firmware.....	65
3.1.2.3 – Connessione diretta Ethernet .....	66
3.1.2.3.1 – Hardware e configurazione degli indirizzi .....	66
3.1.2.3.2 – Settaggio della rete in Windows .....	66
3.1.2.3.3 – Settaggio del Controller .....	67
3.1.3 – Il computer e la scheda video: <u>HP Compaq 8710w</u> .....	68
3.1.3.1 – <u>HP Compaq 8710w</u> : Specifiche inerenti il sistema Butterfly .....	68
3.1.4 – La scheda audio: <u>RME Hammerfall DSP Multiface</u> .....	69
3.1.4.1 <u>RME Hammerfall DSP Multiface</u> : Specifiche inerenti il sistema Butterfly.....	69
3.1.5 – Il pedale MIDI: <u>StuArt MB5</u> .....	70
3.1.6 – Il mixer digitale: <u>Yamaha01v</u> .....	70
3.1.7 – Il proiettore video: <u>Toshiba TDP S-20</u> .....	70
3.2 – Scelta dei componenti software.....	71
3.2.1 – Il sistema operativo: <u>Windows XP sp2</u> .....	71
3.2.2 – Il protocollo di comunicazione: <u>Open Sound Control</u> .....	71
3.2.2.1 – Interfacciamento OSC .....	73
3.2.3 – Il programma di sintesi e composizione algoritmica: <u>Max-MSP</u> .....	74
3.2.3.1 – Operazioni preliminari per l’utilizzo di OSC in <u>Max-MSP</u> .....	74
3.2.4 – Il programma di sintesi grafica vettoriale: <u>FLASH</u> .....	76
3.2.4.1 – L’interfaccia di rete per <u>FLASH</u> : XMLSocket e Floc.....	76

3.3 – Realizzazione dei componenti hardware mancanti .....	77
3.3.1 – Progettazione dell'interfaccia fisica: <u>LittleWings</u> .....	77
3.3.1.1 Disegno tecnico dei supporti removibili contenenti i quattro sensori..	81
3.3.2 – Realizzazione dell'interfaccia fisica: <u>LittleWings</u> .....	89
3.3.2.1 – Meccanismo di calibrazione a viti .....	90
3.3.3 – Progettazione della scheda di condizionamento del segnale <u>C8 [L.Seno]</u> .....	91
3.3.4 – Realizzazione della scheda di condizionamento del segnale <u>C8</u> .....	95
3.3.5 – Progettazione del contenitore/supporto dell'interfaccia <u>LittleWings</u> .....	99
3.3.5.1 – Cavi e connettori.....	99
3.3.5.2 – Disegno delle aperture da praticare nel contenitore.....	100
3.3.6 – Realizzazione finale dell'interfaccia <u>LittleWings</u> .....	105
3.3.6.1 – Alcune foto della realizzazione.....	106
3.4 – Realizzazione dei componenti software mancanti.....	109
3.4.1 – Algoritmo di connessione OSC in Max-MSP.....	109
3.4.2 – Codice per la connessione OSC in Flash: <u>FLOSC [B.Chun]</u> .....	111
3.4.2.1 – Batch file per avviare il server Java .....	111
3.4.2.1.1 – Problemi con il file batch per l'avvio di Flosc .....	112
3.4.2.2 – Codice Flosc in ActionScript di Flash.....	112
<b>4 – TECNICA</b> .....	<b>116</b>
4.1 – Tecnica e tecnologie.....	117
4.2 – La forma e la percezione.....	118
4.2.1 – Vincoli formali nel Butterfly-system.....	119
4.3 – Tecniche generali audio.....	121
4.3.1 – Riscaldamento valori digitali.....	121
4.3.2 – Controllo inizio e fine dati in arrivo.....	121
4.3.3 – Bilanciamento semplice e condizionato.....	123
4.3.4 – Select patch, start/stop, player wave, mute.....	123
4.3.5 – Operatore logico a due margini (3RngTog).....	125
4.3.6 – Alcune specifiche su moduli di Max.....	125
4.3.6.1 – Line, Line~, pak, pack, set.....	125
4.3.6.2 – function.....	126
4.3.6.2.1 – setdomain e setrange.....	126
4.4 – Tecniche generali grafiche.....	127
4.4.1 – Ancora su FLOSC.....	127
4.4.2 – Layers .....	128
4.4.3 – Nidificazione di filmati e simboli.....	128
4.4.4 – Fade in/out.....	128
4.4.5 – Alcune specifiche sugli strumenti di Flash.....	129
4.4.5.1 – Punto di registrazione.....	129
4.4.5.2 – Come creare un filmato complesso e trasformarlo in simbolo.....	129
4.4.5.2.1 – Import movieclip (symbol da documento *.FLA).....	129
4.4.5.3 – Istanze di uno stesso simbolo su layer diversi.....	130
4.4.6 – <u>Flash Action Script 2.0</u> .....	131
4.4.6.1 – Variabili.....	131
4.4.6.1.1 – Variabili legate a testi dinamici.....	131
4.4.6.2 – Operatori composti.....	131
4.4.6.3 – Function (generica).....	132
4.4.6.4 – Ciclo For.....	132
4.4.6.5 – Condizioni .....	132
4.4.6.5.1 – if – else – if else – or.....	132
4.4.6.5.2 – Switch.....	132
4.4.6.6 – Get Property e Set Property.....	133
4.4.6.7 – Rotation.....	133
4.4.6.8 – Utilizzo del mouse come interfaccia.....	133
4.4.6.9 – onEnterFrame.....	134

4.4.6.10 – Gradienti e colori.....	134
4.4.6.10.1 – Colors :GradientBevelFilter.colors property.....	135
<b>5 – COMPOSIZIONE .....</b>	<b>138</b>
5.1 – Effimere soddisfazioni.....	139
5.2 – “L.194/78” primo pezzo.....	139
5.2.1 – La fine di alcune idee.....	139
5.2.2 – Ideazione ed intervento terapeutico.....	140
5.2.3 – Algoritmi.....	142
5.2.3.1 – Prima scena “Click”.....	142
5.2.3.2 – Seconda scena “PulseTrain”.....	143
5.2.3.3 – Terza scena “PointSolo”.....	144
5.2.3.4 – Quarta scena “Biliardo”.....	145
5.2.3.5 – P005 e seguenti.....	149
5.3 – “LittleWings” secondo pezzo.....	150
5.3.1 – Informel.....	151
5.3.2 – BigBang.....	154
5.3.3 – Scacciapensieri.....	156
5.3.4 – Ruote metalliche.....	157
5.3.5 – Sfere.....	159
5.3.6 – Matrix4D.....	161
5.3.7 – Finale.....	166
<b>6 – PERFORMANCE.....</b>	<b>167</b>
6.1 – Boot sequence.....	168
6.2 – Setup.....	168
6.2.1 – Setup Laptop.....	168
6.2.2 – Setup iniziale.....	168
6.3 – Standby.....	169
6.3.1 – Standby system (30’-60’).....	169
6.3.2 – Wake-up system.....	169
6.4 – Problemi e soluzioni.....	169
6.5 – Utilizzo di <u>LittleWings</u> .....	170
6.5.1 – Montaggio.....	170
6.5.2 – Calibrazione.....	170
6.5.3 – Esecuzione.....	170
<b>7 – CONCLUSIONI.....</b>	<b>171</b>
7.1 – Work in progress.....	172
7.2 – Miglioramenti e futuri sviluppi.....	172
7.3 – Mandala o 3D?.....	172
<b>8 – RIFERIMENTI.....</b>	<b>173</b>
8.1 – Bibliografia.....	174
8.2 – Links.....	174

*dedicata a chi la leggerà*



# 1 – IDEAZIONE

## 1.1 Quale strumento e per fare quale musica?

Chi deride la ricerca musicale e quella tecnologica connessa forse dimentica che essa è una delle poche costanti della storia musicale. Inimmaginabile che quindi si arresti e si contenti di utilizzare lo strumentario ed il linguaggio esistente. Lo strumentario infatti non è neutro ed utilizzabile in qualsiasi modo per ottenere qualsiasi risultato fonico. Ogni strumento ha da sempre privilegiato determinate tecniche ed impedito le altre. Lungi da me il sogno di creare lo strumento musicalmente onnipotente, tuttavia ormai da tempo non riesco più ad immaginare della mia musica eseguita e scritta per lo strumentario classico.

Pur essendo la composizione che si avvale dell'elettronica non una disciplina a se, io personalmente cerco un rapporto diretto con la sintesi ed il controllo del suono e quindi mi avvalgo esclusivamente di mezzi elettronici, anche se il materiale elettronicamente trattato potrà essere anche di natura strumentale.

Tra le innumerevoli possibilità di fare e di intendere la musica, io ho preso in considerazione quella "classica" del rapporto unidirezionale tra compositore/interprete e pubblico. Analizzando la situazione di questo tipo di fruizione musicale si distinguono tre componenti:

1. *l'ambiente direzionato che contiene il pubblico seduto e rivolto verso la scena*
2. *la scena contenente l'interprete rivolta verso il pubblico*
3. *un sistema di diffusione/rinforzo del suono.*

Il pubblico è seduto e rivolto in una sola direzione perché è in attesa di un evento determinato nello spazio e nel tempo, evento di cui esiste un responsabile principale (l'autore), coadiuvato o meno da interpreti. In sala possono essere stati distribuiti programmi di sala contenenti notizie sull'autore e sul brano.

La presenza o meno di queste note di sala da sola si presterebbe ad una propria trattazione, investendo la sfera del rapporto tra l'opera e le sue appendici testuali. Già il solo "titolare-nominare" un brano e rendere pubblico tale titolo al momento dell'ascolto indirizza tale ascolto su una strada più o meno ristretta in funzione della quantità di ambiguità presente nel titolo stesso. A differenza di Olivier Messiaen, famoso per l'importanza data a tutto il materiale extra-musicale, dal titolo alle indicazioni e controindicazioni, io confesso di non aver ancora risolto personalmente questo dilemma, combattuto da una parte dall'urgenza comunicativa e dall'altra dalla convinzione dell'impossibilità di qualsiasi comunicazione che valichi il superficiale, vacuo ed autoreferente scambio di *nomi*. Tentare di comunicare aggiungendo utili informazioni che focalizzino precisamente l'attenzione e instradino correttamente l'ascolto o arrendersi alla constatazione che la comunicazione/espressione altro non è che una **azione** che può provocare **reazioni** del tutto imponderabili? O meglio, si deve decidere tra l'uso della retorica (nascondendo quindi il proprio reale pensiero) con l'intento di provocare una specifica **reazione** desiderata o continuare a tentare un'eroica sincerità affidandosi alla casualità delle possibili reazioni e a statisticamente inconsistenti probabilità di eccitare altrui risonanze.

In passato e nel presente credo che tutti i compositori abbiano fatto un uso cosciente delle tecniche retoriche di seduzione artistica e nei migliori casi le abbiano

utilizzate per convogliare funzionalmente le proprie esigenze espressive. Schoenberg pur ricostruendo un sistema di altezze estraneo alla tradizione tonale non ha rinunciato ad una dinamica emozionalmente vissuta e non calcolata. Edgar Varese abbracciando il rumore lo ha reso seducente e strutturalmente significativa, privandolo della sua natura caotica e sovversiva. I pochi coraggiosi che hanno voluto rinunciare alle ben note strategie del passato hanno dovuto aspettare decenni prima di poter incontrare una certa accettazione prima e condivisione dopo (Anton Webern, La Monte Young, Cecil Taylor).

L'auralizzazione del suono sarà effettuata in modo da avvolgere il pubblico senza però costringerlo ad inseguirlo dietro le proprie spalle. Questo perché di fronte si troverà un interprete ed uno schermo di proiezione. Lo schermo e la creazione video/grafica è per me necessaria in quanto non è possibile ipotizzare un ascolto collettivo ideale che escluda qualsiasi percezione visiva (eccezion fatta per i non vedenti). E sapendo che quella visiva resta per l'essere umano la fonte primaria di rapporto col mondo, non credo sia più possibile ignorarla durante la proposta musicale.

Stabilito il campo di azione (sala da concerto/ ambiente statico e direzionato) e le modalità generali di proposizione della mia opera (diffusione prevalentemente frontale con presenza umana visibile), ci accingiamo ad analizzare gli elementi costitutivi di questo genere di spettacolo, lo stato delle cose e gli antecedenti.

## 1.2 La musica elettronica in real-time

La mia ricerca compositiva ha da molti anni privilegiato la creazione digitale del suono per quanto riguarda le composizioni chiuse, ma non disdegna la collaborazione artistica con altri compositori/interpreti per composizioni aperte, nelle quali si può indagare proprio uno degli aspetti dell'esistenza a me più caro, la possibilità di comunicazione.

### 1.2.1 Precedenti generali

La pratica sempre più diffusa dell'elaborazione e sintesi in tempo reale della musica elettronica ha da molto tempo posto in evidenza la necessità di dotare la musica elettronica di uno strumento di controllo della sintesi digitale algoritmica più evoluto e soddisfacente delle periferiche di I/O di base (mouse e tastiera). Le soluzioni adottate si sono mosse essenzialmente in questi percorsi:

1. La ricerca di un sistema basato sul controllo parametrico puntuale di basso livello essenzialmente MIDI based. In questo caso il gesto dell'interprete informatico è nascosto e non significativo. La generazione ed elaborazione audio è efficacemente controllata ma necessita di un virtuosismo fisico e mentale perché l'idea interpretativa deve essere tradotta in una serie complessa e a volte parallela di variazioni parametriche di genere diverso che si presentano agli occhi e alle mani dell'interprete stesso come una serie di *faders* o *knobs* distinte solo da etichette, e che non restituiscono allo stesso nessuna informazione dell'azione musicale effettuata. Il pubblico non avverte alcun lavoro in tempo reale e riceve informazioni solo dagli altoparlanti. In sintesi si può dire che si sono solo aggiunte ulteriori interfacce ma dello stesso tipo e con gli stessi limiti di quelle esistenti. Un leggero miglioramento è stato

quello subito adottato di agire non su controlli singoli ma accoppiati in una matrice quadrata dove le singole coordinate della posizione corrente del mouse controllassero più parametri contemporaneamente. Ovviamente la maggiore potenza del gesto che agisce in parallelo su più parametri satura presto le potenzialità espressive, non potendo più variare un singolo parametro e lasciare gli altri costanti. Un importante passo in avanti su questa tipologia di controllo è stato fatto recentemente con la creazione e commercializzazione di sistemi configurabili basati su una interfaccia grafica sensibile al tatto, touchscreen. In queste periferiche le dieci dita delle mani

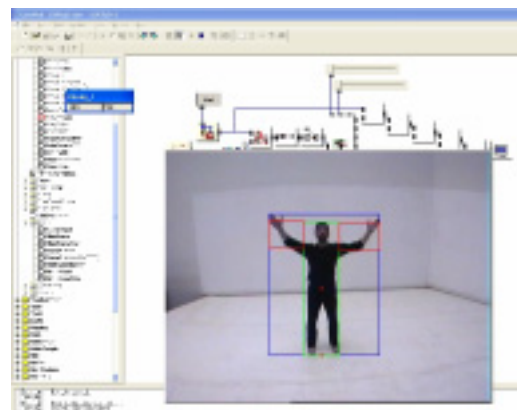


Lemur della Jazzmutant

possono controllare dieci parametri (o in parallelo loro multipli) contemporaneamente, ma anche un singolo parametro per volta. L'informazione di stato, cioè la possibilità di avere sotto lo sguardo la quantità (in forma grafica e/o numerica) della posizione raggiunta da ogni controllo virtuale, garantisce un alto

grado di visione globale dello stato presente. Uno dei limiti di tale sistema è quello di essere una interfaccia virtuale, cioè senza alcuna fisicità del controllo che restituisca in parte l'importanza acustica del gesto, p. es. un passaggio repentino dalla minima alla massima ampiezza è possibile semplicemente allargando la distanza tra l'indice ed il pollice o saltando con la punta di uno stesso dito tra i due estremi. In qualsiasi strumento fisico una tale variazione implica invece un notevole cambiamento fisico (aumento di pressione, di slancio, di posizione etc.) che tende ad unirsi all'emotività dell'interprete in modo tale da poter essere gestita e calibrata con più genuina convinzione e consequenzialità, sia per la quantità parametrica sia per la correlazione che nello strumento fisico ogni variazione interessa più parametri correlati (agogica, dinamica, timbrica)

2. Proprio per tentare di recuperare una maggiore fisicità del gesto si è basata una ricerca di un sistema nel quale l'interprete informatico si trasformi in attore sul palco e gestisca il sistema (una parte di esso) mediante una serie rigidamente codificata e mappata di gesti in campo aperto o agendo su sensori dislocati nello spazio circostante o direttamente sul proprio corpo o su uno strumento appositamente realizzato. La rilevazione delle variazioni da esso effettuate è a carico di sistemi vari, quali la ripresa video, l'interferenza con campi magnetici, la varianza di riflessioni di radiazioni infrarosse e nel caso dello strumento nelle tradizionali rilevazioni dei cambi di stato dei tasti, chiavi etc. Il tipo di controlli è in questo caso di più alto livello potendo essere associate ad un



Eyes Web: rilevazione dati sensori

singolo gesto più sezioni dell'algoritmo implementato, ma soprattutto una attenta scelta e calibrazione degli accoppiamenti sensori/parametri può arrivare a rendere realistico, perché polisemico e correlato, ogni gesto complesso. Il principale limite risiede nel rischio di comunicare mediante il corpo la meccanicità della traduzione parametrica, spesso costringendolo a gesti innaturali e assolutamente privi di un qualche contenuto espressivo e artistico. L'interprete più che libero di esprimersi attraverso la propria fisicità è portato a sentirsi in una gabbia virtuale che lo osserva incessantemente rendendolo più che controllore dell'algoritmo, controllato dallo stesso. Il pubblico viene inevitabilmente dirottato dall'auspicabile ascolto/apprezzamento della musica alla affannosa ricerca di nessi logico/funzionali e/o semantici tra quanto vede e quanto ascolta. Sarà allora necessaria una approfondita disanima del compromesso ottimale tra la quantità di controlli applicabili e la semplicità e naturalezza di movimento dell'interprete sul palco, tenendo conto che ogni movimento dovrà restituire una variazione acustica di importanza paragonabile a quella della visione del movimento stesso. Ma qui si entra già nelle scelte estetiche di ogni autore.

Doveroso ricordare chi di queste specifiche problematiche si occupa ormai da molti anni in Italia. I compositori Roberto Doati e Paolo Pachini, del Conservatorio di Trieste, attivi nella sperimentazione multimediale e il Dipartimento di Informatica, Sistemistica e Telematica dell'Università di Genova [DIST] e che ha centrato la propria ricerca proprio sulla psicologia del fare e dell'ascoltare musica, tendendo via via a liberare l'interprete musicale dal maggior numero di vincoli fisici (cavi e sensori) e dall'andare incontro all'utenza proponendo soluzioni interattive sempre più *user-friendly*. Tale ultima tendenza, mediata dall'industria che con essa mira alla costante espansione del mercato, mi trova personalmente molto contrario, in quanto ritengo che se la facilità di utilizzo avvicini un maggior numero di utenti, lo spessore espressivo ne venga irrimediabilmente compromesso. Questo può ancora andare bene se l'oggetto della facilitazione d'uso è un cellulare e il messaggio è un SMS del tipo "*Butta la pasta*" ma diventa via via proibitivo se gli oggetti pretendono essere portatori di un'espressione artistica. Come diceva Umberto Eco riguardo il suo best-seller "Il nome della rosa" se un lettore riesce a sostenere la lettura delle prime cento pagine, piene di riferimenti storici reali ed inventati, ma comunque tutt'altro che avvincenti, in qualche modo guadagna il diritto e la concentrazione per poter proseguire con soddisfazione nella lettura di quello che comunque è un intrigante *giallo* storico. Siamo circondati da trappole tecnologiche che ci sorridono "gratuitamente" e invece di invitarci a scalare "la montagna sacra" ci fanno dolcemente scivolare verso "la pozzanghera profana" della noia derivante dalla saturazione delle percezioni e dalla nullità dei contenuti.

Il confine tra intrattenimento (*entertainment*) e espressione artistica coinvolgente (*involvement*) è tutto qui: nello sforzo che concerne ogni conquista o meno. Con ciò non voglio assolutamente dire che il fruitore debba essere culturalmente attrezzato, anzi, anche un bimbo in età prescolare (ahimè ormai di tre anni al massimo) può e dovrebbe trovare sensazioni coinvolgenti. Ma la loro natura anarchica metterebbe a dura prova qualsiasi composizione, diventando essa stessa espressione animale di libertà. Quindi limiterei l'accesso solo a fruitori educati alla convivenza sociale. (Se non fosse vietato dalla legge, proporrei come prezzo da pagare per un concerto, una libbra di carne, perché l'arte non si compra col sudore della fronte o i proventi dello sfruttamento altrui, ma con il proprio sangue).



Studio 1 su volumi adattivi (2006) L. Galizia – M. Lupone

Una posizione carica di responsabilità verso i fruitori della musica elettronica è quella assunta da Michelangelo Lupone, che da anni ricerca i modi di conciliare l'ottimizzazione tecnologica della diffusione della propria ed altrui produzione contemporanea e la discrezione di proporla nel modo meno invasivo e più trasparente possibile. *L'involvement* realizzato in sempre nuove installazioni d'arte coinvolge prima di tutto

artisti plastici in soffici esplosioni della bellezza delle loro creazioni, che non vengono semplicemente rivestite di suoni a loro estranei e quindi accostati, ma bensì messe esse stesse in grado di vibrare, di suonare, di cantare con le proprie corde lignee, metalliche o addirittura cartacee. In secondo luogo si preoccupa di abbattere le barriere architettoniche e iniziatiche della fruizione musicale, ponendo nelle stesse mani del fruitore la capacità di dialogare con l'opera stessa, nella convinzione, suppongo, che solo un'opera che si accorga della nostra presenza, che ci ascolti essa stessa e si avvolga intorno a noi possa tentare ancora una volta un possibile contatto di sensibilità e pensieri. Come nei rapporti autentici, l'ascolto realmente aperto dell'interlocutore è la condizione unica di un possibile contatto.

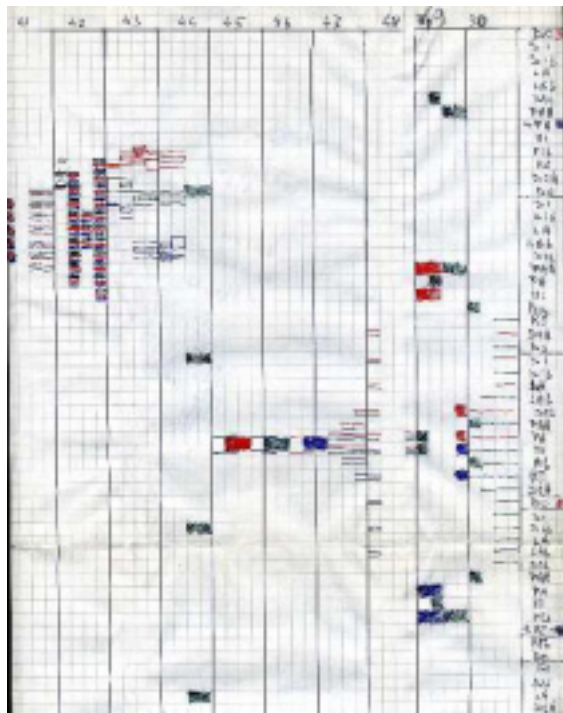
## 1.2.2 Precedenti personali: realizzazioni attinenti le problematiche della performance elettronica e il rapporto suono-luce

L'esigenza di "dare un colore al suono" posso dire di averla praticamente sempre avuta, sin dai tempi dei miei studi di composizione (1976-1986). Mi piace considerare questo lavoro sul quale è incentrata la tesi come un passaggio di questa ricerca quasi trentennale. Non per compiacimento personale, ma a supporto delle tesi espresse, ho ritenuto opportuno citare brevemente ed in ordine cronologico le esperienze musicali che reputo significative cioè intimamente e molto poco coscientemente legate al doppio filo della ricerca di modi di pensare la musica e di proporla.

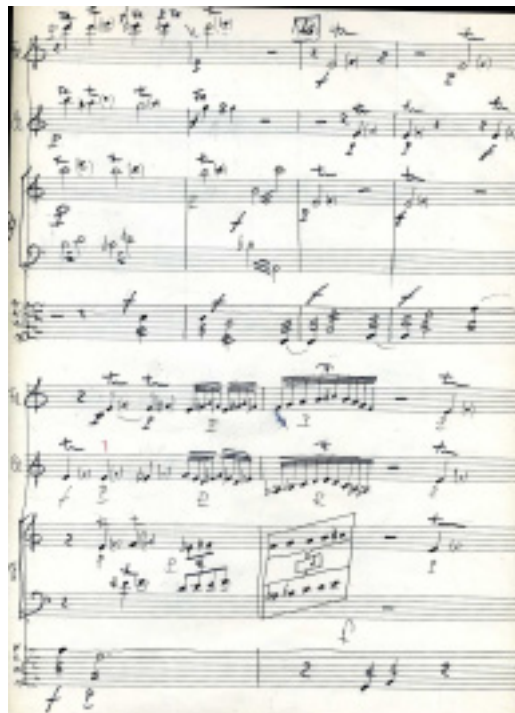
Parallelamente ad una ricerca più politicizzata, incentrata sulle possibilità di "democratizzazione" del fare musica, mediante le sperimentazioni di improvvisazioni collettive con compagini corali e poi strumentali nei primi anni '80, dove l'influsso degli ascolti del free-jazz (Cecil Taylor principalmente) e della musica di Stockhausen aveva instillato l'ipotesi di una più aperta creazione-fruizione musicale, praticavo anche una ricerca di tipo strutturalista nelle mie prime composizioni. Dopo aver brevemente *assaggiato* le asperità ricche di libertà sonora ma povere di gratificazioni acustiche della dodecafonia, tentai una composizione "cromatica" in tutti i sensi.

### 1.2.2.1 “A” (1979) per flauto, clarinetto, pianoforte e percussioni

La partitura fu scritta su dei fogli protocollo quadrettati, mediante una loro assegnazione cromatica, cioè per semitoni, semplicemente dal basso (La0) al rigo superiore (Do8). Il disegno della partitura, effettuato con colori acrilici di 4 tinte, fu effettuato considerandone solo l'aspetto coloristico e formale. Detta partitura grafica fu quindi tradotta in note musicali, associando ogni quadratino orizzontale alla durata del secondo, e realizzando quindi fasce sonore mediante le diverse possibilità tecniche degli strumenti (suoni tenuti, trilli e frullati per i due fiati e trilli, tremoli per le percussioni ed il pianoforte). La dinamica era ristretta in due piani sonori (*p* e *f*), simboleggiati graficamente dai tratti sottili e spessi.



Partitura grafica (b.41-50)



Partitura trad.(b.43-48)

Da questa esperienza scolastica trattenni dentro di me la consapevolezza di poter pensare la musica partendo non da suggestioni ma da formalizzazioni extramusicali conservando relazioni visive, quali la simmetria e la disposizione nell'ambito delle altezze, nonché quella intervallare. Ovviamente i limiti erano dati dal comportamento logaritmico dell'orecchio che non poteva rendere le dimensioni metriche della partitura in maniera fedele. Purtroppo tale episodio compositivo non fu affatto incoraggiato nè minimamente apprezzato ed i miei studi furono indirizzati verso più tranquille strade notazionali e contrappuntistiche.

Iscrittomi come uditore nell'anno precedente (1978) alla scuola di Musica Elettronica del Conservatorio di Pescara nella classe di Riccardo Bianchini, a causa di un furto della strumentazione, nel 1979 venne chiuso il corso, appena in tempo per incontrare per la prima volta Michelangelo Lupone. Per continuare gli studi di pianoforte e composizione nella mia città dovetti rinunciare a dar seguito al mio interesse per lo studio della musica elettronica, che veniva così interrotto per 20 anni.



### **1.2.2.2 “Mariti ciechi” “Femmine folli” “Giglio infranto” “Rapacità” (1980-1986)**

Nei primi anni '80 mi specializzai nell'accompagnamento al pianoforte dal vivo di film muti. I primi due di Erich Von Stroheim con musiche mie, per lo più improvvisate su dei canovacci preparati, “Broken blossom” di David Griffith con partitura originale e “Greed” sempre di Griffith con mie improvvisazioni, effettuate con le spalle allo schermo.

Queste esperienze estemporanee giovanili mi hanno fatto conoscere la problematica della musica da film, i ritmi della narrazione e i cliché utilizzati. La tecnica pianistica in questi casi era legata all'armonia tradizionale, ma lo studio nelle prove di come rendere al meglio ed in maniera sincronica la drammaticità delle scene formarono in me quasi una abitudine alla “sonorizzazione” delle immagini in movimento. Sono dell'epoca i primi tentativi di musicare riprese video casalinghe in VHS.

### **1.2.2.3 “Cuscinetti a sfera celeste” (1984)**

E' la prima performance da me realizzata con un pianoforte, un nastro registrato con un multitracce, un sintetizzatore analogico polifonico, un microfono ed un delay, il tutto amplificato e diffuso con l'HiFi domestico in una sala pubblica ricavata da un bagno penale borbonico. La cito in questo contesto perché la ritengo importante per la mia formazione di ideatore di una situazione musical-teatrale (recitavo anche un mio testo) che per la prima volta presentava un lavoro completamente autogestito e costruito con materiale elettronico. C'era comunque l'idea di creare una situazione complessiva di cui la musica fosse l'aspetto principale ma l'attore rivestisse un ruolo più complesso di quello di esecutore. Sono dello stesso periodo alcune trasmissioni radiofoniche di brani elettronici.

Accantonata per quasi vent'anni l'idea di realizzare una musica visibile, in favore di varie esperienze musicali in qualità di pianista e compositore, quasi sempre coinvolgenti altre persone, sul finire del millennio mi riavvicino alla musica elettronica prima come dilettante e quindi come studente. In precedenza, nel 1987 ho la fortuna di conoscere Pietro Grossi e Armando Gentilucci con i quali seguo un seminario di elettronica e di composizione. Non avendo neanche iniziali conoscenze di elettronica ed informatica, perdo l'occasione di frequentare i centri di Pisa, Milano, Napoli e Roma dove si inizia a sperimentare il real-time. Successivamente inizio la frequentazione con l'informatica musicale *domestica* del periodo (1988), quella resa disponibile dal mercato, che consente solo una gestione di dati MIDI per il controllo di sintesi hardware in wave-table e FM. Sono di questo periodo composizioni per elettronica e strumenti, che però non coinvolgono l'aspetto performance/multimediale. Intorno al 1995 passo dalla piattaforma Atari a quella Windows e prima con Cool Edit e quindi C-Sound inizio a realizzare musica digitalmente. Finalmente, nel 2001 riesco a riprendere i miei studi di musica elettronica con Michelangelo Lupone in questo Conservatorio.

### **1.2.2.4 “Inutile voltarsi, impossibile vederli” (2002)**

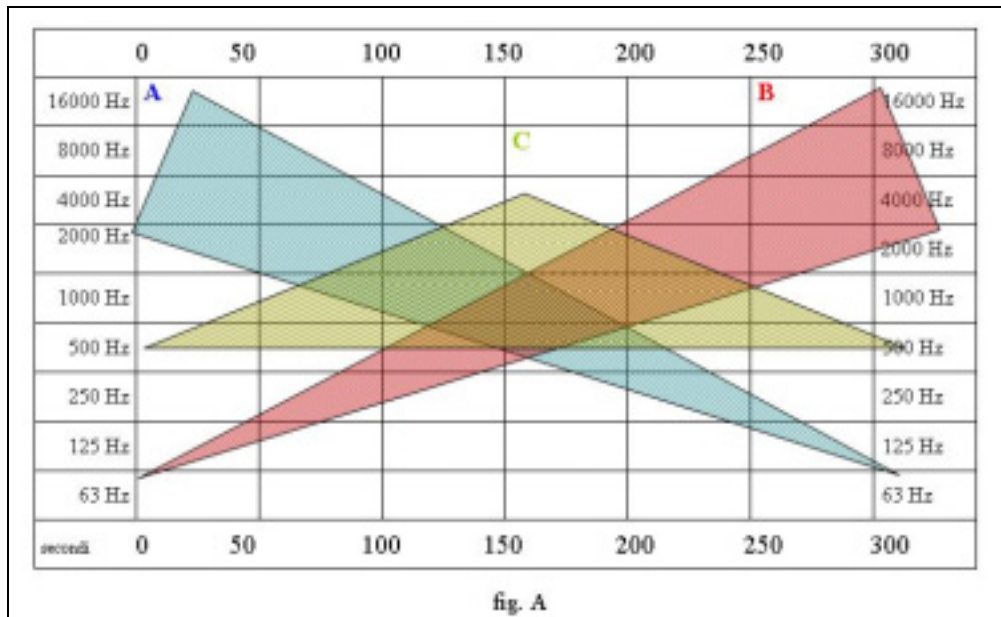
Quello del titolo, pur essendo un brano di musica elettronica senza grafica o video, è un brano il cui stesso titolo e l'idea dalla quale scaturisce torna a riaffacciarsi: l'impossibilità di vedere i suoni, ed in questo caso l'aspetto inquietante di sentire con essi “oscure presenze” senza che gli oggetti-animali che li producono



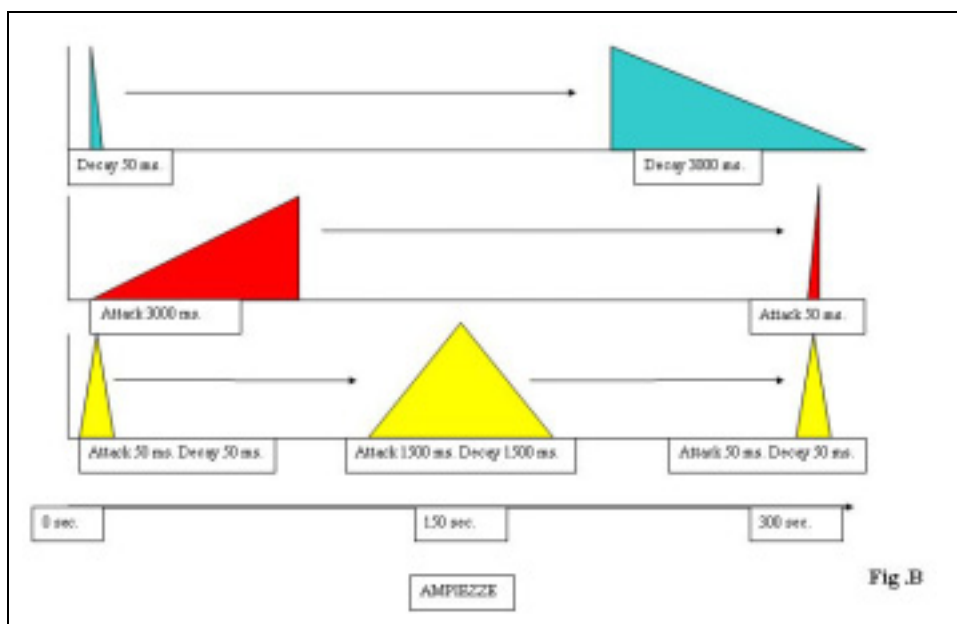
possano rivelarsi. Il brano si interrompe bruscamente ... per la morte del protagonista-ascoltatore.

### 1.2.2.5 "Trilateri" (Marzo 2003)

E' questo un brano di musica elettronica dove si tenta di seguire ancora una volta una formalizzazione grafica della composizione, questa volta incentrando l'attenzione sul triangolo. Non è ancora il momento del tentativo di associazione suono-colore ma solo di un modo di procedere dalla grafica, come nel 1979. Di seguito i grafici dell'andamento delle 3 bande frequenziali nelle quali si addensano gli eventi sonori.



Nella figura seguente la variazione delle tre serie di involuipi triangolari delle ampiezze (attacco e decadimento) realizzati variando continuamente le rispettive durate.

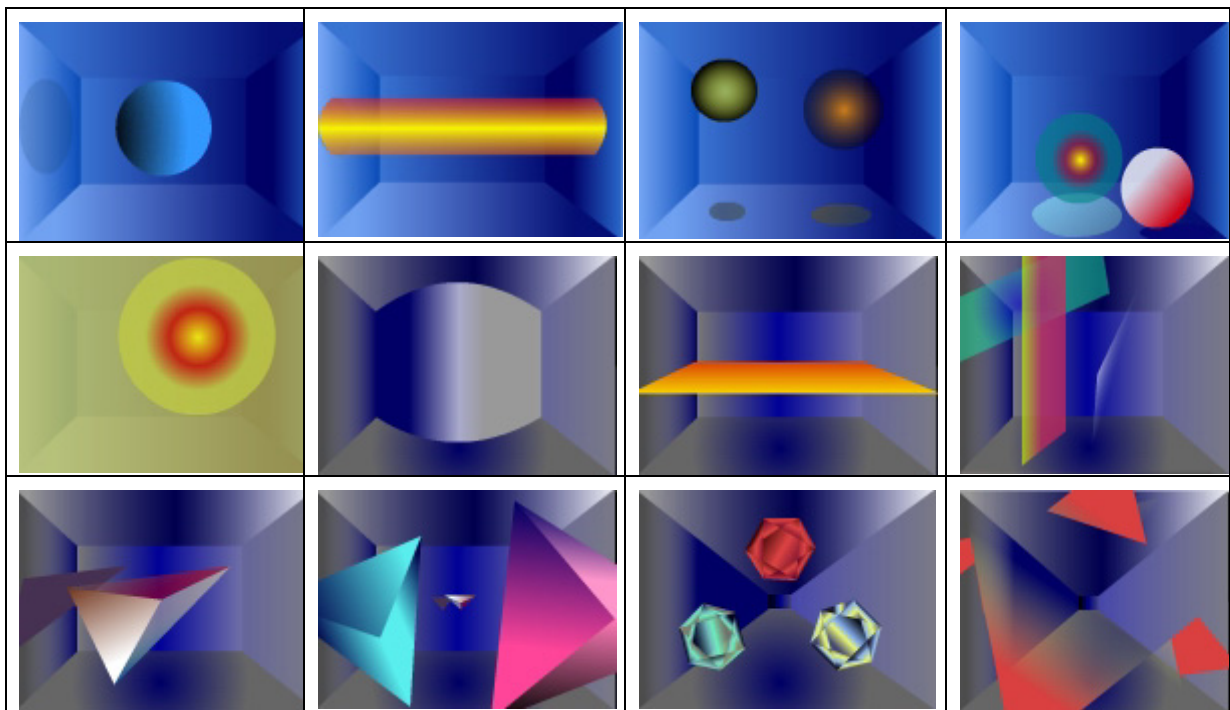


### 1.2.2.6 “Mediateca” (aprile 2003)

L'anno seguente è il momento della riproposizione completa del desiderio per lungo tempo accantonato. Per l'inaugurazione di una Mediateca realizzo un filmato con il programma di grafica vettoriale bidimensionale Flash<sup>1</sup>, simulando graficamente la tridimensionalità e realizzando una prima forma di associazioni dirette audio-grafiche. Esse consistono nella simulazione stereofonica del movimento spaziale degli oggetti e della camera virtuale che le contiene. Ogni evento ha un suo specifico suono, compreso il gelido ambiente. La sensazione di prossimità crescente degli oggetti è data dal diminuire della riverberazione e dall'aumentare dell'ampiezza dei suoni. Gli oggetti sono semplici solidi geometrici, roteanti e cangianti cromaticamente. Unica eccezione uno sciame di formiche che si spande sul pavimento virtuale.

Grazie all'accelerazione dovuta alla committenza riesco ad apprendere i rudimenti della programmazione grafica in Flash, che mi tornerà utilissima nella realizzazione del presente progetto. Inoltre affronto le problematiche inerenti alle libere associazioni audio-grafiche.

Di seguito alcuni frames del filmato:



### 1.2.2.7 “Unreal-Time” (maggio 2003)

In coda ad un concerto nel quale venivano presentate le produzioni di musica elettronica mie e di un altro autore, effettuate con il controllo della spazializzazione mediante 4 canali audio, abbiamo tentato con due computer una interazione digitale tra le personali prerogative assegnate loro.

Gli algoritmi creati mi serviranno per un lavoro di interpretazione in un brano per PC e orchestra seguente.

---

<sup>1</sup> Flash, della Macromedia, oggi Di Adobe è il programma utilizzato per la realizzazione del presente progetto “Butterfly-system”. Quella cui qui si accenna è il primo tentativo in tempo differito.

### 1.2.2.8 “Out of control” (marzo 2004)

A 20anni di distanza, ritento una performance solitaria in real-time, questa volta utilizzando le risorse digitali, nello specifico un computer che con un programma di sintesi algoritmica e dei *plugin* mi rende possibile un controllo in tempo reale dell'esecuzione di un brano di Chopin realizzato digitalmente da Pietro Grossi. Pur essendo sul palco con il portatile, la performance è tanto affascinante quanto guardare un impiegato lavorare al PC<sup>2</sup>. Non viene resa conoscibile alcuna relazione tra le operazioni e la musica elaborata e l'unico feedback del pubblico consiste in un “oh” alla presentazione finale del brano originale di Chopin che viene spogliato gradualmente dei filtraggi. L'operazione mi serve a verificare la stabilità del sistema e le possibilità di seguire una partitura, nonché i modi di realizzazione della partitura operativa stessa.

### 1.2.2.9 “Le metamorfosi” (aprile 2004)

L'indagine sulla forma dello spettacolo questa volta si sofferma sull'utilizzo teatrale e sulla parola delle capacità elaborative digitali. In questo caso sono però presenti un flautista ed un attrice, che giustificano in qualche modo la visione di ciò che si ascolta, rendendo palesi tutte le trasformazioni elettroniche, del flauto e della voce realizzate in tempo reale.



### 1.2.2.10 “Motus nervi et cordis” (giugno 2004)

Insieme ad un chitarrista (Pablo Montagne) anch'esso studente di musica elettronica realizzo un concerto con musiche elaborate in real-time. Oltre al mouse su PC laptop, utilizzo una macchina (Alesis AirFX) per il filtraggio del segnale audio che si avvale di sensori a raggi infrarossi. E' un modo per iniziare ad avvicinarmi alle problematiche del gesto musicale e della sua spettacolarizzazione. La particolarità della macchina non consente ancora una ben visibile e significativa correlazione tra il gesto aereo (nel range di soli 20 cm. circa) e la sua funzione. Delle riprese del concerto ho realizzato un video.

Di seguito due foto della periferica utilizzata con la mano sinistra.



<sup>2</sup> Ringrazio W. Ciancusi per avermelo fatto notare.

### 1.2.2.11 “Film” (novembre 2004)

Una svolta importante nella mia ricerca avviene con la sonorizzazione di un film muto di S.Becket – A.Schneider degli anni '60. Pur essendo voluto muto dagli autori, la mia interpretazione di questo film, la cui trama consiste nel progressivo nascondersi alle persone e agli oggetti da parte del protagonista, mira ad amplificare il concetto di sottrazione aggiungendo tutti i suoni realistici delle azioni e quelli da me immaginati per ogni oggetto, ambiente e animale, per poi farli sottrarre contestualmente alle azioni del protagonista. Si è operata una sorta di “rendering” sonoro totale delle immagini, calibrandone le energie acustiche in riferimento proporzionale alle superfici occupate nei frames. Le uniche musiche strutturate (tonalmente, anzi modalmente) sono le citazioni di Satie in relazione ai ricordi suscitati dalla visione di fotografie del protagonista. Tutto il resto del lavoro presenta fasce sonore non articolate ma proporzionalmente, dinamicamente e sincronicamente calibrate.



### 1.2.2.12 “Piet” [progetto] (dicembre 2004)

Questa volta è la pittura di Piet Mondrian, con la sua negazione della terza dimensione e di qualsiasi altra sfumatura corporea od antropomorfa ad affascinare la mia curiosità e a spingermi a progettare un brano audio-grafico su una sua composizione pittorica resa dinamica mediante la grafica vettoriale. Linee nere che scivolando nel quadro ed intersecandosi creano e variano continuamente le aree di colori saturi. Realizzata una parte del filmato con il programma di grafica vettoriale Flash, la composizione musicale, dopo aver stabilito associazioni tra colori e loro superfici si è arresa di fronte alla difficoltà di rendere in qualche modo le caratteristiche linee nere: silenzi ritmici? Interruzioni?

Di seguito alcuni frames del filmato in sequenza:

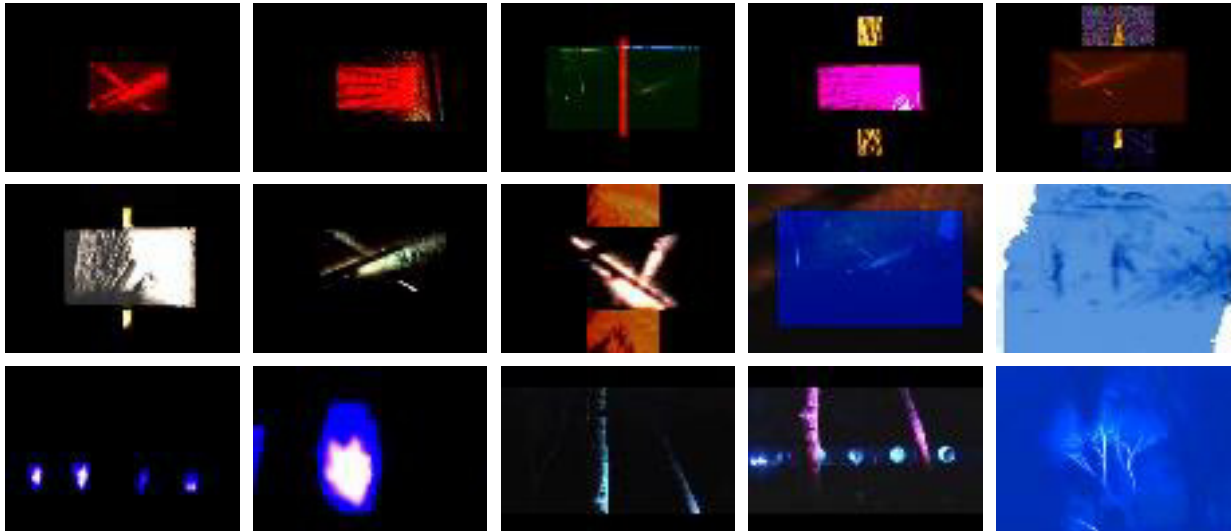




### 1.2.2.14 “Laria” (giugno 2006)

L'ultimo lavoro attinente alla relazione suono-luce è un brano per suoni elettronici dove ad una sezione iniziale di costrizione su un solo canale audio segue una espansione ed invasione dell'intero campo acustico. La rappresentazione prevedeva una regia luci composita. In seguito ne ho elaborato una composizione audio-video che ne amplifica i concetti salvaguardando la sincronia e le relazioni timbriche.

Alcuni frames del filmato realizzato sul brano audio:

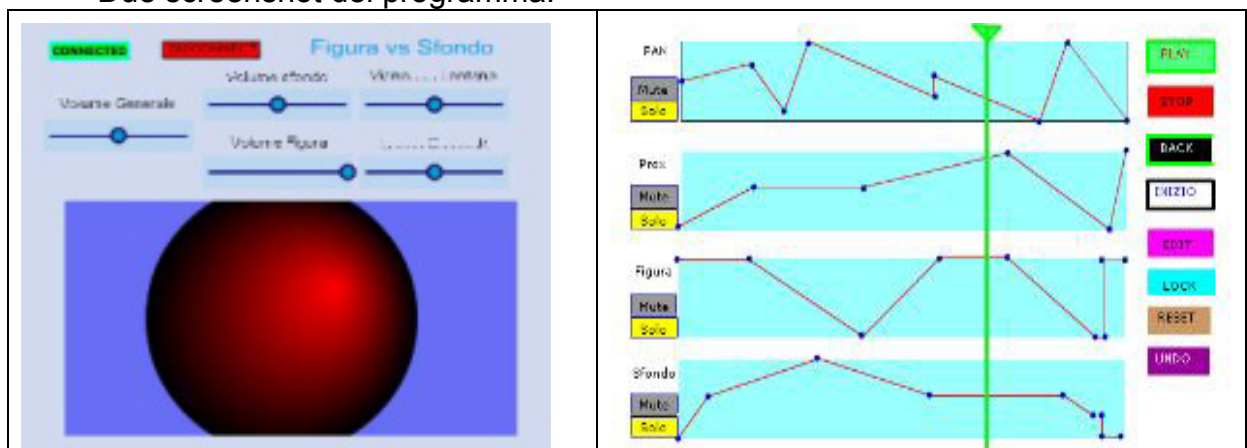


### 1.2.2.15 “Progetto educativo di approccio formale alla musica elettronica, un sistema plurimediale integrato” (marzo 2007)

Già in fase di ideazione del presente progetto e nell'ambito del corso di e-Learning tenuto dal Prof. M. Giordano ho realizzato un sistema software per sperimentare un diverso modo di proporre l'educazione alla musica in generale, ed in particolare alla musica elettronica.

Trattasi di un interfaccia software su connessione con protocollo Open Sound Control che già utilizza i tre componenti software del presente sistema: la grafica vettoriale di Flash, la composizione algoritmica di Max ed il protocollo OSC. In questo caso l'interfaccia fisica è quella tradizionale del mouse che agisce su fader virtuali.

Due screenshot del programma:



Dalla fine del 2006 a metà del 2008 ho lavorato alla realizzazione del presente progetto.

### 1.2.3 Una possibile sintesi

Delle tipologie su esposte riguardanti i precedenti sforzi di creazione di sistemi di controllo [1.2.1], la mia proposta, maturata nel corso di quasi un trentennio [1.2.2], tende a conservare di entrambi quelle qualità che meglio siano percepibili dal pubblico. Ovvero il mio lavoro ha come punto di partenza proprio la percezione dell'evento musicale. Ma la percezione dell'evento musicale avviene sempre in un contesto "visibile" cioè raramente si ascolta la musica al buio assoluto. E' quindi a mio avviso indispensabile valutare l'importanza che ha per la percezione complessiva di qualsiasi evento, anche e soprattutto il contesto visivo dell'ascolto. Questo sarà costituito da una ambientazione fissa (un auditorium con la propria risposta acustica) o variabile ( un percorso all'interno di una installazione), da interpreti visibili o nascosti (strumentisti, interpreti informatici) e da opzionali diffusori acustici (*audio-reinforcement* o funzione diffusione supporto digitale e sintesi *real time*) di varia forma e tecnologia, anch'essi visibili o celati allo sguardo.

Non avendo dietro di me una quantità sufficiente di lavoro per poter spingermi nella pratica di una più "democratica" e rischiosa scelta di coinvolgimento nella fruizione musicale (come ritengo essere quella di Lupone), nella presente trattazione si concentrerà l'attenzione solo su di uno di questi aspetti, quello dell'interprete visibile, perché, quando gli altri aspetti rientrano in situazioni non estreme (*grand canyon*, parco sottomarino, 4000 m. di quota etc.) a mio avviso è il percettivamente più importante nella fruizione musicale. Nel momento in cui c'è uno o più interpreti visibili entra in gioco un altro aspetto del fare arte, che possiamo riassumere nella parola "Gesto".

## 1.3 Considerazioni sul gesto, lo strumento e la visione della musica.

Il gesto umano è parte del linguaggio del corpo, la parte del linguaggio non-verbale maggiormente studiata. A manifestarlo sono le mani, il nostro principale mezzo di interazione fisica col mondo, l'unico che può modificarlo in maniera certa<sup>3</sup>.

Per estensione si dà alla parola *gesto* la proprietà di simboleggiare la continuità e l'andamento di una qualsiasi attività, sportiva (gesto atletico), artistica (il gesto musicale, la pennellata di un pittore). In questo modo ci si riferisce ad una azione fisica o mentale che immediatamente vuol farci individuare una personalità, un carattere che appunto caratterizza una persona. I gesti di una persona, quindi, trasmettono volontariamente o meno, un suo stato psichico, comunicano inconsapevolmente o creativamente un proprio pensiero. Sono cioè del tipo immediato<sup>4</sup>, non hanno bisogno di ulteriori traduzioni ma necessitano, come tutto ciò che è umano, di interpretazioni e prima di tutto di volontà/desiderio di ricezione.

E' stata proprio la sua immediatezza a fare entrare la musica in contatto con il gesto, per la loro comune natura *altra* dal linguaggio verbale, quest'ultimo dotato di regole condivise e funzionali. Ma la musica è fatta da strumenti, ad eccezione del canto che è fatto dal corpo reso strumento, e gli strumenti necessitano di tempo per

---

<sup>3</sup> Il linguaggio verbale ha solo qualche remota probabilità di influenzare indirettamente delle modifiche parziali dello stato fisico, quando invece entrano in gioco gli arti il mondo inizia a tremare.

<sup>4</sup> Il linguaggio dei gesti dei non udenti, essendo del tipo artificiale e cioè mediato dai diversi linguaggi verbali (ne esiste uno diverso in ogni lingua), non interessa il presente scritto.

essere costruiti, compresi ed appresi, di tradizioni secolari. La loro *immediatezza* è appannaggio di pochissimi eletti che riescono ad incontrarli e *indossarli* con facilità. Quel che più ci interessa evidenziare è che la natura dei gesti strumentali non amplifica quella di gesti *immediati* come si è detto intimamente legati al pensiero ma li costringe a posizioni funzionali e appunto *mediate*. Lo strumento, del resto, già prima partecipa alla scelta della creazione imponendone ambiti insuperabili (la tastiera limita gli intervalli superiori alla nona ed incoraggia le ottave). Quando il gesto incontra la musica, passa prima attraverso la penna, che *stringe* in mano, poi attraverso l'archetto che *stringe* tra le dita e solo alla fine riacquista una certa libera immediatezza nel gesto direttoriale o del bambino che ascolta la musica, o dell'ascoltatore adulto che batte il tempo o si muove su di esso.

Quindi il gesto è primariamente visto dall'esterno e, se desiderato, *immediatamente* compreso, ma può anche essere assorbito ed in parte replicato, al fine del naturale piacere che si prova nella condivisione.

Ma il gesto dell'interprete musicale ha una qualche valenza comunicativa? E' cioè rispettoso o partecipe della sua natura di linguaggio non verbale? Io propendo per la negazione se non assoluta, certamente quasi totale di questa affermazione. E' sufficiente guardare in TV una qualsiasi ripresa video di uno o più strumentisti avendo escluso l'audio, per rendersi conto che è impossibile ricevere informazioni dalla sola visione dei gesti degli interpreti: non l'autore della musica, non il tempo o l'andamento, neanche l'intensità per non parlare delle altezze. Solo un occhio allenato riuscirà a ricavare alcune indicazioni dai gesti del direttore d'orchestra. L'unica comunicazione visiva che i gesti degli interpreti ci comunicheranno è quella che stanno facendo musica, con maggiore o minore partecipazione. Questa, che è apparentemente una considerazione banale ed ovvia, a mio avviso è una delle principali cause del disamore verso la musica tradizionale ed ancor più verso quella che fa uso della musica elettronica. A differenza della più abile musica ritmica, che continua a escogitare sempre più imponenti messe in scena per sopperire alla povertà delle formule ritmico-tonali, i concerti di musica *snobbano* con sufficienza la legittima aspettativa di un pubblico che desidera essere coinvolto con tutti i sensi dalla produzione musicale confidando, a mio avviso in maniera eccessiva, sul potere seduttivo della *bella musica*. Tuttalpiù in occasione di trasmissioni televisive, si tenta di sopperire con visioni coreografiche, secondo i dettami di una impresentabile estetica della fruizione ottocentesca, quella del *Grand Opera* e dei balletti.

Non capisco come si riesca a seguire le iperboliche evoluzioni della musica quando di fronte si trova un violinista calvo, sudato e corrucciato, senza che il nostro pensiero sia inevitabilmente dirottato sull'aspetto *circense* dell'esibizione, con pensieri d'ammirazione per la bravura, considerazioni sulla mole di lavoro fatta dal musicista e paragoni con altri, appunti di sartoria sullo smoking e *gossip* sulle storie sentimentali dell'anziano mostro sacro.<sup>5</sup>

E' vero anche, come in precedenza ricordato, che le percezioni sensoriali non sono a "tenuta stagna", cioè semplicemente additive, ma interagiscono tra loro. Ma come interagiscono tra loro? Quando si guarda un interprete che usa uno strumento, solo in parte si svelano i meccanismi fisici che sono alla base della produzione dei suoni, il "fascino" di questa visione è paragonabile a quello che cattura l'attenzione dei pensionati nei lavori in corso, la curiosità di vedere un martello pneumatico o un *caterpillar* a lavoro. Cioè l'interazione percettiva tra il "vedere come fa a suonare" e quella sonora si trova comunque sbilanciata e lacerata tra un meccanismo più o meno semplice e la meraviglia suscitata dalla musica, col risultato che gran parte

---

<sup>5</sup> Io personalmente ai concerti chiudo gli occhi a meno che la musica sia brutta e le interpreti belle. Quando raramente capita una contingenza positiva di questi elementi sono continuamente distratto da entrambe.



della capacità (e volontà) attenta dell'ascoltatore è sprecato nel tentativo indotto della correlazione tra i sensi, che ricordo, per tutti gli animali con sistema nervoso centrale, essere di vitale importanza.

Queste considerazioni sono state alla base di tutto il lavoro che mi ha portato a progettare il sistema "Butterfly". La necessità di ripensare con modestia la modalità di creazione, interpretazione e fruizione del fatto musicale, pur restando nell'ottica di un interprete di fronte ad un pubblico seduto, quindi non ancora interessandomi alle più moderne e innovative scelte di liberazione dalla forma tradizionalmente frontale e unidirezionata, scelte che sono proprie delle *performance* e delle installazioni d'arte. Volevo ancora indagare le possibilità di dare un *corpo* congruo e non posticcio *all'anima* musicale ed ipotizzare un eventuale esportazione delle idee e delle tecniche acquisite anche per la fruizione di musiche diverse dalle mie, magari quelle del periodo classico o romantico, o in modo più filologicamente corretto, quelle di altri compositori contemporanei.

## 1.4 Relazioni possibili audio-grafiche ed implicazioni psicologiche

Una correlazione di tipo fisico tra segnale audio percepito e segnale ottico percepito risulta presto impraticabile.

La considerazione che si tratta in entrambi i casi di fenomeni ondulatori aventi determinate bande frequenziali si scontra con la diversità intrinseca dei fenomeni, da una parte di tipo cinetico e dall'altra di tipo elettromagnetico. Dove sembra di incontrare una certa analogia è nella loro interazione col mondo fisico e nella loro possibile strutturazione.

Considerando la normale alla superficie dell'onda che si propaga, la propagazione dell'onda stessa può essere rappresentata da "raggi acustici", che descrivono piuttosto bene fenomeni come la riflessione, la rifrazione e la diffrazione. Il caso più esemplificativo è dato dalla riflessione delle onde sonore, per le quali vale la Legge di Snell.

### 1.4.1 SINESTESIA: metafora, fenomeno percettivo o patologia?

#### 1.4.1.1 Come nasce

Trattando di ideazione di un sistema per la creazione artistica che coinvolge le relazioni percettive acustico-visive, la presente tesi non può ignorare la sinestesia.

Essa nasce sicuramente come figura retorica che consiste nell'accostamento di termini che riguardano due sfere sensoriali diverse. In questa veste è diffusissima in tutti i discorsi che riguardano qualsiasi sfera (*le luci e le ombre ... della sua storia ... della sua musica ... un discorso a forti tinte, il tocco limpido, il fraseggiare luminoso e coloristico ... il sapore antico della sua musica, etc.*).

Difficile invece risalire all'uso di questo termine di origine greca (dal greco *syn*= *assieme* e *aisthesis*=*percezione*; significherebbe letteralmente "percepire assieme, insieme". E' il contrario di *anestesia*, dal greco *anaesthesia* = mancanza della facoltà di sentire, condizione in cui predomina la totale assenza di risposte a qualsiasi stimolo esterno) come indicatore di un particolare fenomeno percettivo in cui uno stimolo, appartenente ad una determinata sfera sensoriale, non evoca solo la percezione nel senso direttamente interessato, bensì suscita percezioni esperite tramite uno (o più) diversi sensi. Pare essere del

1812 una tesi in medicina sull'albinismo ad opera di un laureando albino, nella quale afferma di percepire, insieme alla sorella anch'essa albina, delle formazioni colorate, coatte e costanti contestualmente all'ascolto di numeri, lettere (sillabe) e di suoni di varia natura. E' il primo esempio di sinestesia percettiva nella letteratura scientifica.

Di certo sul finire del secolo XIX la stessa letteratura annoverava centinaia di titoli e l'interesse cresce notevolmente, tanto da rendere necessario nel Congresso della Psicologia Fisiologica del 1890 il tentativo di standardizzare la terminologia sul tema della sinestesia. Fondamentale "*Audition Colorée*" di Mendoza (1890), ricerca sul fenomeno sinestetico più diffuso, quello dell'ascolto colorato.

All'affacciarsi del Novecento la sinestesia venne indagata in tutta la sua tridimensionalità e totalità di esperienza soggettiva, realmente possibile, esistente da Clavière a Krohn, da Argelander a Mahling.

Fu negli anni '30 del Novecento, con il sopraggiungere della corrente del comportamentismo in psicologia, che la sinestesia venne additata come un'esperienza troppo soggettiva, non degna di essere oggetto di studio accuratamente scientifico.

Bisognerà aspettare la fine degli anni Settanta, per vedere lo sforzo di degnare la sinestesia di uno sguardo da parte del mondo delle scienze con la pubblicazione di un libro di Lawrence Marks, "*The Union Of The Senses*"; egli sosteneva che questo fenomeno poteva far luce sulle basi percettive della metafora omonima, e persino sull'apprendimento del linguaggio.

In neurologia, grazie al progresso e agli sviluppi di essa e di affinate tecniche nella medicina nucleare, come ad esempio la PET, tomografia ad emissione di positroni - che la sinestesia, oggi, gode di un rinnovato interesse tra gli scienziati e di una seria investigazione.

Sarà anche Cytowic, negli anni Ottanta, a cimentarsi nel tentativo di riportare in auge la sinestesia grazie ai suoi studi. "*Synaesthesia, a union of the senses*" (1989) e "*The man who tasted shapes*".

Nel 1987, Simon Baron-Cohen e altri compirono dei progressi riguardo la verifica della *genuinità* del fenomeno sinestetico e delle esperienze soggettive dei sinesteti, grazie ad un esperimento che dimostrava in maniera eclatante quanto la sinestesia fosse reale (*Baron-Cohen, Harrison, 1997*).

Dagli anni Novanta in poi, gli studi e le ricerche sulla sinestesia godettero di una maggiore proliferazione, soprattutto con l'apporto congiunto offerto da diverse discipline, dalla psicologia cognitiva alle neuroscienze. Avvalendosi, inoltre, della CT (computer tomography), della fMRI (Magnetic Resonance Imaging), della medicina nucleare e della PET (grazie alla quale è possibile monitorare l'attività cerebrale durante la manifestazione di una percezione sinestetica, verificando la tangibilità del fenomeno) è possibile ora investigare ed avere una concretezza di dati sul fenomeno sinestetico, evitando che tutti gli studi vengano considerati alla stregua di teorizzazioni filosofiche.

#### **1.4.1.2 In cosa consiste**

Si tratta di una singola stimolazione che riguarda una precisa modalità sensoriale; ma, assieme alla risposta del senso direttamente interessato e preposto alla percezione di quel dato stimolo, se ne ottiene un'altra (o altre) relativa allo stesso senso, o in sensi "non appropriati", cioè non direttamente interessati dallo stimolo percettivo primario, che riguardano una modalità sensoriale differente.

Questo significa che le persone interessate da questo fenomeno, dette sinesteti, possono percepire uno stimolo esperendolo in una modalità "poli-sensoriale": per esempio, un suono può generare la percezione non solo acustica, ma anche visiva di esso, evocando dei colori o delle forme dette FOTISMI.

Esempi di sinestesia sono, ad esempio, associazioni suono-colore, suono-gusto, o suono-tatto. In questo caso si parla di *sinestesia cross-modale* o inter-modale.

Solitamente la sinestesia interessa *due modalità* sensoriali di qualsiasi tipo, sebbene nella pratica vi siano combinazioni più comuni e frequenti di altre come appunto l'ascolto colorato (*Baron-Cohen, Harrison 1997*).

Le principali caratteristiche con cui il fenomeno si presenta secondo i sinesteti sarebbero le seguenti:

1. l'esperienza è involontaria ed automatica, cioè sorge contemporaneamente allo stimolo, come una specie di riflesso.
2. la sensazione ha carattere di realtà, cioè è percepita come proveniente dallo stimolo e non come frutto della fantasia.<sup>6</sup>
3. la sensazione è altamente consistente, cioè si riproduce identica a distanza di tempo.
4. generalmente la sensazione è rigida ed unidirezionale, cioè l'associazione si stabilisce tra una data modalità sensoriale (ad esempio acustica) ed un'altra (ad esempio visiva) ma non viceversa" (*Dogana, 1993, p. 20*).<sup>7</sup>

Sulle cause di questo fenomeno si tendono a scartare le prime ipotesi di scarse differenziazioni fisiologiche con "contatti" tra i diversi canali sensoriali, in favore di implicazioni psicologiche (*Marks, 1975*). Ciò che rende arduo il suo studio è l'estrema soggettività di questi fenomeni: in ognuno dei sinesteti ad un dato stimolo si verifica lo stesso fenomeno anche in tempi diversi, ma allo stesso stimolo ogni sinesteta presenta risposte fenomeniche diverse, anche molto contrastanti.

Una delle poche relazioni condivise sembra essere la sensazione di maggiore luminosità dello stesso colore con il raddoppio della frequenza acustica. Quindi un Do centrale che per un sinesteta può essere associato al rosso ed un altro al viola, per entrambi il Do dell'ottava superiore risulta essere dello stesso colore ma più luminoso. A questo punto si dovrebbe supporre che un glissato di frequenze susciti nei sinesteti una sorta di arcobaleno che si ripete ad ogni ottava schiarendosi verso l'acuto e scurendosi verso il grave.

Altre emergenze scaturite della sperimentazione rivelano che, negli stessi soggetti dotati, i suoni alti tendono a produrre immagini appuntite ed angolose, mentre i suoni bassi immagini arrotondate e massicce (*Marks, 1979*), anche la dimensione dei fotismi dipenderebbe direttamente dall'intensità percepita. Infatti, "*la sinestesia non consiste in un'associazione casuale fra dimensioni isolate che appartengono a due domini sensoriali diversi, ma esprime piuttosto l'esistenza di una correlazione fra tali dimensioni*" (*Marks, 1975, p. 308*). Si verrebbero quindi a creare conflitti dimensionali nei casi di suoni acuti e forti o gravi e deboli. [Da notare che nel presente progetto artistico e non psicologico sperimentale si tenta di associare il parametro intensità sonora alla trasparenza più che alla dimensione].

La percezione sinestetica viene scomposta in due elementi da Grossenbacher, interrelati tra loro, l'*induttore* e il *concorrente*, i quali designano rispettivamente *l'evento che induce la percezione e gli attributi sensoriali indotti sinestheticamente*.

Grossenbacher opera una ulteriore distinzione entro la sinestesia, che riguarda la tipologia di induttore e il *modo* in cui suscita il concorrente.

---

<sup>6</sup> Esistono sinesteti che proiettano il percepito al di fuori di se percependolo a poca distanza, ed altri che invece lo vivono come sensazione interna.

<sup>7</sup> Una eccezione a questa sorta di "regola" è stata riscontrata da Baron-Cohen (1996), esaminando un soggetto sinesteta il quale non solo sperimentava la visione di colori quando udiva suoni, ma anche viceversa: quando vedeva un colore *sentiva un suono*. Una patologia, che gli provocava notevoli disturbi, vertigini, nausea e stress.

L'induttore può essere *percettivo* o *concettuale*: nel primo caso, il concorrente è indotto a causa della *percezione di determinati stimoli sensoriali*; nel secondo caso, il concorrente viene richiamato solo tramite l'atto del *pensare ad un particolare concetto*, senza che il soggetto sia realmente esposto ad alcuno stimolo.

La capacità di associare modalità sensoriali diverse, comunque, è facoltà comune anche in soggetti non fortemente sinestetici: un esempio è quello dell'attribuzione più o meno universale di una sensazione termica ai colori; ad esempio il giallo, rosso o arancione sono chiamati "caldi", mentre il nero, il blu e il verde sono considerati "freddi". E' anche vero che ciò potrebbe essere associato ai colori giallo-rosso del fuoco di legna e a quello dell'azzurro del mare, con temperature medie estremamente diverse tra loro. Elementi questi universalmente esperiti e condivisi da sempre<sup>8</sup>. Da notare che invece le percezioni cromatiche dei sinesteti legati alle rispettive sensazioni di freddo e caldo sono estremamente varie (nero, bianco, giallo, rosso...).

### 1.4.1.3 L'ascolto colorato e i fotismi

Chi possiede questa speciale percezione dei suoni riesce a "vedere" dei colori, sia ascoltando una nota musicale, sia ascoltando una canzone o un brano, e anche quando sente un rumore non classificabile come suono. L'associazione del colore al suono o rumore è immediata e involontaria, insopprimibile e non ignorabile. E' come se il suono fosse realmente quel colore, come se quei due elementi corrispondenti a due diversi registri sensoriali costituissero due aspetti di uno stesso fenomeno.

I sinesteti che possiedono il tipo di sinestesia chiamato "*coloured-hearing*", ossia ascolto colorato, non solo sono in grado, talvolta, di percepire un colore (all'esterno del proprio corpo e precisamente collocato oppure nell'occhio della mente), ma spesso *vedono* anche dei *fotismi*.

I fotismi sono delle forme elementari, che non costituiscono percetti di livello elevato (ad esempio un paesaggio, un oggetto o un volto); sono semplici, costituiti da linee, da curve, o da spigolature.

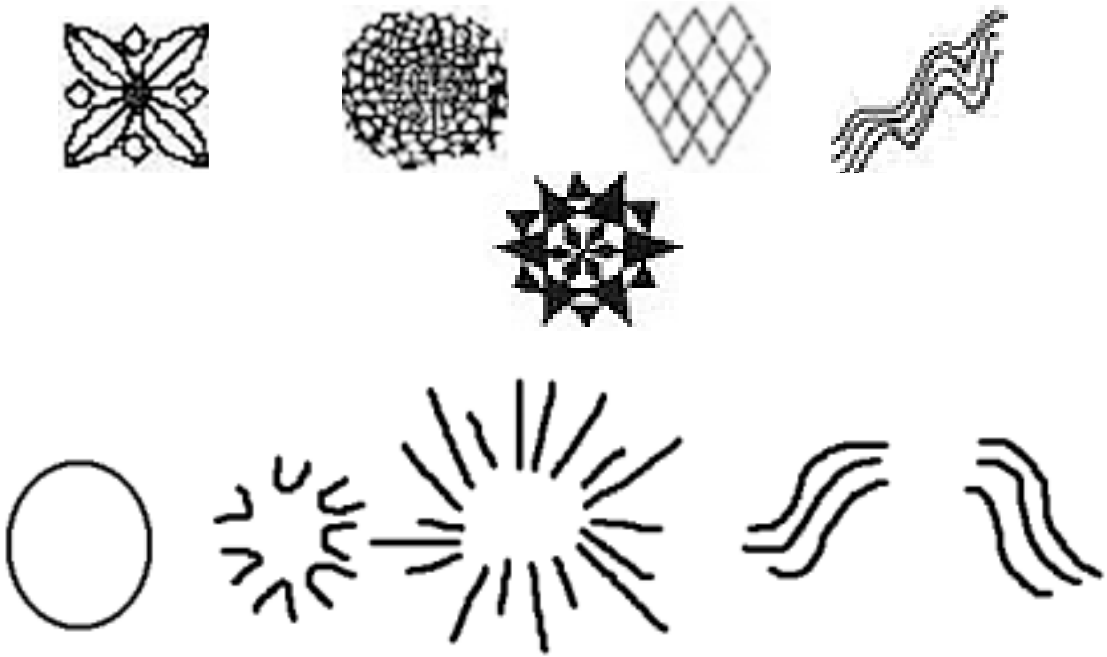
Tali fotismi possono variare in ampiezza e in luminosità, sempre stimolati e condizionati dall'induttore, che è il suono, la musica, il rumore, oppure anche una voce.

Ascoltando una canzone o un brano di qualsiasi genere musicale, i suoni di ciascuno strumento (e anche la voce) sono immediatamente un colore, variando e susseguendosi al ritmo del tempo musicale; la loro taglia corrisponde sovente al volume del suono che è associato al colore, quindi suoni predominanti tenderanno a possedere delle tinte che dominano in ampiezza il campo visivo (o la sensazione interna, l'occhio della mente).

Anche i fotismi e la loro forma ed estensione variano a seconda del volume e del tempo musicale: più il ritmo sarà veloce, più il fotismo assumerà una forma tagliente e spigolosa.

---

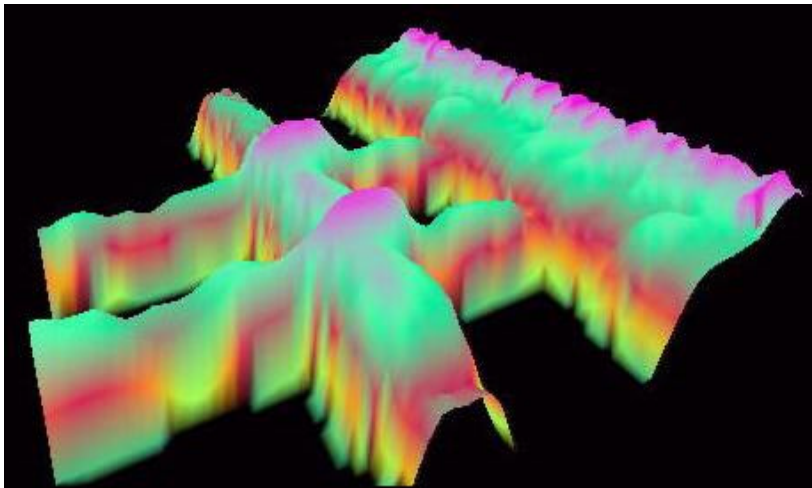
<sup>8</sup> La combustione di gas ed altre sostanze spesso causa fiamme di colori "freddi" e l'acqua della pentola che bolle o trasformata in ghiaccio continua a essere trasparente e cromaticamente inoffensiva, ma sono esperienze recentissime rispetto alla storia dell'umanità.



Griglie, ragnatele, linee, celle come i nidi d'ape, onde sinuose che si duplicano bilateralmente partendo dal centro, figure circolari rotanti... questo è essenzialmente il catalogo dei fotismi di alcuni sinesteti dall'ascolto colorato.

La grandezza e la loro luminosità dipende, come per il colore delle vocali, dal suono cui vengono involontariamente associati e cui corrispondono; tanto più basso sarà il tono musicale, tanto più largo sarà il fotismo percepito dal sinesteta. Se la musica invece sarà veloce, il fotismo assumerà forme taglienti e spigolose; i toni più elevati produrranno immagini più piccole.

#### 1.4.1.4 Sinestesia e informatica



Da metà degli anni '90 con la disponibilità commerciale delle prime schede audio si avviano sperimentazioni per sostenere il computer come lettore multimediale. Il primo software degno di nota e sviluppato fino al 1998 è CTHUGHA (*pron. katuga*) dell'australiano Kevin "Zaph" Burfitt. Non sono riuscito a

procurarmi una copia del programma freeware, ma dalle poche informazioni disponibili e dai *screenshot* sul sito [www.afn.org/~cthugha/](http://www.afn.org/~cthugha/), pare essere basato sulla colorazione e disposizione spaziale (nel 1997 anche 3D come nella foto iniziale) dell'andamento temporale delle ampiezze, cioè una sorta di oscilloscopio cromatico virtuale.



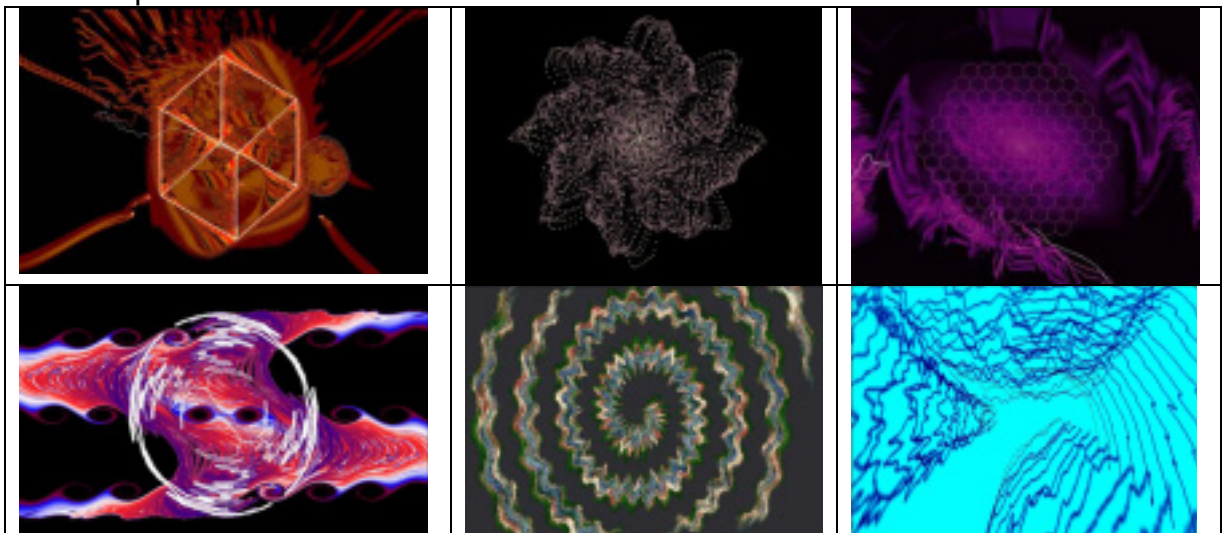
L'abbandono del progetto dipese quasi certamente dalla nascita dei cloni, il più noto dei quali è quello integrato nel Windows Media Player della Microsoft.

Una buona realizzazione è Acid Warp, disponibile per Windows in modalità DOS.

Ricordiamo che il termine warp sta ad indicare una distorsione curvilinea o curvatura.

Irreperibili le informazioni su "Plasmat". Sono tutte realizzazioni che comunque deludono dal punto di vista delle correlazioni, assolutamente casuali ed incostanti. Un modo per riempire lo spazio visivo durante un ascolto. Niente di più. Anzi la critica è quella di deviare l'attenzione, con le informazioni aggiunte e non attinenti, dalle informazioni contenute nella musica.

G-Force, anche come plugin di WMP ed altri lettori o screensaver, attualmente in commercio è nient'altro che uno sviluppo di Cthugha con possibilità di variazione di alcuni parametri.



Probabilmente solo Synaesthesia di Paul Harris si discosta dalla strada di Cthugha, nella combinazione delle informazioni provenienti dalla frequenza e dal bilanciamento stereo. Purtroppo non ne esiste una versione Windows.

Gli effetti grafici, affiancati ai programmi per la riproduzione della musica sul computer, si accostano notevolmente a ciò che accade durante una sinestesia che coinvolge udito e vista. Ricordiamo che i sinesteti non gradiscono affatto simili visioni, in quanto vengono a sovrapporsi ed entrano in conflitto con le loro personali ed ineludibili percezioni visive.

## 1.4.2 Considerazioni sulla fisiologia della vista e dell'udito

Le funzioni biologiche dei sensi hanno essenzialmente due scopi: quella della percezione dello spazio circostante e quella della localizzazione degli oggetti e del loro movimento. La prima, la percezione dello spazio circostante, serve a conoscere e valutare le possibilità di movimento (principalmente fuga o inseguimento ma anche esplorative) e la seconda la posizione di ciò che può essere utile o pericoloso (cibo/prede e pericolo/predatori ma anche acquisizione di strumenti).

Con il diminuire della capacità olfattiva e la necessità di contatto per l'utilizzo del gusto e del tatto, gli unici sensi rimasti all'uomo moderno per conoscere il mondo a distanza di sicurezza sono la vista e l'udito. La prima, di certo più importante e sviluppata, ha bisogno della luce (in quantità minima e non eccessiva) ed è limitata dalla presenza di ostacoli sul suo cammino ma può coprire in campo aperto svariati chilometri, invece la seconda, l'udito, è totalmente impedita solo dalla presenza di vento forte, o da ostacoli di massa e dimensioni importanti. Anch'essa ha una grande capacità esplorativa spaziale, ma certamente inferiore a quella visiva ed anche penalizzata dall'inferiore velocità di diffusione del suono rispetto alla luce [lampi e tuoni, entrambi distanti anche molti chilometri giungono ai sensi in tempi molto diversi].

E' quasi sempre l'udito a guidare lo sguardo nella ricerca dell'oggetto, raramente il contrario. [Quando si cerca di carpire una voce tra le altre, seguendone il movimento labiale e le altre comunicazioni visive non-verbali]. E' anche vero che la localizzazione acustica (zone di circa 15-20° in orizzontale e anche maggiori in verticale) è decisamente più rozza ed approssimata di quella visiva, tant'è vero che una volta individuata uditivamente la zona approssimativa di provenienza è sempre e solo lo sguardo a poter individuare esattamente l'oggetto sonoro. Questa particolare considerazione, quella della guida acustica allo sguardo, e del successivo affinamento visivo della stessa ricerca, è alla base dell'esigenza di fornire un colore ai suoni. Quando ascoltiamo musica, il nostro sguardo è inevitabilmente attratto dalla sua fonte e essendo il suono privo di luce, cade sugli strumenti, le voci o gli altoparlanti. Essendo la musica un'opera di fantasia, essa ha ben poco a che vedere con la sua origine meccanica. Viceversa, se si sta cacciando una preda volatile o terrestre, l'animale che produce il suono coincide perfettamente con l'obiettivo della nostra ricerca.

L'esperienza limite sempre più difficilmente raggiungibile del buio assoluto è angosciante proprio per l'impossibilità di far assolvere allo sguardo questo fondamentale compito biologico. Per avere una minima coscienza dello spazio circostante non è sufficiente una sola fonte luminosa statica diretta (p.es. una stella) perché di essa non saremo in grado di valutarne né la direzione (essendo percettivamente ferma) né la lontananza (potrebbe essere piccola e vicina o enorme e distante). Neanche una seconda fonte luminosa simile alla precedente aiuterebbe la nostra percezione spaziale. Ciò che è necessario ad una minima valutazione spaziale è la riflessione luminosa. Essa solo ci informa dello spazio circostante, svelandolo in parte con le superfici riflettenti. Intendo dire che più che la luce diretta di una sorgente luminosa, è quella riflessa a fornirci informazioni sulla quantità stessa della sorgente luminosa (e proporzionalmente sulla sua distanza relativa), sulla forma della zona riflettente e la sua distanza relativa ed un'altra serie di informazioni accessorie dipendenti dai singoli fenomeni. Da notare che le informazioni che ci giungono dal suono riflesso hanno esattamente la stessa valenza, anche il colore che dipende dalle proprietà riflessive della materia quando

colpite dalla luce, ha un suo equivalente acustico, derivato dalle proprietà riflesse della forma e del materiale riflettente l'energia acustica radiante.

La percezione dello spazio acustico deriva dall'interazione tra il complesso insieme dei fenomeni di produzione, riflessione, rifrazione, interferenza e smorzamento con conseguenti fenomeni di attenuazione e rinforzo delle onde acustiche nello spazio fisico da una parte e dall'altra tra il complesso insieme dei fenomeni attivi e passivi del sistema percettivo sensoriale e nervoso dell'ascoltatore.

Pur esulando lo studio dell'ascolto spaziale dalla presente trattazione, dobbiamo brevemente ricordare la capacità del cervello di ipotizzare la provenienza di un suono mediante la differenza di fase tra le stimolazioni delle due orecchie, coadiuvata dall'analisi del filtraggio operato dagli stessi organi (padiglioni auricolari) e dal resto del corpo (testa e busto) sui suoni, che lo mette in grado di percepire se un suono è frontale o posteriore, se viene dal basso o dall'alto.

Per concludere, molto prematuramente, un discorso complesso di cui si sa ancora molto poco con certezza, vorrei aggiungere che un'altra relazione sensoriale è data da un meccanismo simile di aggiustamento percettivo. La differente velocità delle due irradiazioni, quella luminosa e quella acustica, non presenta percettivamente sostanziali differenze, perché i nostri sensi sono diversamente tarati per entrambe. Quando infatti si spegne la luce repentinamente, l'occhio percepisce non un brusco cambiamento luminoso ma una veloce dissolvenza al nero, dovuta ai fenomeni di isteresi. Similmente in ambienti chiusi riverberanti, la minore isteresi dell'orecchio è per così dire compensata dal fenomeno della riverberazione, la cui coda assolve al "compito" di interpolare i valori estremi di energie.

### **1.4.3 Effetti di fusione: differenze incolmabili**

Nella fusione di segnali audio ogni formante conserva parte della propria informazione all'interno della risultante. Viceversa nella fusione cromatica il colore risultante risulta solo imparentato con i propri genitori.

La percezione audio può porre una separazione di piani o prestarsi ad un indagine temporale del contenuto spettrale, ma mai così netta come in un quadro di Mondrian, perché anche inviando due segnali audio spettralmente distanti, coerenti e statici a due canali diversi, indipendentemente assegnati a due altoparlanti distanti tra loro, la fusione tra le onde acustiche nel medesimo ambiente determina l'insorgenza di fenomeni acustici quali battimenti e effetti comb che non hanno corrispondenti nella percezione delle aree a colore sature di un dipinto di Mondrian, dove il rosso, pur insidiato da un altro colore da tutti i lati, resta individualmente percepibile ed immediatamente localizzabile e definito. Il suono complesso deve essere inteso sempre come sovrapposizione di piani trasparenti e traslucidi (con proprietà di modificare il suono che traspare), mai come oggetti completamente separati nello spazio. Le uniche separazioni possibili saranno quelle tra primo piano e sfondo e quelle operate nel dominio del tempo, per esempio l'alternarsi tra due canali portanti suoni diversi sarà percettivamente simile al lampeggiare alterno di due luci con colori diversi, ma non se contemporaneamente accese. Il motivo di questa sostanziale differenza consiste nel fatto che la visione eccita aree sensoriali distinte e diversificate cromaticamente della retina [ricettori del verde e del rosso in zone localmente diverse], viceversa l'energia acustica giunge fusa e indifferenziata al timpano, diversificata solo dalla sua tempovarianza, dall'intensità relativa delle parziali e dalla binauralità.



## 1.4.4 Analogie strutturali

Le analogie strutturali sono invece assai più complesse da analizzare in quanto coinvolgono la necessaria attività umana. Mettendo da parte questa, cioè riferendoci a casi di formazioni naturali, notiamo come i fenomeni caotici non producono mai percezioni correlabili istintivamente. Per esempio il rumore del mare in tempesta risulta percettivamente molto più minaccioso della sola sua visione, che assume una colorazione pressoché uniforme o comunque non altrettanto drammatica. Così anche una esplosione colpirà il nostro senso acustico molto più di quello ottico (a meno che non abbia un rilascio di energia luminosa ed avvenga nel buio, teniamo anche presente le diverse velocità di propagazione del suono e della luce!). Uno dei fenomeni fisici naturali che produce una paragonabile sensazione acustica ed ottica è forse quello del fulmine notturno, a patto di essere molto vicino alla scarica elettrostatica. In questo caso la struttura che li rende equivalenti è quella dell'inviluppo temporale, la repentina mutazione di pressione da una parte e di quantità di luce dall'altra. Un'eruzione vulcanica invece è otticamente spaventosa, cromaticamente importante, ma acusticamente debole. Il terremoto viene percepito acusticamente e fisicamente dal corpo, trattandosi di frequenze vicine alla soglia di udibilità, ma visivamente è percepito solo per i suoi effetti sulle strutture.

Bisogna subito dire che la percezione umana non è analizzabile separatamente, in quanto il cervello integra gli stimoli sensoriali e ne costruisce le sue librerie di riferimento. Ma ciò che colpisce resta l'assoluta disparità di importanza che ogni evento procura ai nostri sensi: sarebbe bello prendere il sole sdraiati sul pack nell'antartide, nel silenzio più assoluto e nella luce più intensa ed accecante se non fosse per la sensazione di freddo "bruciante".

La bellezza visiva del fuoco non è paragonabile al crepitio del combusto e soprattutto al dolore capace di causare. Quella dell'arcobaleno o di una notte stellata o di un tramonto tropicale non hanno alcuna altra presenza sensoriale se non quella visiva.

Ma è proprio la disparità di sensazioni prodotte da uno stesso fenomeno che ci giungono alla mente ad essere interessante e a costituire la ricchezza della vita sensibile.

Come abbiamo visto ogni fenomeno fisico presenta almeno uno stimolo "piacevole", una sua possibile via di fascinazione. Voglio dire che forse il tentativo di creare artificialmente una correlazione tra i due sensi, ottico ed acustico, è quasi sicuramente "innaturale". Ma scopriamo perché per me resta importante e cosa si è fatto nel tempo.

## 1.4.5 Passeggiata storica

### 1.4.5.1 Fisica

Isaac Newton creò una corrispondenza tra i colori dello spettro e le note musicali, avvalendosi del prisma e scomponendo la luce nei sette colori fondamentali, che assegnò a ciascuna nota. (Newton, "Optics" 1704)

Nel 1742 Bertrand Castel, matematico e fisico, viene ricordato per le sue sperimentazioni che consistevano nell'associazione di luci colorate ai suoni, prodotti simultaneamente grazie al cosiddetto "*clavecin oculaire*".

### 1.4.5.2 Musica, pittura e teatro

Ma quando si parla di opera d'arte multimediale l'antecedente storico per eccellenza è sicuramente Richard Wagner con l'opera d'arte totale.

...

*Il concetto wagneriano di opera d'arte totale, basato sulla fusione dei singoli elementi artistici, viene però duramente criticato: l'arte monumentale nasce infatti facendo leva proprio sulla differenza tra le arti e non in virtù dell'abolizione di tale differenza. La possibilità dell'arte monumentale sta nella equivalenza di tutti i mezzi artistici sul piano dell'interiorità, integrando, e non fondendo, i singoli elementi (suono, colore, parola, ecc.): si tratta di una sintesi di componenti paritetiche. Non si vuole musicare la pittura o dipingere la musica, bensì ricercare un denominatore comune per tutte le arti che garantisca la traducibilità di un genere artistico in un altro. Quello che viene soprattutto contestato a Wagner è l'aver assommato (su un piano dunque meramente esteriore) effetti aventi il medesimo obiettivo, facendo leva su quello che Kandinskij definisce principio della ripetizione (per esempio quando si sottolinea in musica una forte emozione con un fortissimo). In una concezione dell'arte fondata sulla «necessità interiore» il potenziamento viene invece ottenuto, e spesso in maniera più efficace, tramite «contrasti e composizioni compresi tra i due estremi della cooperazione e dell'azione contraria» oppure lasciando simpateticamente agire il suono interiore dell'opera sulla sensibilità dello spettatore. La percezione dello spettatore è un elemento molto importante nella fruizione di un'opera astratta in quanto a essa il pittore affida il compito di completare il proprio lavoro, di colmare quel vuoto creativo che separa l'opera dallo stadio definitivo di composizione.*

...

*In un capitolo dello Spirituale nell'arte intitolato «Linguaggio dei colori», Kandinskij elabora una vera e propria «teoria armonica» del colore fino a giungere a un accostamento programmatico dei colori con i timbri prodotti da particolari strumenti musicali: il giallo alla tromba, l'azzurro chiaro al flauto, il bianco alla pausa creativa, simbolo di un silenzio carico di nuove possibilità espressive [M. Battistini]*

Da queste riflessioni viene fuori molto bene il dilemma che si pone di fronte a chi intende occuparsi di creazioni multimediali: sommare percezioni diverse ma con dinamiche parallele e sincroniche al fine di potenziare il messaggio o assecondare la naturalezza della percezione poliedrica, giocando con i contrasti sensoriali e lasciando allo spettatore il compito di ricomporre una propria immagine complessiva?

Vie intermedie risulterebbero fuorvianti, perché creerebbero delle attese di continuità che improvvisamente verrebbero disattese. Questo perché nel primo caso, la ricerca della fusione di percezioni diverse per essere funzionale deve tendere a nascondere la propria eterogeneità, nel momento in cui quest'ultima viene invece resa palese diventa impossibile ricostruirla in un secondo tempo, perché viene meno la fiducia accordata istintivamente dallo spettatore alla finzione messa in atto.

Come precedenti di interessi multisensoriali vanno citate anche le esperienze di musicisti come Scriabin e Rimsky-Korsakov e la capacità sinestetica di Olivier Messiaen, che informava gli ascoltatori con titoli ed altre indicazioni coloristiche, ma essendo per lui la presenza dei colori una costante compagna dei suoni, non li avvicinò mai a sperimentazioni in tal senso, pur interessandosi al concetto wagneriano di arte totale nell'ultimo periodo della sua carriera.

In teatro, a cavallo tra Ottocento e Novecento, furono impiegati dei sistemi per

diffondere anche i profumi durante le esibizioni di orchestre, per coinvolgere non solo l'udito, ma anche l'olfatto.

### 1.4.5.3 Cinema e Video



V. Eggeling (frame da "Symphonie Diagonale", 1924)

Ma il momento a noi più vicino è quello della nascita del cinema, delle immagini in movimento. Non possiamo non citare almeno i nomi di chi utilizzò il nuovo medium con intenti artistici sperimentandone le peculiari possibilità e tentandone già la correlazione musicale: Walter Ruttmann con le sue Lichtspiel Opus I-IV(1921-1925) ed in seguito con Acciaio su soggetto di Pirandello e con musiche di Malipiero, Viking Eggeling (Symphonie Diagonale 1924), Hans Richter con la collaborazione musicale di Milhaud e Hindemith per finire con uno degli

artisti totali più in sintonia con la visione in movimento delle arti plastiche: Laszlo Moholy-Nagy.

Ma vediamo come il cinema non sperimentale si è rapportato con la musica nel tempo.

Essendo in un primo momento, nel cinema muto, le pellicole proiettate con poche risorse economiche, la necessità di integrarne la mancanza espressiva della voce, sostituita con stringati sottotitoli pieni di esclamativi, diede al pianista di sala il peso di sostenere la proiezione. Questi, improvvisando armonicamente e ritmicamente al piano, si trovò a fondare dei *clichè* buoni per tutte le occasioni emotivamente simili. L'utilizzo della settima diminuita come *passpartout* per le scene drammatiche, il pedale di tonica per le scene funebri o pre-funebri o comunque conclusive di una vicenda, ed altre innumerevoli situazioni concitate sottolineate da figurazioni ritmiche puntate e dense di eventi/note in tempo veloce, come pure languide e cromatiche melodie nelle scene affettive.

Inconsapevolmente quei pianisti creavano una sorta di parasemantica musical-drammatica, ovviamente sulla scia del melodramma romantico, ma attingendo alla riserva della musica coeva (ragtime-jazz-folk).

Con l'aumentare del budget disponibile per film più importanti, si commissionò la creazione di una vera e propria colonna sonora. E' questo il momento che segna, a mio avviso, la separazione linguistica tra l'immagine ed il suono musicale, ovvero, dopo un primo momento di forzosa ed economica integrazione di basso livello, l'utilizzo di compositori decreta il riappropriarsi della musica di una propria linea linguistica indipendente, ove solo pochi e rari momenti si abbassano ad incontrare sincronicamente gli avvenimenti filmici.

L'innovazione della banda ottica laterale alla pellicola permise di incidere le registrazioni delle voci e dei suoni negli anni '30. La musica per il cinema continuò sulla scia della composizione separata temporalmente e linguisticamente. Solo alcune pellicole di tipo sperimentali tentarono soluzioni alternative, sia sincroniche

che “coloristiche”.

Un fondamentale recupero della sincronia è da assegnare al cinema d’animazione da metà degli anni ’30 in poi, dove la musica di tipo soprattutto rumoristica ottenne enorme importanza nel genere, prima esaltando la fisicità dei movimenti virtuali, quindi ingigandandola fino a deformarne l’universo fisico rappresentato: la comicità viene dall’inserimento di un evento plausibile e verosimile in un contesto inverosimile o viceversa. Come al solito è il contrasto, un riferimento, un *ground*, che genera informazione.

Questa enorme iconografia del cinema d’animazione, con l’andare del tempo ha creato o ricreato una serie di *clichè*, non più legati solo al ritmo e all’armonia ma anche al timbro. (vibrafono e arpa per scene subacquee etc.)

Con il diventare un’industria il cinema perde completamente la sua iniziale artisticità, dovuta alla singola mano dello sceneggiatore-regista-montatore, e l’allontanamento tra immagini e musica diventa sempre più profondo, con l’intento di dare alla musica non il compito di sottolineare la drammaticità delle scene, ma quello di aggiungere informazioni non comunicabili dalle sole immagini. Si va cioè ricreando una finzione del mondo psichicamente vissuto, ove una porta che sbatte non fa più “Sbaam!” ma piuttosto narra la tristezza di chi resta a guardare chi se ne è andato, oppure un tuffo non fa più “Splash” ma tenta di suggerirci le sensazioni di freschezza, gioia, felicità, piacere derivanti da esso.

Altro utilizzo della musica è stato quello descrittivo/geografico, l’utilizzo cioè di strumenti o musiche tipiche di particolari e ben individuabili regioni del mondo.

Riepilogando l’avventura del cinema possiamo dire che si sono avuti due utilizzi del suono, uno umoristico / parodistico / descrittivo e l’altro emotivo / melodrammatico / psicologico. Due utilizzi sempre ben separati, quasi mai fusi (un corno alpino descrive immediatamente l’altitudine del posto, ma molto difficilmente potrà *narrare* le angosce del viaggiatore, viceversa un violino gitano oltre a palesare le origini dei personaggi potrà evocarne l’umore).

Che mi risulti mai si è interessato il cinema di larga distribuzione di possibili associazioni suono-colore o forma musicale - forma delle immagini, se non marginalmente, restando sempre ben saldo l’uso del cinema per la narrazione di una storia, quasi un surrogato del teatro.

Nella produzione di audio-video dagli anni ’60 ad oggi, la sincronia è stata praticamente sempre ignorata privilegiando la possibilità, come sopra descritta, di aggiungere informazioni mediante contrasto più che ridondanza. Anzi spesso si è usata espressivamente la consuetudine televisiva dell’audio sincrono per desincronizzare vistosamente l’audio (parlato e umoristico) dalle immagini, con l’intento di decostruire l’evento rendendolo palesemente irreali (meccanismo dell’estraniamento) e a volte grottesco (meccanismo della decontestualizzazione comica).

Va anche aggiunto che molto raramente si sono viste opere le cui parti video ed audio fossero opera della stessa mano, aumentando ulteriormente la distanza tra le componenti.

E’ questo un punto importante per la mia riflessione in quanto ritengo necessaria la creazione dell’intera opera da parte del singolo autore.

## 1.5 Delle opere d'Autore e delle opere di più Autori

Nel campo multimediale o poli-mediale, è ormai abitudine consolidata la formazione di equipe ristrette di artisti specializzati. I vantaggi di simile abitudine risiede nel potersi avvalere di specifiche competenze difficilmente ottenibili singolarmente, di riuscire a giungere alla creazione dell'opera in minor tempo e poter proseguire la ricerca più speditamente, inseguendo la "cresta della novità" e non ultimo la maggiore probabilità di entrare in contatto con il pubblico presentando un'opera che già contiene diverse e personali interpretazioni. Per fare un esempio, il cinema di Nanni Moretti è obiettivamente sgradevole perché è molto legato alla personalità del suo autore, che sceglie cast, fotografia musiche etc. Solo apprezzando la scorbutica personalità di Moretti ci si può avvicinare ed apprezzare il suo cinema. Viceversa *l'universalità* delle opere della ditta Spielberg (Dreamworks: un nome che tradisce l'intento retorico) è garantita dalla molteplicità delle mani che vi lavorano ai massimi livelli.

Sembrerebbe che per meglio comunicare ed essere recepiti sia necessario diluire la propria sensibilità con quella degli altri già nell'atto creativo, una sorta di pre-apprezzamento pre-digeribilità e quindi omogeneizzazione del prodotto artistico.

Il fatto è che a mio avviso, procedendo per questa strada e proiettandola alle sue massime conseguenze (cioè pressochè identiche all'opera cinematografica di Spielberg) si addiène alla banalizzazione dei contenuti comunicati: è perfettamente comprensibile ciò che è già universalmente risaputo. Allora l'opera risolve in pieno la sua funzione catartica, garantisce cioè il coinvolgimento emotivo e la proiezione al di fuori di noi stessi con conseguente superamento, ma fallisce altrettanto completamente la funzione sovversiva, quella di proporre una alternativa veramente *altra* al pensiero. E per essere "altra" l'alternativa deve essere "singola" cioè personale, perché solo ciò che è legato all'unicità dell'esperienza esistenziale di una singola persona può risultarci in parte "alieno" e quindi ad alto valore informativo. Se sono in tanti a fare l'opera la nostra percezione sarà costantemente attratta dall'empatia verso quello che di noi stessi vediamo riflettibile nell'opera, andando inconsciamente a pescare nei singoli contributi personali quelli più affini. Viceversa se posti di fronte ad un'opera "dura come la roccia" non scomponibile in contributi personali, le reazioni possibili, oltre al più probabile rifiuto, sono quelle dell'accettazione condizionata e critica. Simili problematiche si riscontrano nei rapporti personali, è facile andare d'accordo se si mette da parte il proprio orgoglio e "si vestono gli abiti dell'altro" ma così facendo si nega all'altro la possibilità di confronto, mettendolo di fronte ad uno specchio che riflette e consolida il proprio narcisismo. E' vero anche che ponendosi in aperto scontro nei confronti dell'altro il risultato è pressochè identico eccezion fatta per il cattivo umore guadagnato da entrambi. L'unico vantaggio è che una volta finita "*la singular tenzone*" verbale, nell'intimo di ognuno, a mente fredda verranno più attentamente vagliati i "nuovi" punti di vista effettivamente conosciuti e inevitabilmente verranno in gran parte silenziosamente acquisiti.

La vera conoscenza è possibile solo nella fatica, nella sofferenza e nello scontro. I neuroni amano stare tranquillamente allacciati ai vicini, per creare nuovi collegamenti sinaptici è necessario un qualche piccolo trauma.

Sono queste le riflessioni che mi hanno portato ad affrontare singolarmente l'arduo compito della creazione di un sistema di integrazione audio-grafico, e di prevederne l'utilizzo personale come unico interprete. Non quindi un frutto di egocentrismo, ma la necessità di responsabilizzazione totale ed unitaria della parte creativa. Tutti gli inevitabili e anzi ricercati contributi di terzi sono stati filtrati, digeriti e ri-modellati dalla mia sensibilità e con un'unica direzione espressiva.

## 1.6 Dubbi residui: meglio un libro o un ologramma dinamico con audio 24.1?

A mio avviso paradigmatico di tutta l'evoluzione tecnologica dell'*entertainment*, nel contesto della trattazione, il passaggio tecnologico nel cinema dal bianco / nero al colore nella seconda metà degli anni 30 negli USA ed in generale.

La perfetta percezione delle forme e dei volumi data dal colore ma possibile già con il B/N non ha suscitato come mi sarebbe parso giusto una sorpresa notevole, né nei tecnici né nel pubblico. Del resto la fotografia e ancor prima il disegno monocromatico erano già patrimonio comune. Secondo me la possibilità di accettazione e comprensione di rappresentazioni monocromatiche della realtà, per un sistema percettivo visivo basato sulla tricromia dei ricettori, oltre ad essere dovuta dalla presenza attigua dei ricettori di luminosità, anche dalla filogenesi dell'umanità. Con la comparsa tecnologica del colore nel cinema e nella televisione, già disponibile ovviamente nelle arti figurative da millenni, molti reagirono negativamente, continuando a sostenere la "poesia" del B/N. Questo perché anche la percezione visiva, come tutte le altre, ha una componente creativo-ricostruttiva: la mente, oltre a capovolgere la visione retinica, ricostruisce le informazioni mancanti, e più esse mancano, maggiore diventa la sua libertà creativo-immaginativa. Solo grazie a questo potenziale è possibile vedere forme antropomorfe nelle nuvole o nelle montagne. Questa constatazione depone tutt'altro che a favore di quelle ricerche come la presente che tendono ad aumentare la quantità di informazioni percettive, ingenerando una pericolosa apatia immaginativa e quindi creativa dello spettatore. E' un po' quello che avviene nella seduzione della moda femminile, nascondere al fine di esaltare l'immaginazione e quindi il desiderio maschile. La tanto decantata immersione totale (esperienza totale e via dicendo) inseguita dall'industria dell'*entertainment*, rischia a lungo andare di rendere sterile proprio quell'immaginazione che vorrebbe gratificare. Paradossalmente il miglioramento tecnologico, persino l'alta definizione, invece di migliorare le condizioni di vita o per lo meno di divertimento, annichiscono le residue capacità immaginative della mente. Molto meglio un buon romanzo, dove da asettici caratteri neri su sfondo bianco possano emergere paesaggi personali ed emozioni sconvolgenti.

Probabilmente le tavole di Rorschach in futuro saranno usate per determinare la quantità residua d'immaginazione, calcolata in base ai tempi di reazione, dei pazienti afflitti dalla *multimedial full-immersion syndrome*.<sup>9</sup>

*Quanto tempo ti necessita per vedere qualcosa in questa "macchia"?*



<sup>9</sup> Nome di fantasia, al momento della stesura.

## **2 – PROGETTAZIONE**

## 2.1 Obiettivi

Realizzare un sistema hardware e software di controllo di algoritmi di sintesi audio/grafici per scopi interpretativi / improvvisativi. Un interfaccia fisica gestirà contemporaneamente sia la sintesi audio che la sintesi grafico-vettoriale o video.

La funzione della sintesi grafica sarà essenzialmente quella di comunicare il gesto musicale derivato da quello fisico in maniera metaforica, quindi privata della fisicità distraente perché povera di informazioni delle mani che scorrono potenziometri o variatori di impedenze di varia natura, e resa più vicina alla figura musicale che sta uscendo dagli altoparlanti. Una sorta di mediazione quindi tra l'atto fisico di per se visivamente insignificante e il gesto musicale che ne risulta. Una guida visiva dell'evoluzione dinamica del suono che "amplifica" nella luce e nel colore il gesto dell'interprete. In un primo momento era mia intenzione limitare l'importanza della grafica al solo scopo su esposto. In seguito, con il procedere delle conquiste tecniche e tecnologiche, le potenzialità grafiche del sistema hanno meritato una più libera azione all'interno del progetto, fino, in molti casi, ad indirizzarne la creazione compositiva stessa. Ciò è stato voluto perché resta per me una costante il desiderio di realizzare un'opera con la fusione tra i media già in fase compositiva, cioè senza dare ad una delle componenti la priorità assoluta. Pensare la musica come luci e colori, e pensare il colore come variazione spettrale del suono.

L'obiettivo si può quindi condensare nel desiderio di creare un sistema che permettesse l'interpretazione di una composizione nata con l'inseguire un'emozione e "rivestirla" di suoni e luci.

## 2.2 Generalità del sistema

### 2.2.1 Sottosistema hardware

- Il centro del sistema non può che essere un elaboratore di dati, dotato di processore adeguato e scheda video con memoria on-board, nonché di scheda audio con input MIDI.
- L'interfaccia fisica che controllerà l'evoluzione audio-visiva dovrà essere connessa al computer. Essa dovrà consentire molteplici variazioni parametriche parallele mediante gesti che conservino una certa fluidità e coerenza alle trasformazioni prodotte. Dovrà essere uno strumento adoperato e non indossato dall'interprete, con una tecnica di utilizzo di difficoltà non trascendentale. Ad essa si affiancherà una semplice pedaliera MIDI per i controlli di tipo on/off e incrementali.
- Un sistema di amplificazione e auralizzazione del segnale audio
- Un sistema di proiezione del segnale video.

### 2.2.2 Sottosistema software

- Software di rete: **Windows, Java**.
- Software di sintesi audio: **Max-MSP** con OSC support
- Software di sintesi grafica: **Flash** con Flosc Code.



## 2.2.3 Rilevazione dati ed interfacciamento

Si rende necessario

- un hardware di rilevazione dati (elettronica sensoristica),
- un hardware di conversione dati analogici in dati digitali,
- un protocollo e l'hardware necessario alla trasmissione e ricezione dati tra le macchine
- un algoritmo in grado di comunicare in rete
- un codice in grado di comunicare in local host, sullo stesso computer.

## 2.3 Gerarchia procedurale

Trattandosi di un sistema parzialmente nuovo per quanto riguarda le funzionalità richieste, ma completamente nuovo per le modalità e finalità personali, si è scelto di procedere nella progettazione dando priorità all'hardware necessario, in modo da adeguare e cercare in un secondo tempo o contestualmente le necessità di programmazione.

Ciò che nella lista 2.2.1 risulta subito cruciale perché direttamente responsabile delle modalità di controllo è l'interfaccia fisica. Solo dopo aver focalizzato il tipo e le specifiche richieste sarà possibile progettare il sistema costruito su sua misura, ma che non dovrà dipenderne eccessivamente, pena seguirne l'inevitabile obsolescenza.

### 2.3.1 Interfaccia fisica: obiettivi e problematiche

Compito dell'interfaccia fisica è quello di gestire contemporaneamente sia la sintesi audio che la sintesi grafico-vettoriale che in qualche modo visualizza il gesto musicale interpretativo.

La sua forma deve essere del tipo *strumento* e non *abito*, per poterla utilizzare con libertà senza diventarne schiavo. Gli *strumenti* vengono adoperati con le mani, che restano i nostri principali *ponti* attivi con il mondo, per la loro capacità di movimenti fini e complessi<sup>10</sup>.

Le mani e gli arti che le supportano sono due, sono disposte e si muovono simmetricamente. Il loro utilizzo è sempre stato sbilanciato rispetto alla loro apparente e sostanziale uguaglianza. Almeno da quando l'uomo ha iniziato ad usare strumenti che richiedessero lo sviluppo di una qualche abilità e di una generica precisione.<sup>11</sup>

La tecnica pianistica, che a mio avviso resta quella che ha sondato e sonda le straordinarie capacità di precisione dei movimenti delle 10 dita e delle articolazioni superiori connesse, si è sempre dovuta confrontare con una pre-esistente

---

<sup>10</sup> Oggi con le attuali ricerche bioniche è possibile prevedere incredibili scenari dove tutti i cambi di stato del nostro corpo (temperatura, umidità, battito cardiaco, tensioni cerebrali etc.) possano, opportunamente rilevati, diventare segnali che attraverso una conversione ed elaborazione digitale siano in grado di muovere servomeccanismi di qualsiasi tipo. Resta inalterata comunque la supremazia delle nostre articolazioni, finché non riusciremo a sudare o a muovere le palpebre con la stessa precisione e facilità con la quale inseriamo ... un filo nella cruna di un ago o saldiamo un resistore da 1/4".

<sup>11</sup> Mi riferisco all'arco che per la sua asimmetria di utilizzo, a differenza della clava, deve avere inciso molto sullo sviluppo cerebrale della lateralità, iniziando così un percorso di specializzazione di un lato a scapito dell'altro.

configurazione non solo morfologica del pianoforte ma soprattutto musicale. Intendo dire, che nonostante fosse possibile sviluppare una reale autonomia e parità di abilità di entrambe le mani, la quasi totalità della letteratura tecnico-strumentale assumeva come paradigmatica la direzionalità sinistra-destra basso-alto. In questo modo ha sempre sottoposto a schiavitù la mano sinistra (e nei mancini suppongo la destra), che per esercizio si doveva sempre conformare ai profili melodici e alle esigenze armonico-contrappuntistiche della destra. Mi riferisco al moto parallelo d'ottava terza e sesta. Questo aspetto che a mio avviso origina spesso abilità inferiori della seconda mano, è stato per me motivo di studio che ha portato alla realizzazione di un libro di esercizi simmetrici "**Manus docet**"<sup>12</sup>, nella convinzione che una parità di conduzione e controllo di entrambe le mani, messe di fronte ad identiche situazioni e difficoltà topologiche, possano incrementare le capacità mediante l'applicazione delle soluzioni migliori prospettate dalla mano più abile a quella in difficoltà.

Questi precedenti studi pianistici mi hanno immediatamente fatto propendere per una progettazione simmetrica dell'interfaccia che ponesse le mani di fronte a identiche possibilità di movimenti.

### 2.3.1.1 Quantità di controlli *versus* qualità del controllo

Esiste una proporzione empirica tra gli elementi di questo conflitto. Ovvero l'essere umano messo di fronte a degli oggetti per una determinata durata, riuscirà a scoprirne l'utilizzo o il significato in maniera proporzionata alla loro quantità. La sterminata distesa di 88 tasti del pianoforte diventa meno spaventosa quando si intuisce o si apprende che non solo tutti i tasti effettuano lo stesso movimento e hanno lo stesso significato funzionale<sup>13</sup> ma anche che sono una ripetizione seriale legata a 12 suoni-simboli. Viceversa negli strumenti a bocchino il principale banco di prova risiede nell'eccitatore stesso della colonna d'aria e nelle sue modalità vibrazionali.

Comunque tutti gli strumenti musicali tradizionali hanno una tecnica complessa che sostanzialmente si occupa da una parte del controllo della qualità di emissione del suono e dall'altra della topologia di individuazione delle altezze. Ove, come nel pianoforte, la prima è quasi nulla, si ha la possibilità di concentrarsi sul maggior numero di suoni/tasti da abbassare contemporaneamente.

Traducendo questa problematica nella scelta dell'interfaccia da progettare, ho inizialmente scelto di privilegiare la quantità dei controlli, al fine di poter ottenere una molteplicità di situazioni legata alla elevata capacità combinatoria di essi. Poi, riflettendoci, ho cercato di immaginare in quale stato di imbarazzo avrei posto l'interprete messo di fronte ad una selva di sensori ognuno dei quali con assegnazioni diverse e contrastanti e la cui contemporanea attuazione era generatrice di ulteriori diversità. Proprio la situazione che volevo evitare eliminando mouse, fader e knob.

Decisi quindi di ridurre all'essenziale il numero di controlli, di renderli disponibili tutti contemporaneamente per poter conservare la potenzialità dei gesti delle mani senza obbligarle ad eccessive torsioni ed evoluzioni.

---

<sup>12</sup> Per saperne di più e scaricare il libro in PDF: <http://www.vincenzogrossi.it/dd/mdmain1.htm>

<sup>13</sup> Pur con le differenze di forma dei 12 tasti, la difficoltà è causata dalla loro disposizione non dal movimento del singolo tasto. Del resto è pur vero che basta una tastiera poco equilibrata a mettere in crisi anche un abile pianista.

### 2.3.1.2 Informazione di stato

Ogni periferica che vuol inviare all'utente un'informazione sullo stato raggiunto può essere fisica (leds) o virtuale (oggetto grafico). Per iniziare si utilizzerà l'interfaccia grafica stessa o meglio la visualizzazione del gesto come segnalazione dello stato del controllo. In questo caso si parla di un'informazione sullo stato relativo e non assoluto. Il nuovo intervento sullo stesso controllo raggiungerà i nuovi valori dopo un'opportuna interpolazione automatica.

Deve esserci una relazione tra il gesto esecutivo e il risultato fonico, la gestualità deve essere attinente al suono generato e non alla modalità parametrica, altrimenti il rischio è quello di trasferire sulla figura umana, rivelandola, la disprezzata "artificialità" del suono.

### 2.3.1.3 Feedback fisico e devianza

La fisicità di uno strumento trasmette al suo utilizzatore informazioni ulteriori oltre quelle dello stato raggiunto, come il raggiungimento di un limite e l'andamento lineare o meno della resistenza opposta alle forze applicate secondo le zone di applicazione.

E' da queste informazioni, denominate feedback fisico, che si percepisce come la macchina risponde alle sollecitazioni e con quale devianza rispetto alle aspettative. Quest'ultima è responsabile della qualità dello strumento stesso.

### 2.3.1.4 Hardware e Software per la rilevazione dei dati

Per questo oggetto si sono prese in considerazione le disponibilità attuali presenti sul mercato, che si possono condensare in:

1. periferica con tecnologie varie con controlli continui elettronici [lineari e rotativi] e interruttori [on/off e increment +/-] protocollo MIDI.
2. periferica basate su sensori fisici o tecnologie touch-screen, liberamente configurabili via software, con protocolli di connessione vari (MIDI – OSC – proprietari) e varie risoluzioni (7-10-12-14-16bit)

L'interfaccia analogica con il sistema digitale è stata scelta di tipo sensoristico, per conservare come più volte detto una fisicità del gesto ed una sua immediatezza.

Avendo come di consueto ognuno dei prodotti in commercio pregi e difetti, e soprattutto limiti dovuti al target sempre dettato dall'utenza di consumo (*Djeeing*, *Vjeeing* etc.) , si è deciso di provvedere alla progettazione e realizzazione di un'interfaccia mirata e tarata alle specifiche esigenze del sistema.

Gli avanzamenti tecnologici provocano come di consueto continui ripensamenti e si accetta tale situazione iniziando a costruire oggi quello che completato dopo sei mesi è già tecnologicamente superato, nella consapevolezza imprescindibile comunque che non la tecnologia ispira l'operare e che questo ne è, se di qualche valore, sempre perfettamente scindibile. Guai legare la propria evoluzione di ricerca musicale alle tendenze commerciali del mercato, pena scomparire con la loro obsolescenza.<sup>14</sup>

---

<sup>14</sup> La propria ricerca, iniziata nel mio caso con nastri a cassette di 1/8 di pollice e continuata servendosi delle tecnologie via rete disponibili alla grande utenza, incontra sempre lo stesso problema, la necessità di conversione ai nuovi formati per non perderne le possibilità riproduttive e la conseguente possibilità di *rivisitazione* del prodotto con le attuali tecnologie. Solo una chiara visione e ricordo del proprio operare può impedire di *espandere* un pezzo monofonico su due canali, ma è lecito fare a meno del fruscio dei nastri che per tanto tempo ci ha ossessionato.

## 2.3.2 Connessioni e trasmissione dati: ipotesi di lavoro

Dovendo l'interfaccia fisica controllare le sintesi audio e grafiche e dovendo queste essere correlate si presentano tre possibili scenari.

1. L'interfaccia invia i propri dati parallelamente ai due software preposti alle sintesi residenti su due computer connessi anch'essi tra essi. La connessione che meglio si presta a tale compito è quella di una rete locale (LAN) su HUB, con indirizzi specificati per ognuno dei tre componenti. Il protocollo di trasmissione può essere OSC. La separazione residenziale dei software consente una maggior capacità di calcolo ed una divisione del rischio di crash di sistema, ma può produrre tempi di latenza superiore dovuti alle risoluzioni di conflitti di rete ed alla velocità supportate dal protocollo e hardware Ethernet. Infatti un dato messaggio proveniente dall'interfaccia che raggiunga entrambi i software non può essere rielaborato da uno dei due per essere inoltrato all'altro, pena un'ambiguità del comando ed una imprevedibilità del risultato.
2. L'interfaccia invia i propri dati ad uno dei due software su uno dei due computer che li gestisce, insieme a quelli creati autonomamente, per l'invio al secondo software su secondo computer. In questo caso si riduce il numero di possibili conflitti di trasmissione, assegnando ad uno dei due software l'incarico gestionale della trasmissione ricezione dati.
3. Come al precedente punto 2 ma con i due software residenti su di un solo computer. I vantaggi sono evidenti in termini di diminuzione della latenza e scompare la necessità dell'HUB di rete, potendo effettuare una connessione diretta tra l'interfaccia e l'unico computer. Si richiede solo un computer con processore adeguato e scheda grafica con memoria e capacità di calcolo sufficiente.

Sarà quest'ultima ipotesi ad essere seguita, anche per le raccomandazioni in proposito dettate dalla scelta della scheda di conversione A/D e trasmissione "Make Controller Kit" (vd. 2.6)

## 2.3.3 Tipologie dei software di sintesi audio-grafiche

Tra le varie realizzazioni testate nel tempo quelle che hanno meglio risposto in termini di stabilità e plasticità sono Max-MSP per l'algoritmica audio (che non implementa un sequencer pre-costituito ma dispone degli script necessari alla trasmissione-ricezione dati in forma di pacchetti UDP). La disponibilità maggiore in campo video-grafico sarà materia di sperimentazioni nell'immediato futuro. Volendo affrontare però un tema così complesso come la correlazione audio-grafica in tempo reale ho preferito utilizzare esclusivamente un software che oltre ad essere già praticato consente una notevole capacità di programmazione con linguaggio proprietario (Action Script) simile al C ed in parte ai programmi *object oriented*.

Inoltre dispone del codice necessario alla trasmissione-ricezione dati XML, che con codice esterno di Ben Chun possono essere utilizzati per il protocollo OSC.

E' un programma di sintesi e gestione grafica vettoriale e non implementa capacità di rendering 3D. Con una maggiore dose di lavoro si può comunque ovviare manualmente a tale mancanza costruendo empiricamente proiezioni su superfici di oggetti solidi.

### **2.3.4 La composizione, ultimo e principale obiettivo**

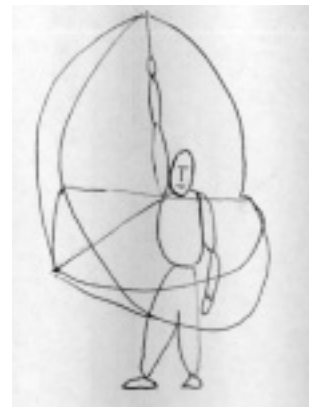
Una volta realizzate e connesse le periferiche necessarie, si aprirà la fase di sperimentazione ed esplorazione delle caratteristiche dinamiche e spaziali del sistema. Il lavoro di ideazione dei pezzi dovrà necessariamente viaggiare in parallelo tra i due media impiegati, avvalendosi della straordinaria possibilità di inferenza reciproca, cioè potendo pensare la musica in funzione di un'idea grafica e viceversa.

Ma le relative considerazioni, in parte già esaminate nei precedenti capitoli, saranno ulteriormente sviscerate in seguito alla sperimentazione del nuovo strumento di creazione disponibile.

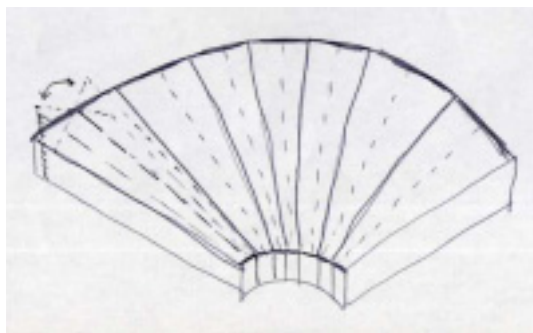
## 2.4 Interfaccia fisica: ipotesi di lavoro

Sono così scaturite quattro ipotesi di lavoro che si sono avvicinate nel tempo, lasciando ogni volta parte di se.

1. Proprio partendo dalla constatazione che le qualità dirette del *gesto* sono conservate in parte dal solo gesto direttoriale, ho pensato ad un interfaccia che ne prendesse le sembianze: una bacchetta, già amplificatore-protesi della conduzione gestuale della musica che diventa gesto creativo e comunicativo al tempo stesso. Un condensarsi delle qualità musicali in un gesto facilmente interpretabile che indirizza l'ascoltatore verso una possibilità in più di ingresso nel mio mondo musicale. Questo *gesto direttoriale* è facilmente associabile ad alcuni parametri di alto livello come andamento e densità degli eventi sonori, disposizione spaziale sia psico-acustica in un sistema audio multicanale che in ambito metaforico (alto = range acuto o forte intensità ecc.).



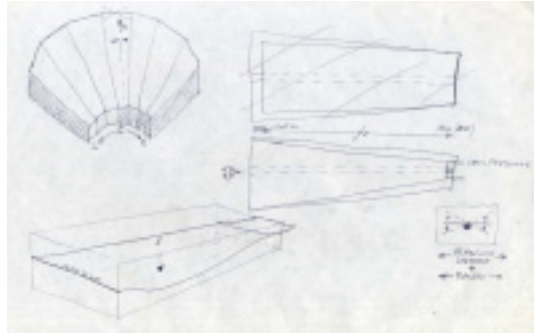
Tecnicamente si era vista la possibilità di utilizzo di sensori giroscopici per la rilevazione delle velocità angolari e deduzione della posizione in tre coordinate x,y,z o la mappatura di una ripresa video mediante l'utilizzo della tecnologia software di Eyes-web. Mi sono però reso presto conto che i vantaggi erano equivalenti agli svantaggi, in quanto l'estrema libertà del gesto di tipo direttoriale o coreografico era molto difficilmente associabile ad un sistema parametrico complesso senza correre il rischio di legare l'interprete alla capacità interpretativa del sistema con le sue specifiche richieste di range predeterminati, di univocità delle relazioni logiche<sup>15</sup>. Non è solo un problema di taratura e di tolleranza o plasticità del sistema è un caricare su di un gesto articolamente semplice da fare con una sua plasticità un potere semantico comunque troppo rigido o in alternativa poco correlato. Intendo dire che difficilmente si riesce a scegliere l'assegnazione parametrica anche di alto livello ad un comune "bel" gesto morbido e sinuoso senza essere o semplicisticamente didascalici (come disegnare un profilo frequenziale di un glissato) o troppo lontanamente correlato (come disegnare un profilo spettrale di un suono).



2. La seconda ipotesi di lavoro parte quindi dalla rinuncia di trovare un gesto che contemporaneamente sia costruttivo ed evocativo-comunicativo. Rinuncia che ha come presupposto anche la decisione di separare totalmente l'interfaccia dall'interprete, ponendola quindi con le

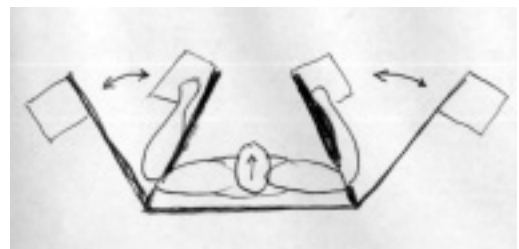
<sup>15</sup> Un movimento lento ed orizzontale del braccio doveva essere assegnato ad uno specifico comportamento musicale dove invece esso, nella interpretazione dei gesti, può benissimo evocare, a seconda del contesto, diverse figure psichiche come calma o continuità o lentezza. Sceglierne una avrebbe escluso le altre dalla libreria comunicativa, dovendo necessariamente salvaguardare l'univocità del codice.

classiche funzioni strumentali e non con funzione di protesi come la bacchetta o l'invisibile abito come quella derivante da una ripresa video. Uno strumento da maneggiare con cura e abilità ma senza esservi inchiodato o appeso. Si fa strada l'idea di sostituire alla visione del semplice gesto nella sua

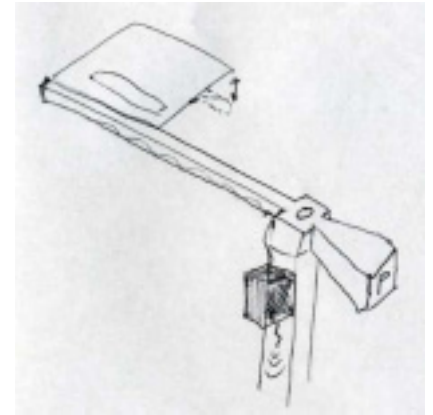
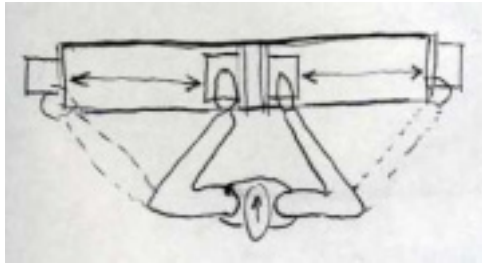


problematica nudità una sua visualizzazione metaforica, che ne assuma tutte le valenze energetiche e plastiche, senza che diventino un ingombrante fardello parametrico astratto. Prende quindi forma l'idea di una tastiera modulare disposta ad anfiteatro, in cui tutti i moduli sono identici nella costruzione ma i cui segnali di controllo possono essere liberamente ed indipendentemente assegnabili. La disposizione ad anfiteatro è ritenuta più ergonomica di quella delle comuni tastiere ed ha una sua iniziale disposizione ad essere interpretata come spazializzazione sia degli oggetti/eventi sonori sia della loro struttura spettrale o di qualsiasi altra associazione di caratteristiche di tipo monodimensionale (A-B, Left-Right, Alto-basso etc.). Rispetto alla precedente opzione come si è detto viene a perdersi la visibilità esterna del gesto (comunque recuperabile con una ripresa video in diretta) ma si può lavorare con oggetti solidi e fissi quindi capaci di contenere molte più funzioni di una bacchetta. Il termine tastiera è molto riduttivo essendo il disegno preparatorio del modulo base legato non alla semplice pressione del grande tasto quadrangolare ma alla sua flessione nei due sensi laterali nonché alla pressione esercitata nella sua lunghezza. Le considerazioni che hanno fatto decadere questa opzione sono di tipo economico, di economia dei mezzi e della capacità di calcolo e anche di portafoglio, in quanto si è considerato che era possibile ottimizzare il sistema sostituendo ad una tastiera con dei tasti la cui utilizzazione statistica sarebbe stata di  $1/n$  Tasti con una unica coppia di tasti alla quale un'ulteriore funzione avrebbe assegnato l'equivalente disposizione spaziale. Invece di 6\_8 lastre fisse una coppia di lastre mobili nelle due direzioni laterali.

3. La terza ipotesi di lavoro vede quindi il condensarsi della tastiera modulare in due superfici utilizzabili dalle mani e la sostituzione della loro moltiplicazione spaziale con un nuovo controllo di tipo rotativo azionato dalla rotazione degli avambracci.

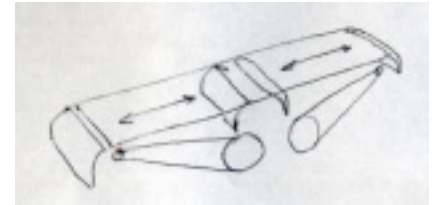
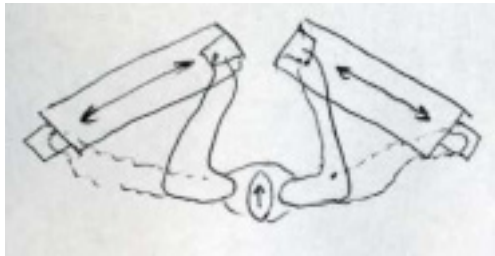


Lo strumento, perché ormai le caratteristiche dell'interfaccia sono diventate tali da essere paragonabili per feedback fisici a questa categoria, si pensò potesse meglio essere utilizzato in posizione eretta, con due bracci mobili dotati di limitatori di estensione meccanici solidali tra loro tramite un asse posizionato alle spalle dell'esecutore .



Risultò presto evidente che la realizzazione di tali bracci rotanti presentava problemi meccanici ed ingegneristici al di sopra delle mie possibilità, dovendo sopportare carichi e forze non facilmente quantificabili, neanche approssimativamente.

La necessità di ridurre i tempi di realizzazione e il coinvolgimento di persone diverse, portò all'abbandono della rotazione degli avambracci e la loro sostituzione con un potenziometro rotativo.



4. La quarta ipotesi di lavoro ha dato vita alla preparazione del progetto di interfaccia fisica denominato “**Little wings**” in omaggio a Jimi Hendrix<sup>16</sup> e per le piccole ali che sporgono dal corpo massiccio del contenitore.

Hendrix a mio avviso è tra i primi musicisti che si confrontarono creativamente con il legame gesto-suono utilizzando i nuovi strumenti. E' proprio il parossismo, certamente spettacolare ma anche fisico-funzionale, delle sue esecuzioni-esibizioni pubbliche a far esplodere la necessità di dare ad una musica dirompente un gesto diverso da quello compassato ed accademico. A capire che la creazione e l'interpretazione del pensiero musicale possono essere meglio trasmesse se quello che si guarda è il più possibile coerente con quello che si ascolta. Da questo punto di vista forse solo la musica barocca, a mio avviso, andrebbe suonata in forme ingessate ed eleganti.

La progettazione dell'interfaccia fisica sarà iniziata dopo l'acquisizione, lo studio e la sperimentazione dei componenti necessari (sensori, materiali di supporto, contenitori).

Detta progettazione subirà notevoli modifiche in corso d'opera proprio a causa dell'evidenziazione dei limiti di alcuni componenti.

Per questo motivo ho ritenuto opportuno esporla nel capitolo di questa trattazione inerente la realizzazione dell'hardware mancante (vd. **3.3.1** Progettazione dell'interfaccia fisica: LittleWings).

<sup>16</sup> “Little Wing” è il titolo di un noto e bel brano di Hendrix.



## 2.5 La conversione A/D ed il trasferimento dati.

Una volta ottenuti dai sensori, tramite opportuna alimentazione ed utilizzo, dei valori in uscita espressi come variazioni di tensione, è necessario convertire questi dati analogici in dati digitali per poter essere usati in ambiente software. Le possibilità sono di due tipi:

1. Utilizzo del convertitore della scheda audio, quindi acquisizione della variazione di tensione come se fosse un segnale audio e sua elaborazione diretta da parte del software di sintesi audio
2. Utilizzo di un convertitore hardware esterno al computer e suo successivo collegamento per il trasferimento dei dati.

Nella musica in real-time è di vitale importanza per la riduzione dei tempi di ritardo (latenza) tra azione fisica di controllo e risposta audio del sistema che i compiti elaborativi siano ben distribuiti tra l'hardware ed il software. Un hardware specifico che si occupa della conversione A/D si è quindi subito reso necessario, per non caricare già in partenza il software di compiti di conversione ed adattamento.

## **3 – REALIZZAZIONE**

## 3.1 Scelta dei componenti hardware

Di seguito sono trattate tutte le componenti hardware utilizzate nella realizzazione del sistema. Alcune, come i sensori e il Make Controller Kit, risulteranno determinanti nella progettazione dell'interfaccia fisica, altre come il computer, la scheda audio, il pedale MIDI, il Mixer, il proiettore video, vengono riportate solo come testimonianza del test di sistema e quindi sostituibili facilmente con altre macchine similari.

### 3.1.1 La sensoristica

Per utilizzare in elettronica e quindi anche in informatica le informazioni sui cambiamenti di stato nel mondo fisico, a qualsiasi scopo e poterli misurare è necessario l'utilizzo di sensori.

Con il termine **sensori** in questa trattazione ci riferiamo solo a quei dispositivi che trasformano una grandezza fisica che si vuole misurare in un segnale elettrico (variazioni di tensione).<sup>17</sup>

Tra le innumerevoli tipologie presenti sul mercato, i sensori presi in considerazione sono stati quelli del tipo resistivo, cioè in grado di modificare, con una certa linearità, una tensione applicata in ingresso ottenendo variazioni di resistenze. I valori in tensione dovranno poi essere convertiti in dati numerici per poter essere utilizzati nei software.

Questi sensori sono costituiti da un sottile film polimerico ed hanno la proprietà di mostrare un decremento nella resistenza in concomitanza dell'aumento della forza applicata alla superficie attiva.

Per le informazioni tecniche sui sensori consultare la guida ai Force Sensing Resistor (FSR) della Interlink Electronics ([www.interlinkelectronics.com](http://www.interlinkelectronics.com)). Sono informazioni essenziali sull'utilizzo e le specifiche tecniche, ma non ho ritenuto giusto tradurle e copiarle nel presente lavoro.

Non essendo stato possibile reperire, al momento della realizzazione, dei sensori in grado di rilevare contemporaneamente punto di contatto (sensore lineare) e pressione esercitata su di esso (sensore di pressione) si è dovuta modificare la progettazione dell'interfaccia e reperire i singoli sensori necessari.

Essi sono nello specifico:

1. SLIDE V.2.1 della Infusion System - Sensore lineare sensibile al punto di contatto con soglia di ingresso.
2. TOUCH v.1.4 della Infusion System - Sensore di pressione sensibile alla pressione esercitata
3. BEND SENSOR (wired) della Making Things - Sensore di flessione sensibile alla flessione esercitata
4. POTENZIOMETER 10 k della Making Things - Potenzimetro rotativo

La loro scelta è stata dettata dall'esigenza di ottenere 3 valori distinti dal movimento/pressione/flessione di ogni mano più un quarto valore sostitutivo della rotazione degli avambracci azionabile come manopola. Si sono scartati sensori di maggiore lunghezza per essere adoperati in poco spazio. Essendone prevista l'implementazione su due superfici sensibili per le mani, ne sono state acquistate 2 coppie per ogni sensore più uno di riserva per ciascun tipo.

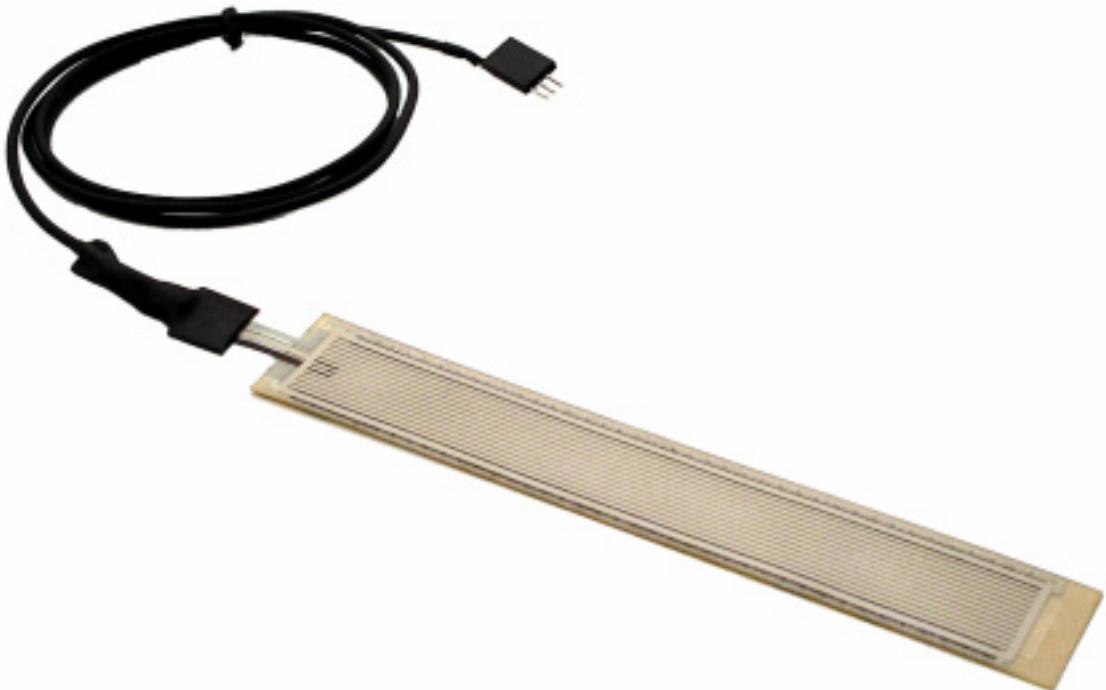
---

<sup>17</sup> Non tutti i sensori restituiscono un valore elettrico, come per esempio il termometro a mercurio.

### 3.1.1.1 SLIDE V.2.1

#### Technical specifications

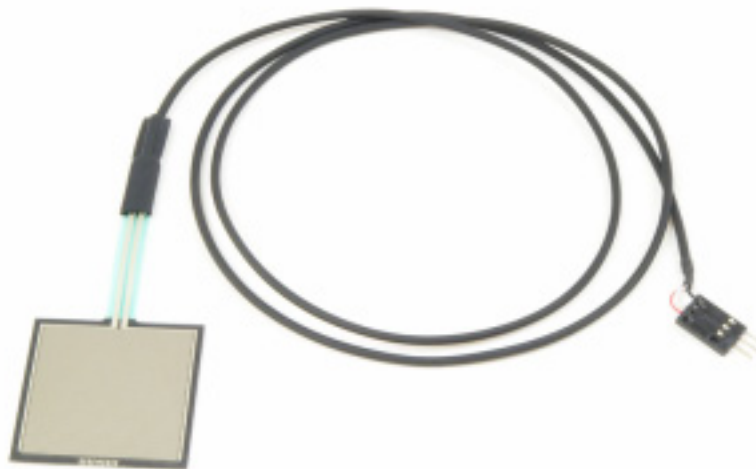
<b>Product</b>	Slide sensor		
<b>Version</b>	2.1 (2005) - for older versions search the obsolete section of our website		
<b>Sensing parameter</b>	single-axis contact position		
<b>Sensing method</b>	resistive		
<b>Active area</b>	140 x 20 mm (5.51 x 0.79 inch)		
<b>Range</b>	140 mm (5.51 inch)		
<b>Output resistance</b>	approx. 20 Kohm (no force applied), 0 to 3 Kohm when moving from one end to the other		
<b>Calibration</b>			
cm	inch	Voltage (use 5 V power supply)	7-bit MIDI value (use 'no processing' editor preset)
no contact	no contact	0.039	1
0	0	0.039	1
140	5.51	5.00	127
<b>Power supply</b>	1.0 to 10 V DC, 3 mA at 5 V		
<b>Sensor dimensions</b>	158 x 26 x 0.6 mm (6.22 x 1.0 x 0.02 inch)		
<b>Weight</b>	15 g (0.5 oz), incl. Cable		
<b>Cable</b>	1.0 m (39 inch), shielded, red wire = power, black wire = ground, white wire = sensor output, maximum extension 30 m (98 ft)		
<b>Connector</b>	male plug with a row of 3 pins spaced 2.54 mm (0.100 inch)		



### 3.1.1.2 TOUCH v.1.4

#### Technical specifications

<b>Product</b>	Touch sensor			
<b>Version</b>	1.4.1 (October 2005)			
<b>Sensing parameter</b>	contact pressure			
<b>Sensing method</b>	zero-travel force sensitive resistor			
<b>Active area</b>	37.5 x 37.5 mm (1.48 x 1.48 inch)			
<b>Range (recommended)</b>	less than 0.98 N (0.10 Kg, 0.22 lb) to greater than 98 N (10 Kg, 22 lb) or less than $9.8 \times 10^2$ Pa (0.010 Kg/cm <sup>2</sup> , 0.14 psi) to greater than $9.81 \times 10^5$ Pa (10.0 Kg/cm <sup>2</sup> , 142 psi)			
<b>Response time</b>	1 to 2 ms (mechanical)			
<b>Output resistance</b>	more than 100 Kohm (at 0 N) to 200 Ohm (at 98 N) in parallel with 47 Kohm			
<b>Calibration</b> (force applied evenly across entire sensor surface)				
N	Kg	lb	Voltage (use 5 V power supply)	7-bit MIDI value (use 'no processing' editor preset)
0	0	0	0.00	0
0.98	0.10	0.22	3.15	80
9.8	1.0	2.2	4.92	125
98	10	22	5.00	127
<b>Calibration</b> (force applied per unit of surface area)				
Pa	PSI	Voltage (use 5 V power supply)		7-bit MIDI value (use 'no processing' editor preset)
0	0	0.00		0
$6.8 \times 10^2$	0.098	3.15		80
$6.8 \times 10^3$	0.98	4.92		125
$6.8 \times 10^4$	9.8	5.00		127
<b>Power supply</b>	1.0 to 10 V DC, 0.1 mA at 5 V			
<b>Operating temperature</b>	-20° to 100° C (-4° to 212° C)			
<b>Sensor dimensions</b>	38 x 38 x 0.5 mm (1.5 x 1.5 x 0.02 inch)			
<b>Weight</b>	15 g (0.5 oz), incl. Cable			
<b>Cable</b>	1.0 m (39 inch), shielded, red wire = power, black wire = ground, white wire = sensor output, maximum extension 30 m (98 ft)			
<b>Connector</b>	male plug with 3 pins in a row spaced 2.54 mm (0.100 inch)			



### 3.1.1.3 BEND SENSOR (wired)

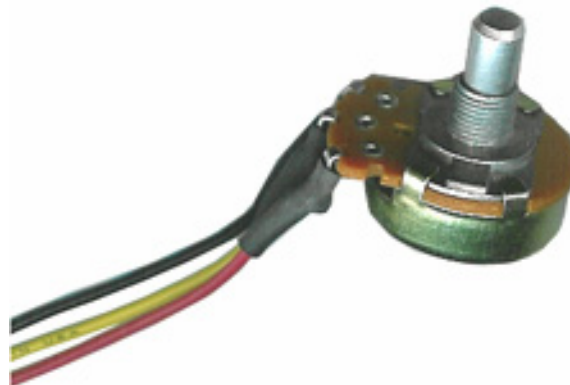


The bend sensor is composed of tiny patches of carbon that change resistance when bent. Its maximum resistance is between 30k and 40k ohms, proportional to how far it's bent. Its nominal resistance is 10k ohms. Resistor and wire assembled, ready for immediate connection to the Make Controller.

The sensor has three wires, which should be connected to the following locations:

- **3.3V** (red) to a Vin on the Controller set to 3.3V.
- **Ground** (black) to a GND connector on the Controller.
- **Signal** (yellow) to an analog input, 0-7.

### 3.1.1.4 POTENZIOMETRO 10 k



10K Potentiometer. Wire assembled, ready for immediate connection to the Make Controller.

The sensor has three wires, which should be connected to the following locations:

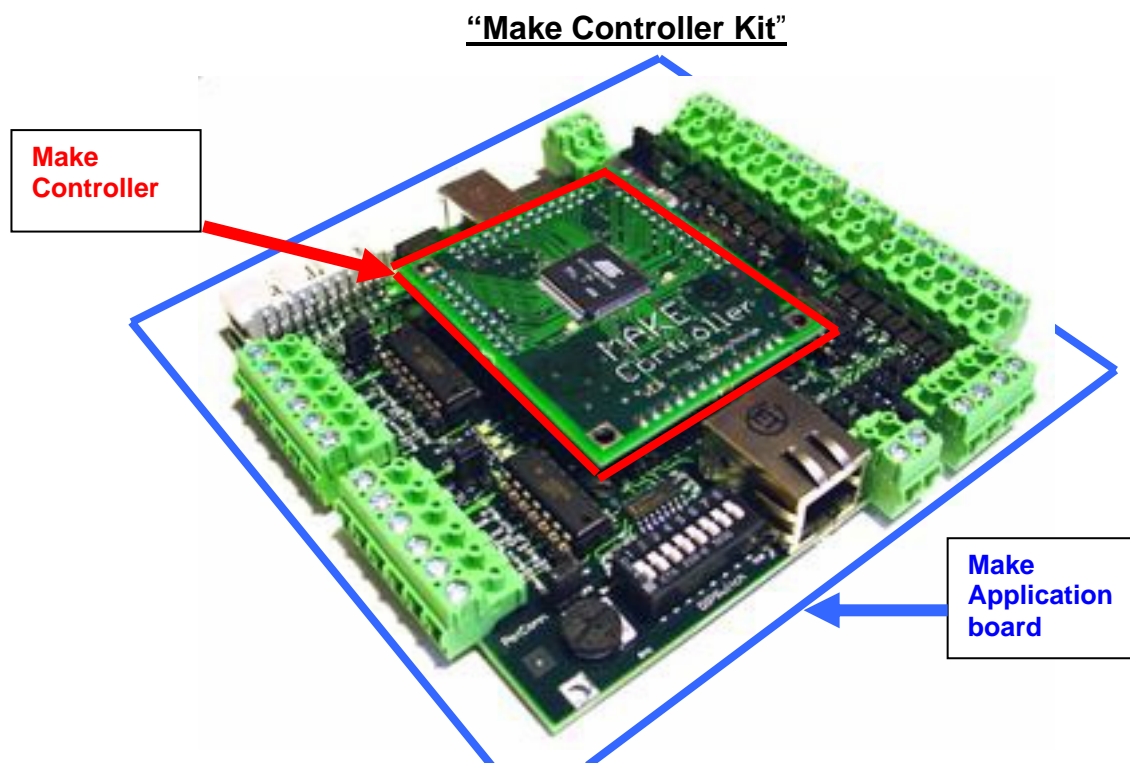
- **3.3V** (red) to a Vin on the Controller set to 3.3V.
- **Ground** (black) to a GND connector on the Controller.
- **Signal** (yellow) to an analog input, 0-7.

### 3.1.2 La conversione A/D e la comunicazione: “Make Controller Kit”

Varie aziende si occupano o si sono in passato occupate di fornire kit hardware sperimentali dotati di sistemi per la conversione ed il trasferimento dati digitali, la maggior parte di esse provvede una macchina pronta al collegamento con il computer, tramite connessioni *firewire* della Apple (IEEE1394b, 786.432 Mb/s), Gigabit Ethernet (IEEE802.3z, 1000 Mb/s) o USB2.0 (480 Mb/s)<sup>18</sup>.

Si è scelto di utilizzare una scheda “nuda”, di costo contenuto, ma in grado di effettuare la conversione richiesta e di gestire i protocolli USB ed Ethernet per la trasmissione e ricezione dati. A determinare la scelta fu anche il supporto per il protocollo **OpenSoundControl** già utilizzato in precedenza e dalle ottime prestazioni.

Detta scheda, fornita di porte analogiche e digitali è la “**Make Controller**” montata su “**Make application board**” prodotte dalla **MakingThings** con il nome di



Il **Make Controller Kit** consiste in una scheda figlia (**Make Controller**) montata su di una scheda madre (**Make Application Board**) come intuibile dalla foto precedente.

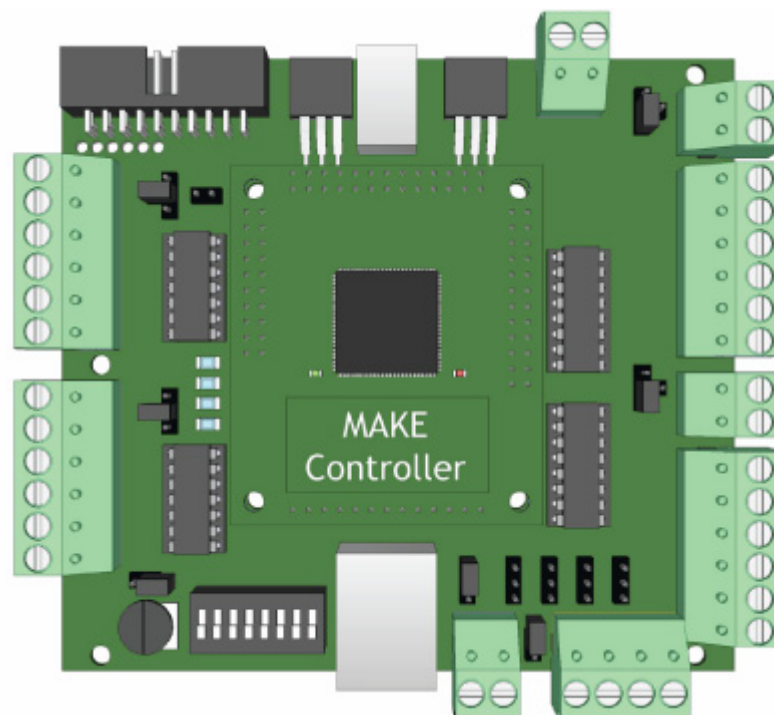
Per evitare facili confusioni le informazioni riportate di seguito saranno contrassegnate dai colori assegnati, rosso per il Controller e blu per il supporto-conessioni.

<sup>18</sup> Prossimamente (2009) USB3.0, 4,8 Gbit/s, 10 volte l'USB2.0

### 3.1.2.1 Make Application Board

La **Make Application Board** fornisce una robusta interfaccia tra la **Make Controller** e i dispositivi reali. Essa offre una varietà di I/O e opzioni di connettività:

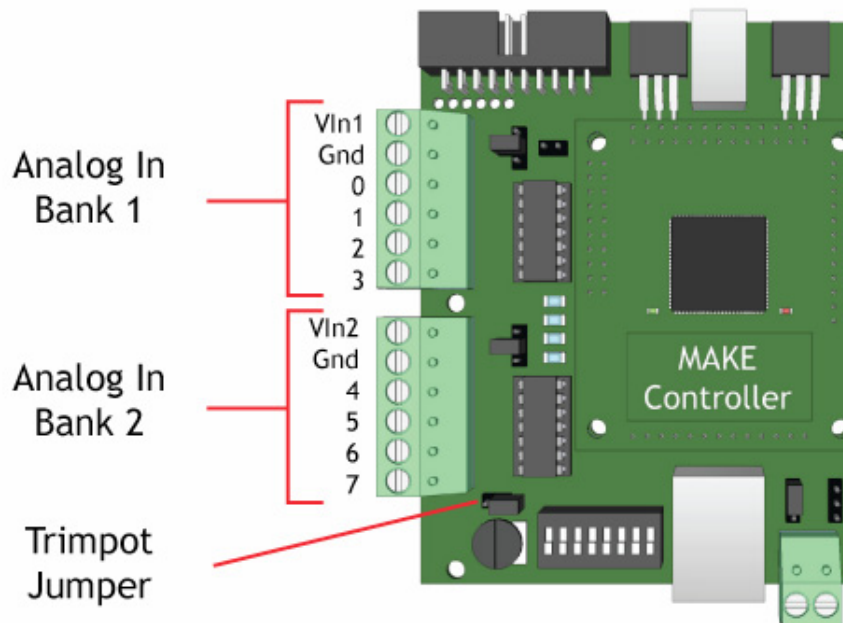
- 8 inputs analogici - convertiti con risoluzione a 10bit e circuito di protezione.
- 8 outputs a corrente elevata - fino ad 1A ciascuno. Può essere combinata per guidare motori a corrente diretta e/o motori passo/passo.
- 4 outputs per servo-motori - controlla fino a 4 standard hobby servo-motori.
- 4 LEDs di stato - fornisce un controllo visivo per i programmi.
- Commutatori DIP e trimpot - per semplici configurazioni manuali.



#### 3.1.2.1.1 Input analogici

La **Make Application Board** fornisce 8 inputs analogici, per leggere i valori dai sensori ed altri dispositivi di input. Ogni input legge voltaggi da 0 a 3.3 V, e li converte in valori digitali con risoluzione di 10 bits, dando valori da 0 a 1023.

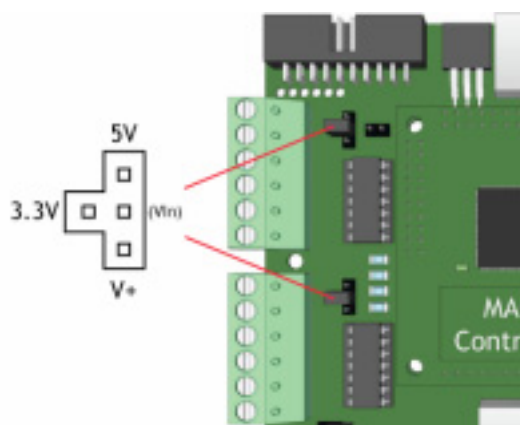




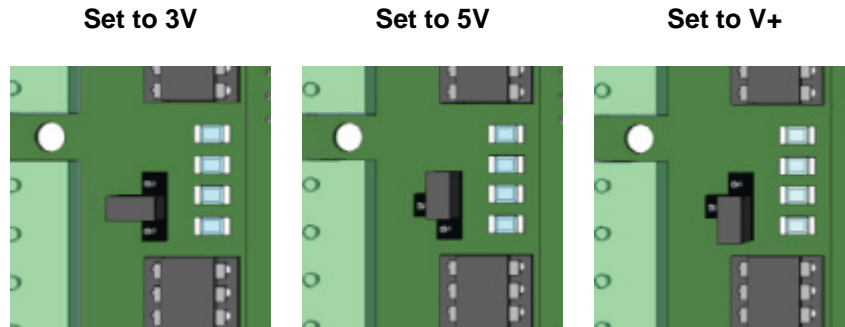
L'input analogico n.7 è normalmente connesso al trimpot, ma si può disconnettere per utilizzarlo con l'ottavo dispositivo esterno in input. Per far questo rimuovere il jumper di plastica nera [Trimpot Jumper in figura connesso al trimpot] e lasciarlo inserito solo al centrale. **Questa opzione è importante sottolinearla per evitare di perdere tempo qualora si connettessero tutti e 8 gli input notando la mancanza di segnale dell'ottavo!**

### 3.1.2.1.2 Alimentazione degli input analogici

Ogni banco di input analogici ha una coppia tensione/terra (**Vin** e **Gnd**) sul connettore per fornire l'alimentazione alla dispositivo d'input connessa alla scheda. Di default, la scheda è settata per fornire **3.3V** a **Vin**, ma può facilmente essere cambiata configurando il jumper di selezione voltaggio. Il voltaggio per ogni banco di input può essere settato indipendentemente ad uno dei tre valori: **3.3V**, **5V**, o **V+**. Il voltaggio selezionato sarà disponibile al connettore **Vin** - **Vin1** per il primo banco e **Vin2** per il secondo.



Per cambiare la posizione del jumper, sollevarlo dalla scheda e riposizionarlo nel verso del voltaggio desiderato. Un lato del jumper deve sempre essere connesso col pin centrale.



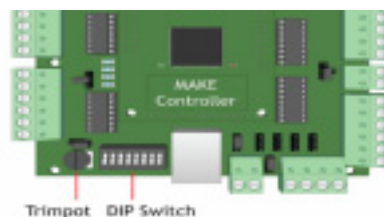
### 3.1.2.1.3 Voltaggi di ingresso

Siccome la **Make Controller** lavora a 3.3 V, ogni ingresso accetta segnali da 0 a 3.3V. Sebbene siano protetti contro sovratensioni, è meglio evitare tensioni al di fuori di questo range. Gli ingressi sono calibrati con tolleranza 5V, per non rischiare di danneggiare la scheda, ma gli input sono comunque sensibili solo nel range 0-3.3V.

Per i sensori di tipo resistivo, ciò non causa un problema, potendoli alimentare direttamente dalla scheda con i 3.3V forniti. Invece per i dispositivi che necessitano di 5V in uscita, è necessario applicare un circuito per giungere al voltaggio appropriato.

*(Specifiche di dispositivi presenti omesse perché non utilizzati nel progetto)*

- Digital Outputs*
- PWM*
- Digital Out Power*
- Servo Power & connect.*
- Dip Switch & Trimpot →*

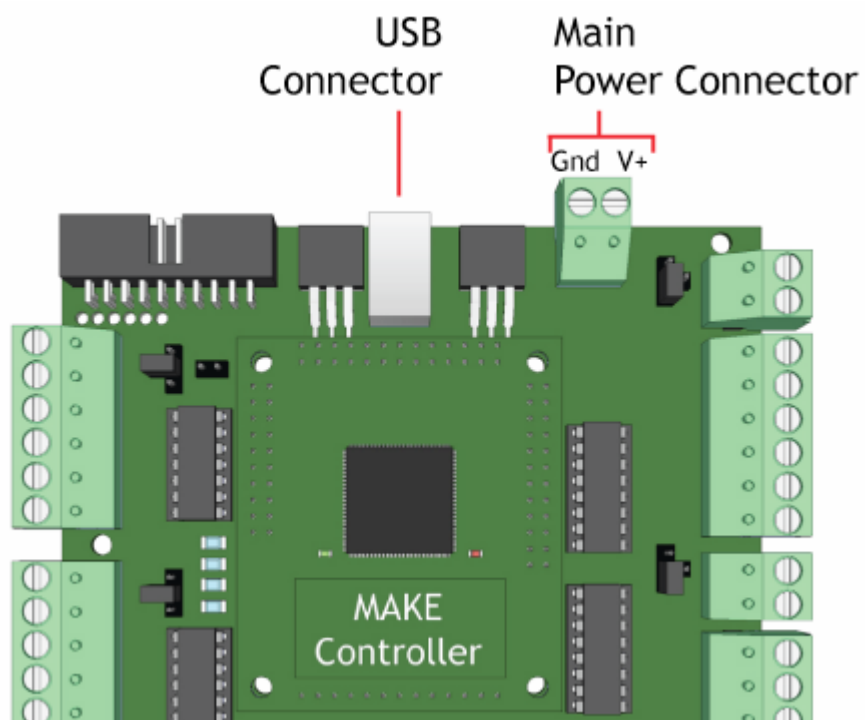


### 3.1.2.1.4 Alimentazione (9V stabilizzata)

L'alimentazione può essere fornita alla **Make Application Board** dal **connettore principale d'alimentazione**, o dal **connettore USB**. Generalmente l'alimentazione proveniente dalla connessione USB basta solo per alimentare la scheda (utile per la programmazione ed i test), ma non per usare dispositivi più potenti di un LED. E' invece necessario utilizzare un alimentatore esterno da 9V stabilizzato connesso al connettore principale. La scheda sopporta tensioni tra i 6 ed i 12V, ma i voltaggi superiori caricano di lavoro la scheda costretta a ridurli alle linee di 3.3 e 5V necessarie all'elettronica della scheda.

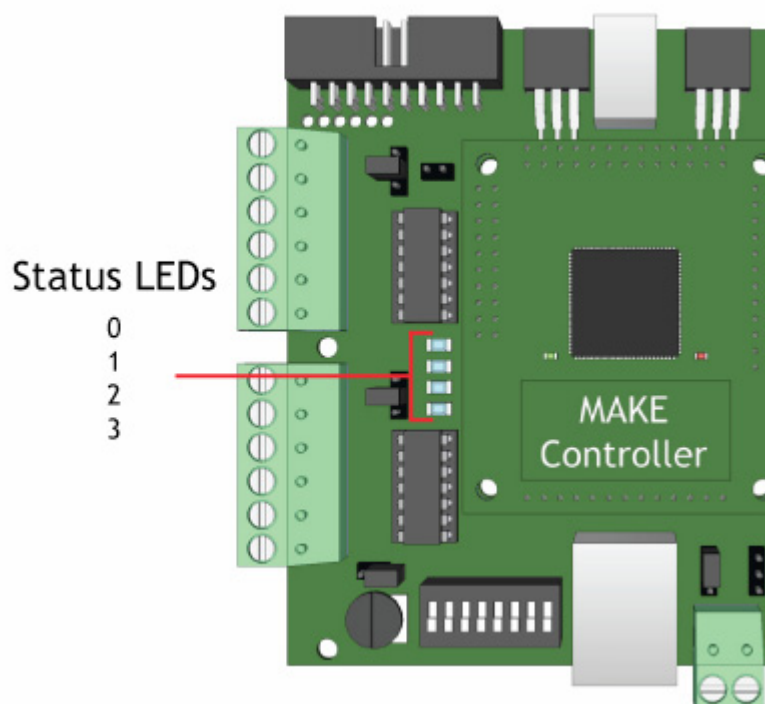
L'alimentatore deve fornire corrente sufficiente per azionare i dispositivi connesse alla scheda. I due sensori lineari assorbono ciascuno 3mA, il TouchPad quadrato per la pressione 0.1 mA. Non sono disponibili gli amperaggi del potenziometro e del sensore di flessione, ma si suppone siano simili ai precedenti. Si ha quindi una richiesta di corrente massima complessiva degli otto sensori inferiore ai 20mA. L'alimentatore utilizzato supporta un carico di 1.1A. Ma sarebbero sufficienti anche soli 100 mA. Nella figura il connettore USB e quello bipolare per la tensione

cui collegare l'alimentatore esterno.



### 3.1.2.1.5 Componenti per l'utente: Status LEDs

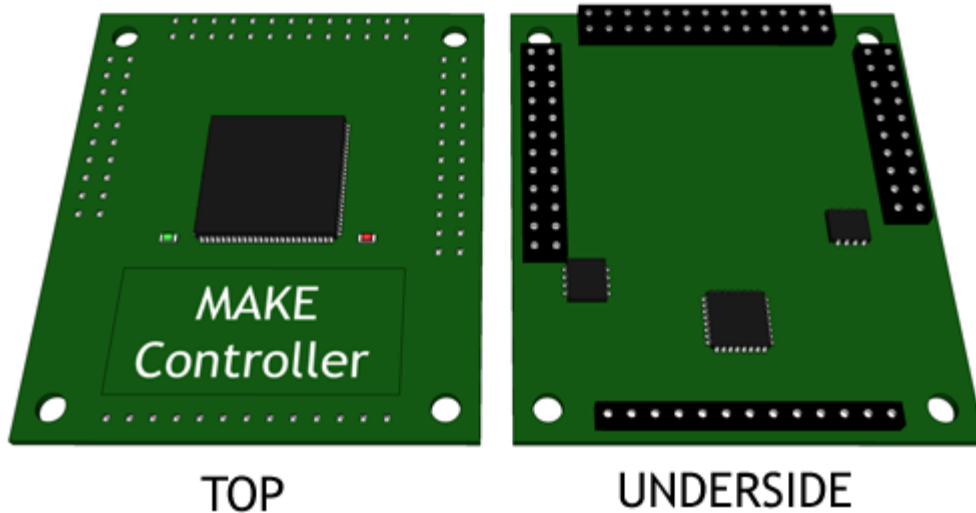
Quattro LEDs di stato Verdi forniscono un *feedback* per semplici test di programmazione[[vd.OSC sample](#)].



### 3.1.2.2 Make Controller

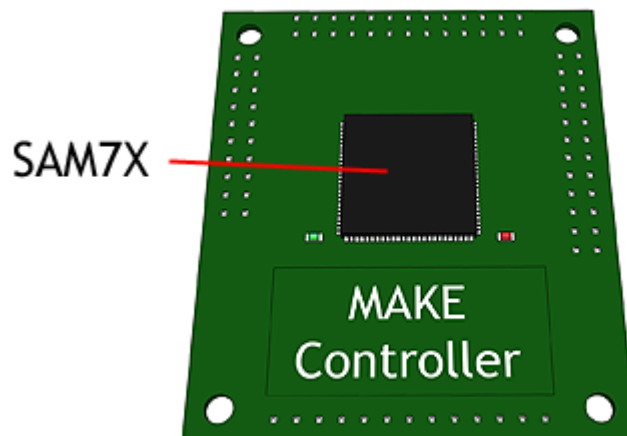
#### 3.1.2.2.1 Panoramica del Controller

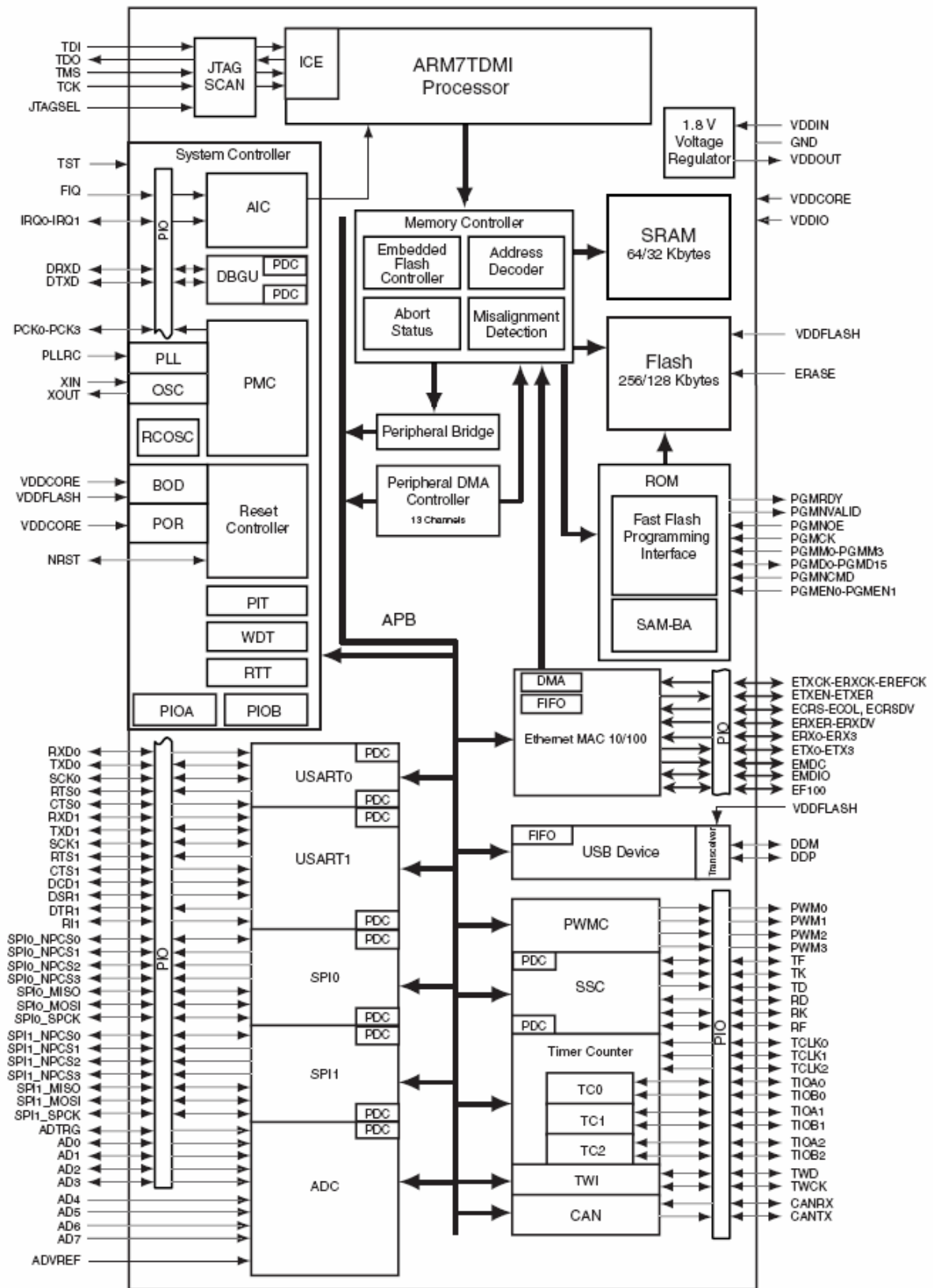
La **Make Controller** è costruita intorno al processore **Atmel AT91SAM7X256**, e aggiunge i componenti essenziali (come il cristallo, il regolatore di voltaggio, filtri a capacitore, etc.) richiesti per funzionare, e che utilizzano connessioni inferiori a quelli standard di 0.1 pollice. L'idea dietro questo disegno è quella di fornire in modo accessibile manualmente la potenza di calcolo del SAM7X evitando l'assemblaggio di componenti a saldatura laser.



#### 3.1.2.2.2 SAM7X

La Atmel AT91SAM7X256, o SAM7X, è un processore a 32 bit basato sull'ARM7 che gira fino a 55MHz/48MIPs. Ha 64K di RAM e 256K di memoria Flash, con una notevole lista di dispositivi sulla scheda: Ethernet, USB 2.0, RS232, SPI, TWI (o I2C), JTAG, CAN, come anche una generosa fornitura di I/O generiche.

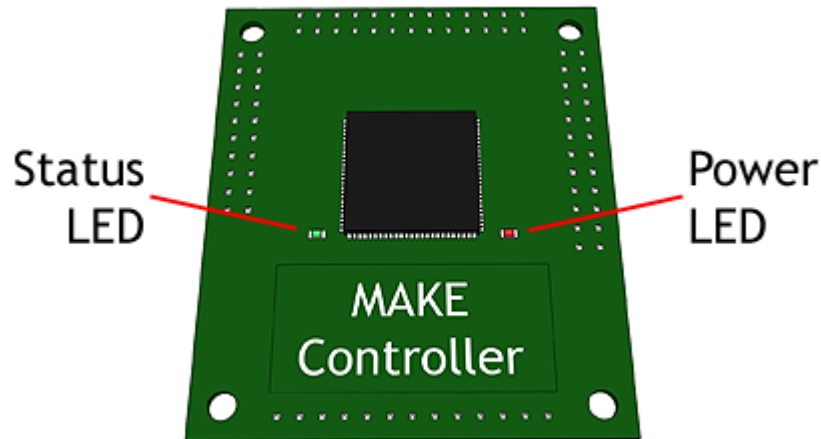




Block Diagram 1

### 3.1.2.2.3 LED's

Ci sono due LEDs sulla scheda controller. Uno (power LED) segnala la presenza di alimentazione, e l'altro (status LED) è disponibile per il processore da usare. Nel firmware utilizzato nel presente progetto la sua è una routine di lampeggiamento (~1s), la cui interruzione indica un overflow di dati.



### 3.1.2.2.4 Alimentazione

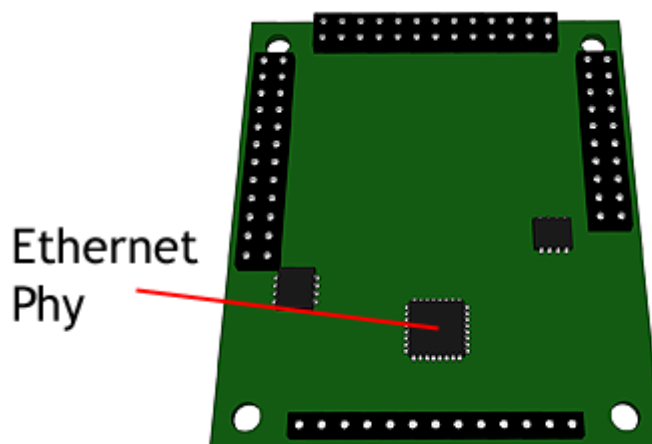
La SAM7X gira a 3.3V (e internamente a 1.8V). La **Make Controller** può essere alimentata da una sorgente a 3.3V (come quando è connessa alla [Make Application Board](#)) o può girare dal connettore V+ (un alimentatore DC da 7V a 12V). In questo caso, un piccolo regolatore sulla scheda fornisce i 3.3V. Il regolatore sulla scheda può fornire 3.3V per un limitato numero di circuiti esterni.

La linea a 3.3V è protetta contro momentanee tensioni superiori da un diodo TVS ma esso non durerà a lungo se la sovratensione si protrae per più di un secondo o due.

### 3.1.2.2.5 Ethernet PHY

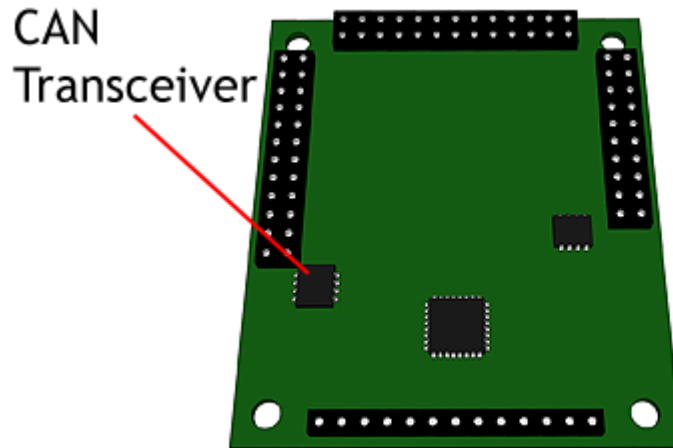
La SAM7X fornisce un Ethernet MAC, un chip addizionale è richiesto per PHY (physical) layer - il DM9161A della Davicom. Con questo chip sulla **Make Controller**, è possibile includere un normale connettore Ethernet sulla scheda madre, com'è la [MAKE Application Board](#), e collegarla senza ulteriori problemi.

Questo chip è posto nel lato inferiore della **Make Controller**.



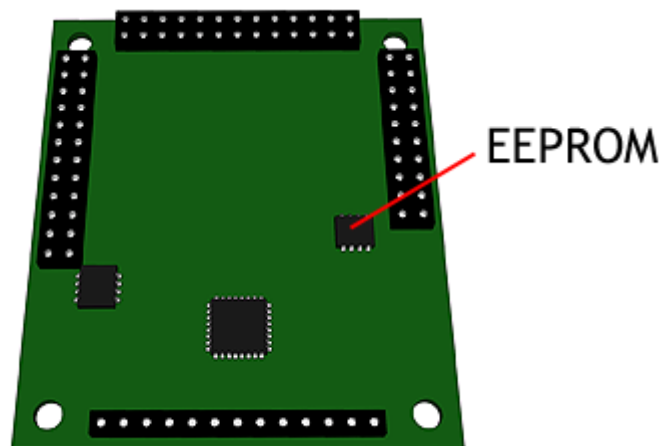
### 3.1.2.2.6 CAN

CAN (Controller Area Network) è un protocollo di rete seriale usato per la comunicazione da scheda a scheda quando si creano una rete di multipli MAKE Controllers Kit. Il SAM7X ha un controller CAN sulla scheda, e l'aggiunta di un chip CAN *transricevente* significa che la scheda di controllo può essere CAN-abilitata usando solo 2 linee di segnale. Un connettore a 4 posizioni sulla [MAKE Application Board](#) collega abbastanza facilmente il [MAKE Controller](#) alla rete CAN.



### 3.1.2.2.7 EEPROM

In aggiunta alla memoria sul chip del SAM7X, sul [Make Controller](#) ci sono ulteriori 32kB di EEPROM per fornire un immagazzinamento permanente *byte-by-byte*. Mentre la Flash memory sul chip è abbondante, essa può solo essere scritta un settore (256 bytes) alla volta, e solo da codice girante in RAM, ecco perché un'altra forma di memorizzazione permanente è molto utile. EEPROM può essere scritta un byte alla volta ed è usata per immagazzinare cose come il numero seriale della scheda, l'indirizzo IP assegnato, etc.



### 3.1.2.2.8 USB

Il SAM7X ha un controller USB 2.0 interno, e con esso il [MAKE Controller](#) diviene un dispositivo USB a piena velocità, trasferendo dati a 12 Mbits/sec. USB è usato per due propositi sulla scheda di controllo - uno è quello di permettere generici input ed outputs. L'altro è quello di fornire un accesso per programmare velocemente il controller.

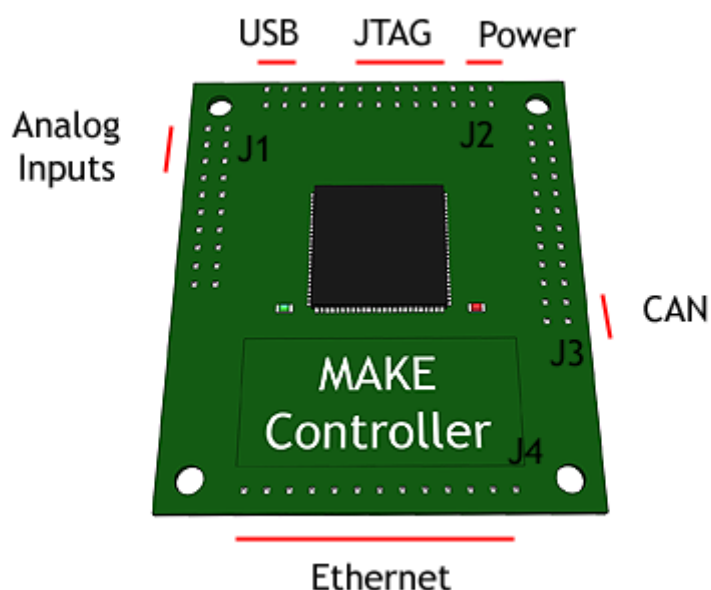


### 3.1.2.2.9 JTAG

Il SAM7X Controller ha una struttura JTAG, che permettono ad hardware esterno di esaminare lo stato del processore, avviarlo e fermarlo, etc. Le linee richieste per il JTAG sono portate fuori attraverso i connettori della [MAKE Application Board](#).

### 3.1.2.2.10 Input / Output

In aggiunta ai dispositivi descritti sopra, il chip SAM7X fornisce molti più dispositivi e generali contatti di IO. L'accesso a molti di questi è fornito sui connettori J1 - J4.



### 3.1.2.2.11 Firmware

Il SAM7X è basato sul celebre core ARM 7 TDMI. C'è una catena di strumenti *open source* (GCC etc).

Tutte le caratteristiche *hardware* sulla scheda richiedono un *software* per settarle e farle lavorare. Ci sono due possibili approcci alla programmazione del **MAKE Controller Kit**, una inizia dal minimo progetto che presenta il minimo codice di base dal quale il microcontroller può essere programmato, oppure un opposto approccio può essere preso, per iniziare con un progetto veramente completo che la MakeThings fornisce chiamato "heavy".

Usare il minimo "tiny" richiederà una più approfondita comprensione del processore essendoci meno facilitazioni, ma per quelli che vogliono strizzare ogni possibilità ulteriore di velocità e flessibilità dal loro sistema, questo sarà il modo giusto.

Per altri che vogliono beneficiare del sistema operativo, dello stack TCP/IP e altre facilitazioni, il più pesante codice di base "heavy" sarà il modo migliore di iniziare.



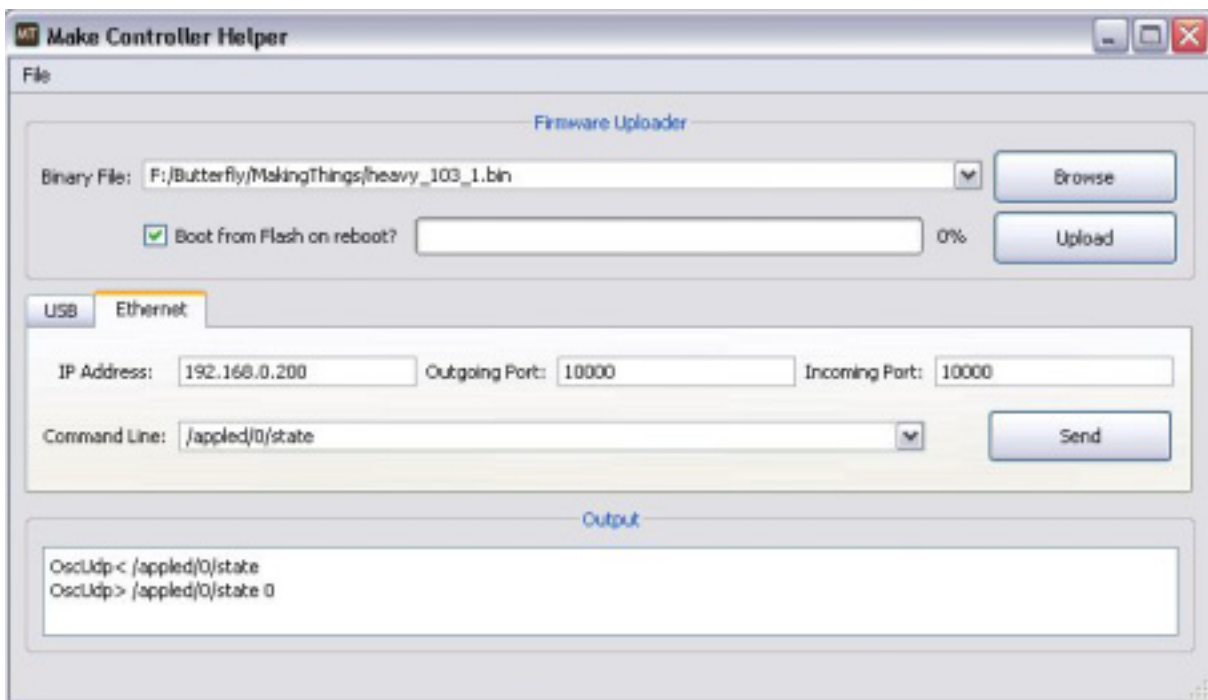
### 3.1.2.2.12 McHelper: programmazione della scheda e upload Firmware

L'**McHelper** è un'applicazione che consente di caricare nuovo codice sul MakeControllerKit ed effettuare un semplice debugging.

Dopo aver scaricato il file di setup **mchelper-setup-v1.2.3.exe** (o successivo) ed installato è necessario procurarsi anche il file binario **heavy.bin** o scriverne uno nuovo.<sup>19</sup>

Alimentare il **MakeControllerKit**, connetterlo al PC tramite la porta USB, selezionare il file **heavy.bin**, premere "upload" per inviarlo alla scheda, resettare l'alimentazione della scheda ed essa farà girare il nuovo programma.

Di seguito la schermata con gli indirizzi Ethernet, per la connessione USB leggere il manuale fornito con l'installazione.



Per testare la comunicazione USB o Ethernet, dopo la connessione fisica e l'inserimento degli opportuni indirizzi di rete (vd. cap. precedente), si può utilizzare la linea di comando diretto per inserire un messaggio OSC. Nella fig. precedente il messaggio si riferisce alla richiesta di stato del led 1 (0\_3 = 1\_4) e nella finestra di output appare l'invio del messaggio e la risposta del MakeCK in questo caso "0" cioè spento.

Questi i messaggi di esempio relativi ai led:

- /appled/0/state 1** = accendi il primo led
- /appled/0/state 0** = spegni il primo led
- /appled/0/state** = il primo led è acceso o spento?

<sup>19</sup> Il link della Making Things è nel capitolo finale Bibliografia e Links

### 3.1.2.3 Connessione diretta Ethernet

Normalmente, il **Make Controller Kit** è pronto a connettersi ad un router o accedere alla propria rete locale e comunicare con tutti i dispositivi connessi.

Comunque, c'è qualche volta un inconveniente a connettere la scheda ad un *router*, e bisogna preferire la connessione diretta col computer. In questo modo si perdono i benefici di essere connessi ad una rete ma si ottengono quelli di una veloce connessione Ethernet. Questa è una guida per settare una connessione Ethernet diretta tra il computer e il **Make Controller Kit** (in questo paragrafo 2.6.3 lo chiameremo semplicemente Controller).

#### 3.1.2.3.1 Hardware e configurazione degli indirizzi

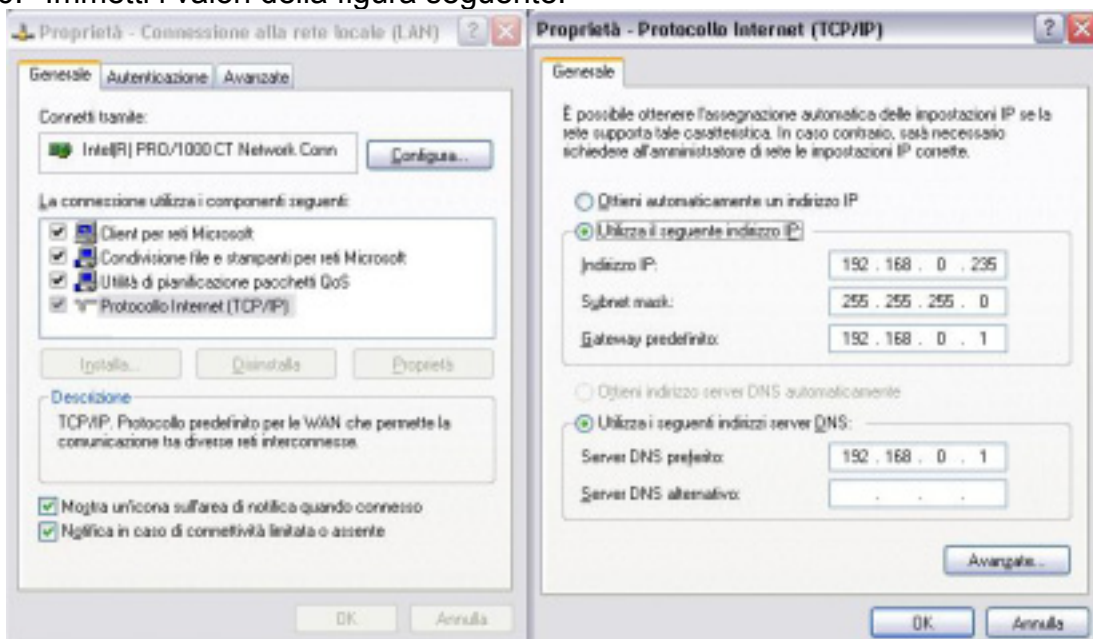
E' necessario solo un cavo Ethernet ed ovviamente una porta sul computer.

Se il PC ha solo un'interfaccia cablata con un interfaccia Ethernet, si può cambiare il settaggio di rete sul PC per trasmettere solo al Controller.

Se si hanno sia interfacce cablate che *wireless*, si vuol tenere la connessione internet wireless e utilizzare una connessione cablata con il Controller. Per far questo il Controller ed il PC devono essere configurati per comunicare su una *subnet* che non è usata dalla rete *wireless*. Molte reti, compreso il Controller stesso sono configurati in *default* per comunicare su una *subnet* con indirizzo 192.168.0.xxx. Se il *router* è su una differente subnet, bisogna sceglierne una diversa per collegare il Controller.

#### 3.1.2.3.2 Settaggio della rete in Windows

1. Dallo **Start Menu**, selezionare **Connetti a**.
2. Seleziona **Mostra tutte le connessioni**.
3. Doppio-click su **Connessione alla rete locale (LAN)**.
4. Seleziona **Protocollo internet (TCP/IP)**, e click **Proprietà** (scheda: **Generale**)
5. Seleziona **“Utilizza il seguente indirizzo IP” (192-168-0-235) Subnet (255.255.255.0) Gateway predefinito (192.168.0.1)**
6. Immetti i valori della figura seguente:



Settaggio della rete locale di Windows per comunicare direttamente con il **Controller**.

Il valore “**235**”, nella figura precedente, è arbitrario, è sufficiente sceglierlo in modo che non sia uguale ad un altro in rete.

Altrimenti, per preservare altre connessioni come quella internet condivisa, scegliere un'altra subnet tipo: **192.168.5.xxx** dove il **5** è arbitrario. Sarà necessario cambiare anche il **gateway predefinito** in **192.168.5.1**, o l'equivalente cambiando il **5**.

In questo caso, essendo l'**IP** predefinito della scheda **192.168.0.200** si deve cambiarlo sulla scheda stessa, come spiegato di seguito.

### 3.1.2.3.3 Settaggio del Controller

L'indirizzo predefinito del Controller è **192.168.0.200** . Per cambiarlo, eventualmente si intenda utilizzare per il PC **192.168.5.2** con Gateway predefinito **192.168.5.1**, è necessario che anche la scheda abbia I primi tre numeri identici ed il quarto differente p.es. **192.168.5.200**

Per cambiare l'indirizzo della scheda tramite OSC<sup>20</sup> utilizzando l'**mchelper**<sup>21</sup>, ci sono 4 messaggi da inviare alla scheda:

1. **/network/address 192.168.5.200** → cambia l'indirizzo IP.
2. **/network/gateway 192.168.5.1** → cambia il gateway.
3. **/network/valid 1** → salva questi cambiamenti.
4. **/system/reset 1** → resetta il controller.

---

<sup>20</sup> Open Sound Control, vedi capitolo dedicato a questo protocollo.

<sup>21</sup> Piccolo software fornito dalla MakingThings onSite dedicato alla programmazione del Controller.

### 3.1.3 Il computer e la scheda video: HP Compaq 8710w

La scelta dell'elaboratore idoneo a garantire prestazioni adeguate al sistema nel quale viene inserito è stata fatta avendo quattro richieste specifiche:

1. Per una massima portabilità ci si è subito orientati ad un modello laptop. Il peso non costituiva un problema, contrariamente all'attuale tendenza alla massima leggerezza e maneggevolezza.
2. La presenza di un processore Dual Core, "SantaRosa", per le ottime prestazioni multimediali promesse.
3. Avendo a disposizione già una scheda audio in buone condizioni ma costituita da un box esterno per gli I/O ed una scheda (cardbus) in formato PCMCIA II da collegare, una esigenza era quella della presenza di tale alloggiamento e porta predisposti. Tale caratteristica, ormai soppiantata dalle schede di nuova generazione (Express Card), ovviamente incompatibili con il formato PCMCIA II, ha costituito un discreto problema.
4. Era necessaria una scheda video di ottime prestazioni e dotata di memoria *on-board* di almeno 256 MB.

Ci si orientò quindi su uno dei pochissimi modelli rispondenti a queste specifiche: la Workstation HP Compaq 8710w.

#### 3.1.3.1 HP Compaq 8710w: Specifiche inerenti il sistema Butterfly



Processore	Intel Core2 Duo 2.4 GHz
RAM	DDR2 SDRAM 2GB
HD	120 GB 7200 rpm
Chipset	Mobile Intel 965PM Express
Slot	PCMCIA I/II
Porte	VGA, HDMI, Firewire (IEEE1394a), RJ45, 6 USB 2.0
Scheda Video	NVIDIA Quadro FX 1600M – 256 MB memory – Core 625 MHz – Pipelines 32 - Bus Memory 128 bit – Velocità memory 800 MHz
Schermo	TFT 17" Risoluzioni 800*600 → 1680*1050 (nativa)

### 3.1.4 – La scheda audio: RME Hammerfall DSP Multiface

Il sistema RME Hammerfall DSP è composto dalla scheda esterna di I/O “Multiface” e dalla scheda in formato PCMCIA “Cardbus”. Esse sono connesse tra loro da un cavo Firewire1394 che utilizza un protocollo dedicato, proprietario.



RME multiface I/O box (fronte e retro sovrapposti)



Cardbus PCMCIA II,  
con cavo Firewire  
collegato

#### 3.1.4.1 RME Hammerfall DSP Multiface: Specifiche inerenti il sistema Butterfly

- 8 ingressi e 8 uscite analogiche, ¼” TRS Jack bal/unbal
- 8 canali (4 coppie) I/O su optical TOSLINK (ADAT)
- 2 canali (una coppia) I/O su SPDIF coassiale
- ASIO2
- MIDI I/O

##### AD

- Resolution AD: 24 Bit
- Signal to Noise ratio: 101 dB RMS unweighted, 106 dBA
- Input Line: ¼” TRS jack, servo balanced

##### DA

- Resolution DA: 24 Bit
- Signal to Noise ratio: 108 dB RMS unweighted, 111 dBA (unmuted)
- Output Line: ¼” TRS jack, servo balanced

Il sistema utilizzerà solo una coppia di canali su Adat optical e l’ingresso MIDI.

### 3.1.5 – Il pedale MIDI: StuArt MB5



La pedaliera MIDI utilizzata ha 5 pulsanti program e 2 avanzamento bank. Ciò significa che si possono inviare gruppi di 5 Messaggi di canale (0-4 o 5-9 e via dicendo). La scelta è caduta su questo prodotto per le ridotte dimensioni (la maggior parte delle altre fornisce dieci pulsanti program + altri di bank) perchè facilita l'individuazione essendo il numero dei pulsanti uguale alle dita di una mano<sup>22</sup>. L'iniziale richiesta della progettazione del resto prevedeva un semplice pulsante di incremento program.

Unico neo della periferica la rumorosità dei bottoni metallici. Per ovviare a tale ingombrante presenza si è provveduto ad imbottire di gommapiuma il contenitore e ad inserire anelli di materiali incoerenti.

### 3.1.6– Il mixer digitale: Yamaha01v



Il mixer digitale Yamaha01v utilizzato aveva la scheda opzionale di connessione ottica TOSLINK (Adat) 8Ch IN/OUT, utilizzata proprio per il collegamento alla scheda audio.

### 3.1.7 – Il proiettore video: Toshiba TDP-S20



#### **Caratteristiche inerenti le specifiche di sistema:**

Luminosità immagine [Lumen] 1400 Lumen  
Contrasto 2000:1  
Fattore di zoom 1.2x  
Min Dimensione Immagine 0.762 m  
Max Dimensione Immagine 7.62 m  
Risoluzione Max 800x600 pixel  
Min Distanza Schermo 1.15 m  
Max Distanza Schermo 10 m  
Segnale video analogico RGB  
Interfacce VGA , Mini-DIN , RCA , S-video

<sup>22</sup> E' noto che l'essere umano riesce a contare "immediatamente" cioè con un solo sguardo, un gruppo di oggetti non superiori a 5.

## 3.2 – Scelta dei componenti software

Anche in questo caso, come per le componenti hardware, molte di quelle che seguono possono essere sostituite a seconda delle condizioni generali ed operative nonché artistiche. Restano capisaldi difficilmente sostituibili il protocollo OSC e il software audio Max-MSP. Quest'ultimo però può essere sostituito da Pure Data (PD) e un po' più difficilmente da CPS.

### 3.2.1 – Il sistema operativo: Windows XP sp2

Nel momento in cui si affronta una impegnativa ed innovativa realizzazione come quella presente, alcune cose presenti vengono utilizzate per non dover confrontarsi con un mondo completamente nuovo. Tra queste abbiamo già notato macchine facenti parte del parco in mio possesso, e aggiungiamo anche il sistema operativo, di certo tra i meno benvisti in ambito professionale, ma che almeno in questa versione (XP sp2) è risultato nel tempo di una certa affidabilità. In fase di performance, per ridurre lo sperpero di potenza di calcolo e memoria, si è provveduto a terminare tutti i processi non legati al sistema (antivirus, firewall, screensaver etc.).

Testato in precedenza, non si è ritenuto opportuno aggiornarne il Service Pack e tantomeno azzardare l'upgrade a Vista.

### 3.2.2 – Il protocollo di comunicazione: Open Sound Control

**Open Sound Control** (in seguito **OSC**), ideato da Matthew Wright e Adrian Freed [1997 CNMAT *Center for New Music & Audio Technologies* del Dipartimento di Musica dell'Università di Berkeley], è un protocollo di comunicazione a pacchetto definito in modo indipendente rispetto al livello di trasporto, anche se le implementazioni più diffuse attualmente si basano sui protocolli TCP/IP e UDP/IP. Più esattamente è un protocollo a livello applicazione, a differenza del UDP che è a livello trasporto.

Le principali implementazioni attualmente disponibili vanno dai bus ad alta velocità (es. PCI) alle infrastrutture di rete seriale come *Ethernet* (10-100-1000 Mb/s), USB, IEEE1394 (*Firewire*).

**OSC** può facilmente sostituire il protocollo MIDI, conservandone le ben documentate capacità, ma ottimizzandone la trasmissione con l'avvalersi delle nuove tecnologie e senza i limiti dell'hardware MIDI. Infatti la velocità di trasmissione delle tecnologie moderne parte da 10 Mb/s in confronto ai solo 31.25 Kb/s del MIDI, in più codifica i dati a 32 o 64 bit, fa uso di nomi simbolici e di messaggi di *time-tag*. Non utilizza un flusso continuo di dati su una connessione stabilita ma dei pacchetti di dati (*datagrams*) inviati serialmente e ricostruiti a valle come tali. Ciò permette di non dover supporre che il ricevente conservi lo status precedente fino al nuovo messaggio, essendo contenute in ogni pacchetto tutte le informazioni necessarie alla sua funzione, comprese quelle necessarie alla sincronizzazione degli eventi.

La sua sintassi si basa su una struttura di indirizzi con architettura ad albero ed è simile a quella in uso nelle URL, con una *root* (radice) e varie ramificazioni. Si avvale di caratteri speciali per l'invio di un singolo messaggio a più indirizzi di un ramo (?, \*, [, ], {, and }) e di *query* di status o di *data*.



I dati numerici sono rappresentati con 4byte (32bit) o 8(64bit) in complemento a due per quanto riguarda gli interi e nello standard IEEE per quelli invirgola mobile; le stringhe utilizzano caratteri ASCII non-null, seguiti da questi fino a completare la lunghezza di un multiplo di 4 byte.

Ogni messaggio (*message*) è formato da una stringa che contiene indirizzo e nome del messaggio seguita dall'argomento del messaggio in forma di dati binari.

Ogni pacchetto contiene o un singolo messaggio o un insieme di questi chiamato bundle (#bundle + 64bit fixed point time tag + ogni numero di messaggi o bundles, ognuno preceduto da un numero intero a 4 byte per il conteggio). Quindi ogni bundle può contenerne altri. La funzione dei bundle è quella di fornire dei dati di eventi che devono essere eseguiti simultaneamente in un dato momento, opportunamente sincronizzati dal server mediante i time tag forniti dal client. I time-tag sono formati da un numero a virgola fissa di 64 bit, di cui i primi 32 descrivono il numero di secondi a partire dalla mezzanotte del 1° gennaio del 1900 e gli ultimi 32 specificano le frazioni di secondo fino ai 200 picosecondi. OSC non fornisce un meccanismo autonomo di sincronizzazione del clock ma utilizza la rappresentazione del tempo nelle reti NTP [Network Time Protocol] o SNTP [Simple ...]. Per una corretta interpretazione il ricevente necessita di un real-time scheduler.

Il modello di un messaggio **OSC** è un set gerarchico di oggetti dinamici indirizzato tramite un codice simile a quello delle URL (p.es. **/voices/drone-b/resonators/3/set-Q 234**) senza le limitazioni di altri protocolli. La prima parte è l'indirizzo, seguito da uno spazio e dall'argomento del messaggio (in questo esempio il numero 234)

p.es. nella sintassi applicata al **Make Controller Kit**

1. **"appled/0/state 1"** comanda alla scheda di accendere il primo led di controllo
2. **"appled/0/state 0"** comanda alla scheda di spegnere il primo led di controllo
3. **"appled/0/state"** richiede alla scheda l'attuale stato on/off dello stesso led

Per permettere l'indirizzamento di un messaggio a più destinatari, il ricevente distribuisce a valle tutti i messaggi a tutti gli indirizzi corrispondenti a quello indicato dal messaggio. Ciò è possibile anche senza conoscere gli indirizzi esatti del ricevente, indicando una parola riservata [\*] che ha la funzione di "jolly", tipica dei S.O. nel *searching*. Nello specifico abbiamo a disposizione:

1. " ? " che corrisponde ad un singolo carattere eccetto "/" riservato all'indirizzamento.
2. " \* " che corrisponde a nessuno o più caratteri eccetto "/"
3. "[ (string) ]" un insieme di caratteri corrisponde ad ogni carattere della stringa
4. " (FirstChar) - (LastChar) " indica un intervallo di caratteri cui corrispondere
5. "[! (string)]" indica tutti i caratteri non compresi nella stringa
6. "{(string1),(stringN), ...}" corrisponde a tutti i membri della lista separata da virgole.

Altra possibilità è data dal meccanismo di richiesta e risposta. Il richiedente invia un messaggio di richiesta con la sintassi generale e riceve un messaggio di risposta da tutti i riceventi corrispondenti all'indirizzo. Il meccanismo è tale che in caso di messaggi a più destinatari può ricevere risposte da ciascuno prima di aver inviato il successivo.



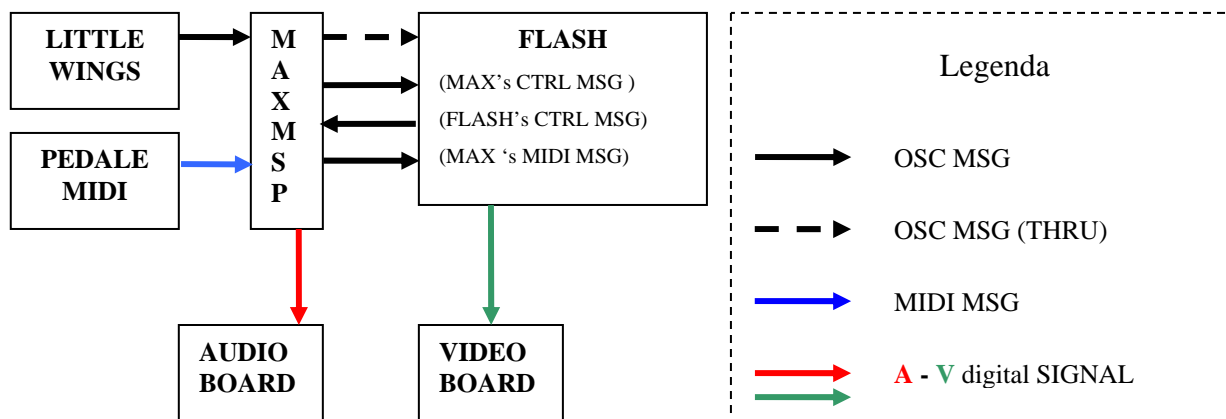
Per esempio il messaggio di richiesta del valore corrente inviato da MAX-MSP all'interfaccia che sarà gestita dal **Make Controller Kit** è: “ **analogin\*/value** ” dove **analogin** è il nome simbolico riconosciuto dalla scheda della Makethings , contenente la serie di 8 porte analogiche ai quali sono collegati i sensori. La presenza della parola riservata “\*” fa sì che il messaggio di richiesta del valore corrente ( /value ) venga ricevuto da tutti e 8 i canali del ADC (i valori analogici degli 8 sensori convertiti digitalmente dalla scheda). Le risposte alla richiesta verranno indirizzate da MAX-MSP mediante i ricevitori **osc-route** “**analogin/0/value**” ... “**analogin/7/value**”. Più approfondite spiegazioni saranno fornite nella trattazione inerente Max-MSP.

### 3.2.2.1 –Interfacciamento OSC

Il flusso delle informazioni di controllo provenienti dall'interfaccia, come abbiamo detto, sarà indirizzato a Max-MSP che le utilizzerà e le inoltrerà come messaggi al software di sintesi grafica.

Pur potendo il protocollo OSC inviare messaggi in parallelo ad entrambi i software si è preferito creare una gerarchia in modo da evitare conflitti e poter gestire in sequenza unidirezionata il sistema. I messaggi OSC provenienti dall'interfaccia **Little Wings** raggiungono MAX e vengono all'occorrenza rinviati a FLASH (msg through). MAX potrà variare o aggiungere messaggi OSC ed inviarli a FLASH per la sintesi grafica. Anche FLASH potrà generare propri messaggi OSC (per esempio segnali di sincronizzazione *bang* legati a cicli di frames o all'intero filmato) ed inviarli a MAX. I dati MIDI provenienti dalla pedaliera verranno interpretati e utilizzati da Max e da questo stesso software inoltrati, opportunamente ricodificati in OSC, a Flash.

Di seguito un diagramma di flusso dei segnali. **LittleWings** è il nome dell'interfaccia fisica appositamente creata, **Max-MSP** il software audio, **Flash** quello grafico, in nero i collegamenti che utilizzano il protocollo **OSC**, in blu il **MIDI**, in rosso il **segnale audio** (2ch) in verde quello **video**:



I messaggi MIDI del pedale, controlleranno l'avvio di patches di MAX e tradotti in messaggi OSC (→) inviati anche a FLASH. Maggiori e più dettagliate specifiche in seguito.

### 3.2.3 – Il programma di sintesi algoritmica: Max-MSP

In questo capitolo si introduce solo il programma utilizzato e le sue principali caratteristiche inerenti il sistema. La parte inerente l'interfacciamento OSC è nel capitolo Realizzazione dei componenti software mancanti [3.4.1].

Dopo aver utilizzato per anni CPS di Niels Morisse, un programma ispirato al presente Max, sono trasmigrato a questo potente software di programmazione algoritmica, non solo musicale, ma utilizzabile in molte applicazioni. Detto cambiamento è dovuto al rallentamento dell'upgrade di CPS e alla continua e precedente ulteriore diffusione in campo della musica elettronica di Max. Un passaggio "intermedio" è stato quello con PD (PureData) "fratellastro" di Max.

Il programma verrà utilizzato sia per la gestione dei dati provenienti dall'interfaccia fisica e loro inoltre a Flash che per il loro utilizzo come parametri negli algoritmi creati.

Scritto in linguaggio C, è continuamente, sin dall'inizio, arricchito di "externals", codici mirati a specifiche utilizzazioni. Tra questi vi sono i componenti per l'utilizzo in Max del protocollo OSC utilizzato nel sistema Butterfly.

La versione di Max-MSP utilizzata nel presente sistema è la 4.5.7. Gli external OSC quali **udpwrite** e **udpread** sono stati rimpiazzati dai più recenti (2005) **udpsend** e **udpreceive**

#### 3.2.3.1 Operazioni preliminari per l'utilizzo di OSC in Max-MSP

Dopo l'installazione del programma è necessario aggiungere i files relativi ad OSC.<sup>23</sup>

Quelli utilizzati nel sistema Butterfly sono:

- **OpenSoundControl**: l'interfaccia bidirezionale Max-OSC
- **osc-route**: per l'indirizzamento dei messaggi provenienti da **OpenSoundControl**
- **udpsend**: per l'invio di messaggi ad un dato indirizzo e porta (in **Butterfly** l'indirizzo local host, cioè il computer stesso e la porta relativa all'interfacciamento con Flash.
- **udpreceive**: per il ricevimento di messaggi su una data porta con un etichetta (in **Butterfly** la porta 10000 e l'etichetta "MakeCtrl" che contraddistinguono le impostazioni di default del MakeControllerKit.)

Detti componenti vanno copiati nella cartella "C:\Programmi\File comuni\Cycling '74\externals", per poter essere riconosciuti ed utilizzati da Max.

Di seguito gli externals OSC come appaiono e come vengono collegati in Max-MSP nel sistema **Butterfly**.

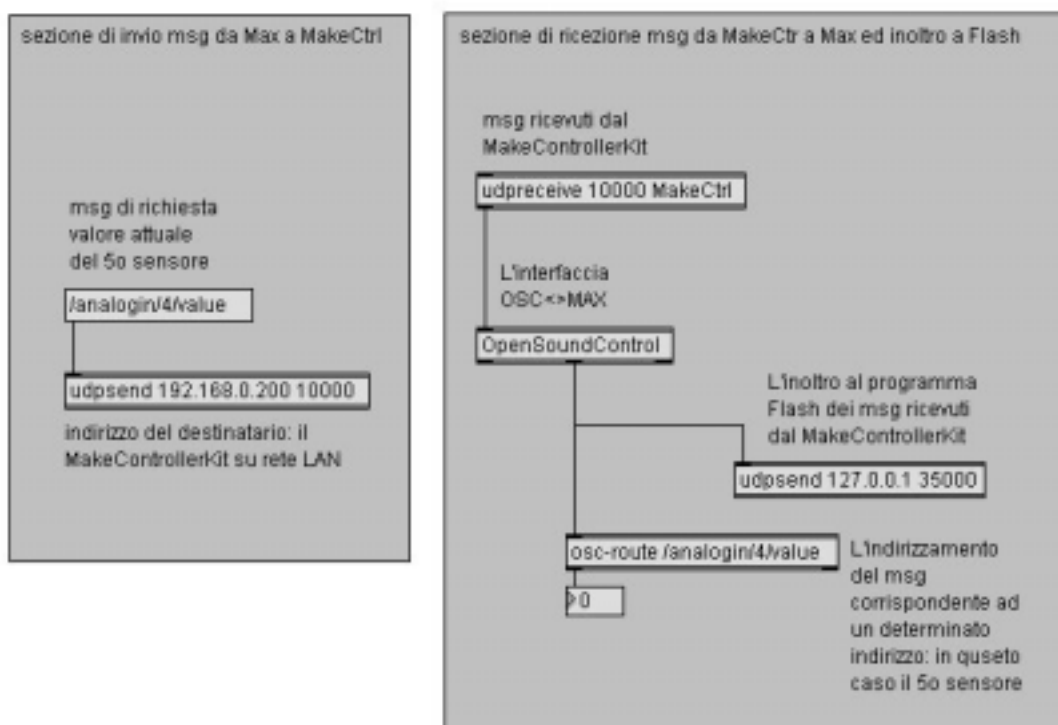
La sezione destra attiene alla routine di richiesta dati<sup>24</sup> provenienti dalla conversione A/D delle tensioni momentanee nei sensori. Nell'esempio, un messaggio "*analogin/4/value*" indica l'indirizzo generale degli input analogici ai quali sono applicati gli 8 sensori (*analogin*), il numero d'ordine del sensore (da 0 a 7 il "4" è associato al 5° sensore), e per finire il nome simbolico (*value*) che indica il valore richiesto.

<sup>23</sup> Gli indirizzi internet dove detti files sono reperibili sono riportati nel capitolo finale "Bibliografia e Links"

<sup>24</sup> Detta routine è stata resa superflua da un recente upgrade del software di controllo del MakeControllerKit

Detto messaggio è inviato (tramite l'esternal udp send) all'indirizzo di rete predefinito del MakeControllerKit (192.168.0.200) nonché alla porta prevista (10000).

La sezione destra invece contiene il ricevimento delle "risposte" alle interrogazioni precedenti, provenienti dal MakeControllerKit, la "traduzione" in linguaggio "Max" (OpenSoundControl), ed il parallelo indirizzamento ad un oggetto che riconosce lo specifico indirizzo utilizzandone il valore ricevuto e l'inoltro dello stesso messaggio all'interfaccia Flosc utilizzata da Flash per i propri compiti. In questo caso l'indirizzo 127.0.0.1 è quello predefinito di local host e la porta liberamente scelta 35000, che deve essere inserita nel *batch* che crea il Java Server per Flosc (vd. cap seguente) .



Da notare che questi oggetti in Max rappresentano pezzi di codice i cui collegamenti virtuali sono a volte palesi ed altre sottintesi. P.es il modulo OpenSoundControl rappresenta il codice necessario non solo alla traduzione da OSC a Max (come sembrerebbe dal collegamento della linea) ma anche il contrario. E per questo è assente nel riquadro sinistro. Purchè il modulo sia presente nello stesso patcher o in un patcher padre (come avverrà in un brano per il sistema **Butterfly**).

L'algoritmo completo per la rete OSC-Max-Flash nel cap. 3.4.1

## 3.2.4 – Il programma di sintesi grafica vettoriale: FLASH

Le peculiarità del programma dalle quali dipende la scelta per l'impiego in questo sistema sono principalmente quattro.

1. Le possibilità di creazioni di forme tramite la sintesi vettoriale e non bitmap, con la conseguente facilità di modifica
2. La capacità di creare filmati di animazione interattivi, cioè il cui svolgimento nel tempo può essere continuamente modificato
3. L'implementazione di un linguaggio, l'**ActionScript**, che permette una potente capacità di gestione algoritmica del filmato e connessioni di rete.
4. La disponibilità di un codice per l'utilizzo del protocollo OSC per la comunicazione con hardware e software diversi (FLOSC).

Inoltre i filmati realizzati, organizzabili con struttura ad albero e richiamabili immediatamente tra loro, vengono eseguiti da un player, SFWPlayer, sufficientemente potente anche se non esente da crash.

### 3.2.4.1 - L'interfaccia di rete per FLASH: XMLSocket e Floc

Per poter utilizzare la potenza di **ActionScript (2.0)** in una rete gestita dal protocollo OSC è indispensabile appoggiarsi all'esistente capacità di gestione dell'XML. Prima di tutto è necessario installare la libreria Floc!

**Floc** è un server Java che può comunicare con qualsiasi cosa che utilizzi il protocollo OSC. Quindi **Max-MSP** ed altri programmi. Esso può anche comunicare con Flash usando la caratteristica disponibile in **Flash** del XMLSocket. Esso è un oggetto dell'**ActionScript** di **Flash** che implementa una porta client che permette al computer dove risiede e gira il FlashPlayer di comunicare con un computer server identificato da un indirizzo IP o da un nome di dominio.

Il protocollo usato da XMLSocket è un flusso full-duplex (bidirezionale) TCP/IP. Ogni messaggio XML è un documento completo XML terminato da uno zero byte. Un numero illimitato di messaggi XML possono essere mandati o ricevuti sopra una singola connessione alla porta XMLSocket. A differenza di una connessione basata su http, che richiede un frequente *polling* (interrogazione) tra i comunicanti, una connessione basata sul XMLSocket è sempre aperta per mandare o ricevere messaggi immediatamente.

Per utilizzare **XMLSocket object** in **Flash ActionScript** si deve prima usare il costruttore "new XMLSocket" e quindi connetterlo ad una porta superiore o uguale al numero 1024 tramite la funzione "XMLSocket.connect" (per motive di sicurezza le porte inferiori utilizzate da http, FTP e Telnet sono vietate).

**Floc** funziona come un gateway tra OSC e Flash, un applicazione scritta in Java (necessita quindi l'installazione della runtime Java) che manda e riceve pacchetti via protocollo UDP, traducendo bidirezionalmente i pacchetti binari di OSC e una codifica XML di questi, e manda e riceve documenti XML via TCP in modo da essere compatibile con le caratteristiche della XMLSocket di **Flash**. In breve un interfaccia (un gateway o bridge) tra il vettore UDP di OSC e quello TCP di **Flash**. **Floc** può accettare simultanei input da multiple connessioni OSC e clients di Flash e può inviare in parallelo (*broadcast*) messaggi a multipli clients Flash ma senza effettuare alcun indirizzamento (*routing*).

### 3.3 Realizzazione dei componenti hardware mancanti

Acquisiti i materiali è iniziata subito la sperimentazione connettendo i sensori al **MakeControllerKit** e questo al PC su rete LAN connessione Ethernet. Sul PC si era realizzata in **Max** una patch di test contenente i componenti OSC precedentemente illustrati e una casella numerica per valutare la risposta digitale del sensore scelto.

I problemi principali incontrati durante la sperimentazione sono stati due:

1. La difficoltà di ottenere un'omogenea pressione del sensore **Touch** contemporanea a quella dello **Slider**. Quest'ultimo fu subito sovrapposto ad un supporto che poggiava sull'altro cercando mediante una leva di vari materiali il modo e la posizione migliore per il contatto che non doveva essere continuativo in stato di riposo.
2. Il sensore **Bend** (sensore attivato dalla sua flessione da 0 a 130°) rivelò una risposta molto limitata in range (valori digitali da 200 a 600 invece che i 1024 promessi).

A questo punto si avevano a disposizione 4 coppie di sensori diversi che dovevano venire assemblati su un supporto adatto per poter essere maneggiati nel modo migliore.

Il Prof. Lorenzo Seno si prestò generosamente a disegnare una scheda accessoria in grado di risolvere il problema della risposta inadeguata del **Bend Sensor**, mediante il condizionamento degli otto segnali con la sottrazione di un offset e la loro amplificazione in corrente continua. Scheda che provvede in uscita i segnali analogici così condizionati. A causa della non reperibilità di alcuni componenti [trimmer verticali] la configurazione finale differisce un poco dal disegno originale [vd. 3.3.3].

#### 3.3.1 Progettazione dell'interfaccia fisica: LittleWings

La progettazione della superficie sulla quale montare i sensori ha richiesto molto tempo per la necessità di testare i comportamenti dei componenti e le loro interazioni.

I primi tentativi hanno richiesto la temporanea "messa in sicurezza" della scheda in un provvisorio contenitore plastico. In esso sono stati montati connettori mammuth per evitare di agire direttamente e continuamente su quelli verdi montati direttamente sulla scheda (Make Application Board).



Mentre i sensori sono stati montati su un supporto metallico di lunghezza adatta ed incasato in un “sarcofago” di cartone di cui purtroppo non resta testimonianza fotografica.

Stabilite le misure idonee per la finestratura e testati i materiali adatti alla costruzione delle ali flessibili si è proceduto al disegno della superficie sensibile e la sua stratificazione. L'utilizzo di questa procedura, quella di pensare il contenitore dei sensori a strati, è stato dettato dalla possibilità di determinare gli spessori necessari e poterli sagomare secondo le necessità. Una delle quali, quella del sensore di pressione era di restare a contatto di una pressione atmosferica di riferimento, quindi senza impedimenti o pressioni involontarie. L'altra necessità della stratificazione è dovuta alla natura artigianale della realizzazione che non consente una pressofusione plastica presagomata. I fogli e le lastre plastiche sarebbero quindi dovute essere smontabili facilmente per gli aggiustamenti necessari in corso d'opera. Quindi vennero pensati con due modalità assemblative:

1. viti e bulloni per il montaggio a “sandwich” dei pezzi rigidi
2. nastro biadesivo per quelli flessibili

Di seguito tra parentesi quadre ed in **grassetto** i riferimenti alla nomenclatura dei pezzi costituenti la superficie sensibile ed i relativi sensori secondo le schede grafiche dell'allegato SensorLinearBox5, dove “**P**” indica un componente in plex e “**S**” un sensore.

Per motivi di leggerezza e praticità di ritaglio si è scelto come materiale rigido di supporto un plexiglass bianco opaco dello spessore di 3mm [**P1\_15** e **Base**] ed uno flessibile [**Ala Inferiore**]. I quattro sensori [**S1\_S4**] sono disposti tre sulla parte rigida ed uno su di un foglio flessibile trasparente ma sufficientemente elastico da ritrovare una volta rilasciato la sua posizione iniziale [**Ala Inferiore**] rivestito di un foglio di acetato [**Ala Superiore**]. Il sensore di posizione [**S1**] è ingabbiato nella struttura lasciando la parte sensibile disponibile direttamente al tocco del pollice o dell'indice mediante una finestra precisamente dimensionata<sup>25</sup> alla sua larghezza presente nella superficie superiore [**P15**]. La finestra è sagomata in modo da permettere un comodo sfioramento da dita di dimensioni maggiori. Il sensore [**S1**] aderisce tramite la propria base adesiva *Scotch* ad una barra di ferro zincato [**F**] delle stesse dimensioni. La barra metallica ha la parte inferiore appoggiata su di uno spessore di 3mm ed ingabbiata in parallelo come livello mentre a  $\frac{3}{4}$  della sua lunghezza poggia su un cuscinetto di gommapiuma [**sp**]<sup>26</sup>, che le consente una volta premuta una limitata escursione e rilasciata il ritorno relativo. La barra metallica consente di trasmettere la pressione esercitata su punti diversi del sensore [**S1**] con sufficiente omogeneità. Funzioni software di trasferimento sono state adottate per correggere l'inevitabile ma ridottissima differenza tra le pressioni nei punti estremi. La funzione del cuscinetto è ovviamente quella di contrastare la spinta di pressione che si trasferisce, mediante [**P1**] un quadratino di plexiglass sagomato a cuneo per fornire un contatto totale della sua superficie, sul sensore di pressione [**S2**] su di esso appoggiato. La barra metallica [**F**] con il sensore di posizione solidale [**S1**] è parzialmente sostenuta da una ghiera metallica [**F2**] che mediante due viti viene avvicinata o allontanata dalla superficie superiore in modo da calibrare (mediante un piccolo giravite a croce) la

---

<sup>25</sup> Lo sfioramento oltre i limiti predisposti provoca una assoluta imprevedibilità dei valori risultanti

<sup>26</sup> La scelta della gommapiuma si è rilevata vincente su altri prodotti elastici quali molle ed elementi plastici e metallici.

distanza alla quale il sensore di pressione non reagisce, cioè quando non si esercita alcuna pressione.

**P2** e **P3** bloccano l'escursione longitudinale superiore della barra con sensore. **P1** è incollato sul retro di **F** e serve per trasmettere la pressione, esercitata su **S1**, sul sensore di pressione **S2** sottostante. Questo sensore di forma quadrata è disposto a 90° rispetto a **S1** per via dei conduttori che devono restare, come da indicazioni del costruttore, a contatto con l'aria per avere il riferimento della pressione atmosferica. Per questo motivo l'intera struttura non è sigillata ma si sovrappone per sei strati (4 di plexiglass da 3mm, uno di acetato e uno di trasparente flessibile). Il sensore di pressione **S2** è bloccato alla base da nastro biadesivo lungo i 4 lati non essendo esso stesso fornito di adesivo come **S1**.

Il terzo sensore, quello di flessione **S3** è sistemato mediante biadesivo tra l'Ala Superiore e quella Inferiore, la prima in acetato che lo protegge da graffi e da spostamenti e la seconda in un foglio trasparente più resistente. La sua resistenza è dovuta alla necessità di mettere in flessione il sensore anche piegando il foglio lontano da esso. Anch'esso è perpendicolare ad **S1** essendo destinato alla flessione del foglio doppio mediante pressione delle ultime tre dita della mano. I pezzi da **P4** a **P11** sono semplici supporti che limitano l'escursioni della coda del sensore, comunque bloccato in più punti (anche da un filo metallico in verde agganciato a due scanalature) e ne consentono il passaggio della medesima sotto la barra **F**.

Il quarto sensore, il potenziometro rotativo **P4** è avvitato a **P15** e data l'altezza del suo contenitore necessita, oltre al piccolo foro rettangolare per il fermo, anche di una sede circolare nella **Base** che di poco supera. Del resto la **Base** è distaccata dalla box sottostante dallo spessore dei controdadi che fermano i bulloni di aggancio alla stessa.

I piccoli pezzi interni alle due facce principali la superiore **P15** e l'inferiore **Base** sono tra loro incollati e la loro dimensione e disposizione è studiata in modo da essere sufficiente al supporto dei bulloni a vite e relativi dadi. Questi sono di due dimensioni diverse, da 3mm per quelli che saldano le parti tra di esse e da 5 mm, disposti grosso modo agli angoli, che oltre a contribuire alla tenuta della struttura permettono il suo montaggio alla box contenente le schede ed il resto dei componenti. Essi restano montati permanentemente sulla box tramite controdadi dalla sua parte superiore, e su di essi vengono inserite, attraverso i relativi fori, le due superfici sensibili in plexiglass e acetone, permettendone sia il facile e sicuro trasporto che il comodo montaggio. Tali viti a bullone, a differenza delle minori che hanno la vite sulla parte superiore, sono montate al contrario per permettere di chiuderle manualmente mediante comodi dadi arrotondati di ottone. Ad essi (come risulta dal disegno) si sarebbero preferite le antiche *farfalle*, oggi introvabili, figlie di un tempo dove non esistevano trapani-giraviti, e il cui nome e forma sarebbero state più indicate per un sistema di nome **Butterfly**.

Le **ali sup** e **inf** sono collegate tra loro oltre che dai bulloni a vite nelle parti che devono restare ancorate al supporto rigido anche da nastro biadesivo lungo i lati.

I conduttori che escono dai 4 sensori sono opportunamente bloccati ed indirizzati verso la parte superiore. Alla loro uscita dalle superfici sono stati rivestiti da guaine termosensibili.

I due sensori della *Infusion System*, progettati e cablati per essere utilizzati con una loro macchina (*I-Cube X*) avevano i terminali assemblati in un pin tripolare da 2,5 mm di passo. Invece gli altri due sensori avevano le code libere e solo stagnate.

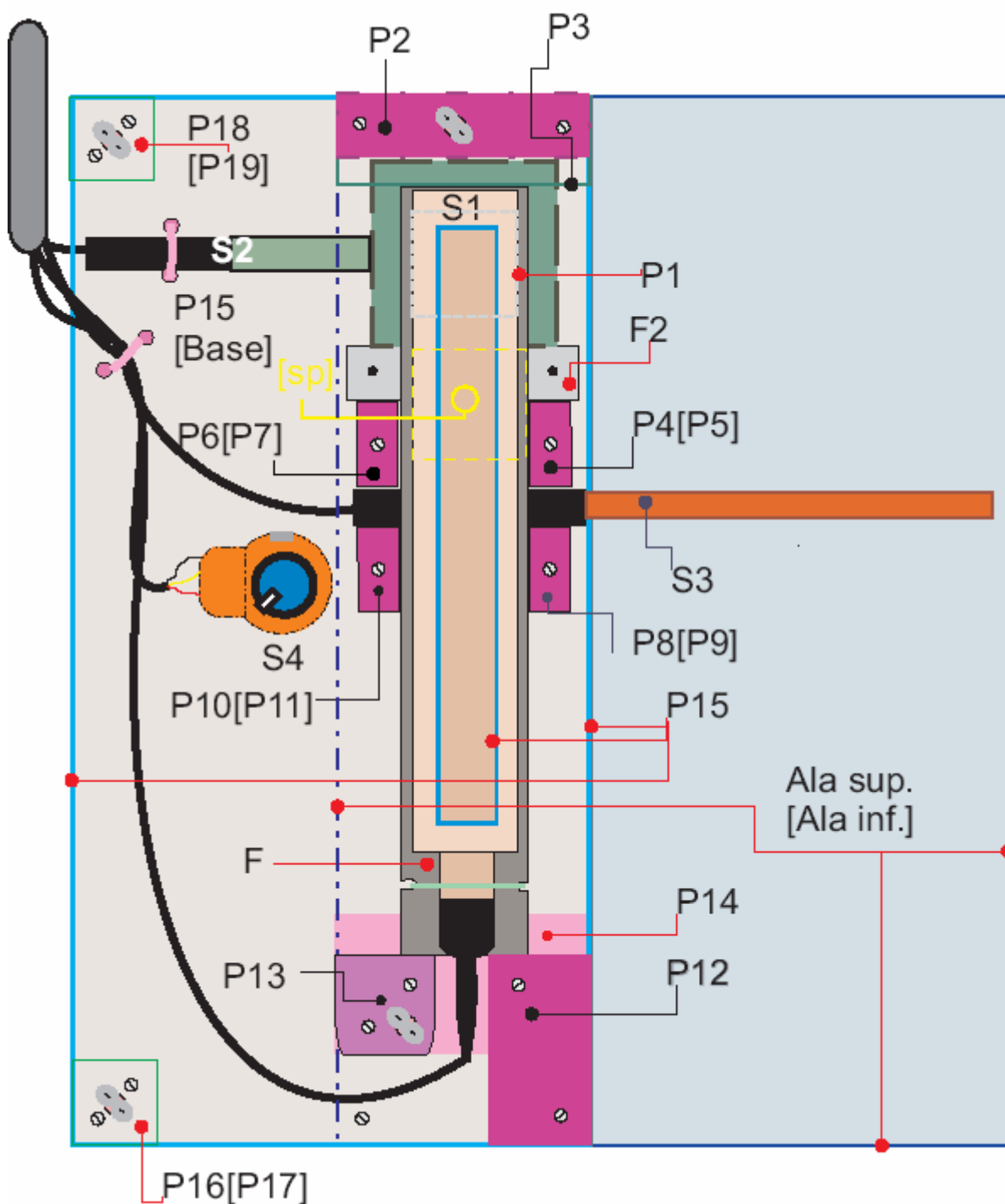
Si è scelto di adoperare delle spine e relative prese **Mini-XLR** pensandole più sicure per il serraggio, ma pur ben serrate dall'attrito non pare usufruiscano del gancio esistente che ha reso celebri ed affidabili le loro sorelle maggiori. A parte l'inevitabile difficoltà di saldatura di componenti così minuti e termo sensibili, la tenuta si è rivelata soddisfacente e l'ingombro minimo.

Abbiamo fino ad adesso esposto le modalità costruttive dell'**Ala Destra** delle superfici sensibili, resta da dire che l'**Ala sinistra** è stata disegnata simmetricamente come da indicazioni iniziali. Il suo potenziometro **S4** secondo le preferenze sarà mutato di direzione mediante inversione software dei dati. La sua posizione di fianco a **S1** ne consente l'utilizzo in contemporanea con esso mediante la sostituzione delle dita.

La larghezza delle ali sporgenti è tale da permettere una flessione non superiore ai 130 gradi consigliata dal fabbricante, flessione comunque limitata dal corpo stesso della box sottostante.



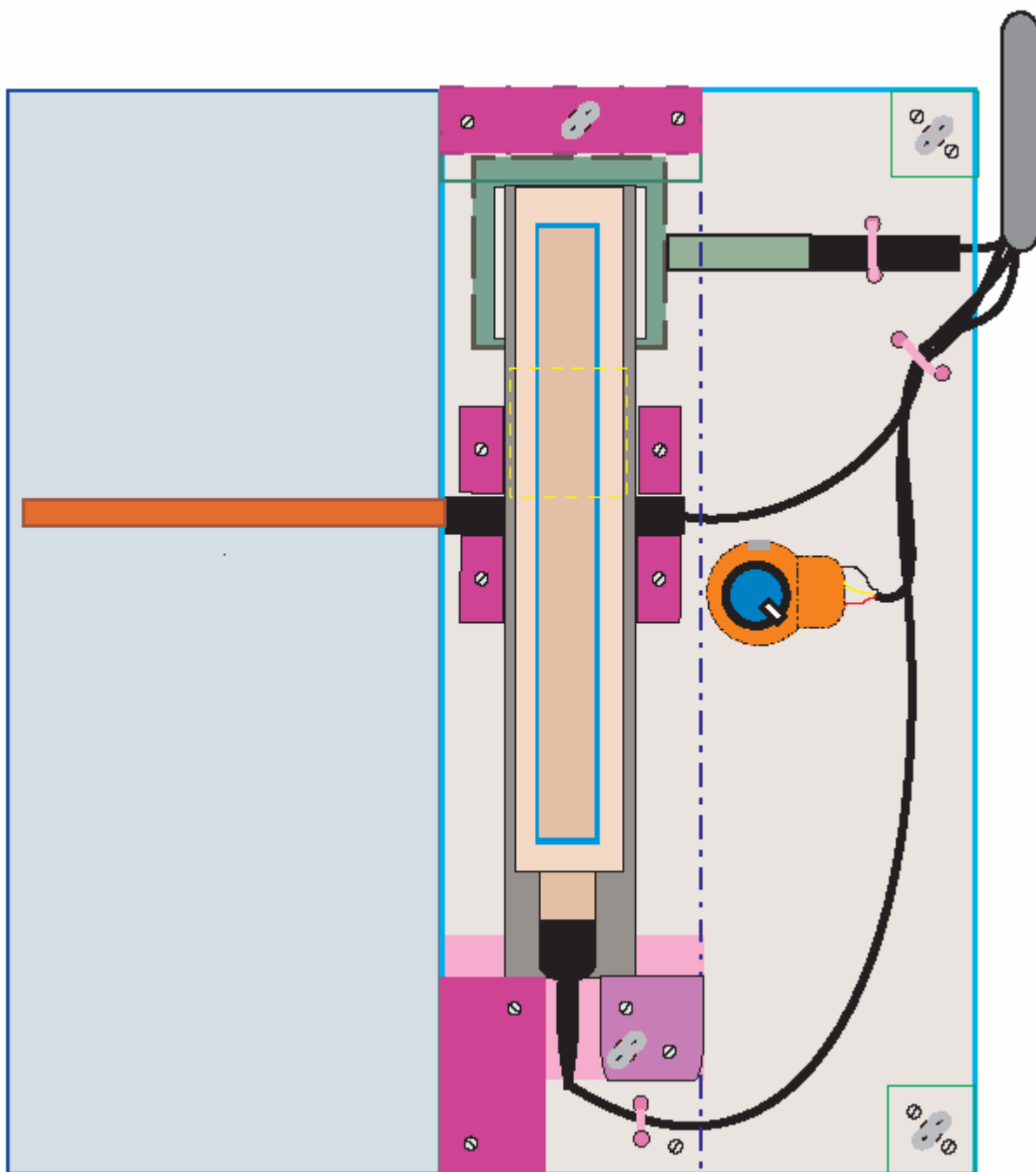
### 3.3.1.1 Disegno tecnico dei supporti removibili contenenti i quattro sensori (S1\_S4)



Visione superiore del supporto [P15 + sottostante Base] destro e dell'**Ala** destra parzialmente sottoposta (le linee tratteggiate indicano i margini degli oggetti invisibili perché sottostanti altri componenti. I punti colorati con le linee che collegano la sigla indicano o il corpo del componente o i suoi bordi. I componenti in plex sono colorati con sfumature bordeaux o beige, i sensori in arancio o verde e le ali in celeste. In nero i cavi. Le sigle tra parentesi quadre indicano componenti non visibili perché sottostanti. P.es. P4 [P5] sono due rettangolini di plex della stessa dimensione sovrapposti ed incollati insieme.

In posizione di riposo P1 deve restare a sfiorare S2.

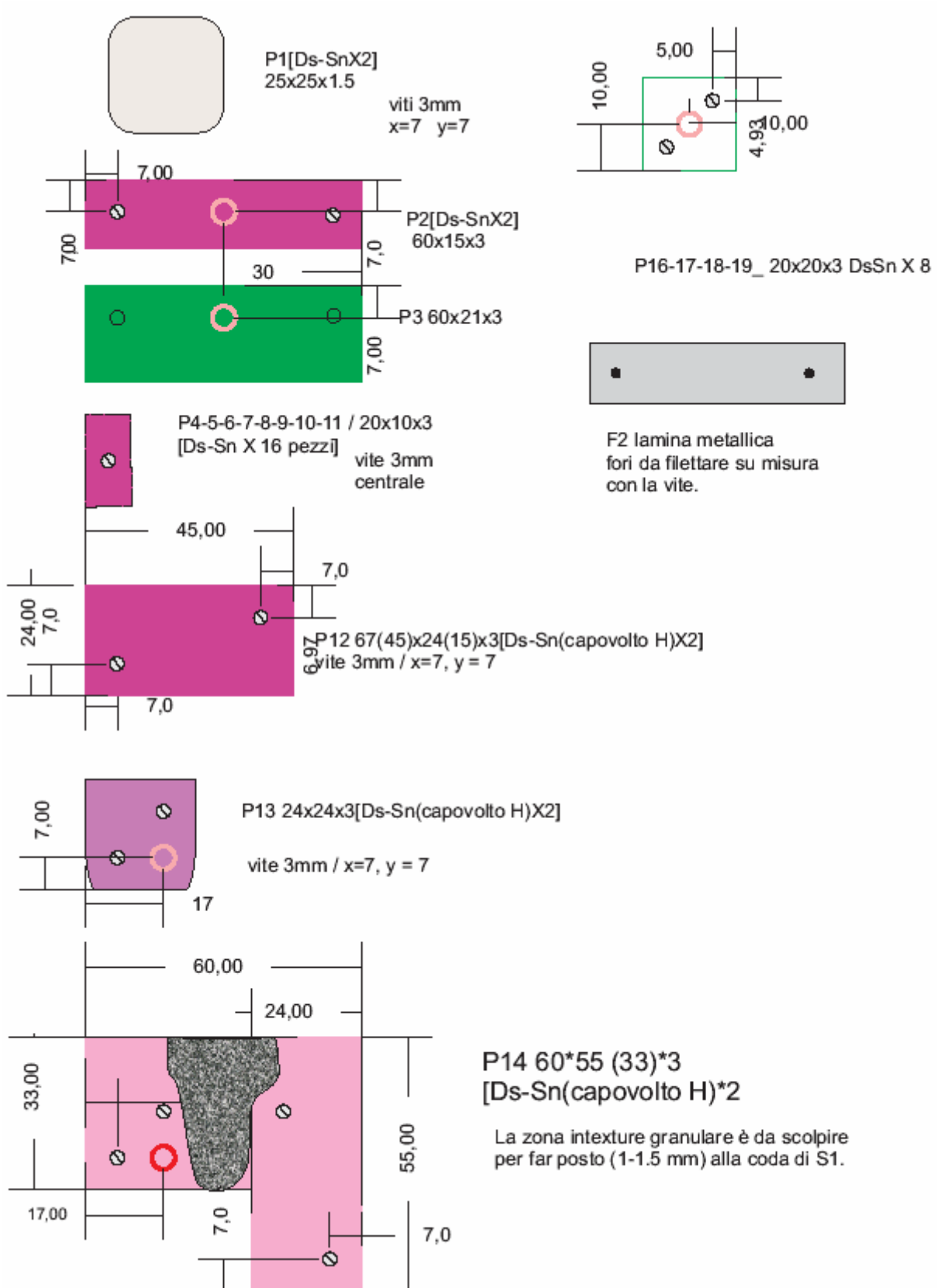




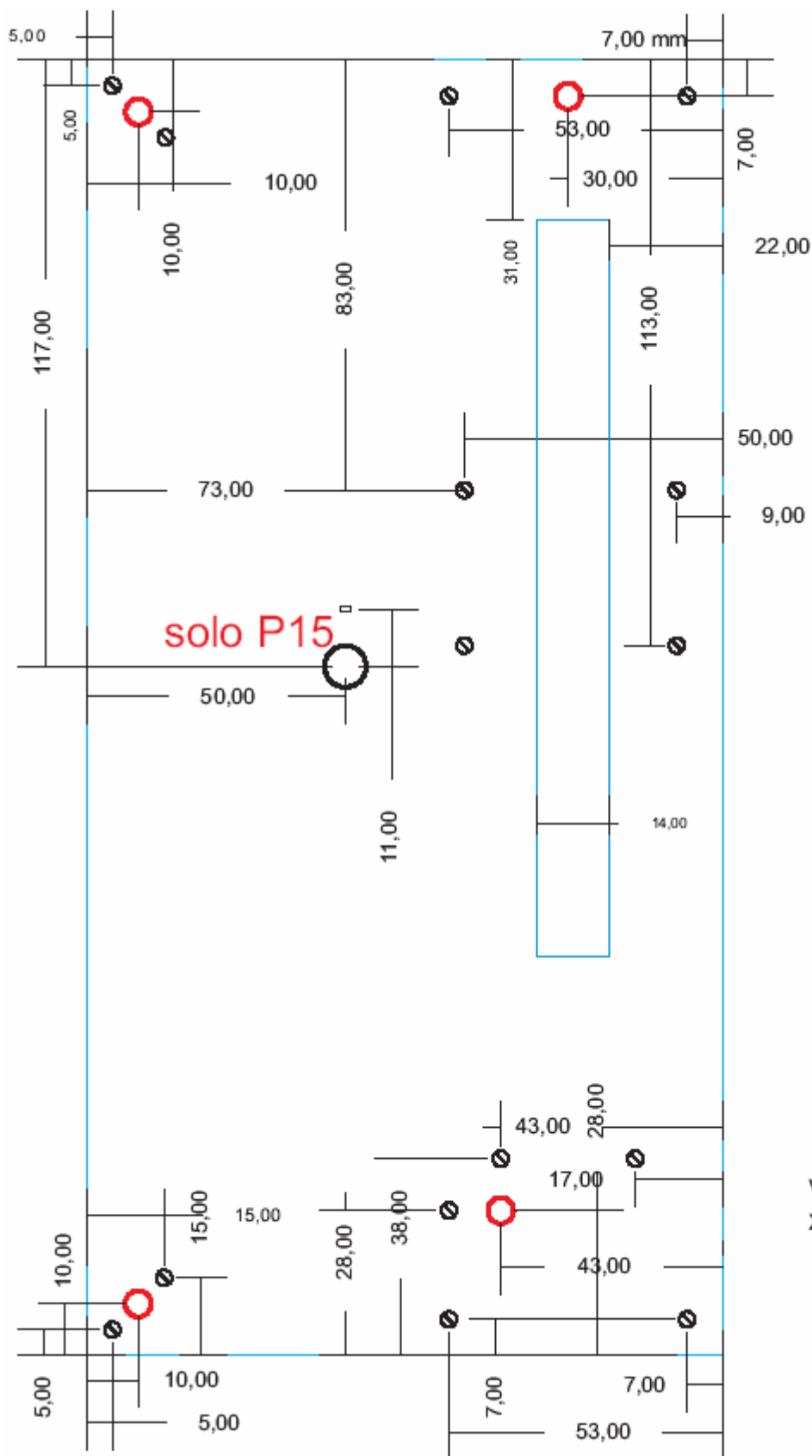
Questa è la visione superiore dell'ala sinistra, costruita a specchio della destra, senza le sigle. Mancano anche le scanalature per il filo di ferro che stringe F alla base e il componente F2 (lamina ricurva di regolazione) con i relativi fori per le viti a croce.

Tutti i disegni sono in scale diverse, non indicate, come le dimensioni, data la natura illustrativa e non tecnica del presente lavoro.

Nella pagina seguente i 9 pezzi di piccole dimensioni necessari (in numero vario), oltre alla Base, P15, F, AlaSup, AlaInf, bulloni (3-5 mm), dadi aperti e ciechi da 5mm, spezzoni di filo di ferro per legare alcune parti ed i cavi.



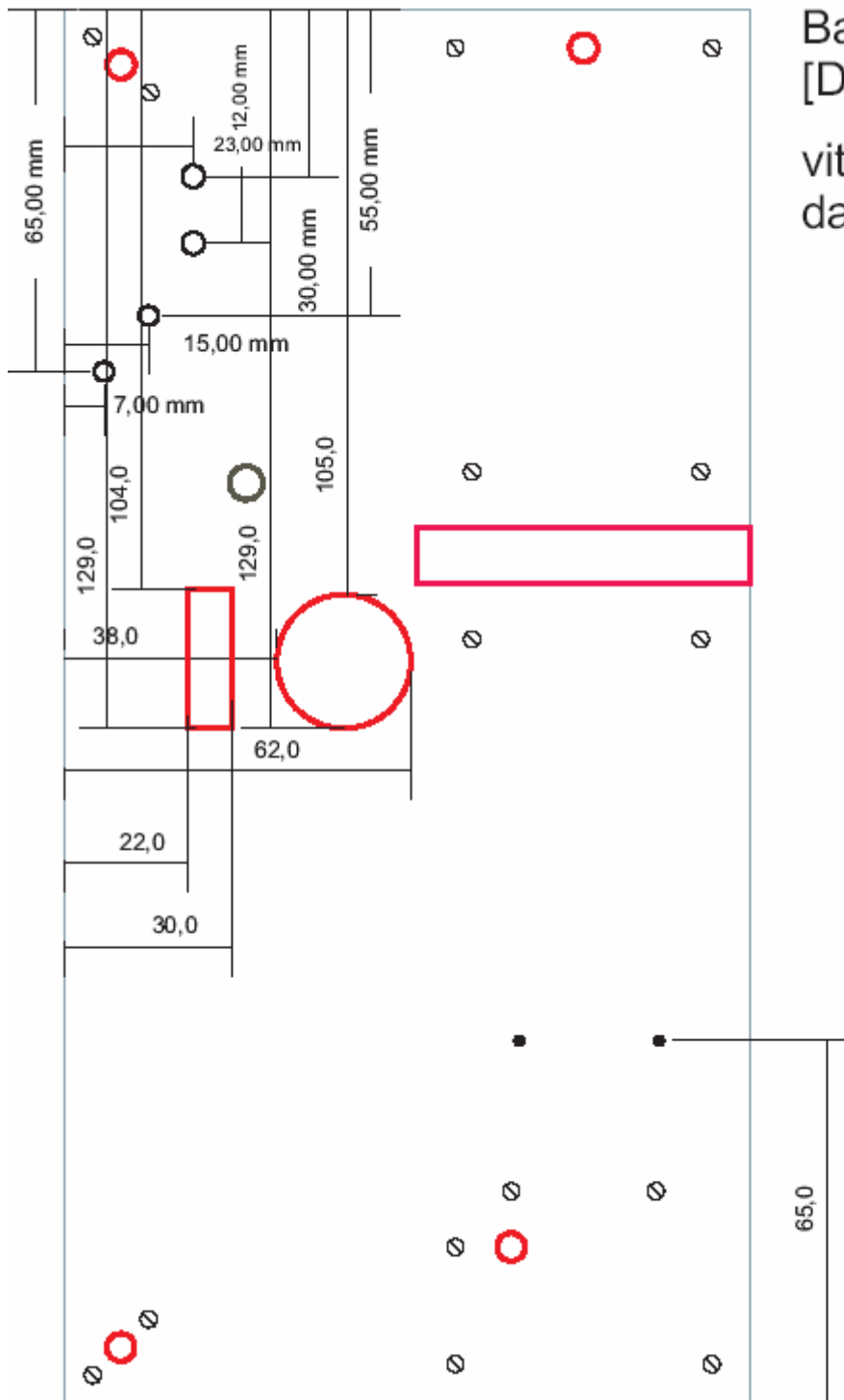
I 9 pezzi diversi necessari con le quote in mm. che indicano dimensioni e le distanze dei fori per le viti da praticare. Nel disegno manca **F**, la barra di ferro zincato che è dimensionata al sensore **S1**. Di ogni pezzo ne sono richiesti un numero indicato nella sigla. P.es. "**P16-17-18-19\_20x20x3 DsSnX 8**" indica che i pezzi identici numerati da P16 a 19 sono di dimensioni 20x20x3 mm. e ne vanno realizzati per le ali Destra e Sinistra in numero di 8. x = ... ed y = ... indicano distanza orizzontale e verticale delle viti dai bordi.



P15 123\*250\*3  
[Ds-Sn(cap.)\*2]

viti 5 mm farfalle  
x=10, y=10

Il disegno superiore di P15, che è la parte superiore del supporto, quella che presenta la finestra rettangolare per S1. Va realizzata in due esemplari (il sinistro è capovolto orizzontalmente). In rosso i fori per bulloni da 5 mm, in nero da 3. In questo pezzo devono essere anche presenti i due fori per le viti di regolazione di **F2**, non presenti in questo disegno ma solo nel primo a pag.78. E' stata una delle modifiche in corso d'opera.

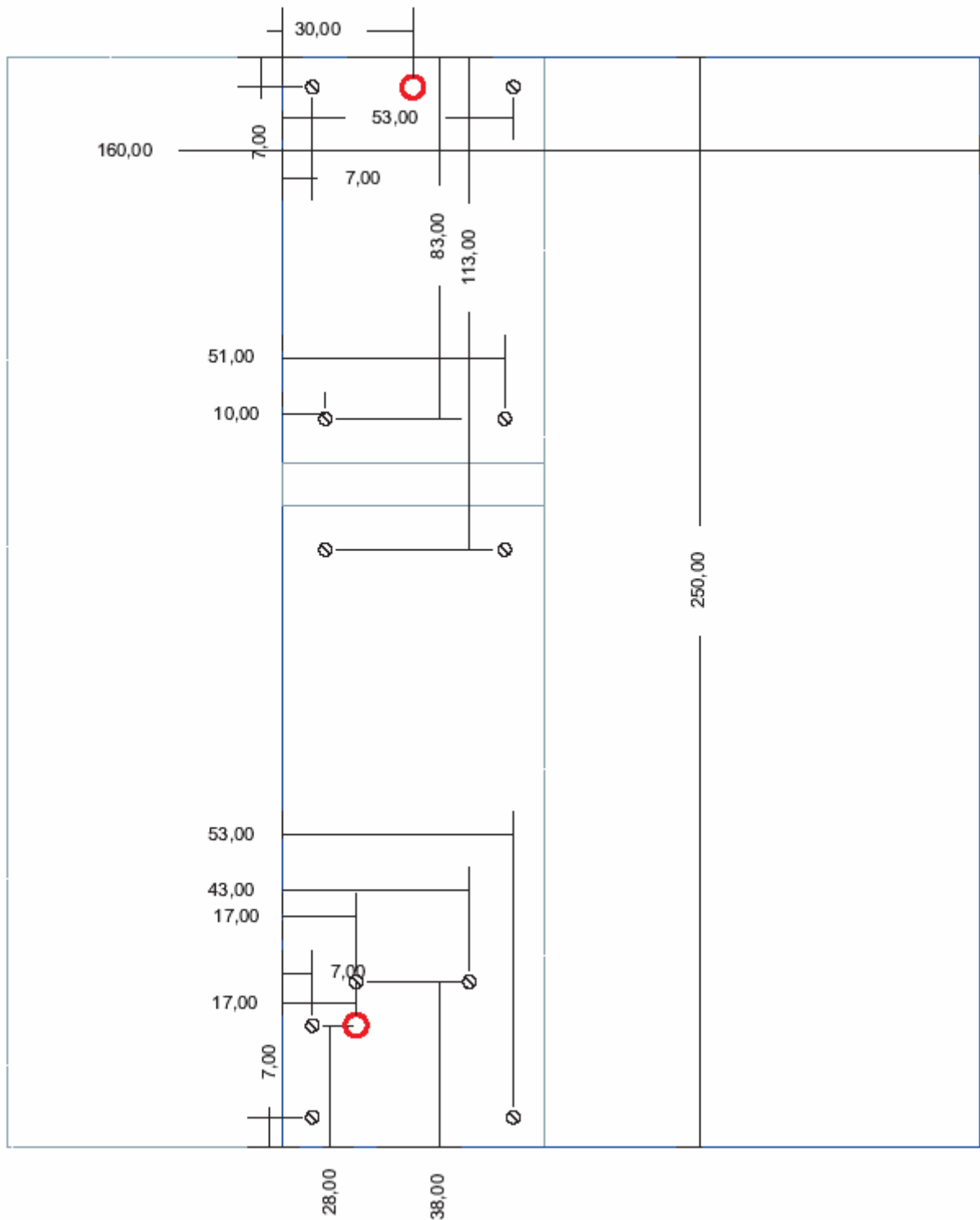


Base 123\*250\*3  
[Ds-Sn(cap.)\*2]

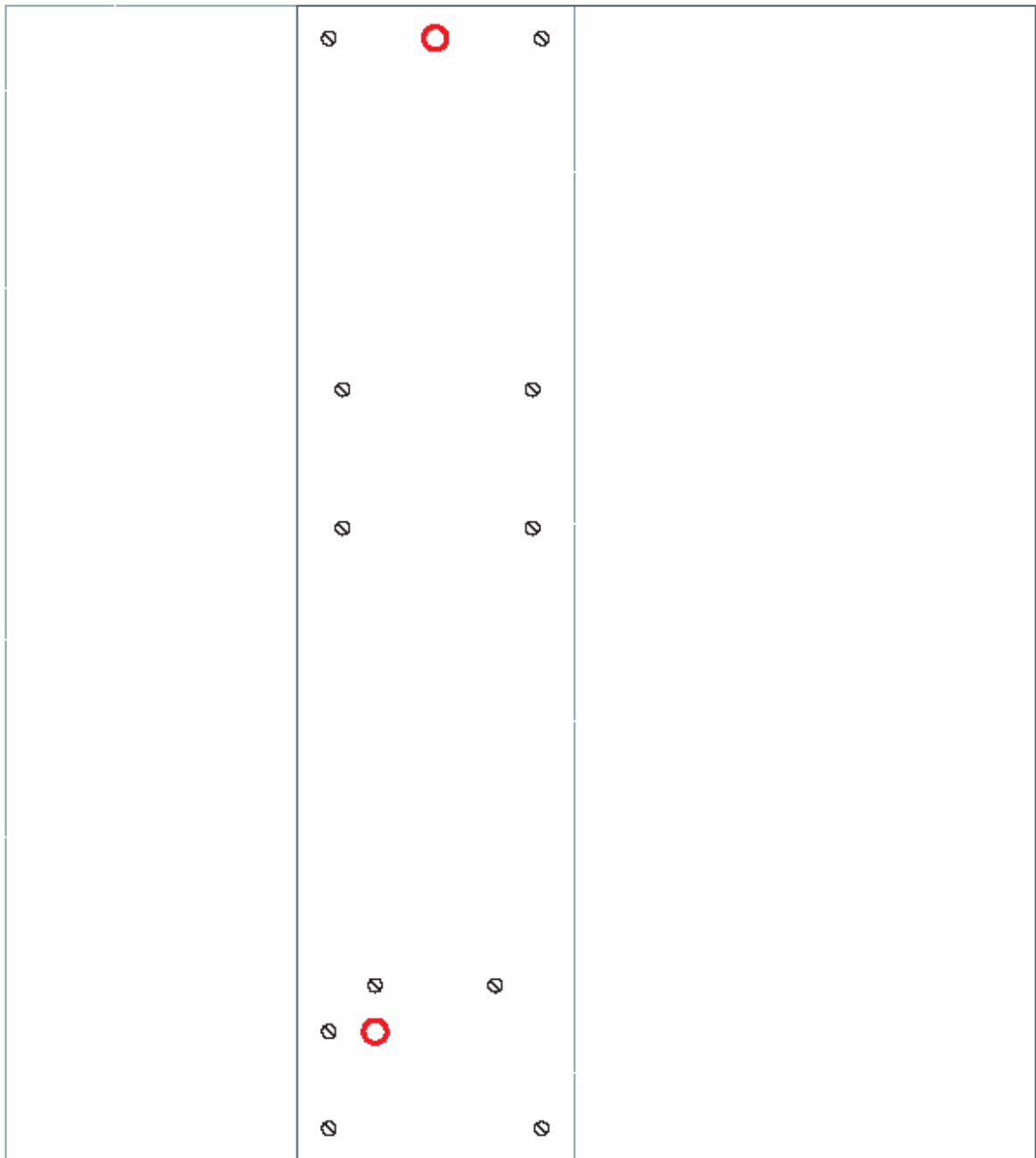
viti 5 mm  
dadi ciechi

Disegno della **Base** in plex, in rosso i fori per bulloni da 5mm e quelli di varie forme da realizzare per dar spazio ai componenti S4 e S2. Si rendono infatti necessari anche i 3 mm. di spessore del pezzo.

In nero i due fori distanti 65 mm. dalla base della figura per l'aggancio di **F** ad essa tramite filo.



Disegno dell'Ala Superiore Ds (160x250mm), in acetone spesso ma flessibile, distinguibile in bordo blu dalla Base con bordo grigio inserita come riferimento.



Ala Inferiore Ds, in acetone sottile, disegnata con bordi neri, per distinguerla dalla Base sempre in grigio. Quest'ala ha la funzione di ingabbiare e proteggere il sensore di flessione S3 e sarà incollata a quella superiore tramite biadesivo.

Nella realizzazione finale le due ali sovrapposte sono state ulteriormente sagomate e dipinte in acrilico al fine di un ulteriore richiamo nominale. [Sia in senso di richiamo al nome "Ali" e "Wings" sia in senso di capacità migliorata di richiamo allo stato di quiete].



### 3.3.2 Realizzazione dell'interfaccia fisica: LittleWings

Si è provveduto ad acquisire i materiali necessari e l'utensileria mancante (minitrapano con mole e frese, morsa, lime, raspe, etc.).

Si è stilata la lista completa dei pezzi da ritagliare, sagomare e forare:

Elenco parti

**P** = parte in plexiglass spessore 3mm

**S1** = Sensore Lineare

**S2** = Sensore pressione

**S3** = Sensor flessione

**S4** = Potenzimetro

**Base** = base in plexiglass 2.5 mm

**F** = Barra di ferro zincato spessore 3 mm porta con adesivo S2

Ds o A = parte destra (mano destra)

Sn o B = parte sinistra ( mano sinistra) [copia speculare della Ds]

**Ala Sup.** = Foglio di acetone destro sottile (posto sopra)

**Ala Inf.** = Foglio di acetone spesso (posto sotto)

**F2** = lamina metallica da piegare e forare per la calibrazione della pressione di S1 su S2

Tutte le misure in mm.

sigla	Lungh max	Larg max	Spess	Da fare
<b>P1</b> (Ds-Sn)	35	35	3 (2.5?)	2
<b>P2</b>	60	15	3	2
<b>P3</b>	60	21	3	2
<b>P4_11</b>	20	10	3	16
<b>P14</b>	60 (24)	55 (33)	3	2
<b>F</b>	183 (181)	30	3	2
<b>Base</b>	250	123	3	2
<b>P15</b>	250	123	3	2
<b>P12</b>	45	24	3	2
<b>P13</b>	24	24	3	2
<b>P16_19</b>	20	20	3	8
<b>AlaSup</b>	250	160	1.5	2
<b>AlaInf</b>	250	160	1~	2
<b>F2</b> [da piegare]			1~	2
Totale pezzi				48
Dimensioni dei Sensori				
<b>S3</b> (9mm ingombro max verticale)	97+55cavo	6 e 8 con cavo	9	
<b>S4</b>			10~12	

Oltre alcuni rifacimenti di piccoli pezzi dovuti ad errate forature, si è dovuta rifare P15, la parte superiore, a causa della rottura per caduta del pezzo non montato. E' da ritenersi la parte più fragile dell'intera struttura, per le forature e finestrate.

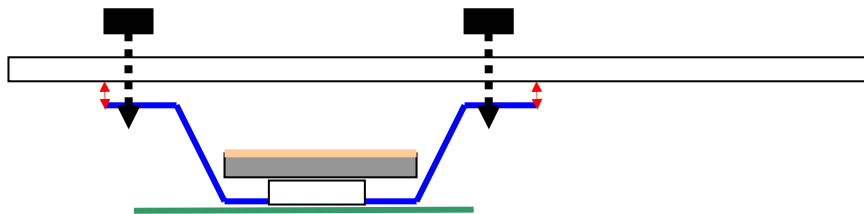
La finestra in P15 è stata smussata nei bordi onde consentire un più agevole scivolamento. I 4 bulloni cechi centrali da 5mm della coppia (2sn della ds e 2 ds della sn) sono risultati leggermente impediti dai vicini bulloni da 3mm.

Il progetto ha anche rivelato una poco agevole montabilità per i 16 bulloni da svitare per ogni ala e soprattutto per la sovrapposizione a sandwich degli elementi. Molti dei quali si sono dovuti incollare.

Come elementi decorativi e di segnatura dei sensori lineari si sono applicati nastri adesivi colorati e di carta ai rispettivi bordi.

### 3.3.2.1 Meccanismo di calibrazione a viti

La modifica al progetto ed al disegno più rilevante è stata quella di realizzare la lamina ricurva per sorreggere **F** che supporta **S1** in modo da poterne calibrare, esternamente al supporto e quindi in fase operativa, la distanza ottimale dal sensore di pressione **S2**. In precedenza era difficile far conservare a **F** la distanza ottimale da **S2**

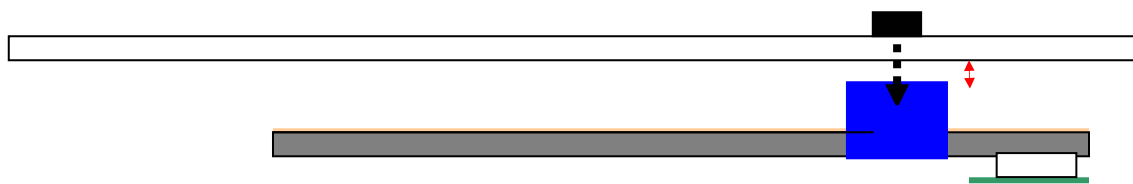


Visione sagittale del meccanismo di calibrazione.

In nero le viti di calibrazione che allontanano ed avvicinano la lamina **F2** (blu e curvata) da **P15** (in bianco). In rosso le frecce indicanti la corsa possibile.

In rosa il sensore lineare **S1** ed in verde quello di pressione **S2**. In bianco anche il quadratino di plex **P1** che insiste direttamente su **S2**.

Per la visione dall'alto → pag.78.



Visione laterale

### 3.3.3 Progettazione della scheda di condizionamento del segnale C8 [L.Seno]

Dal "Manuale del condizionatore 8 canali – Lorenzo Seno (2007)  
*Per gentile concessione dell'autore*

#### Descrizione generale

La scheda C8 è in grado di condizionare 8 canali per segnali lenti in CC per adeguarli ad ingressi analogici unipolari.

Essa consiste di 8 linee di condizionamento identiche. Ciascuna linea permette la sottrazione di un offset e una amplificazione in CC su di un segnale sbilanciato in ingresso, fornendo un segnale sbilanciato in uscita.

Per facilitarne l'uso con sensori potenziometrici, fornisce tre terminali di ingresso per canale: massa, segnale e Vcc.

#### Alimentazione

L'alimentazione è di un solo segno, +5V CC 300 mA stabilizzati, e deve essere provvista esternamente attraverso la morsettiera J2 oppure i due punti JVCC e JGND, rispettivamente per la Vcc e la massa.

La corrente di spunto può essere molto più elevata del valore stazionario a causa del condensatore di filtro C41 sulla Vcc.

La corrente assorbita indicata è per tutti gli otto canali. Montandone un numero inferiore questa può essere scalata proporzionalmente, tenendo conto che il Led di accensione assorbe circa 10 mA.

#### Protezioni

L'alimentazione è protetta contro le inversioni di polarità. Non è protetta contro le sovratensioni. Il limite massimo di tensione applicabile è comunque stabilito dall'operazionale utilizzato.

#### Segnalazioni

Il led verde LED2 segnala la corretta polarità di alimentazione e la sua presenza. Il led rosso LED1 segnala l'inversione di polarità, e va pertanto considerato un allarme.

#### Ingressi

Gli ingressi sono riportati sul connettore J1, SIL con passo 100 mil (2.54 mm), con la seguente mappatura dei terminali:

Pin	Function	Pin	Function	Pin	Function	Pin	Function
1	Ch.1 Vcc	7	Ch.3 Vcc	13	Ch.5 Vcc	19	Ch.7 Vcc
2	Ch.1 Sign.	8	Ch.3 Sign.	14	Ch.5 Sign.	20	Ch.7 Sign.
3	Ch.1 GND	9	Ch.3 GND	15	Ch.5 GND	21	Ch.7 GND
4	Ch.2 Vcc	10	Ch.4 Vcc	16	Ch.6 Vcc	22	Ch.8 Vcc
5	Ch.2 Sign.	11	Ch.4 Sign.	17	Ch.6 Sign.	23	Ch.8 Sign.
6	Ch.2 GND	12	Ch.4 GND	18	Ch.6 GND	24	Ch.8 GND

## Circuito di condizionamento

I circuiti di condizionamento sono identici per tutti i canali. Pertanto nella descrizione che segue si farà riferimento ai componenti del primo, le considerazioni restando le stesse per tutti.

Il circuito effettua nell'ordine dall'ingresso all'uscita le seguenti operazioni:

- Filtraggio passa basso con filtro passivo a tre poli
- Sottrazione dell'offset (stabilito con il trimmer R52)
- Guadagno di un fattore circa 2
- Filtraggio attivo passa basso ad un polo
- Attenuazione (stabilita dal trimmer R67)
- Guadagno per un fattore variabile in funzione della posizione del jumper
- sul connettore J11
- Filtraggio passa basso ad un polo.

Trattandosi di un circuito a singola alimentazione, lo stadio di sottrazione dell'offset rettificherebbe le componenti alternate (quali residui della 50 Hz) eventualmente presenti sugli ingressi, generando un offset in uscita. Pertanto si è provveduto al filtraggio a tre poli del segnale di ingresso prima della sottrazione dell'offset-

Con i valori dei componenti indicati, il circuito assicura una soppressione della 50 Hz superiore a 20 dB con tempo di salita di circa 130 msec, con J11 in posizione 1 (guadagno del secondo stadio pari a 2) ed attenuazione pari 0.5 (guadagno totale 2).

### Primo stadio

Nel primo stadio si trova il filtro passa basso di ingresso a doppio P costituito dalle resistenze R1, R2, R3, R4 da 124 kOhm e i capacitori C1, C2 e C3 da 30 nF. Segue un sottrattore attivo passa basso con rete di controreazione RC costituita da R6 e C5 sul ramo di controreazione e da R5 e C4 sul ramo verso massa del partitore di ingresso. L'amplificatore utilizzato è la prima sezione di U1.

Volendo alterare i valori, si deve tenere conto che per un corretto funzionamento del circuito devono valere le seguenti relazioni:

$$R1 + R2 + R3 + R4 = R51$$

$$R5 = R6$$

$$C4 = C5$$

$$R51 \gg R52$$

In queste condizioni il guadagno in CC dello stadio è:  $G = R6/R51$

Il guadagno nominale del primo stadio, con i valori indicati, è pertanto:  $G = 2$ .

### Secondo stadio

Il secondo stadio è un attenuatore variabile costituito dal trimmer R67 ( $0 < A < 1$ ), seguito da un guadagno non invertente che utilizza la seconda sezione di U1. Con questa configurazione è pertanto possibile sia ottenere un guadagno sia una attenuazione complessivi.

La posizione del jumper sul connettore J11 determina il guadagno del secondo stadio. Le posizioni possibili sono tre:

A) assente: Resistenza verso massa  $R_m = \infty$

B) connessione tra centrale (2) e piedino 1:  $R_m = 47.5 \text{ K}\Omega$

C) connessione tra centrale (2) e piedino 3:  $R_m = 4.75 \text{ K}\Omega$

Il guadagno del secondo stadio è dato da:

$$G = 1 + R69 / R_m$$

La posizione del jumper in J11 determina pertanto il guadagno massimo del secondo stadio (ottenibile con il trimmer R67 in posizione tutta alta).

Inoltre, la posizione del jumper modifica il comportamento in frequenza del 2° stadio. Questo è caratterizzato da una coppia zero-polo, come segue:

$$v_p = \frac{1}{2\pi R69 \cdot C} \quad v_z = G v_p$$

e dunque da un guadagno asintotico (per frequenza infinita) pari a 1. Con i valori indicati,  $v_p = 33.5 \text{ Hz}$

La situazione complessiva è pertanto questa:..

	Range di guadagno 2° stadio	Range di guadagno totale	Freq. di zero (Hz)
A	0 ÷ 1	0 ÷ 2	∞
B	0 ÷ 2	0 ÷ 4	77 Hz
C	0 ÷ 11	0 ÷ 22	368 Hz

## Uscite

Queste sono disponibili su due tipi di connettori, J35 (SIL passo 100 mil, pari a 2.54 mm, 16 poli) e due morsettiere standard a 8 poli passo 5 mm (J27 E J28).

Le mappature sono le seguenti.

<b>J35</b>			
Pin	Function	Pin	Function
1	Ch1 out	9	Ch5 out
2	GND	10	GND
3	Ch2 out	11	Ch6 out
4	GND	12	GND
5	Ch3 out	13	Ch7 out
6	GND	14	GND
7	Ch4 out	15	Ch8 out
8	GND	16	GND

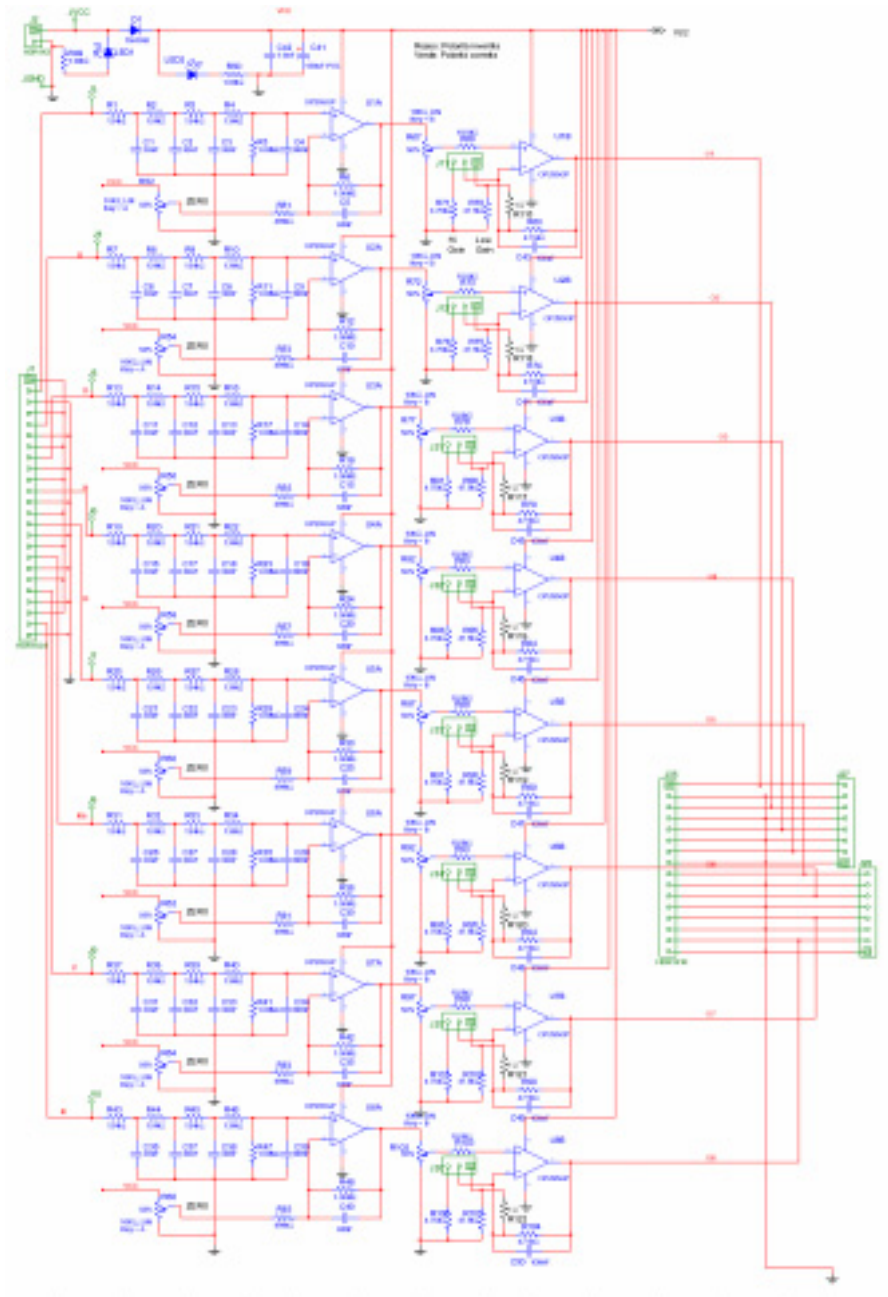
<b>J27</b>			
Pin	Function	Pin	Function
1	GND	5	GND
2	Ch 4 Out	6	Ch 2 Out
3	GND	7	GND
4	Ch 3 Out	8	Ch 1 Out

## J28

Pin	Function	Pin	Function
1	GND	5	GND
2	Ch 8 Out	6	Ch 6 Out
3	GND	7	GND
4	Ch 7 Out	8	Ch 5 Out

L'inversione della piedinatura sulle morsettiere comporta che i canali si succedano nello stesso ordine di J35 e del connettore di ingresso J1.

*(L.Seno 2007 per gentile concessione)*



Il progetto in risoluzione ridotta per ovvi motivi.

### 3.3.4 Realizzazione della scheda di condizionamento del segnale C8

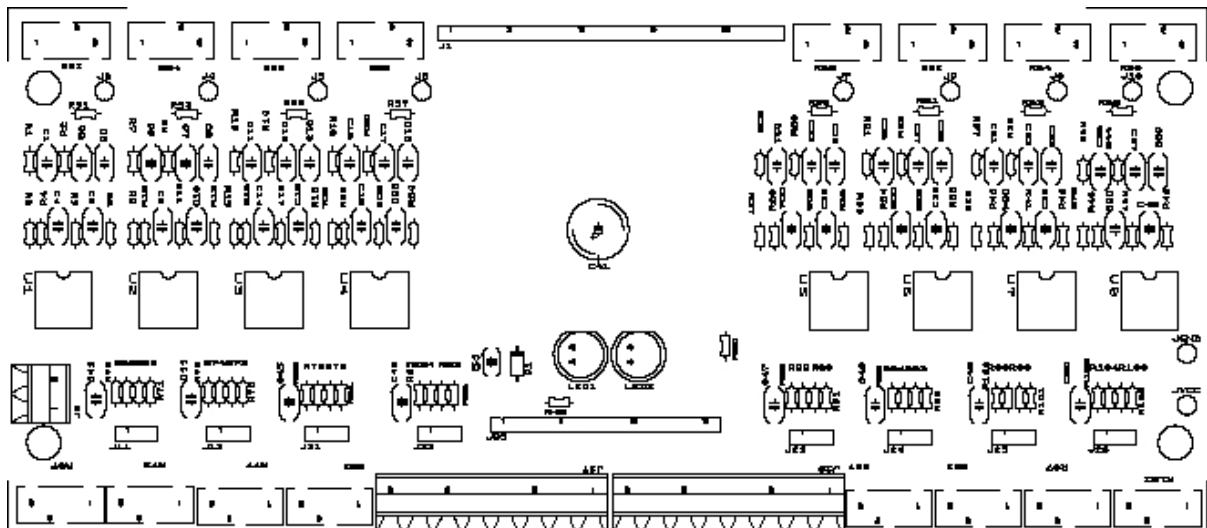
Si è provveduto a far stampare la scheda da uno studio specializzato.

Lista dei componenti necessari alla realizzazione della scheda, alcuni di riserva e vario utilizzo, a destra il riferimento del silk screen:

Qt	Description	RefDes
16	10K_LIN)(Adattatore,nylon,montaggio a pannello,trimmer rettangolare 19mm	R54, R56, R58, R60, R62, R64, R66, R67, R72, R77, R82, R87, R92, R97, R102, R52
16	10K_LIN, 10K_LIN - Trimmer,cermet,lineare,multigirot,regolazione laterale,sigillato,43P,19x6.35x4.8mm,500mW,20K	R54, R56, R58, R60, R62, R64, R66, R67, R72, R77, R82, R87, R92, R97, R102, R52
1	CAP_ELECTROLIT, 100uF-POL -Elettrolitico vert. 0,1MF 63V 4x7mm	C41
1	CAPACITOR, 1.0nF-Condensatore Mylar 1 KpF 100V	C42
8	CAPACITOR, 100nF- Cond. Poliestere 0,1uF 250V Pas:10 da sost. Con passo 5	C43, C44, C45, C46, C47, C48, C49, C50
24	CAPACITOR, 30nF da sostituire con quelli da 22nF	C3, C2, C1, C6, C7, C8, C11, C12, C13, C16, C17, C18, C21, C22, C23, C26, C27, C28, C31, C32, C33, C36, C37, C38
16	CAPACITOR, 68nF Condensatore polies. Box 68 nF 250V passo 10 da sost. Passo 5	C5, C4, C9, C10, C14, C15, C19, C20, C24, C25, C29, C30, C34, C35, C39, C40
3	CONNECTORS, HDR1 terminale PCB,stagnato,fila singola,dritto,36 poli sostituisce tutti gli HDR1Xn	J35, J1,J11, J12, J21, J22, J23, J24, J25, J26 (35poli complessivi)
1	CONNECTORS, HDR1X16	J35
1	CONNECTORS, HDR1X2 2 poli, terminali a vite, PCB, 5mm, AMP 2828362	J2
1	CONNECTORS, HDR1X24	J1
8	CONNECTORS, HDR1X3	J11, J12, J21, J22, J23, J24, J25, J26
2	CONNECTORS, HDR1X8 -8 poli, terminali a vite, PCB, 5mm, AMP 2828368	J27, J28
10	CONNECTORS, TEST_PT1	J3, J4, J5, J6, J7, J8, J9, J10, JGND, JVCC
1	DIODE, 1N4148 Si-Diode SS 100V .2° &lt;4ns	D1
1	Distanziatore Esagonale A=20mm M3	x sovrapposizioni schede
xx	Guaine Termoretraibili silic. D.1,6mm_6,4	Varie dim
2	LED_green 3mm 2,2V 100mW	LED2 + led esterno Make
2	LED_red 3mm 2,2 V 100 mW	LED1 + led esterno Make
1	manopola verde	manopola esterna pot2 Make
2	Manopola Plastica Indice 2,5Cm A.Vite	2 manopole alternative
1	Manopola Rossa con vite lat.	Manopola esterna pot1 Make
10	Mini XLR 3p Presa miniatura pannello	8 (+ 2)
10	Mini XLR 3p Spina miniatura volante	2 di 8 (+ 2)
8	OPAMP, OP295GP IC, amplificatore, reazione di tensione, TS922IN	U1, U2, U3, U4, U5, U6, U7, U8
2	RESISTOR, 1.00kOhm_1%	R49, R50
10	RESISTOR, 1.00kOhm_1%- pellicola metallica,alta stabilit�,0.25W	10

16	RESISTOR, 1.00Mohm_1%	R6, R5, R11, R12, R17, R18, R23, R24, R29, R30, R35, R36, R41, R42, R47, R48
10	RESISTOR, 1.00Mohm_1% -pellicola metallica,alta stabilità,0.25W,1Mohm,1%	10
8	RESISTOR, 10.0kOhm_1%	R68, R73, R78, R83, R88, R93, R98, R103
10	RESISTOR, 10.0kOhm_1%-pellicola metallica,alta stabilità,0.25W,10kOhm,1%	10
32	RESISTOR, 124kOhm_1%	R1, R2, R3, R4, R7, R8, R9, R10, R13, R14, R16, R15, R22, R21, R20, R19, R28, R27, R26, R25, R34, R33, R32, R31, R40, R39, R38, R37, R46, R45, R44, R43
32	RESISTOR, 130kOhm,1% pellicola metallica,alta stabilità,0.25W, (ex 124kOhm)	R1, R2, R3, R4, R7, R8, R9, R10, R13, R14, R16, R15, R22, R21, R20, R19, R28, R27, R26, R25, R34, R33, R32, R31, R40, R39, R38, R37, R46, R45, R44, R43 (32+8 res)
10	RESISTOR, 4.70kOhm_1%-pellicola metallica,alta stabilità,0.25W,4.7kOhm,1%	10 pz reserve
8	RESISTOR, 4.75kOhm_1%	R71, R76, R81, R86, R91, R96, R101, R106
10	RESISTOR, 47.0kOhm_1%- pellicola metallica,alta stabilità,0.25W,47kOhm,1%	10 pz reserve
16	RESISTOR, 47.5kOhm_1%	R70, R69, R75, R74, R79, R80, R84, R85, R89, R90, R94, R95, R99, R100, R104, R105
8	RESISTOR, 499kOhm_1% (sost 510kOhm)	R51, R53, R55, R57, R59, R61, R63, R65
8	RESISTOR, 510kOhm,1% -pellicola metallica,alta stabilità,0.25W, (ex 499kOhm)	R51, R53, R55, R57, R59, R61, R63, R65 (8+2)
1	Box 260x250x90	

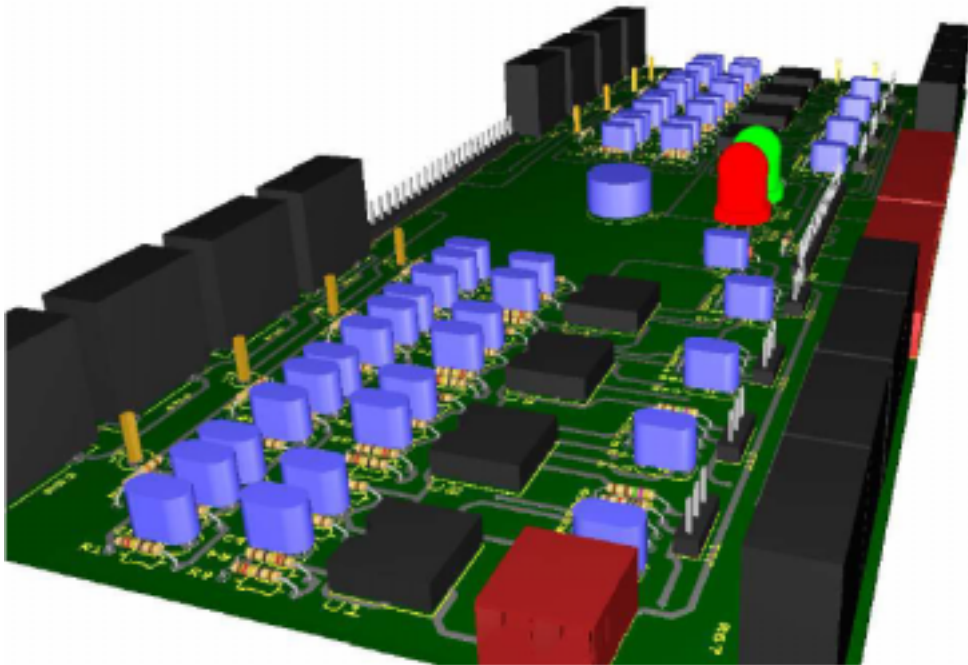
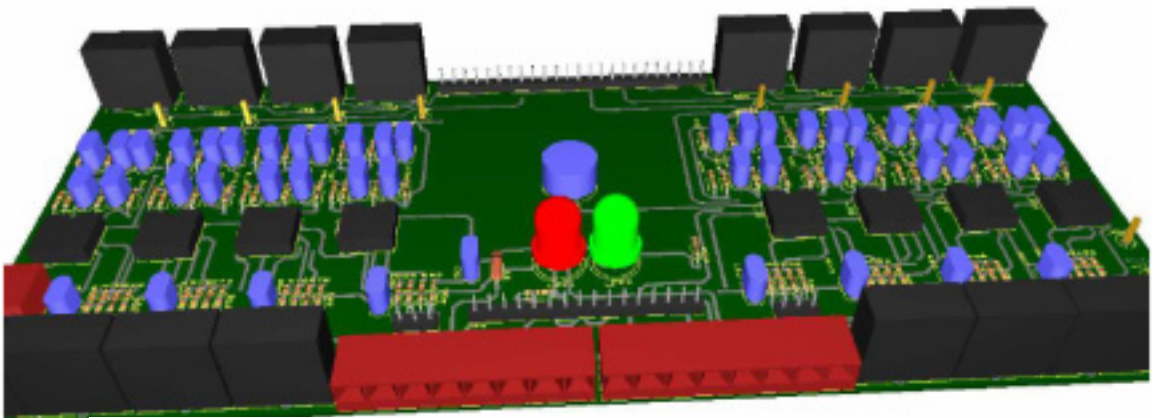
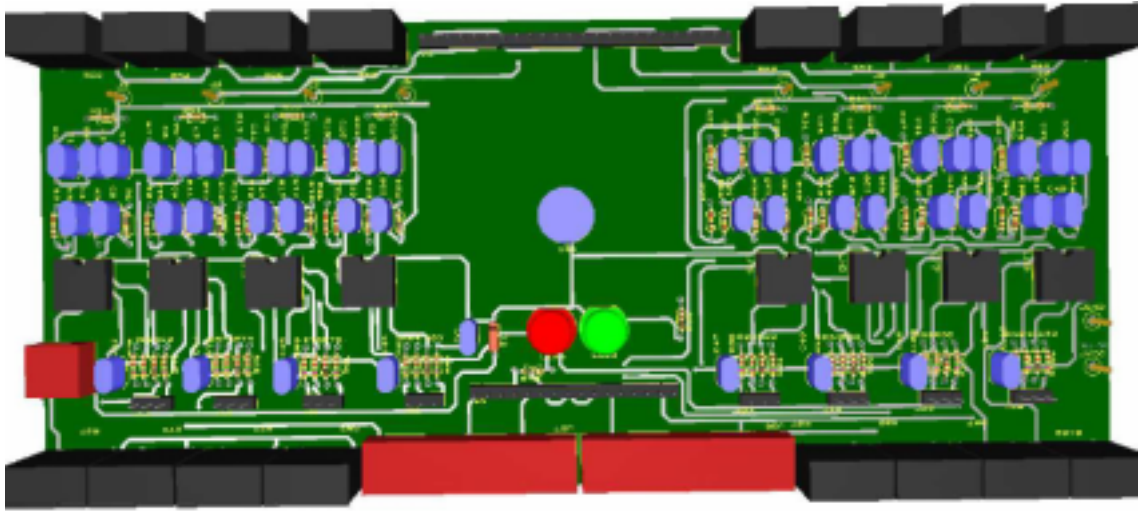
### Silk screen

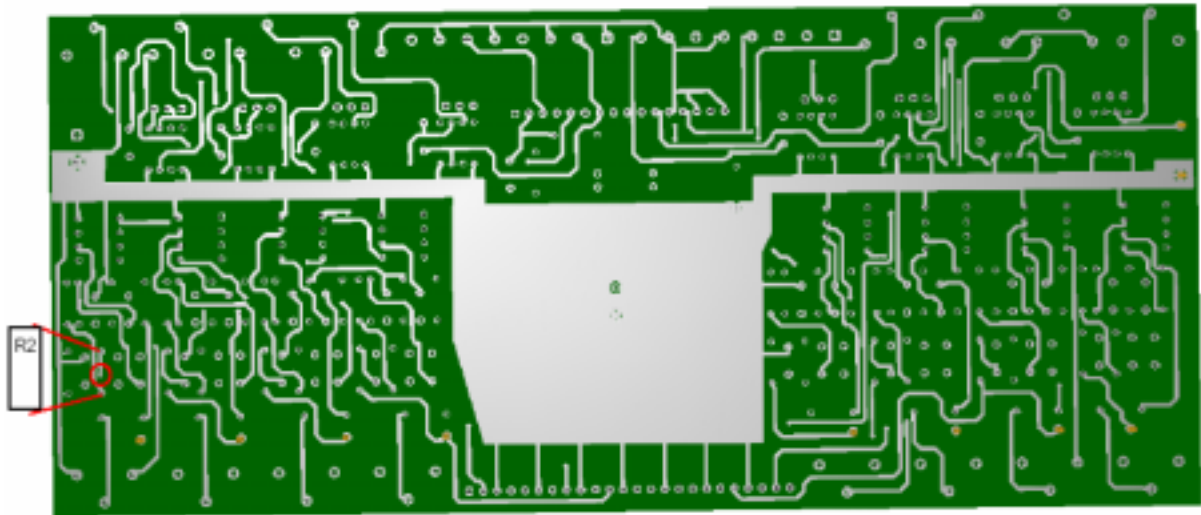


In corso d'opera e di acquisizione componenti alcuni di essi sono stati modificati a seconda delle disponibilità del mercato.

Oltre al silk screen stampato sulla scheda e su fogli diversi per ingrandimenti, per la saldatura dei componenti elettronici si sono utilizzate le visioni d'insieme realizzate dallo stesso autore.



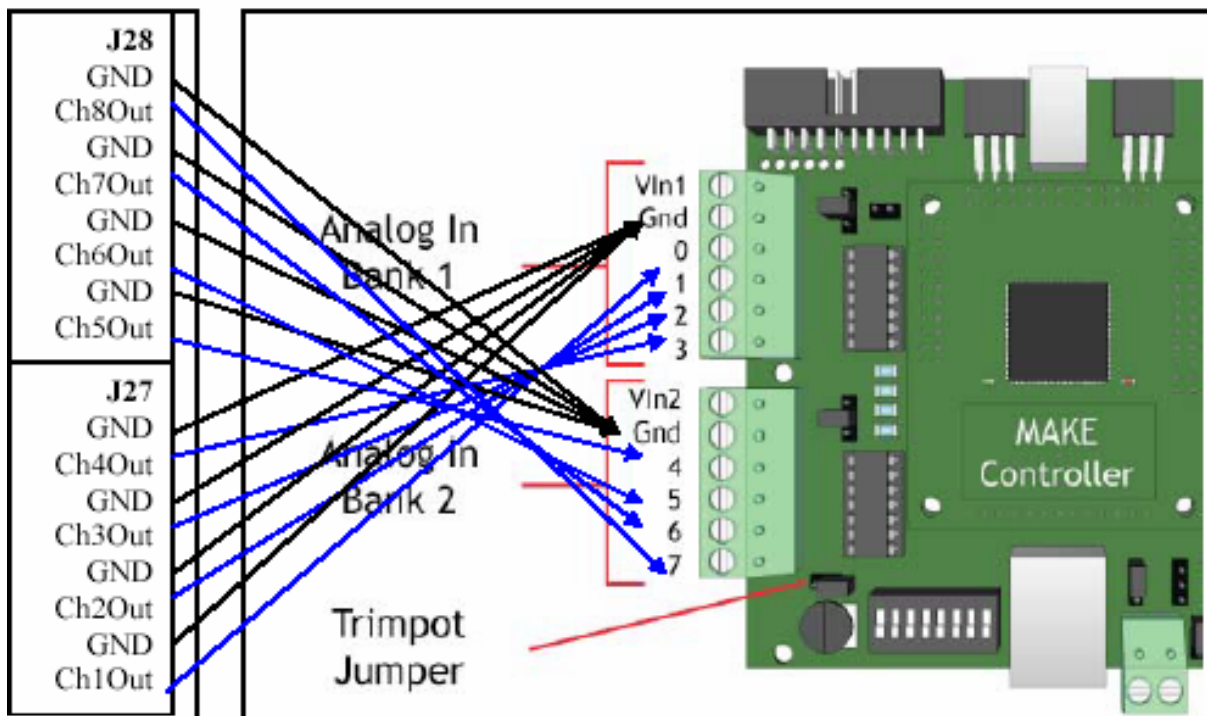




retro della scheda stampata

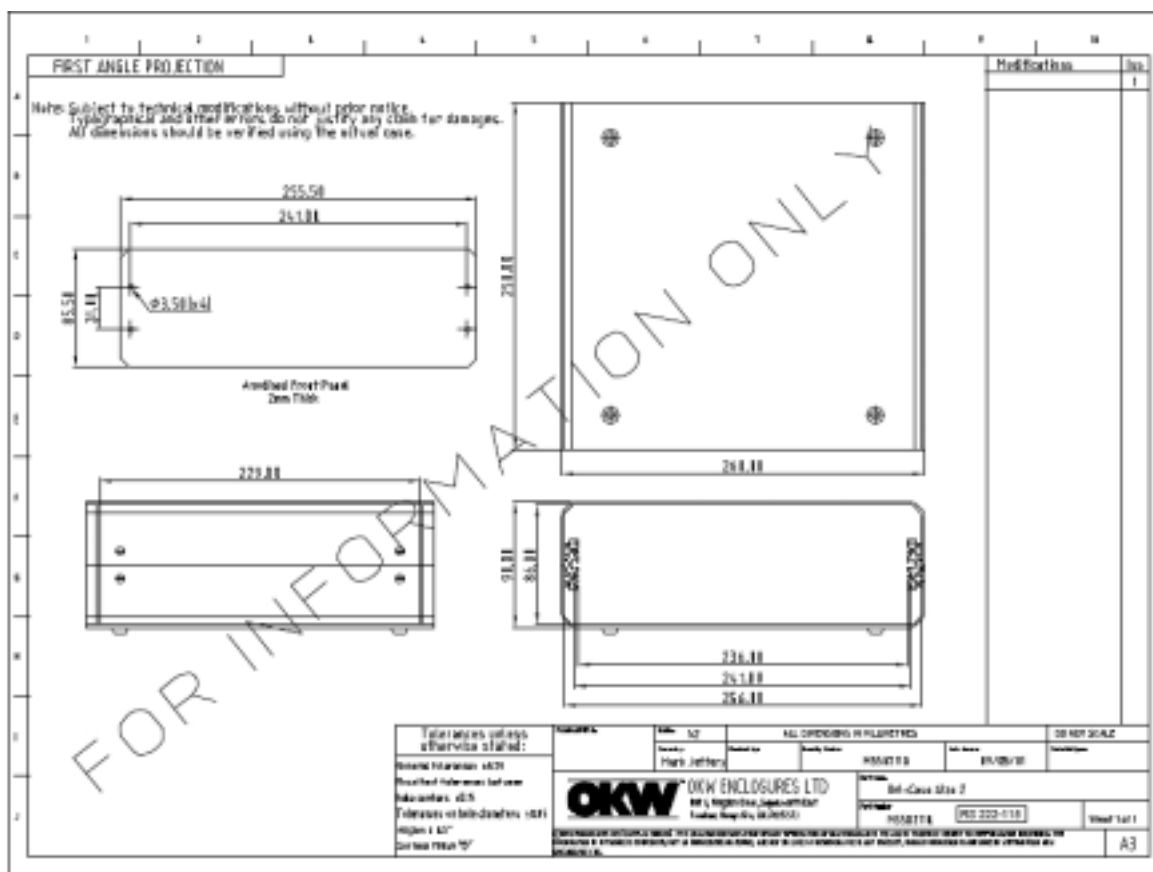
Essendo stata disegnata in previsione di una sua sovrapposizione alla scheda MakeControllerKit, le uscite dei segnali furono pensate in tale funzione e quindi ordinate adeguatamente. In seguito alla realizzazione della scheda ci si accorse che tale sovrapposizione era impraticabile ed io scelsi di affiancarle in un certo ordine.

Di seguito il disegno dei collegamenti tra le uscite dei segnali della C8 agli ingressi degli stessi nella Make. In nero il ground e in blu i segnali analogici



### 3.3.5 Progettazione del contenitore/supporto dell'interfaccia LittleWings

Dovendo contenere due schede non sovrapponibili a causa della presenza di leds indicatori dello stato di servizio su entrambe e sulla non corrispondenza tra le dimensioni per l'inserimento dei fori necessari al fissaggio si è preso in considerazione un contenitore quadrato. In considerazione dell'opportunità di utilizzare il contenitore stesso come sostegno alle superfici sensibili, invece di montarle, come originariamente previsto, su un apposita struttura metallica indipendente, si è trovato un contenitore di dimensioni adeguate. E' anche in base alle sue dimensioni che sono state disegnati i supporti delle ali.



Esso è in alluminio formato da due parti che si collegano tramite sostegni scanalati, pensati per l'inserimento schede, e presenta due pannelli comandi in alluminio leggero.

#### 3.3.5.1 Cavi e connettori



I sensori vanno collegati mediante i tre conduttori di cui dispongono: GND, V, signal.

La scelta per i connettori da inserire nel contenitore e cui collegare i cavi dei sensori è caduta sui connettori tripolari Mini XRL. A sinistra il maschio, a destra la femmina da applicare con due viti al pannello del contenitore.

La cavetteria utilizzata per collegare tutti i componenti risponde alle specifiche AWG 22, con una sezione di 0.35 mm<sup>2</sup> e un

diametro di 0.64 mm.

L'utilizzo di connettori miniaturizzati giustificati dalle minori ed ottimali dimensioni dei cavi, ha portato inevitabilmente qualche difficoltà nella saldatura, considerata anche la ridotta tenuta termica dei componenti plastici.

Si è anche utilizzato per il contenitore un interruttore a bilancia per consentire un rapido reset dell'interfaccia in caso di overflow dei segnali.

Della prima realizzazione si sono utilizzati anche due mammuth per l'alimentazione, portata su uno dei pannelli con apposito connettore circolare.

### 3.3.5.2 Disegno delle aperture da praticare nel contenitore

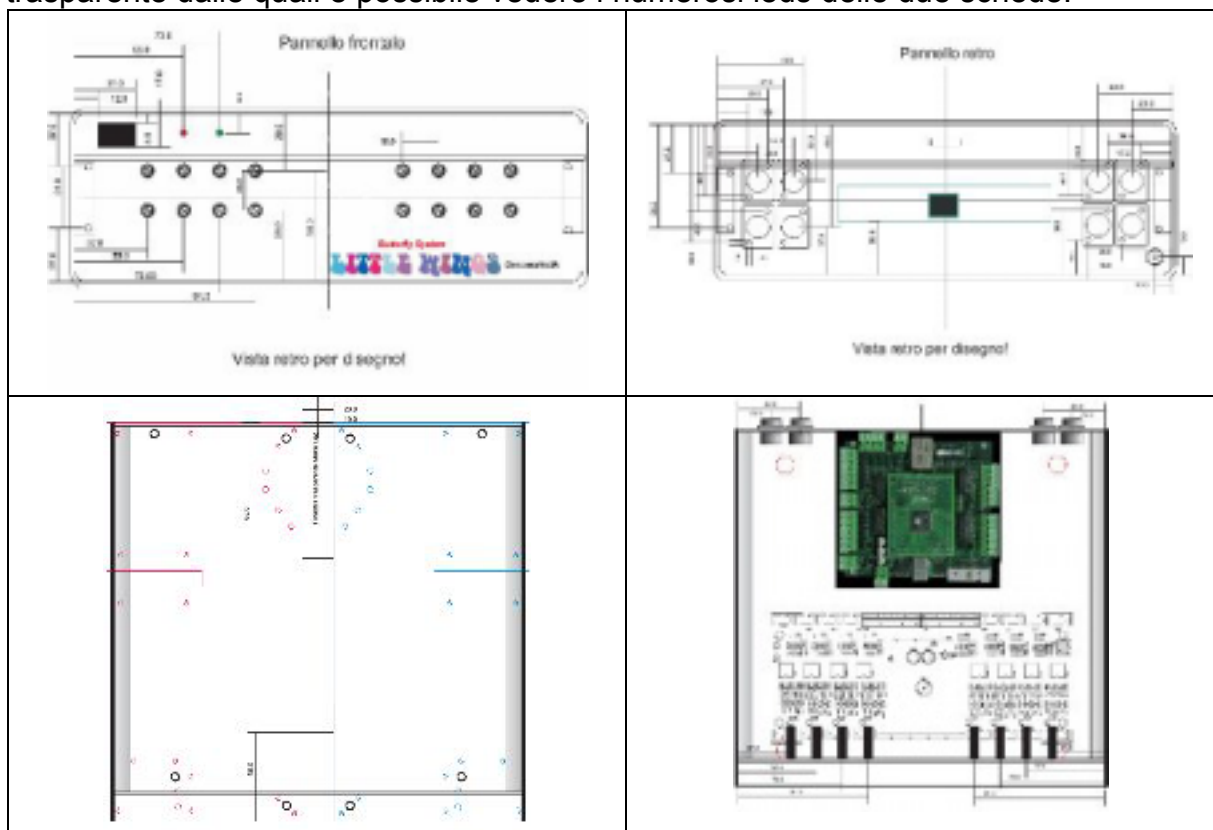
Determinati così gli ultimi componenti del contenitore-sostegno, si è provveduto a disegnarne le aperture necessarie da praticare.

A lungo si è cercata la migliore disposizione possibile dei numerosi elementi che, soprattutto per la scheda MakeControllerKit fornita di porta **USB** e **LAN** sui lati opposti, vincolava uno dei due accessi. Così è stato anche per la scelta della sistemazione delle 8 prese Mini XLR e dei 16 trimmer a vite con supporti orizzontali. Alla fine, come risulta dai disegni del capitolo successivo, si è optato per privilegiare la porta Ethernet posta al centro a media altezza in compagnia delle otto prese XLR disposte a gruppi di 4 ognuna per la sovrastante ala.

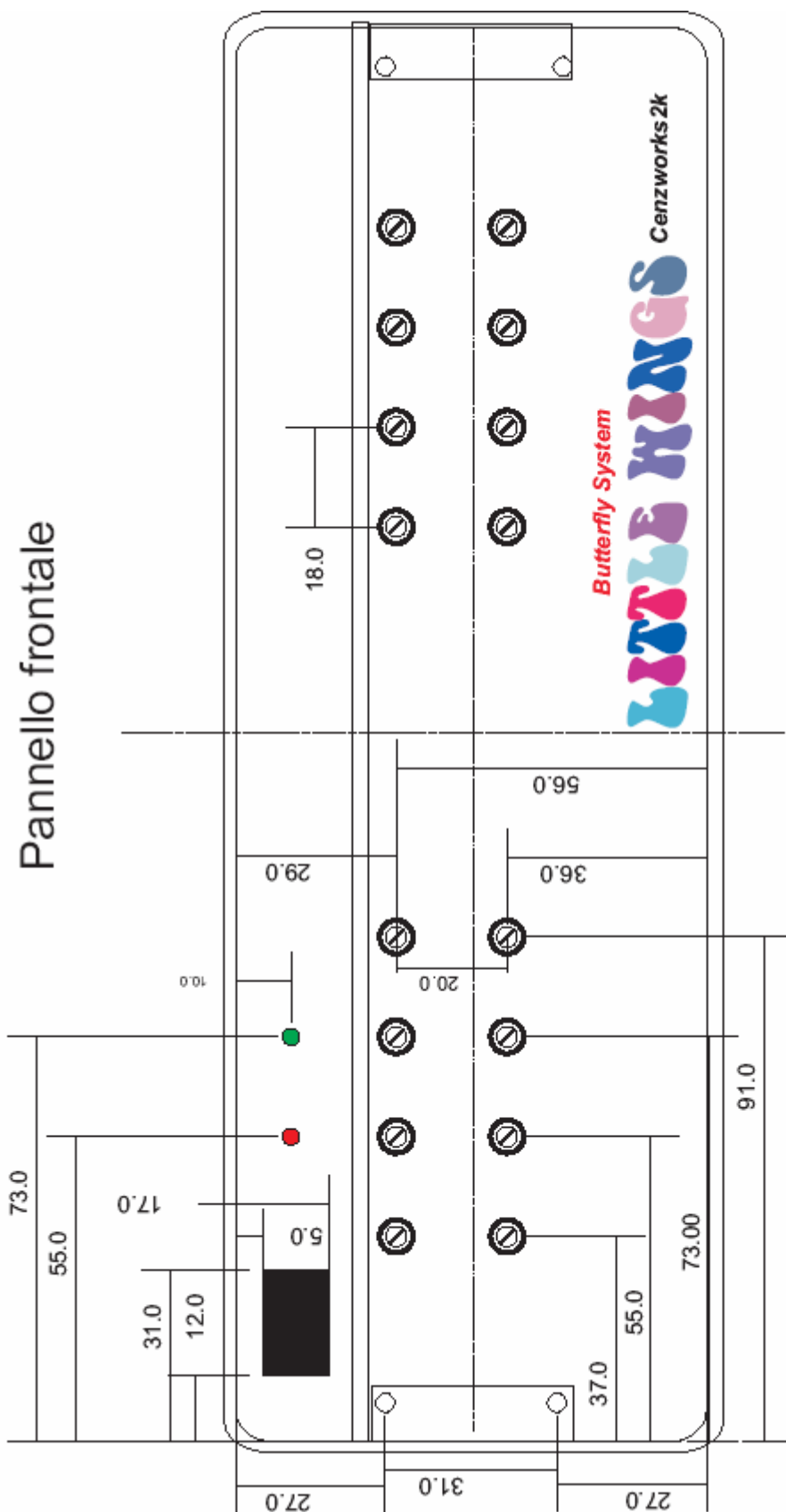
I trimmer hanno invece trovato posto sul frontale anch'essi in due gruppi sovrapposti, zero e fondoscala.

Si è aggiunto anche un piano che permette di inserire e contenere una prolunga USB che ne consente il relativo collegamento esterno, vista la posizione interna nella quale per forza di cose si viene a trovare. Detto collegamento è previsto con un accesso superiore a quello del cavo Ethernet.

La superficie superiore presenta due piccole finestre chiuse con materiale trasparente dalle quali è possibile vedere i numerosi leds delle due schede.

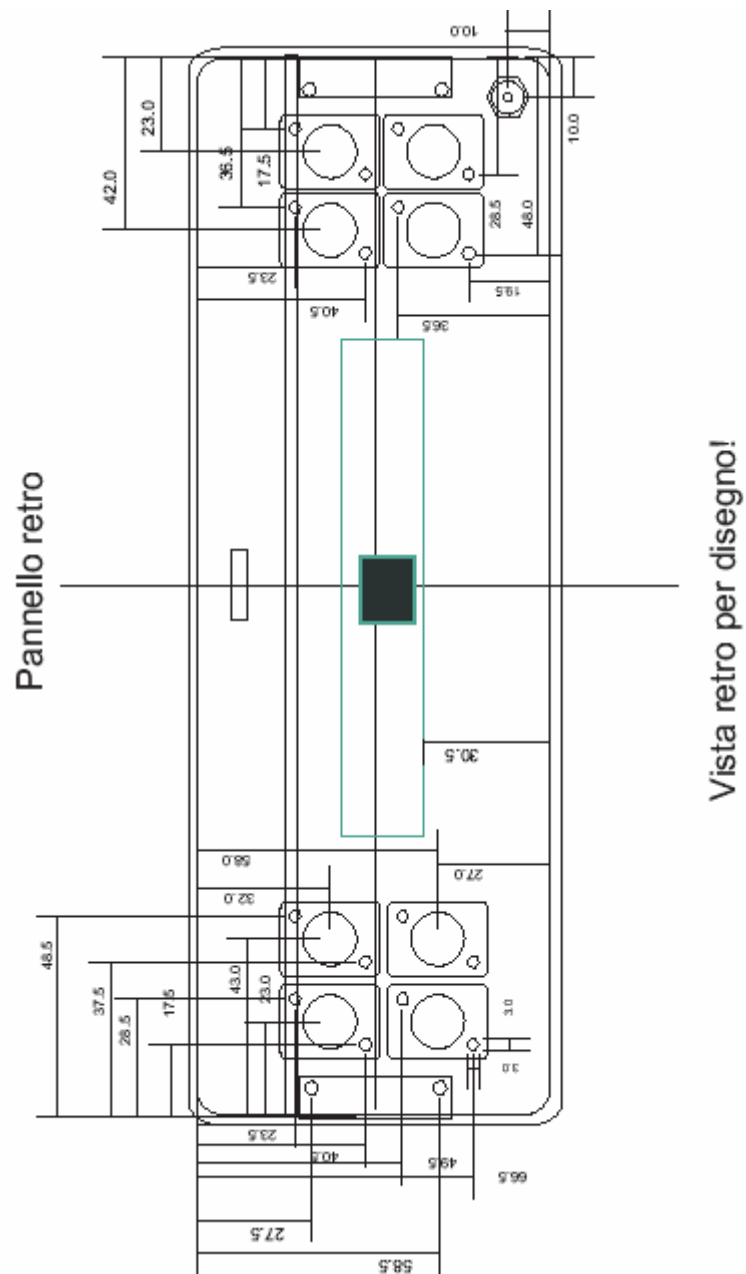


Visione ridotta frontale, retro, superiore, interno.



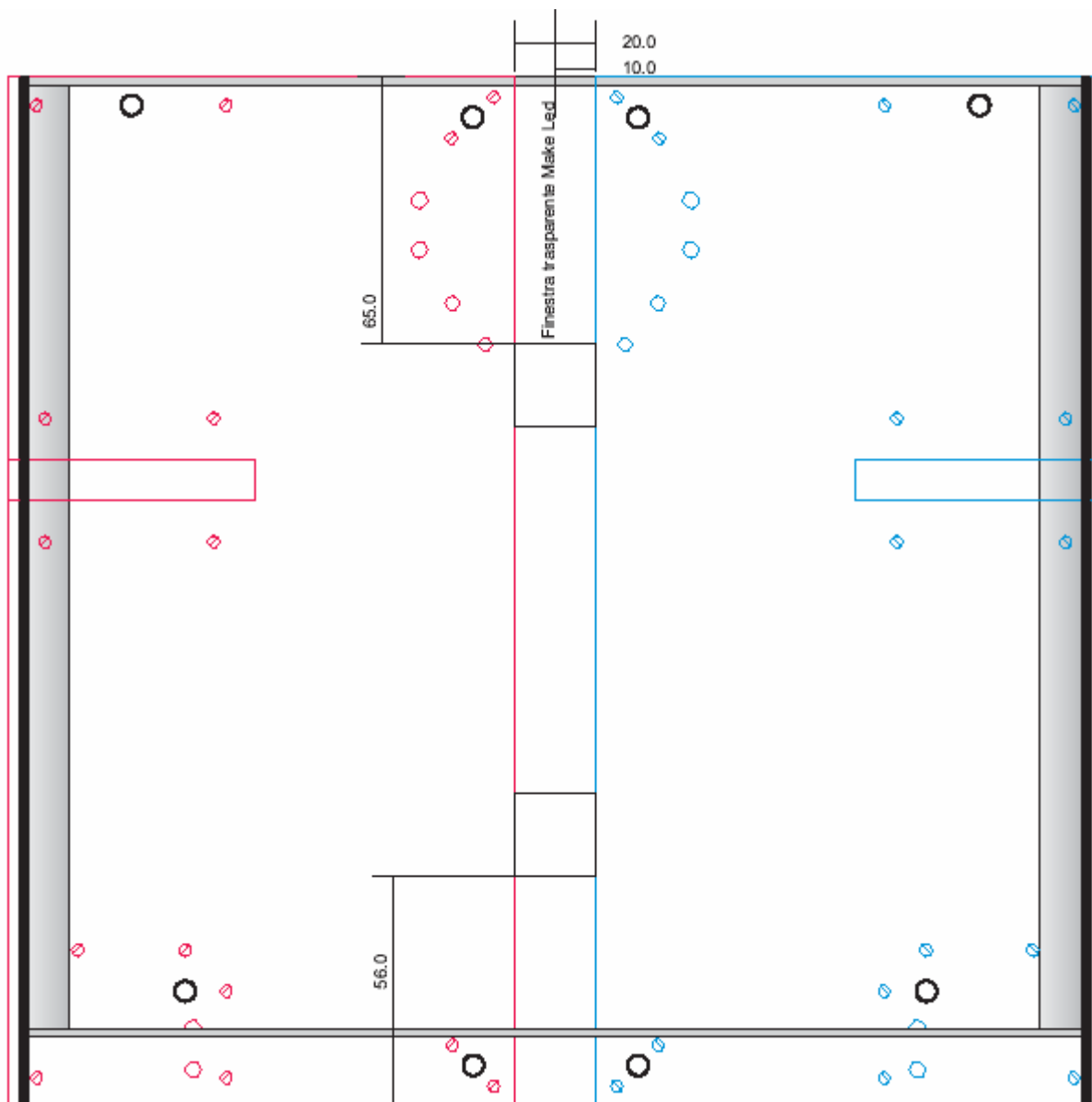
Frontale. Eccetto la scritta (non ancora realizzata), il resto è da intendersi disegnato sul retro del pannello, onde consentirne una più sicura foratura.

Sono presenti i 16 fori circolari per le viti di regolazione dei trimmer, due per i leds (realizzato uno solo) e l'apertura per l'interruttore a bilancia.



Sono visibili i fori per le otto prese da pannello con i sedici fori relativi per le viti, l'apertura centrale in nero/verde per il connettore RJ45 Ethernet, quella per la porta USB ed in basso ad un angolo l'apertura per fissarvi il connettore di alimentazione generale.



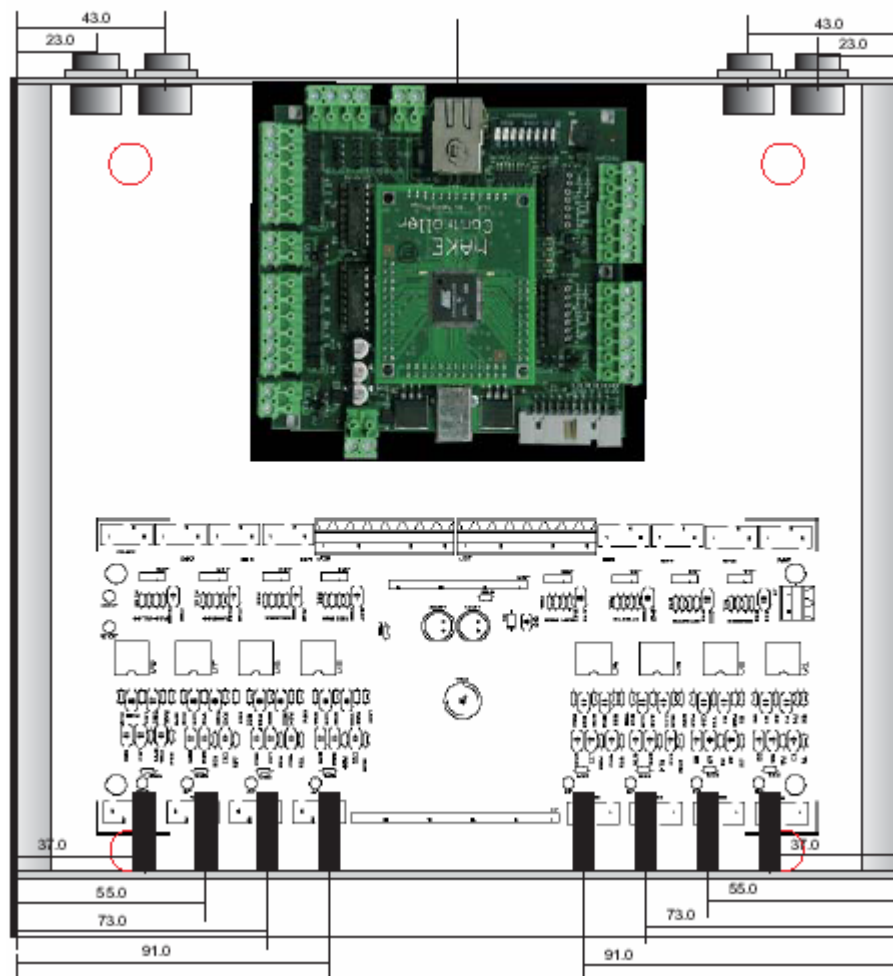


La superficie superiore in alluminio spesso del contenitore deve prevedere il sostegno dei due elementi porta-sensori. Nel disegno questi sono colorati in blu e rosso con le proprie viti. Devono quindi essere praticati 8 fori per i bulloni di sostegno, indicati dai fori in nero-spesso.

Nella zona centrale, rimasta libera dalle superfici adiacenti, vengono praticate due aperture quadrate, ricoperte da materiale trasparente, per la vista dei leds interni.

Le zone laterali sfumate in grigio rappresentano i bordi smussati del contenitore, mentre i bordi neri sono le pareti laterali del contenitore stesso.

Come si può notare le parti colorate, che rappresentano i corpi rigidi dei supporti superiori, sporgono leggermente.



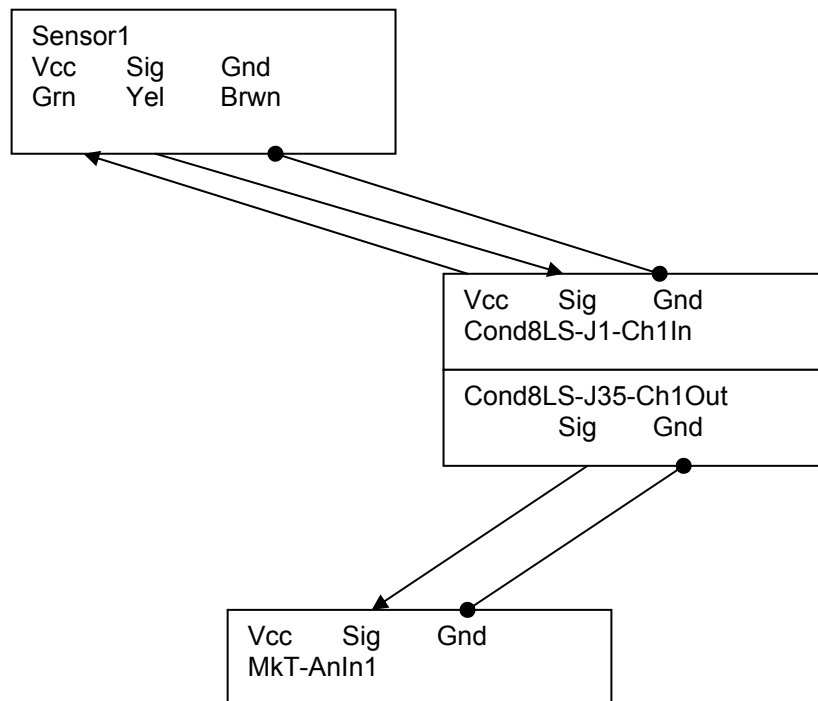
L'interno del contenitore con le due schede (MakeControllerKit e C8) raffigurate. Disegno per valutare l'ingombro delle prese XRL, la distanza del connettore RJ45, l'ingombro dei trimmer e la posizionatura dei fori per le viti di aggancio delle schede al contenitore stesso.

In rosso i 4 piedi in gomma del contenitore.

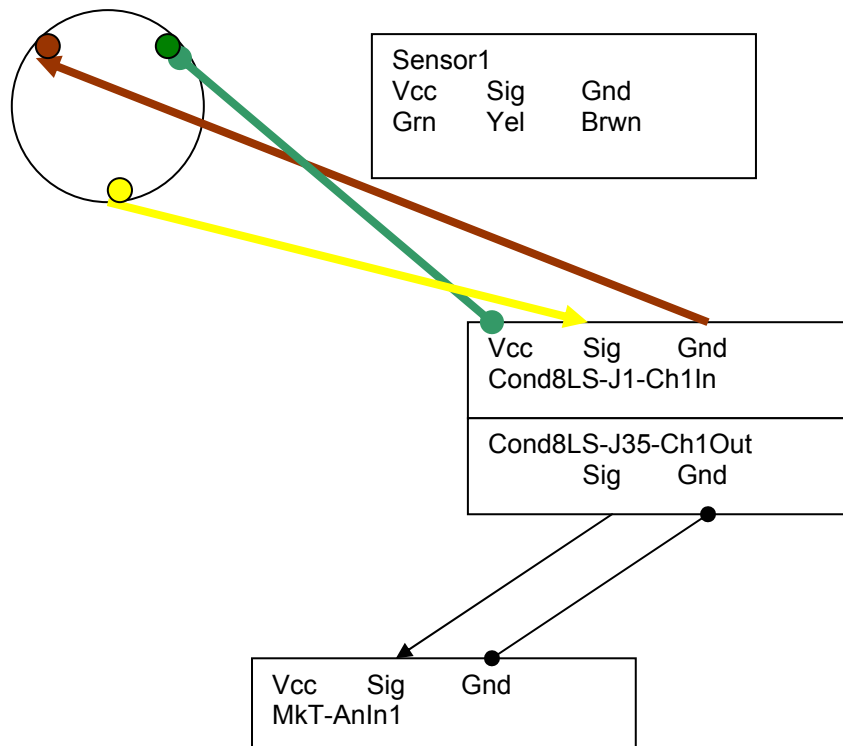


### 3.3.6 Realizzazione finale dell'interfaccia LittleWings

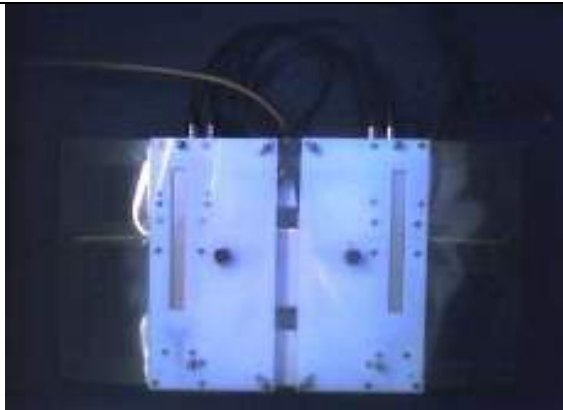
Si sono scelti cavi tricolori (giallo, marrone, verde) per i collegamenti dei sensori sempre AWG22 e le connessioni sono state così formulate



Cond8LS = condizionatore  
MKT = makethings Board



### 3.3.6.1 – Alcune foto della realizzazione



*Visione dall'alto*



*Di lato dal basso*



*Interruttore, gli 8 trimmer di destra e l'ala destra*



*Frontale in piano (particolare strati e bulloni)*



*Retro con connessione Ethernet e 8 cavi inseriti*



*Mini XLR, 2 spine e 4 prese*



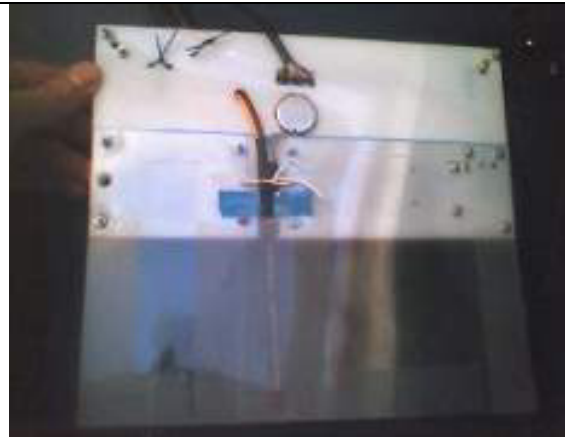
*S1 e S4, particolare delle viti di calibrazione*



*Dall'alto senza le ali con le due finestre monitor led*



*Retro senza Ali*



*Retro dell'Ala, in vista il S3 e S4*

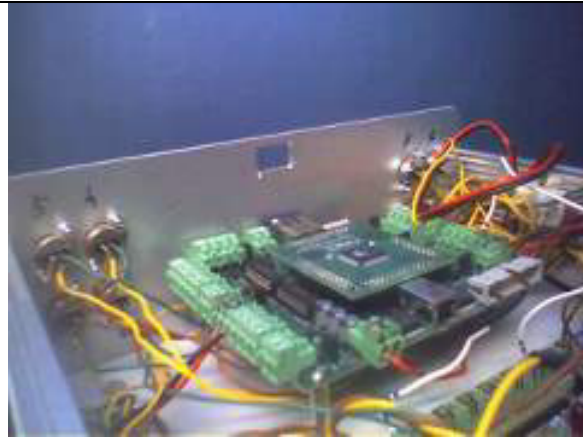


*Coperchio smontato e rovesciato*

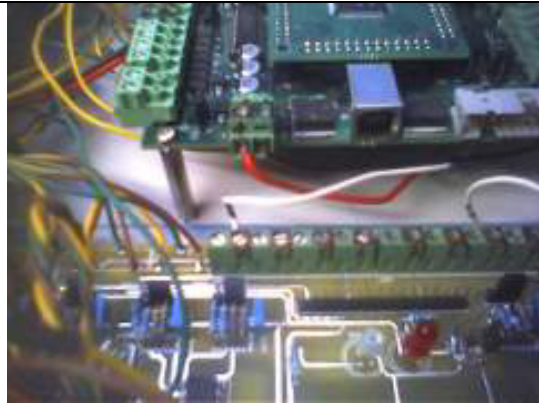


*Scheda C8 e trimmer frontali*

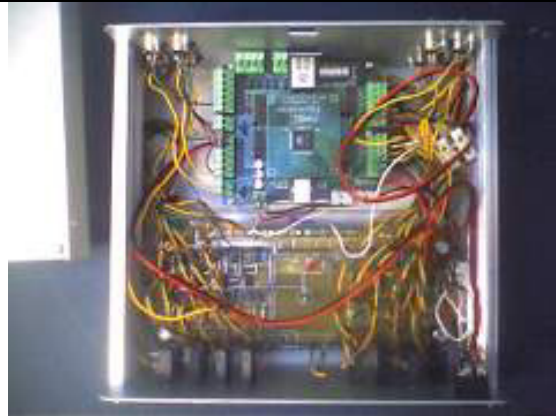
I tre colori dei cavi indicano: giallo= segnale, marrone = ground, verde = tensione; i cavi bianchi portano gli 8 segnali dalla scheda C8 alla Make per la conversione ADC. Il rosso è la tensione 9V ed il nero il ground.



*Scheda Make e Mini XLR*



*Differenza di livello delle due schede*



*Visione dall'alto*



*Aperto con coperchio di lato*

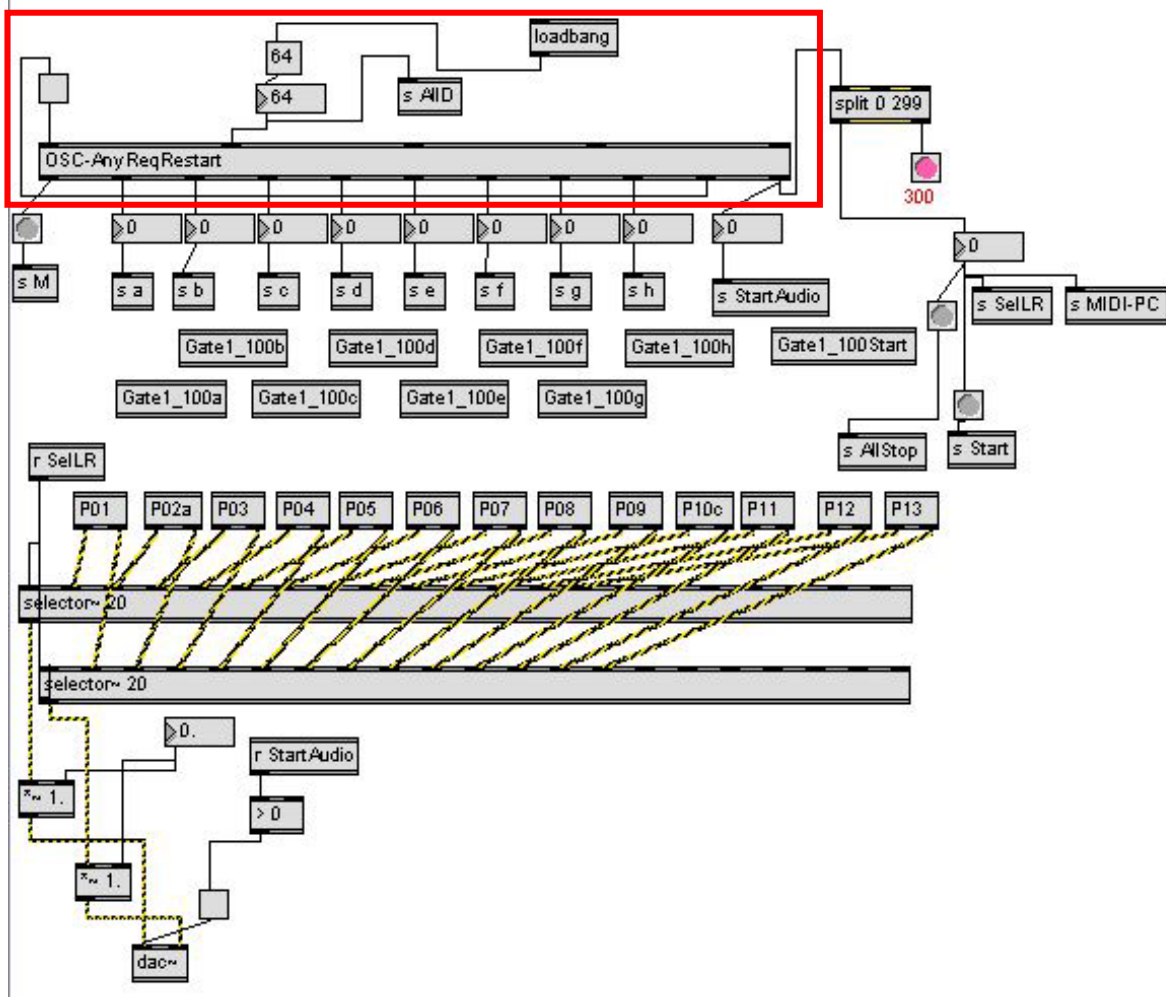
## 3.4 – Realizzazione dei componenti software mancanti

Terminata la realizzazione dell'hardware si procede con quella software di supporto alle composizioni. E precisamente alle interfacce software di **Max** e **Flash** tra esse e in rete **LAN** con **LittleWings**.

### 3.4.1 Algoritmo di connessione OSC in Max-MSP

Di seguito alle procedure di cui alla 3.2.3.1 si formula la struttura di base per le composizioni algoritmiche in Max. Esse prevederanno un algoritmo di connessione (LittleWings - Flash - MIDI), una patch di interfacciamento tra questo e le patch modulari.

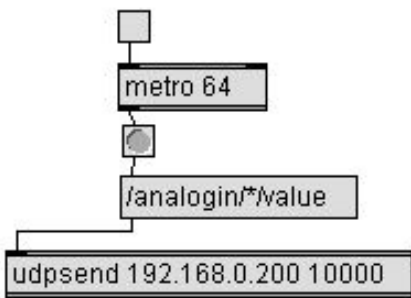
Tale struttura modulare, prevista per cento patch max, ha consentito la sperimentazione di ogni singolo algoritmo in modo indipendente.



Questa è la patch generale in Max-MSP di un mio pezzo per il **BUTTERFLY-system**. In questo capitolo esamineremo solo l'algoritmo "OSC-anyReqRestart", evidenziato nel riquadro in rosso, responsabile della connessione tra i vari software e l'hardware. Su di esso insiste una checkbox per l'avvio del metronomo interno ed un campo numerico per l'impostazione del suo intervallo. Da esso escono i valori parametrici che verranno utilizzati nelle patches P01\_P13 ed altri meccanismi di indirizzamento vario.

In questo cap. il MakeControllerKit sarà siglato "MCK"

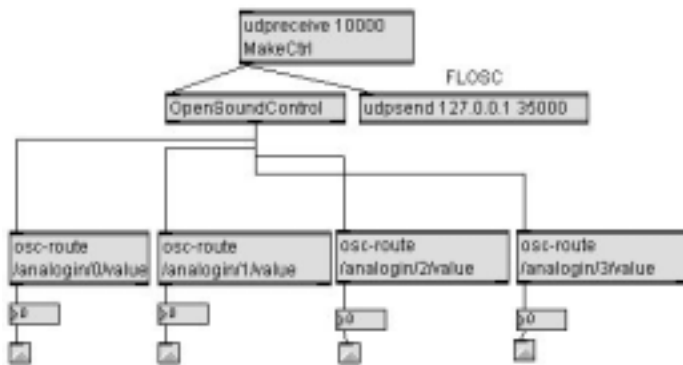




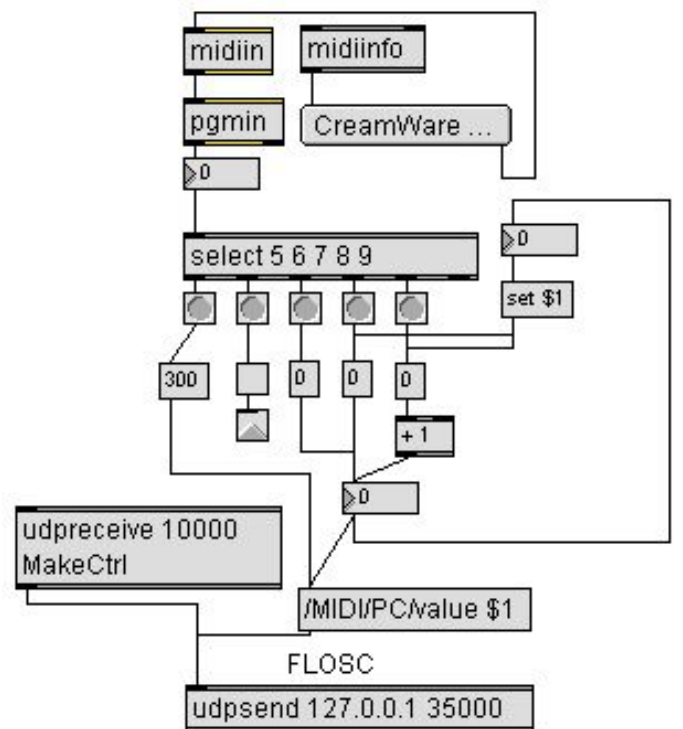
Al suo interno troviamo un metronomo che invia ogni 64 ms. al MCK la richiesta dei valori di tutti (\*) gli input analogici, su indirizzo della rete LAN 192.168.0.200 e porta 10000.

[l'attuale firmware del MCK rende obsoleto ed inutile tale parte, essendo sempre disponibili i valori da essa proveniente, senza polling]

I valori vengono ricevuti da "udpreceive" sulla porta 10000 intestata al MCK



Da questo oggetto vengono inviati in parallelo al traduttore OpenSoundControl e tramite udp send al server java Flosc, su indirizzo locale e porta 35000, Infine i messaggi tradotti da UDP-OSC a Max vengono indirizzati agli otto campi numerici tramite gli osc-route con impostati il filtro dell'indirizzo relativo (analogin/0 ...1... ...7) /value. Nella figura ne sono presenti solo 4 degli 8.



/MIDI/PC/value è l'indirizzo riconosciuto da Flash, il valore proviene dall'algoritmo (\$1=campo numerico collegato)

A sinistra la parte MIDI della patch. "pgmin" fornisce il numero di Program Change in arrivo dal pedale MIDI. La pedaliera utilizza in questo sistema solo il secondo bank di PC (program Change) quindi con valori da 5 a 9. I 5 valori/pulsanti sono così destinati/interpretati:

**5**=invia a Flosc il numero 300\* che servirà come interruttore di visibilità dei parametri in arrivo a Flash nella schermata principale.

**6**=abilita il metronomo di Max

**7**=invia uno "0" per tornare ad una situazione iniziale (audio stop, primo filmato Flash etc.)

**8**=invia il numero di programma attuale, senza l'incremento, per poter riavviare la patch ed il filmato in caso di necessità.

**9**=incrementa di uno il numero di programma attuale, per poter proseguire alla patch ed al filmato successivo.

Il numero 300 è stato scelto per poterlo filtrare con "split 0 299" presente nella finestra iniziale e quindi separarlo dai numeri inferiori che indicheranno sia il numero

di patch di Max che il numero d'ordine dei filmati di Flash.

Inoltre sono presenti oggetti che si incaricano di indirizzare i valori così ottenuti alle singole patches di Max, oggetti "gate" che si occupano di fermare tutte le patches per poterne mandare in linea una alla volta, ed oggetti "gate" che filtrano i messaggi provenienti da Flash indirizzandoli alle patches desiderate etc.

Per ricevere messaggi **OSC** da **Flash**, è necessario un altro modulo **udpreceive** collegato all'ingresso di un altro **OpenSoundControl**. Il primo sarà così settato " *udpreceive 15000 nbufs 20 bufsize 1024*" dove 15000 è il numero di porta sulla quale Max ascolta Flash, 20 è il numero di buffers e 1024 i bit della loro dimensione. Essendo questa possibilità non molto sfruttata, l'ho inserita solo nella patch dove viene utilizzata (P04) e non nel modulo generale sopra descritto.

### 3.4.2 Codice per la connessione OSC in Flash: FLOSC [B.Chun]

Come spiegato in 3.2.4.1 **Flosc** è un server Java scritto da Ben Chun. Per il suo utilizzo in ActionScript di Flash è necessario avviare il server Java.

#### 3.4.2.1 Batch file per avviare il server Java

Prima di tutto si crea un file di testo e lo si salva come batch (\*.bat). Contiene le seguenti quattro righe, solo quelle in grassetto:

<b>ipconfig</b>	<i>opzionale, visualizza gli indirizzi delle attuali connessioni</i>
<b>cd FloscJava</b>	<i>configurabile, sposta il comando nella sottocartella creata, contenente i files java, chiamata da me "FloscJava"</i>
<b>java Gateway 35000 15000</b>	<i>il primo numero(35000) è la porta utilizzata da Flash/Flosc per ricevere i dati provenienti da Max, il secondo )15000 è il numero di porta sulla quale Flash/Flosc invia i propri dati a Max</i>
<b>pause</b>	<i>il programma batch deve restare attivo durante tutta la performance col sistema Butterfly</i>

La cartella FloscJava deve contenere i seguenti files, o più recenti:

.DS_Store	Bytes.class	Bytes.java	Debug.class	Debug.java	flosc.dtd
Gateway.class	Gateway.java	OscMessage.class	OscMessage.java		
OscPacket.class	OscPacket.java	OscServer.class	OscServer.java		
OscSocket.class	OscSocket.java	TcpClient.class	TcpClient.java		
TcpServer.class	TcpServer.java				

Avviato il server java con l'esecuzione del file bat, la finestra dovrebbe indicare questo testo o simile:

```
F:\Butterfly\Flash>ipconfig
Configurazione IP di Windows
Scheda Ethernet Connessione alla rete locale (LAN):
    Stato supporto . . . . . : Supporto disconnesso
Scheda PPP FlyNet Internet Connection:
    Suffisso DNS specifico per connessione:
    Indirizzo IP. . . . . : 151.64.134.91
```

```

Subnet mask . . . . . : 255.255.255.255
Gateway predefinito . . . . . : 151.64.134.91
F:\Butterfly\Flash>cd FloscJava
F:\Butterfly\Flash\FloscJava>java Gateway 35000 15000
Attempting to start OSC / Flash Gateway server
OscServer created...
TcpServer created...

```

Se così non è verificare quanto segue.

### 3.4.2.1 Problemi con il file batch per l'avvio del java server di Flosc

A seguito di alcune installazioni di programma (QuickTime p.es., ma anche mchelper(!)), può saltare la corretta definizione del path di sistema e rendere introvabile la runtime di java e la stessa directory di windows\system32 dove risiedono i comandi DOS.

Se il file di batch non funziona:

- aprire la finestra dos e digitare help

se viene stampato l'insieme dei comandi allora il problema è un altro, inutile continuare a leggere, altrimenti:

è slittata la assegnazione della directory di windows dopo la assegnazione del path, si deve ripristinare il corretto path in chiaro, senza il nome variabile "%systemroot%"

1. aprire il pannello di controllo/sistema/ Proprietà sistema/avanzate/variabili d'ambiente/variabili di sistema/
2. Controllare che la riga sia questa
3. Path: %SystemRoot%\system32;%SystemRoot%; ... etc
4. Eventualmente sia questa e non funziona ancora, sostituire la variabile %SystemRoot% che sta per "c:\windows" la cui assegnazione è slittata dopo e quindi non è riconoscibile (ogni variabile deve essere assegnata prima del suo richiamo) sostituirla con il path in chiaro cioè
5. Path: C:\WINDOWS\SYSTEM32;c:\windows; ...etc.

Non dovrebbe essere necessario riavviare il sistema pr riconoscerla e quindi avviare la runtime di java, l'help di DOS etc.

### 3.4.2.2 Codice Flosc in ActionScript di Flash

Prima di tutto è necessario installare la libreria della MakingThings concernente Flosc!

In grassetto le righe essenziali per la connessione

```

//*****
// CENZWORKS2K 2007-2008 su Flosc di Ben Chun 2003
//*****
import com.makingthings.flosc.*; // import the MakingThings library, installata in precedenza!
Var localAddress = "127.0.0.1"; // your computer's IP address FLASH
var localPort = "15000"; // port dove MAX riceve da FLASH vd. P004 in Max
var remoteAddress = "127.0.0.1"; // MAX local Address
var remotePort = "35000"; //dove MAX udpsend message e flash riceve ed è il server
// java(primo numero)

var flosc:Flosc; // our FLOSC server

onLoad = function( )
{
    Mouse.hide(); //nasconde il puntatore del mouse sullo schermo, comando da
// ripetere in ogni filmato

```



```

fosc = new Fosc( );
fosc.setLocalAddress( localAddress );
fosc.setLocalPort( localPort );
fosc.setRemoteAddress( remoteAddress );
fosc.setRemotePort( remotePort );
fosc.connect( );
fosc.setMessageHandler( onOscMessage ); // set the function onOscMessage to get called
                                         // back when messages arrive
}

//*****per ricevere dati da un swf caricato e mandarli a MAX*****
//La definizione di una funzione deve precedere la sua chiamata!
//definizione: function sendOSC([name, argument, destAddr, destPort]){
function sendOSC(msgNameX, msgArgX, msgNameY, msgArgY){ //definizione della
                                                         //funzione
var oscMX:OscMessage = new OscMessage( );           //definizione variabile e assegnazione
                                                         //nuova
oscMX.address = "/" + msgNameX; //l'indirizzo, non accetta variabili stringa, deve iniziare
//con "/"
oscMX.args[0] = msgArgX; //l'argomento, legato alla X
var oscMY = new OscMessage( ); //secondo diverso messaggio legato alla Y
oscMY.address = "/" + msgNameY;
oscMY.args[0] = msgArgY;
var myOscBundle:Array = new Array( oscMX, oscMY ); //bundle(impacchettamento di
// più messaggi)
fosc.sendBundleToAddress( myOscBundle, remoteAddress, localPort ); //invia ilbundle
}

//un'altra funzione per inviare a Max un messaggio di stop/muting
function sendOSCMute(msgNameMute, msgArgMute ){
var oscMute:OscMessage = new OscMessage( );
oscMute.address = "/" + msgNameMute; //non accetta variabili stringa, deve iniziare con "/"
oscMute.args[0] = msgArgMute;
fosc.sendMessageToAddress( oscMute, remoteAddress, localPort );
}

// un'altra funzione per inviare a Max un messaggio di bang
function sendOSCAmpBang(msgNameAmp, msgArgAmp, msgNameBang, msgArgBang){
var oscMA:OscMessage = new OscMessage( );
oscMA.address = "/" + msgNameAmp;
oscMA.args[0] = msgArgAmp;
var oscMB = new OscMessage( );
oscMB.address = "/" + msgNameBang;
oscMB.args[0] = msgArgBang;
var myOscBundle:Array = new Array( oscMA, oscMB );
fosc.sendBundleToAddress( myOscBundle, remoteAddress, localPort );
}

```

Quello che stiamo illustrando è il codice Action Script contenuto nel primo frame del filmato principale. Esso si occupa della connessione e del caricamento dei filmati composti. Inoltre cura l'interfacciamento in entrata ed uscita con Max. Per poter condividere con i filmati caricati i nomi i messaggi si adopera questa function, chiamata da me *sendOSCInterface()* che a sua volta riceve da *sendOSC()*

```

//Questa function INDISPENSABILE serve per ricevere dal SWF caricato i dati e
//utilizzarli nel principale (sendOSC). Non bypassabile!
Function sendOSCInterface(){
sendOSC(msgNameX, msgArgX, msgNameY, msgArgY );
}

```

```

//simile per altra routine
function sendOSCInterMute(){

```

```

    sendOSCMute(msgNameMute, msgArgMute );
}

```

//simile per altra routine

```

function sendOSCIInterAmpBang(){
    sendOSCAmpBang(msgNameAmp, msgArgAmp, msgNameBang, msgArgBang);
}

```

//Qui inizia la sezione dedicate alla ricezione dei messaggi provenienti da Max

```

onOscMessage = function( address, argument )

```

```

{
    if( address == "/MIDI/PC/value" )    //se l'indirizzo coincide, ...
    {
        PgChange = argument; // assegna alla variabile (anche nel campo testo) il numero di
                               //ProgramChange, il valore del messaggio ricevuto.

        If(PgChange == "0") // scarica l'ultimo movie → sfondo nero iniziale
        {
            unloadMovieNum(1);
        }

        if(PgChange == "1")
        {
            unloadMovieNum(1);
            loadMovieNum("files/coperchio6.swf", 1); //il nome del primo filmato da caricare
        }

        if(PgChange == "2")
        {
            unloadMovieNum(1);
            loadMovieNum("files/Informel09.swf", 1); //il nome del secondo filmato da caricare
        }

        ... omissis.....
        if(PgChange == "3") //... e così via per tutti i filmati seguenti
        {
            unloadMovieNum(1);
            loadMovieNum("files/End.swf", 1);
        }
    }
}

```

Qui invece nel caso specifico del ricevimento del numero 300, dovuto alla pressione sul pedale MIDI del secondo pulsante (ProgramChange = 6). Ha effetto solo se nel filmato precedente esistono campi testo con proprietà "dynamic text" e variabile assegnata

```

if(PgChange == "300") //nasconde o mostra i campi numerici an e MIDI
{
    MIDIchangeTxt._visible = !(getProperty (MIDIchangeTxt,_visible));
    MIDIvalue._visible = !(getProperty (MIDIvalue,_visible));
    lined._visible = !(getProperty (lined,_visible));
    an0Txt._visible = !(getProperty (an0Txt,_visible));
    an1Txt._visible = !(getProperty (an1Txt,_visible));
    an2Txt._visible = !(getProperty (an2Txt,_visible));
    an3Txt._visible = !(getProperty (an3Txt,_visible));
    an4Txt._visible = !(getProperty (an4Txt,_visible));
    an5Txt._visible = !(getProperty (an5Txt,_visible));
    an6Txt._visible = !(getProperty (an6Txt,_visible));
    an7Txt._visible = !(getProperty (an7Txt,_visible));
    trimp0._visible = !(getProperty (trimp0,_visible));
    trimp1._visible = !(getProperty (trimp1,_visible));
}

```

```

    trimp2._visible = !(getProperty (trimp2,_visible));
    trimp3._visible = !(getProperty (trimp3,_visible));
    trimp4._visible = !(getProperty (trimp4,_visible));
    trimp5._visible = !(getProperty (trimp5,_visible));
    trimp6._visible = !(getProperty (trimp6,_visible));
    trimp7._visible = !(getProperty (trimp7,_visible));
}
}

```

Questa è la sezione, sempre annidata in quella dei messaggi in ricezione, che si occupa di assegnare agli argomenti dei messaggi, cioè ai valori parametrici, una variabile globale, riconosciuta quindi da tutti i filmati in tutti i livelli (layers)

```

if( address == "/analogin/0/value" ) //primo sensore
{
    trimpot0 = argument;
    _global.address0 = address;
    _global.trimpot0 = trimpot0;
}
if( address == "/analogin/1/value" )
{
    trimpot1 = argument;
    _global.address1 = address;
    _global.trimpot1 = trimpot1;
}
if( address == "/analogin/2/value" )

```

... e così via fino all'ottavo sensore...

```

if( address == "/analogin/7/value" )
{
    trimpot7 = argument;
    _global.address7 = address;
    _global.trimpot7 = trimpot7;
}
}

```

Il filmato swf di cui abbiamo descritto il codice ActionScript, deve contenere anche un semplice comando: **stop()**; che ha la funzione di far sì che lo stesso filmato composto da un solo frame, venga ripetuto incessantemente dal player, al frame rate impostato, nel mio caso 24 fps. Ho provato "risoluzioni" maggiori e minori ma la ritengo la scelta ottimale.

La definizione delle dimensioni invece della finestra principale, 800x600, deve essere la stessa per tutti i filmati caricati, che ricordiamo andranno semplicemente a sovrapporsi al presente, tant'è che se restano visibili i campi di testo, utilizzati in fase di setup per la calibrazione di **LittleWings**, essi saranno sempre visibili, magari in trasparenza o in parte coperti. Dimensioni maggiori della finestra base possono sovraccaricare il player.

E' importante anche che la risoluzione del filmato coincida con la risoluzione della scheda video, ovvero con le sue impostazioni, altrimenti si avranno ritardi per le successive riduzioni operate dalla scheda (che ricordiamo nativa 1640x1050).

A questo punto il sistema è pronto.

Serve un'idea per la composizione!

## **4 – TECNICA**

## 4.1 – Tecnica e tecnologie

In campo grafico e musicale la tecnica è rimasta quella di sempre, a cambiare sono state principalmente le tecnologie. E' un'affermazione che può essere un po' manichea, come la maggior parte delle mie affermazioni, ma anche se falsa in assoluto ha in un'accezione relativa dei fondamenti ampiamente condivisibili.

Se si ha la fortuna di guardare i primi cortometraggi della storia del cinema, e poi si studiano gli ultimi film di fantascienza, si potrà intravedere una serie di basilari tecniche legate alle immagini in movimento, al cinema in quanto tale, che sono le stesse.

Prendiamo p.es. la tecnica del Morphing, quella che consente una graduale trasformazione di una forma in un'altra, usatissima nei film d'horror e fantasy, ebbene "tecnicamente" è la stessa sia in Metropolis del 1926 (creazione del robot somigliante alla protagonista) che in "Abyss" del 1995 (comparsa dell'alieno acquatico), il primo film ad utilizzare la tecnologia informatica di rendering 3D. La diversità in queste realizzazioni è solo di tipo tecnologico, con interpolazione manuale dei fotogrammi nel primo caso e con la tecnologia digitale nel secondo.

A questo genere appartengono tutte quelle tecniche che giocano sulla trasparenza delle immagini e lo sfruttamento delle limitazioni percettive dell'occhio umano, con iniziare proprio da quella che ha consentito la nascita del cinema e della televisione, l'impossibilità di isolare immagini, distinguerle percettivamente, che si succedono ad una velocità superiore ai 30 fps circa (27-25-24).

In musica il barocco "trillo" ha una funzione simile: la fusione percettiva data dalla rapida successione di altezze vicine diverse.

Nelle composizioni realizzate per il sistema Butterfly si fa ampio uso di queste consolidate tecniche, sfruttando le nuove possibilità di interazione in real-time.

*A differenza di quanto accade nella vita,  
dove spesso le cose non sono quello che sembrano,  
nell'arte le cose sono solo ed esattamente quello che sembrano.*

Questa autocitazione, in realtà, dice tutt'altra cosa...o no?

## 4.2 – La forma e la percezione

Il compositore crea un pezzo mediante una forma complessiva che ne indirizzi il dislocamento del materiale. Detta forma non deve essere percepita in quanto tale, ne durante ne alla fine del pezzo: solo un'analisi musicale approfondita potrà rilevarla.

L'ascolto si muove su di essa ma non la vede ne intravede, come una strada di cui scorge temporanei e locali punti di riferimento senza per questo averne una percezione complessiva.

La forma è quindi un percorso costruito dal compositore, ad suo uso e consumo, che ha la funzione di guidare l'ascoltatore in un viaggio con determinate finalità. Il percorso deve essere funzionale al viaggio e dare in ogni momento quelle indicazioni di posizionamento all'interno di esso, cioè indurre nell'ascoltatore la conoscenza di essere verso l'inizio, metà o la fine del viaggio. E anche verso quali dinamiche tensive ci si sta portando.

Per far questo la forma utilizza linee vettoriali, la cui parziale prevedibilità consente all'ascoltatore di formulare ipotesi sulla loro durata e indirettamente sul loro attuale posizionamento.

*P. es.* un *diminuendo*, inteso in senso lato come caduta di intensità o diminuzione di densità o scemare di energie varie, induce una sensazione di fine (0 = fine) viceversa un *crescendo* prospetta un possibile culmine ad una certa distanza temporale.

Ogni *crescendo* ha un fisiologico limite, insuperabile a meno che non si distribuisca la generale sensazione di crescendo su molteplici livelli. La sua percezione, come molti fenomeni sensoriali, è di tipo logaritmico, cioè ha bisogno di incrementi sempre crescenti per produrre una sensazione uniforme di crescendo naturale. Se il crescendo è lineare, come potrebbe essere l'allagamento di una vasca da bagno da parte di un rubinetto dal flusso costante, la sensazione non viene percepita come crescendo, ma al contrario ci si accorge ad un certo punto ... di avere l'acqua alla gola, presi cioè di sorpresa dal raggiungimento dell'apice, che in musica si avvicina alla soglia del dolore.

Paradossalmente il crescendo dinamico è il più limitato percettivamente perché per essere percepito ha bisogno di tempi molto ristretti di variazione, ed essendo la materia sonora generalmente già involupata in ampiezze ognuna con un proprio picco, sarà presto saturata la gamma dinamica dovendo rapidamente muoversi, crescere, non tra ampiezze medie, non significanti per l'orecchio, ma tra picchi relativi per produrre nell'ascoltatore la sufficiente sensazione di crescendo.

Un modo per ottenere sensazioni sufficientemente lunghe di crescendo è quello di dosare la quantità di informazioni nel tempo. A parità di intensità generale il comparire di una maggiore densità di eventi acustici provvede a fornire nell'ascoltatore una sensazione di "aumento" "accrescimento" energetico che lo continua a condurre "avanti" "più sopra" nel percorso formale. Per questo motivo, dopo un eventuale diminuendo transitorio o addirittura un silenzio, si può recuperare "quota" e riallinearsi alla curva ascendente solo a patto di introdurre nuovi elementi energetici che ne rivitalizzino l'apparente decadimento.

Se unitamente al crescendo gli eventi sonori tendono a diradarsi si ha una percezione di aumento dimensionale, viceversa unitamente ad un accelerando o infittirsi di eventi la sensazione si avvicina a quella fisica legata alla gravitazione, caduta massi, pioggia, frana.

Nell'interpretazione strumentale è buona norma scegliere sempre una delle due

soluzioni agogiche, al fine di aumentarne il realismo e la drammaticità. In mancanza di una seppur minima variazione di tempo qualsiasi crescendo o diminuendo appare inevitabilmente scolastico, letto, suonato, ma non vissuto.

Molto efficaci compositivamente anche gli eccessi opposti, una grande massa di suoni, densissima in un *pp* estremo, come anche un solo suono con la massima energia, per la loro innaturalità colpiscono molto l'immaginario dell'ascoltatore, suggerendo a seconda del suo umore disperate/impotenti moltitudini (minacciose e lontane presenze) o disperate/eroiche solitudini (vincenti e maligne presenze).

La prima sezione nel pp di "O fortuna" di Orff, è un tipico esempio di una moltitudine oppressa dall'infausto fato (ma che medita una rivincita). "All'alba vincerò" della Turandot pucciniana un esempio del secondo tipo.

## 4.2.1 – Vincoli formali nel Butterfly-system

Ogni brano composto per, ed eseguito col sistema **Butterfly**, deve prevedere una serie di sezioni, scene, audio/grafiche che si succedono nel tempo. Ciò è necessario per la gestione modulare del brano, soprattutto per la parte grafica, per non rendere estremamente pesanti il filmato unico ed il relativo codice ActionScript. Il caricamento di filmati Flash e abilitazioni audio di patches Max non comporta alcun ritardo avvertibile né interruzione audio o aberrazioni grafiche di sorta.

Parleremo di **Patch** (patches) in senso di logaritmo audio indipendente di Max " *Named\_Patch.pat* " e **filmato** o meglio **movie** come " *Named\_Movie.swf* " indipendente eseguito dal lettore Flash<sup>27</sup>.

Il sistema nella sua parte software nasce con la predisposizione di 100 algoritmi di tipo modulare per la sintesi audio attivabili in sequenza, quindi singolarmente caricati e scaricati nel e dal kernel.

Così nel codice principale che si occupa del carico/scarico filmati, è sufficiente aggiungere tre righe per aggiungere ogni ulteriore filmato.

La costruzione modulare e indipendente tra la parte audio e grafica, rende possibili l'utilizzo di diverse patches audio sullo stesso filmato o viceversa, anche se per evitare possibili confusioni tra variabili è sempre preferibile scorrere parallelamente nell'avanzamento, e se occorre, salvare con nome diverso il movie o la patch copiata.

Per quanto riguarda la tipologia tecnica delle sezioni (patch più movie) ognuna di esse potrà essere di tre tipi fondamentali:

1. Definita completamente – non accetta parametri esterni in real-time
2. Definita parzialmente – accetta parametri esterni
3. Completamente controllata parametricamente dall'esterno

1 - Nello specifico si intende definita completamente quella sezione del brano nella quale sia la parte grafica che quella relativa alla sintesi audio sono da intendersi come players di un filmato e di un algoritmo audio chiusi parametricamente. La loro struttura potrà prevedere moduli stocastici ma non accetterà variazioni esterne di alcuna delle proprie variabili (o più semplicemente non ci saranno variabili ma solo costanti). In campo grafico si caricherà nel momento esecutivo un filmato SWF con una durata fissa, cioè di un numero predeterminato di frames o comunque con un frame finale nel quale

---

<sup>27</sup> La possibilità di utilizzare le Scene propria di Flash è sconsigliata dagli stessi autori del software per le possibili incongruenze tra le chiamate di movie clips e moduli

è presente il codice **stop()**; Dal lato audio si determinerà l'apertura di una specifica sezione dell'unico algoritmo. In questa tipologia viene meno completamente l'utilizzo di LittleWings, e il sistema diventa, temporaneamente un player multimediale.

2 – Si intende definita parzialmente, una sezione nella quale coesistono letture di file audio o di algoritmi direzionati, irreversibili ma ripetibili (looped), in correlazione con filmati di durata predefinita, direzionati e ripetibili, il tutto insieme ad algoritmi che modifichino i parametri degli stessi filmati e algoritmi grazie a LittleWings. E' di certo la soluzione privilegiata, quella che consente una base multimediale sulla quale operare a piacimento in real-time.

3 – Si intende, infine, Completamente controllata parametricamente dall'esterno, quella sezione le cui patches e i cui filmati iniziano e restano indefinitivamente in silenzio e al buio fintantoché non si agisce su LittleWings. E' il tipo più impegnativo dal punto di vista compositivo ed interpretativo, ma che consente meglio degli altri di garantire una correlazione immediata per il pubblico tra l'azione umana sull'interfaccia ed il sistema.

Questa suddivisione formale non è legata al singolo patch-filmato, nel senso che all'interno di esso possono trovare temporalmente posto ognuna delle esposte tipologie di sezione. P.es. un filmato può iniziare con una sequenza automatica generata randomicamente e, arrivato ad un determinato frame, restare in attesa dell'intervento parametrico operato su LittleWings. Quello che resta precluso al sistema è il settaggio in tempo reale del codice o della parametrica locale, non potendo in fase di esecuzione su un solo computer, visualizzare altro che il filmato Flash attuale ed attivo, e non essendoci altri controlli al di fuori di LittleWings e del pedale MIDI. Ovviamente la catena audio resta indipendente dal sistema nel quale è inserita, consentendo aggiustamenti mixer ed inserimento effetti digitali.



## 4.3 – Tecniche generali audio

### 4.3.1 Riscaldamento valori digitali

I dati inviati da LittleWings hanno tutti un range di 10 bit e vengono interpretati da OSC di Max con questi numeri interi 0\_1023. Per poterli utilizzare come parametri negli algoritmi di Max è necessario prima di tutto riscalarli al range desiderato.

Si usa “**scale 0 1023** *valore\_min valore\_max exp*” dove valore min e max sono i limiti del range desiderato, sia int che float, sia invertiti (da max a min) e dove exp rappresenta il 5° numero opzionale per l'utilizzo di un fattore esponenziale maggiore di 1 nel riscaldamento. Es. “**scale 0 1023 0. 1. 1.05**” riscalda i dati nel range di ampiezze tipiche 0\_1 con una minima curvatura esponenziale.

### 4.3.2 Controllo inizio e fine dati in arrivo

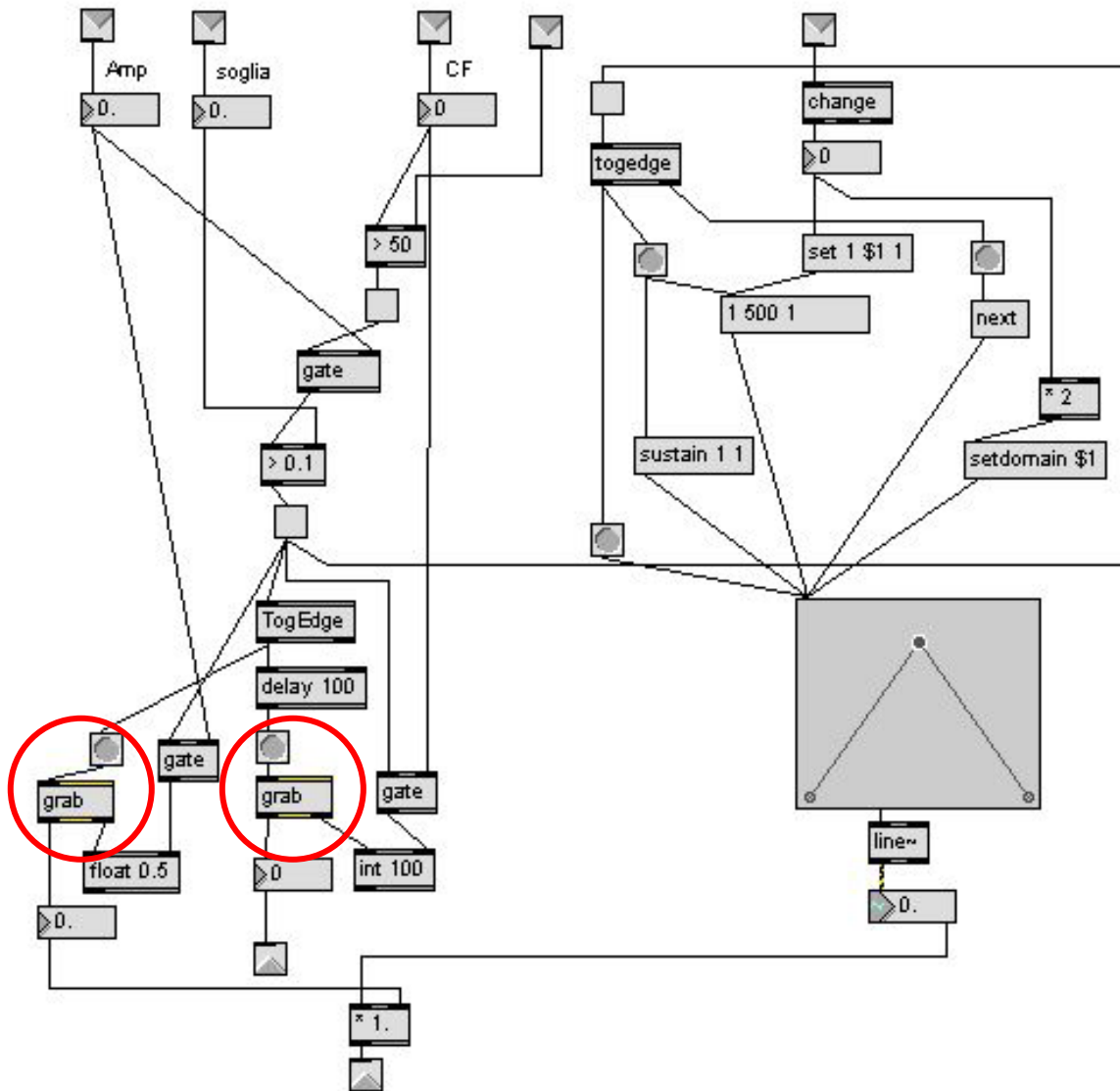
Uno dei problemi principali dell'utilizzo di **LittleWings** ed in genere di qualsiasi sistema sensoristico è rappresentato dai dati inviati al momento del contatto del dito sul sensore e al momento del suo abbandono. Qualsiasi possa essere il tipo di parametri cui si intende inoltrare questi dati essi devono prima essere controllati e quindi in seguito correlati. Per spiegare meglio la problematica ipotizziamo l'uso dei dati provenienti dal sensore lineare come controllo parametrico della frequenza di un suono ed il sottostante sensore di pressione come controllo dell'ampiezza dello stesso suono. Non essendoci possibilità di garantire che detti sensori vengano sollecitati nello stesso identico istante ed anche in questo caso, udremo un tick determinato dal repentino scarto digitale dallo 0 di riposo al valore di ampiezza iniziale e la frequenza del suono balzare da 0 al valore impostato nel riscaldamento. Tutto ciò continuerà con estrema incertezza alla frequenza di campionamento impostata col metronomo (64 ms.) rendendo immediatamente rumoroso ed instabile il suono stesso. Per finire con un inevitabile click determinato al momento del rilascio del dito dal doppio sensore dall'andare a 0 la frequenza o peggio prima l'ampiezza.

Per evitare questi ingestibili flussi di dati si sono utilizzate due risorse di Max, il modulo **rampsmooth~** ed il modulo **grab**.

Con il primo si introduce una continua interpolazione tra le variazioni di dati in arrivo, che in questo modo sono addolciti e con il secondo si cattura (**grab**) l'ampiezza o meglio la frequenza del momento “precedente” al rilascio dei sensori consentendo all'ampiezza di raggiungere lo zero prima che lo faccia la frequenza. A tale scopo sarebbe bastato anche un semplice delay, ma con **grab** abbiamo la possibilità di immagazzinare l'ultimo dato utile ed eventualmente riproporlo come dato iniziale della nuova interpolazione, senza che questa riparta sempre dal valore minimo (0).

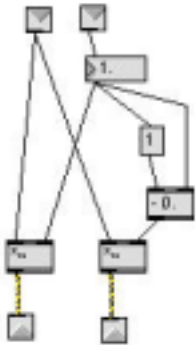
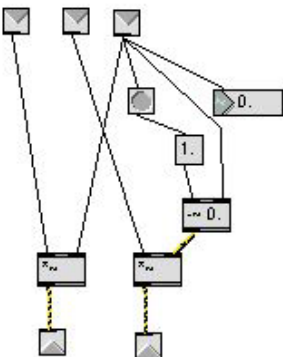
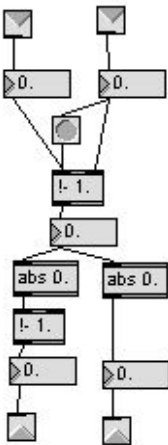
1. **rampsmooth~ 22050 22050** , indica un interpolatore audio tra due campioni del quale si può indicare con il primo argomento il numero di campioni con i quali si giungerà al nuovo valore partendo dal precedente e con il secondo i campioni da utilizzare per l'interpolazione degli ultimi due dati in ingresso. Con valori di 22050 si ottengono fade in e out di 500 ms (1s = FC), sufficienti per attacchi e rilasci morbidi dei suoni. Per calcolare in real time i numeri di campioni si può utilizzare il convertitore **mstosamps~** , mentre per impostare fade in diversi il messaggio **rampup** seguito dal numero di campioni. Similmente **rampdown** *ncampioni* per il fade out.

2. **grab** invece è stato utilizzato in un algoritmo più complesso, creato per il coordinamento dei dati dei sensori lineari e di pressione collegati fisicamente.



Questo è l'algoritmo che correla un involuppo del tipo ASR sulla destra e le variazioni di dati provenienti dai sensori citati. In breve tramite esso possono essere catturati l'ampiezza o la frequenza o entrambi di un momento desiderato (dipende dal delay impostato) e moltiplicarlo per un fadein, sustain, release reimpostato o modificabile da ulteriori parametri in arrivo.

### 4.3.3 Bilanciamento semplice e condizionato

	<p>Per un semplice bilanciamento in ampiezza dei canali controllato con un float di un segnale mono è sufficiente sottrarre ad 1 il valore in ingresso.</p> <p><b>PanMono:</b> l'ingresso audio mono viene indirizzato al canale sinistro e moltiplicato per un fattore 0_1 float, sottratto ad 1 ed inviato al canale destro.</p>
	<p>La versione stereo SR del precedente, notare che l'inlet destro è <i>signal</i>, i primi due sono i canali L e R. Il segnale in ingresso (3° inlet) che controlla il bilanciamento proviene da un sensore, è il risultato di un'operazione o proviene da un filmato Flash (p.es. la coordinata x di un oggetto in movimento randomico)</p>
	<p>Per ottenere un oggetto simile al primo ma reversibile, cioè che accetta un secondo e indipendente controllo del bilanciamento, bisogna utilizzare qualcosa di simile ... o questo algoritmo. I due inlets sono due indipendenti controlli del bilanciamento che interagiscono tra essi fornendo la risultante come controllo del bilanciamento di due canali audio. Questo algoritmo nasce dall'esigenza di controllare una rotazione di un oggetto su un asse fisso e contemporaneamente sulla rotazione dello stesso asse. (pezzo 2 P08, giroscopio)</p>

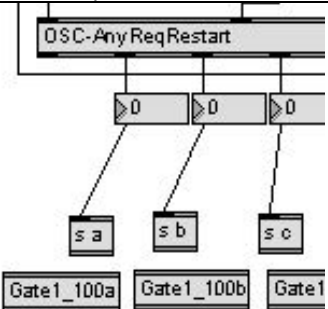
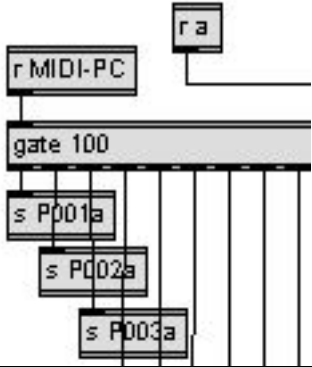
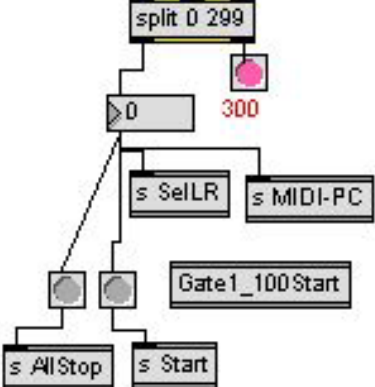
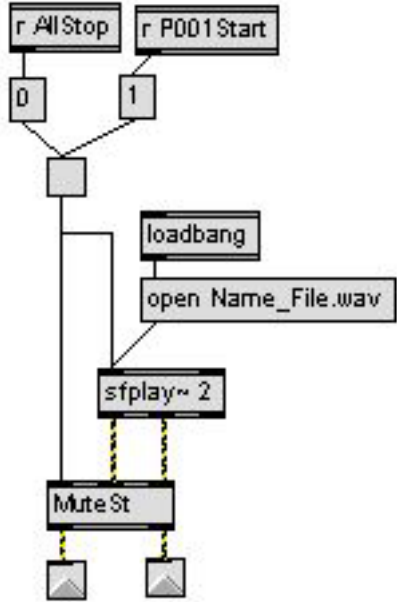
### 4.3.4 Select patch, start/stop, player wave, mute

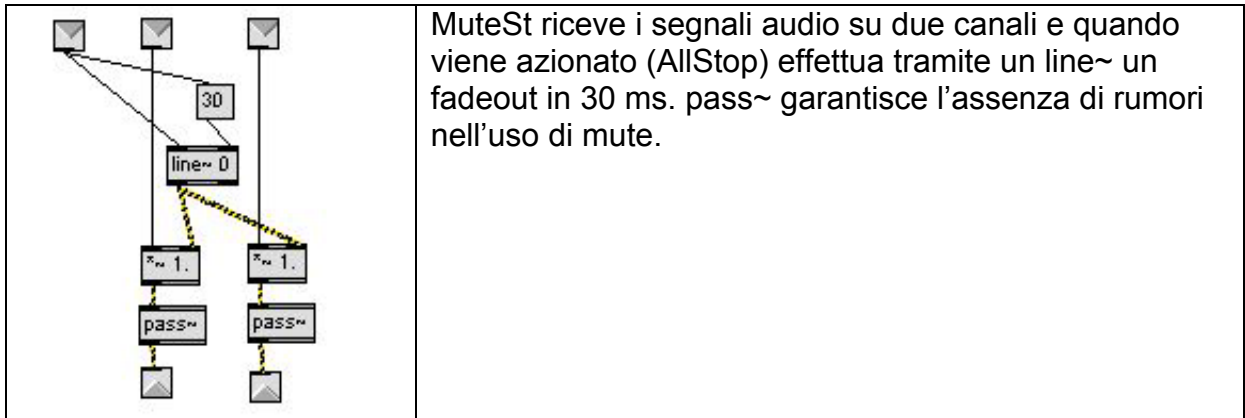
La gestione delle patches, cioè il loro avvicendamento nella ricezione dei dati LittleWings e nel loro singolo ed esclusivo collegamento al DAC, ha esatto una laboriosa preparazione, soprattutto manuale, nel dover gestire 8 flussi di dati parametrici su 100 patches e garantirne la silenziosità e non sovrapposibilità.

Siccome gli stessi problemi si sono presentati anche per la gestione dei *samples* audio (file wave), vengono trattati insieme in questo capitolo.

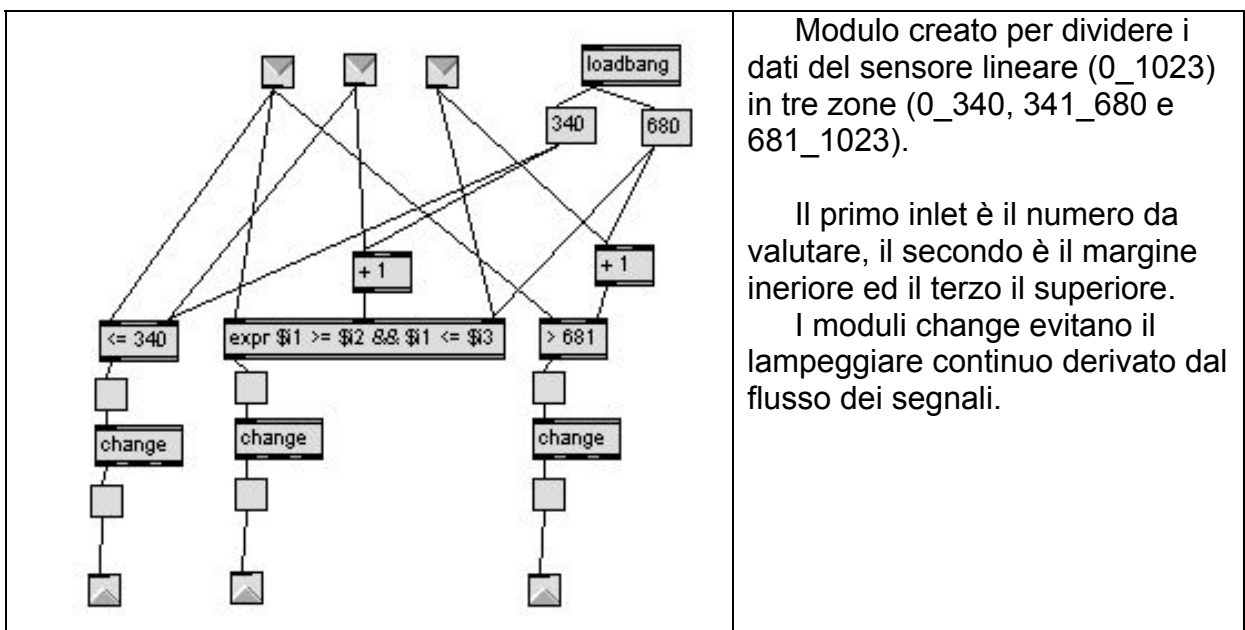
Il modulo **OSCanyReqRestart** esposto in 3.4.1, per la gestione dati OSC, presenta in uscita gli otto valori provenienti dai sensori (nominati a, b, c, d, e, f, g, h). Essi sono mandati non direttamente ad ogni singola patch ma ad otto cancelli (gate) contenenti ognuno 100 collegamenti alle patches. Questo per evitare sovrapposizioni e confusioni. Ogni **gate1\_100°**, **gate1\_100b** etc. è gestito dal ricevimento del Program Change MIDI, o meglio dal meccanismo di incremento della scena

(patch+filmato). Quindi ogni gate riceve il valore di un sensore (**Gate1\_100°** il sensore a) e lo indirizza alla singola patch selezionata dal ProgramChange.

	<p><b>s a</b> = send a, il valore del primo sensore. <b>Gate1_100°</b> è visto al suo interno nella figura sottostante (un particolare, vista la dimensione 100)</p>
	<p><b>R MIDI-PC</b> è il ricevitore del programChange, che in <b>Gate1_100(a_h)</b> controlla l'uscita, cioè seleziona la patch destinataria del parametro a_h. A destra in alto spunta il ricevitore del parametro a che è collegato al inlet destro del gate, fuori campo. Oltre a questi 8 Gate "centenari", ve ne è uno che indirizza lo start patch il <b>Gate1_100Start</b>, qui non raffigurato perché simile ai precedenti.</p>
	<p>Il Program change viene filtrato da <b>split 0 299</b> come sappiamo ed inviato ad un oggetto select (vd. dopo) ai ricevitori MIDI-PC di prima e ad ogni variazione invia in successione un messaggio di stop a tutte le patches ed uno di start alla patch selezionata dal <b>Gate1_100Start</b>. In ogni singola patch esiste un ricevitore del bang <b>AllStop</b> (inviato quindi sempre a tutte) ed uno per lo <b>Start</b>, inviato singolarmente nella successione dell'avanzamento scena. Nella figura seguente vediamo una sintesi di questo meccanismo.</p>
	<p>Interno di ogni singola patch (in questo caso la P001). Il bang "AllStop" invia uno "0" alla checkbox che interrompe l'eventuale lettura del file wave e rende muta l'intera patch, scollegandola virtualmente dal DAC. Prima del MuteSt diciamo l'essenziale per sfplay. All'avvio (loadbang) viene caricato in memoria un file (<i>Name_file</i>). <u>E' necessario che la directory dove risiede tale file sia espressamente indicata nelle opzioni di ricerca file di Max (FilePreferences)!!</u> L'avvio della lettura di tale file è comandato dal ricevitore <b>Pnnn Start</b>. Per garantire l'assenza di click ad avanzamento patch, c'è il modulo basato sul mute di Max: MuteSt.</p>



### 4.3.5 Operatore logico a due margini (3RngTog)



### 4.3.6 Alcune specifiche su moduli di Max

#### 4.3.6.1 Line, Line~, pak, pack, set

Per utilizzare un involuppo di soli tre parametri (init, end e dur) è sufficiente il **LINE** non audio. Altrimenti **line~** che consente di avere :

- init, P1 Dur1 P2 Dur2 (...Pn Durn) con numeri dispari di parametri e virgola solo dopo il primo.
- Accetta input int e float e variabili. Esce con un signal, per convertirlo in float usare **number~** (campo numerico float [.] con "onda verde") e prendere l'output destro. Impostare nel Get-Info il rate di 20ms (minimo) altrimenti i valori in uscita saranno dati a 250ms di periodo.

Per variare i valori interni se ne serve uno solo usare un message box:

Es. **Set** (init) \, (P1) \$1 (P2) \$1 (P3) \$1

dove p1\_3 sono i parametri di esempio. Quindi collegarne l'uscita ad un altro message box sul quale insisterà un bang di invio differito, altrimenti nel caso che

l'involuppo possa essere "sparato" ogni qualvolta arriva un valore lo stesso message box senza **set**.

Per utilizzare più variabili (\$1, \$2, ...\$n) bisogna impacchettarli prima di inviarli al message box **SET**.

Usare **pak** (2 def, 3 ... n) tenendo conto che gli input sono in ordine da sinistra a destra (quindi \$1, \$2 etc). Oppure **pack** se si intende differire il settaggio dall'invio effettivo.

### 4.3.6.2 function

L'help dell'involuppo grafico "**function**" non è chiaro riguardo il settaggio di diversi punti. Un msg "0 1" crea un nuovo punto con coordinate x=0 e y=1. Mentre un msg "0 0 1" modifica le coordinate del primo punto che deve pre-esistere assegnando a x 0 e a y 1. Per modificare i punti 2 3 4 il primo dei tre numeri, il numero d'ordine dei punti, sarà rispettivamente 1 2 3 (-1, numerazione 0 based) ma ogni tripletta di valori (numero d'ordine -1, x, y) deve essere separata da virgole oppure scritta in msg separati p.es :

msg = "0 0 1, 1 20 0, 2 30 1" a tre punti pre-esistenti vengono assegnate le segg. Coordinate : primo punto x=0 y = 1, secondo x=20 y = 0, terzo x = 30 y = 1.  
Oppure tre msg separati : "0 0 1" "1 20 0" "2 30 1"

#### 4.3.6.2.1 setdomain e setrange

Per riusare una stessa funzione con valori uguali per involuppi diversi conviene usare **setdomain** per il range delle x e **setrange** per quello delle y.

Es. Involuppo di 4 punti modificabili con 4 coppie di campi numerici:

Ogni coppia di numeri sarà formata da un integrale ed un float, perché il primo numero modificherà la x, presumibilmente funzione del tempo in ms, e il secondo float per funzioni d'ampiezza (0~1). Detti campi numerici insisteranno su un oggetto "**pak** 0 0." Con i due zeri non è puntato per i due input. L'output si connetterà ad un msg "0 \$1 \$2" dove lo 0 è la costante che indica il numero d'ordine del punto da modificare e i due \$ le variabili.

Quindi per gli altri 3 punti si duplicheranno gli oggetti creati e si modificheranno solo i msg finali con "1 \$1 \$2" "2 \$1 \$2" "3 \$1 \$2" tutti e 4 verranno infine connessi all'input unico sinistro della function.

## 4.4 – Tecniche generali grafiche

### 4.4.1 Ancora su FLOSC

Flash non può connettersi e comandare altri programmi, per questioni di sicurezza, per farlo si deve avviare un programma separato che consiste in un Server Java. [leggere in fondo al cap. altre importantissime notizie sulla sicurezza e l'autorizzazione alla connessione fornita solo on-line!]

Xml socket è una connessione server-client che possono connettersi con server con indirizzo ip e numero porta superiore a 1024. Utilizzata localmente (indirizzo ip 127.0.0.1) con server java che deve avere impostate le porte in ordine di ricezione e trasmissione (35000 15000). In realtà la conoscenza dei due Flosc e della terminologia di Max e della MakeThings, provoca una notevole confusione tra indirizzo/porta locale e remote

- **localAddress** è l'indirizzo IP del PC dove FLOSC e Flash si avviano.

Esso può essere qualsiasi ma è bene sia **192.168.0.201** (desktop) e **192.168.0.202** (laptop) impostando anche i relativi server dns preferiti (identici).

- **localPort** è la porta dove il computer ascolta e riceve i dati - default **10000**.

• **remoteAddress** l'indirizzo di Make Controller Kit con il quale comunica MAX - default **192.168.0.200**. Invece in Flash, esso coincide con l'indirizzo locale, dovendo inviare da Flash a Max sullo stesso computer e quindi **127.0.0.1**.

• **remotePort** è la porta sulla quale il Make Controller ascolta I messaggi in arrivo - il default è **10000**. Invece in flash per porta remota si intende quella creata da MAX per inviare a Flash I propri dati.

**Flosc e Max** creano entrambi dei server esclusivi. P. es. non si può creare su Max un UDPsend 20000 (server UDP con porta 20000) con la stessa porta utilizzata dal server JAVA 35000 20000. Invece è possibile creare più client con la stessa porta o diversa (java 35000 15000, Max UDPreceive 15000 o 35000). In caso di conflitto è il server Java a trasmettere la bella notizia e a disconnettersi.

Ricapitolando, al di là del nome *local* o *remote*, relativo più che altro al task visualizzato (Flash o Max), si possono avere più server con porte diverse e più client anche con la stessa porta. Dove Max riceve "UDPreceive" abbiamo un server, un altro server con relativa porta di ricezione è quello Java *receiveportnumber* *sendtportnumber* (35000 15000) e Flash:

```
localAddress = "127.0.0.1"; // indirizzo del PC
localPort = "15000"; // port dove MAX riceve da FLASH (UDPreceive)
remoteAddress = "127.0.0.1"; // MAX local Address
remotePort = "35000"; //dove MAX udpSEND message e flash riceve ed è il
server java (primo numero)
flosc:Flosc; // il nostro FLOSC server
```

La Make, una volta aperta la connessione, resta disponibile all'invio/ricezione di innumerevoli messaggi senza bisogno di richiesta da parte del client.

Inoltre alla prima connessione su rete lan di Flosc e del player di Flash, la

Macromedia (oggi Adobe) ha pensato bene, sempre per questioni di sicurezza, di impedire la connessione degli oggetti XML senza aver prima dato il proprio formale e macchinoso assenso on-line !

Infatti è necessario connettersi al sito deputato, richiesto dal player e tramite un interfaccia java nel browser, indicare i files o le cartelle cui garantire l'accesso. Quindi è determinante seguire tale procedura la prima volta che ci si connette via LAN col Player, anche se non si è connessi in remoto (internet). Una volta terminata, scegliendo la cartella sul proprio HD che conterrà tutti i filmati che fanno uso di connessione ed averla impostata, si potrà lavorare in santa pace anche disconnessi da internet. Fino alla prossima formattazione dell'HD!

## 4.4.2 Layers

Il programma Flash prevede una timeline verticale ed una gerarchia orizzontale di layers, fogli trasparenti sui quali disegnare gli oggetti e gli sfondi. Il layer superiore è da ritenersi l'ultimo sovrapposto, cioè quello sulla pila di layers e pertanto coprirà, secondo le proprietà e dimensione degli oggetti ivi disegnati, tutti gli altri layers.

Viceversa, se un filmato ne carica un altro, quest'ultimo si sovrapporrà al precedente, senza peraltro sostituirsi.

Il Butterfly-system oltre alla sovrapposizione di layers all'interno dei filmati, sfrutta quest'ultima capacità per conservare la struttura di comunicazione OSC nel filmato principale (sfondo nero con campi numerici nascondibili al ricevimento on/off di un messaggio OSC) e caricare uno alla volta tutti i filmati delle singole scene, ogni volta scaricando il precedente.

## 4.4.3 Nidificazione di filmati e simboli

In Flash, una volta creato un filmato composto da una serie di frames è possibile utilizzarlo come un "oggetto-film" all'interno di un altro filmato, cioè un oggetto che pur continuando ad essere ripetuto per tutta la durata dei suoi frames, subirà le impostazioni delle proprietà come se fosse una semplice forma geometrica. Per capire meglio, immaginiamo di creare un cartone animato di un ape che muove le ali (una decina di frames p.es.), importiamolo in un nuovo filmato come symbol e a questo punto potremo imporgli cambi di dimensione, trasparenza e farlo muovere sulla scena nel tempo, cioè sulla successione temporale fatta di frames del nuovo filmato.

## 4.4.4 Fade in/out

Come per la parte audio, anche in Flash si è reso necessario un meccanismo di assolvenza/dissolvenza ogni qualvolta il sensore veniva toccato/rilasciato, sempre per evitare bruschi ed innaturali cambiamenti di immagine.

Il seguente codice ActionScript agisce su un movie di due secondi, 24 fotogrammi (a 24 fps). Ogni sezione del codice risiede sul frame indicato

**\_global.trimpot0** è il nome della variabile globale dei dati in arrivo dal primo sensore.

```
//frame1 |movie vuoto (in 2nd frame property color alpha = 0
stop());
this.onEnterFrame = function () {
  if (_global.trimpot0 > 0) {
    this.nextFrame();
  }
}
```



```

else if (_global.trimpot0 == 0) {
    this.prevFrame();
}
};
//-----frame 24 |property color alpha = 100
stop();
this.onEnterFrame = function () {
if (_global.trimpot0 < 5) {
    this.nextFrame();}
else {
    stop();
}
}
//-----frame 48 property color alpha = 0
this.onEnterFrame = function () {
gotoAndPlay (1);
}

```

## 4.4.5 Alcune specifiche sugli strumenti di Flash

### 4.4.5.1 Punto di registrazione

Ogni oggetto ha un punto di registrazione che serve come riferimento per tutte le trasformazioni, sia AS che di motion (scale) e shape tween.

In default questo punto risiede, inopinatamente, nell'angolo superiore sinistro, ciò provoca che ridimensionando l'oggetto esso si sposti anche verso destra e in basso. Per ovviare a questo inconveniente è necessario aprire dal menu "Window" il pannello "Info" (se non è già aperto sulla destra) e selezionando l'oggetto, cliccare ed annerire il quadratino centrale dei 9 disponibili.

### 4.4.5.2 Come creare un filmato complesso e trasformarlo in simbolo

Creare il filmato anche su più layers, poi selezionare un layer (in questo modo si selezionano tutti i frames del layer) tenendo premuto SHIFT selezionare uno alla volta anche gli altri eventuali layers poi clic ds su uno dei frame e "copy frames".

Dal menu insert "New symbol" selezionare "export for AS" nominarlo. Nel new symbol creato in modo edit symbol creare tanti layers quanti ne deve contenere (non i frames, non è necessario) selezionarli come in precedenza con SHIFT e clic ds "paste frames". A questo punto è possibile importare nella scena il symbol ed eventualmente copiarlo in un nuovo filmato.

#### 4.4.5.2.1 Import movieclip (symbol da documento \*.FLA)

E' possibile utilizzare come movieclip dei filmati esterni in formato diverso. P. es. \*.swf . E' sufficiente importarli nella libreria o direttamente sullo stage. I filmati swf possono contenere più fotogrammi, l'unico problema riscontrato fino ad adesso è la non importabilità di filmati contenenti forme grafiche con *tweening* (*shape* o *motion*) se sono del tipo gradienti con modifiche del punto focale: si possono creare *tween* tra gradienti con dimensioni e larghezze diverse ma non con variazioni di fuoco.

Modo migliore: creare un nuovo filmato, creare un movieclip, trasformarlo in symbol (movieclip) selezionando "export for action script". A questo punto in qualsiasi altro filmato è possibile caricare nella propria libreria questo symbol in questo modo: creare un nuovo symbol nella libreria, nella finestra "Create new symbol" premere il

pulsante in fondo "Source / Browse (file...)", individuare il documento \*.fla nel quale risiede il symbol desiderato, nella nuova finestra "Select source symbol" selezionare il symbol desiderato.

#### **4.4.5.3 Istanze di uno stesso simbolo su layer diversi**

Quando si instanziano più volte lo stesso simbolo, essi sono autonomi per quanto riguarda l'eventuale motion-tween (no shape), anche distort etc. Invece sono condizionati temporalmente:

- Se sono tutti su un singolo frame in layer diversi vengono eseguiti tutti i frames del simbolo originale
- Se sono tutti su un numero uguale di frames tra loro su layers diversi, si comportano come sopra
- Se il numero di frames è diverso tra i layer, viene eseguito correttamente (con tutti i frames del clip originale) solo l'istanza sul layer col maggior numero di frames, le altre istanze vengono interrotte al numero di frame finale rispetto al clip originale e quindi ripetuti di seguito fino alla fine del layer contenente il numero maggiore di frames. La posizione dei layer è indifferente.

## 4.4.6 Flash Action Script 2.0

### 4.4.6.1 Variabili

Le variabili che devono essere valide per tutto lo script nel filmato devono essere dichiarate prima, al di fuori delle function:

```
var i;  
var xb:Array = new Array();
```

Gli array, così dichiarati, possono contenere infiniti indici (0\_n). Per gestirli si userà un ciclo **for** con indice variabile locale i : for i =0... { xb[i] = ...}.

E' possibile assegnare valori iniziali anche a più indici diversi contemporaneamente:

```
es.   var fb:Array = Array();  
      fb[1] = fb[18] = 0.3;
```

#### 4.4.6.1.1 Variabili legate a testi dinamici

Nel filmato principale, sopra lo sfondo nero, le caselle numeriche con i valori degli ingressi sono dei testi dinamici con assegnazione delle variabili, non dei nomi delle istanze: le assegnazioni delle variabili vanno fatte nella finestra proprietà nel riquadro in basso a ds "**Var:** " .

I nomi delle istanze servono solo ad individuare i campi numerici per poterli fare scomparire all'occorrenza.

### 4.4.6.2 Operatori composti

**"+="** (**-=**)

```
x += y; // x = x + y;
```

**"--" incremento negativo (++ positivo)**

```
var x:Number = 3;  
var y:Number = --x; //y is equal to 2
```

```
var x:Number = 3;  
var y:Number = x--; // y is equal to 3
```

**"!="** **diseguale** (**== uguale**)

```
trace(5 != 8); // returns true  
trace(5 != 5) //returns false
```

**"!"** **inversione booleana**

**! expression** // Inverte il valore Booleano di una variabile o di una espressione

**"||"** **OR**

vera se almeno una delle condizioni è vera, usabile anche ripetuta più volte.

**"&&"** **And**

vera se tutte le condizioni lo sono, usabile anche ripetuta più volte.

### 4.4.6.3 Function (generica)

```
function = nomeFunction(){ //dichiarazione iniziale della funzione, tra le
                          //parentesi{ } il codice da eseguire
                          // ( ) l'eventuale variabile passata
}
```

### 4.4.6.4 Ciclo For

```
for ( i = 0 ; i < 2 ; i++ ){ //i++ incremento positivo della variabile
                               //indice i
...                               //i=0, i=1 e poi esce dal ciclo perché non più minore di 2
}
```

### 4.4.6.5 Condizioni

#### 4.4.6.5.1 if – else – if else – or

```
if (x < 100) {
  statement;
}
else if (y == 101){
  statement;
}
else if (x >= 12 || y <> ...) // || = OR, && = And, != = diverso
gotoAndPlay("startProgram");
}
```

```
if ((passwordTxt.text.length == 0) || (emailTxt.text.length == 0)) {
  gotoAndStop("invalidLogin");
} else if (passwordTxt.text == userID){
  gotoAndPlay("startProgram");
}
```

#### 4.4.6.5.2 Switch

```
switch (nomevariabile da valutare){
case numeroPossibile :

  codice da eseguire

break;
case altroNumeroPossibile:

  altroCodiceDa Eseguire

case...[altri casi seguiti da break]
...
default:

  codiceDaEseguireInCasiNonContemplati

break;
}
```

#### 4.4.6.6 Get Property e Set Property

Ad ogni ripetizione dei filmati swf è possibile modificare le proprietà dei mc. Per farlo però si deve prima conoscere il valore attuale delle proprietà da modificare.

Quindi **getProperty**. Per modificarle **setProperty**.

Per avere il valore attuale di una proprietà di un **mc** istanziato con nome unico, p.es. "mc":

```
nomeVariabileCuiAssegnareilValore = getProperty (nomemovieclip, proprietà);  
es. xDi_mc = getProperty (mc, _x);
```

Per gli array di oggetti:

Nella timeline su uno stesso o diversi livelli copiare uno stesso simbolo dalla libreria e istanziarlo con nomi con indici crescenti del tipo *nome0*, *nome1*, *nome2* ...in modo che sarà possibile gestirli in action script come array ( "nome" + i) [il + associa la stringa alla variabile indice ottenendo la stringa completa del nome indicizzato.]

```
nomeArrayCuiAssegnareilValore = getProperty ("nomearrayMovieClip" + indice), proprietà);  
p. es. b[i] = getProperty ( ( "b" + i ), _x);
```

Per modificare le proprietà occorrono tre valori:

```
setProperty (nomemovieclipOggetto, nomeproprietà, nuovovaloreproprietà);
```

```
es. di array setProperty( ( "b" + i ), _x, ( xb[i] + xbVel[i]));
```

il nome dell'oggetto può quindi essere un nome array e il nuovo valore può essere una espressione.

#### 4.4.6.7 Rotation

E' possibile far ruotare un oggetto (symbol) ad ogni ripetizione del filmato chiedendone prima la attuale posizione (getProperty) e quindi impostarla con setProperty. La rotazione è in gradi da 0 a +/- 180 float. Ai valori maggiori vengono sottratti 360° : p.es +190 = 190-360= -170.

Rotazione positiva in senso orario, negativa in senso antiorario. Da considerare determinante la frame rate: essendo un singolo frame ad essere ripetuto p.es. 24 frame/sec un incremento come nell'esempio seguente di 0.5 significa una rotazione oraria di 12°/sec

```
onEnterFrame = function (){  
setProperty(nomeMovieClip, _rotation, (getProperty(nomeMovieClip, _rotation)+0.5));  
}
```

E' possibile anche controllarne la velocità esternamente (OSC) impostando una variabile al posto della costante (p.es. *nomeVariabile* al posto di 0.5)

#### 4.4.6.8 Utilizzo del mouse come interfaccia

```
Mouse.hide(); //nasconde il puntatore
```

```
Nomevariabile1 = _xmouse; //controllo di due variabili  
Nomevariabile2 = _ymouse;
```

Per controllare un movieclip con il wheelmouse

```
var mouseListener:Object = new Object();
mouseListener.onMouseWheel = function(delta) {
    mc._x += delta;
}
Mouse.addListener(mouseListener);
```

```
// init nuovo oggetto chiamato mouseListener
//onMouseWheel = evento della classe Mouse chiamato dal movimento
//della rotella del mouse
//delta = parametro opzionale di onMouseWheel che aggiunge (vs su) o //sottrae (vs giù) da 1 a
3 linee (+1_+3 - -1__-3) al target
//mc = nome movieclip istanziato
// _x = proprietà (coordinata) del movieclip (_y, _alpha, _width etc.)
// mc._x += delta; => la x di mc = alla x di mc + delta del wheel
```

#### 4.4.6.9 onEnterFrame

All'avvenuto caricamento del file swf vengono eseguiti i codici iniziali non legati a nessun evento (l'evento onLoad non si riferisce al caricamento del swf ma del movieClip e quindi non serve). Quindi viene eseguito l'evento onEnterFrame. Potendo esso contenere movieClip complessi e lunghi con AS in un singolo frame, la ripetizione automatica del filmato avviene al completamento delle azioni richieste all'interno del frame.

come ripetere un'azione ad ogni ripetizione del filmato

```
createEmptyMovieClip("nomemovieclip", getNextHighestDepth());
// oppure con un mc esistente
nomemovieclip.onEnterFrame = function (){
//azione da fare ad ogni ripetizione del frame unico
}
```

#### 4.4.6.10 Gradienti e colori

```
import flash.geom.* // serve per effettuare modifiche al mc
var focale; // variabile del fuoco del mc radiale da -1 a +1 float sposta il
//fuoco nella sfera da sn a ds
var colors:Array = [0x000000, 0x000000,0x000000 ]; // init nero per i 3 canali RGB
Mouse.hide();
onEnterFrame = function (){
    this.createEmptyMovieClip("mc14", 2);
    if( _global.address0 == "/analogin/0/value" )
    {
        ratioY = (_global.trimpot0 / 1023 * 255) ; //variabile di uno dei due gradienti
    }
    if( _global.address1 == "/analogin/1/value" )
    {
        colorR = (0x000000 + (_global.trimpot1 / 1023 * 16581375)) ;
// sommando l'HEX [0x] ad un intero si ottiene uno dei tre colori (quantità) dopo 0x //che indica l'HEX
//gli altri 6 numeri sono
    }
    if( _global.address2 == "/analogin/2/value" )
    {
        ratioX = (_global.trimpot2 / 1023 * 255);
    }
    if( _global.address4 == "/analogin/4/value" )
    {
```

```

        focale = (_global.trimpot4 / 512 - 1) ;
    }
    if( _global.address5 == "/analogin/5/value" )
    {
        colorG = (0x000000 + (_global.trimpot5 / 1023 * 65025)) ;
    }
    if( _global.address7 == "/analogin/7/value" )
    {
        colorB = (0x000000 + (_global.trimpot7 / 1023 * 255)) ;
    }
with (mc14)
{
    colors = [colorR, colorG, colorB];
    fillType = "radial"
    alphas = [100, 100, 100];
    ratios = [0, ratioY , ratioX];
    spreadMethod = "reflect";
//spreadMethod = "pad"; // continuo
    interpolationMethod = "RGB";
    focalPointRatio = focale;
    matrix = new Matrix();
    matrix.createGradientBox(800, 640, Math.PI, 0, 0);
    beginGradientFill(fillType, colors, alphas, ratios, matrix,
        spreadMethod, interpolationMethod, focalPointRatio);
    moveTo(0, 0);
    lineTo(800, 0);
    lineTo(800, 640);
    lineTo(0, 640);
    lineTo(0, 0);
    endFill();
}
}
}

```

#### 4.4.6.10.1 Colors :GradientBevelFilter.colors property

Nelle figure a gradienti, da 2 a più colori, il colore più a sinistra, il primo dei due nell'assegnazione della proprietà **colors**, tre etc, è il colore di riferimento che impegnerà stabilmente la scena almeno nel suo 50% di destra.

Nell'es seg. La proprietà **colors { colors=[xxx, yyy, zzz, ..., nnn] }** è stata suddivisa in due colori con due variabili.

Es:    colorG = (0x00FF00) ; // variabile con assegnazione hex verde (Green)  
       colorB = (0x0000FF) ; //variabile con assegnazione hex Blu  
       **colors = [colorG, colorB]; // colors (GradientBevelFilter.colors property)**

Il secondo colore (nell'es. verde) sfumerà nel blu, all'interno del suo 50% di sinistra in conseguenza dei valori di **ratios (GradientBevelFilter.ratios property)**.

A tre colori:

Es:    colorR = (0xFF0000) ; // variabile con ass. hex rossa (Red)  
       colorG = (0x00FF00) ; // variabile con assegnazione hex verde (Green) colorB =  
       (0x0000FF) ; //variabile con assegnazione hex Blu  
       **colors = [colorR , colorG, colorB];**

Nell'es. seg. la proprietà **ratios**, che è un'array, è suddivisa in due var. che devono susseguirsi incrementandosi (0, 32, 64 .. etc)

```

ratioX = ...; // 0-255
ratioY= ...; // 0-255
ratios = [ratioX, ratioY];

```

con ratioX=0 e ratioY =0 il disegno è diviso a metà tra i due colori. Se i colori son 3 il centrale è solo una riga tra i due altri. Nessuna sfumatura.

Con ratioX=32 e ratio Y = 0 e tre colors il verde centrale sfuma poco nel blu a sn il principale Red è al 50% a ds.

Con due soli colori (G e B principale) il Blu principale sfuma poco nel verde a sn.

Con ratioX al max 255 e sempre ratioY =0, il verde di ds sfuma completamente nel blu di sn assorbendolo tutto.

## Conclusioni

Le *ratios* devono essere incrementali, altrimenti vengono assunti lo stesso come tali.

Per avere una figura che emerga dal nero di fondo si devono usare 3 colori, nero, un altro anche variabile, e nero. Anche più colori con l'avvertenza del primo ed ultimo che devono essere nero, o comunque lo stesso colore del fondo. In questo modo si ha la possibilità di fare una box<sup>28</sup> di varie dimensioni, anche scalabili, che non viene mai vista nei suoi bordi.

Il colore principale, visualizzato a destra nel **fillType** "linear" e al centro in quello "radial" ed inserito per prima in RGB, occuperà sempre almeno il 50% della superficie nel filType "linear".

Trovare una formula che assicuri che la Ratio Z, RatioY e RatioX siano incrementali e con una differenza minima di "10" (passaggio sfumato minimo).

Le condizioni iniziali sono:

$(ratioZ + 10) < ratioY$  con  $ratioZ > 0$

$(ratioY + 10) < ratioX$

$ratioX < 255$

La formula è:

$ratioX = \text{Math.pow}(2, potenza) + 40;$  // con  $41 < ratioX < 253.828$

dove ratioX è l'ultima delle tre ratio che formano l'array **ratios**. E' ottenuta elevando 2 ad una potenza float compresa tra 0 e 7.75 con risultati compresi tra 41 e 253.828. L'offset di +40 è necessario per garantire la condizione  $(ratioY+10) < ratioX$ .

$potenza = variabile2 / rangemax * 7.75;$

in cui *potenza* è l'indice di elevazione a potenza della ratioFinale (ratioX), viene trovato normalizzando ad 1 il valore ottenuto dal sensore (variabile2) e moltiplicandolo per una costante empirica.

$ratioZ = (ratioX / 4) - 10;$

la *ratioZ*, prima delle tre ratio che compongono l'array **ratios**, dipende dalla  $ratioX/4$ . Viene sottratto un offset di 10 derivato dalla necessità di offset a 40 per ratioX.

$ratioY = (variabile1 / rangemax) * (ratioX - ratioZ - 20) + ratioZ + 10;$

questa è la ratio centrale che dipende dalle altre due e dalla variabile del sensore1. Il valore normalizzato ad 1 del sensore viene moltiplicato per la differenza degli altri valori, in modo da ottenere sempre valori compresi tra essi. Dovendo soddisfare le condizioni  $ratioZ + 10 < ratioY$  e  $ratioX + 10 > ratioY$ , si sottrae 20 al

---

<sup>28</sup> In questo capitolo si fa riferimento ad una Box, essendo un rettangolo quello creato dal `movieclipvuoto(empty)`



range utile. Infine si aggiunge l'offset della ratio minore/sottostante/precedente nell'incremento (ratioZ) e si aggiunge ad esso l'offset desiderato di 10 per avere sempre un minimo di sfumatura.

La rotazione e il punto focale sono una coppia di variabili che possono svelare la box nascosta quando l'mc viene traslato!.

Si utilizzeranno i sensori lineare(**S1/5**) e di pressione(**S2/6**), affidando al lineare il valore per il quale si moltiplicheranno le ratio e alla pressione la "larghezza" del RatioY (secondo, centrale)

E' da evitare lo **spreadMethod = "reflect"**; che crea anelli che possono rivelare la box, a meno ch  non occupi l'intero screen e non sia mobile.

Nel **fillType = "linear"** il colore principale   a destra, in **fillType = "radial"** il colore principale   al centro ed il terzo colore occupa lo sfondo.

In **beginGradientFill(fillType, colors, alphas, ratios, matrix, spreadMethod, interpolationMethod, focalPointRatio)**; tutte le propriet  devono essere presenti e correlate

Idem in **matrix.createGradientBox(width, height, rotazione, tx, ty)**; dove tx e ty sono le coordinate della box .

## **5 – COMPOSIZIONE**

## 5.1 Effimere soddisfazioni

Durante le prime sperimentazioni del sistema ci si accorse che la grafica prendeva immediatamente le redini del sistema per quanto riguardava la creatività. Risultava quasi sempre proibitivo immaginare una forma/colore/luce partendo da un algoritmo audio o comunque da un'idea prettamente musicale. Viceversa la scelta di materie sonore con le quali rivestire semplici forme geometriche era molto più immediata ed addirittura istintiva. Sospettamente istintiva!

Per altro tutti gli sforzi fatti per creare lo strumento ed il sistema sembravano frustrati dalle misere realizzazioni che ne iniziavano a venir fuori, anche se erano momenti di estrema felicità riuscire a realizzare codici ed algoritmi che facevano esattamente quello che desideravo. Ma ogni volta il calvario sopportato per realizzarlo e la felicità raggiunta venivano immediatamente annullati dal desiderio di progredire ulteriormente: subito ci si ritrovava di fronte, per l'ennesima volta, ad un nuovo problema e alla sofferenza da affrontare per risolverlo.

Anni di studio della composizione sembravano non aver lasciato alcuna traccia di metodo e ragione, tutto veniva travolto dal desiderio di sperimentare nuove tecniche con la nuova tecnologia.

Finalmente dopo mesi di abbandono del progetto, spesi in ancor meno fortunate imprese, fui costretto a dar forma finita a qualcosa da presentare.

Sorse l'idea, nacque una "storia" e si iniziò a scriverla.

## 5.2 "L.194/78" primo pezzo

### 5.2.1 La fine di alcune idee

La vita ci costringe incessantemente a cercare qualcosa, i più abili o fortunati inquadrano subito l'oggetto della propria ricerca, altri mai e altri pur trovando quel che cercano non se ne avvedono.

La donna è biologicamente indirizzata verso il desiderio dei figli e l'uomo a darglieli, ma forse per la banalità presunta di queste inclinazioni genetiche, entrambi si spingono per mari e per monti nella ricerca che fu di Gilgamesh.

Quando per caso o per volontà concepiscono un bambino il loro mondo individuale si sgonfia per seguire quello del bimbo, per tornare dopo un po' a rigonfiarsi quando il bambino viene catturato dal mondo degli altri. Ma cosa succede a quei mancati genitori che per motivi terapeutici hanno dovuto rinunciare al loro desiderio quando già avevano iniziato a sgonfiare le loro ricerche personali? E cosa può voler dire per loro essere costretti a veder comunque nascere quel loro figlio mancato in nome di una sedicente alleanza tra la scienza (leggi tecnica) e la religione (leggi clero)?

E il figlio nato, se in grado di pensare, che ne pensa?

Il gusto della composizione ci costringe incessantemente a ricercare qualcosa, i più abili o fortunati inquadrano subito l'oggetto della propria ricerca, altri mai e altri pur trovando quel che cercano non se ne avvedono.

Il compositore è professionalmente indirizzato verso il desiderio di dar vita alle proprie idee, ma per evitare le banalità del passato e del presente si spinge per mari e per monti nella ricerca di nuove strade da percorrere.

Quando per caso o per volontà concepiscono un pezzo il loro mondo privato si

dissolve per seguire la sua nascita, per tornare dopo un po' a ricostruirsi quando il pezzo viene eseguito. Ma cosa succede a quegli autori che per motivi di evoluzione della ricerca stessa hanno dovuto rinunciare al loro desiderio di partorire un nuovo brano quando già pregustavano la sua esecuzione? E cosa può voler dire per loro essere costretti da esigenze di varia natura a veder comunque nascere quel loro figlio mancato in nome di superiori e loro stesse esigenze?

E il pezzo abortito, costretto a venire alla luce, che fine fa?

## 5.2.2 Ideazione ed intervento terapeutico

La possibilità data dal sistema di pensare ad un brano multimediale ha determinato per lungo tempo una necessaria acquisizione delle tecniche disponibili, ma non purtroppo una sufficientemente approfondita analisi delle implicazioni semantiche. Solo quando la realizzazione iniziava a configurarsi come *continuum* espressivo, svanita l'euforia dell'esploratore, ci si accorgeva che l'associazione audio-grafica nascondeva molteplici e poco note implicazioni estetiche. L'apparente libertà ideativa che consentiva di pensare ad esso sia partendo da immagini grafiche che sonore, si traduceva nella sconfinata landa di una ricerca senza fine e senza fini.

Presto si intravedeva l'enorme mole di lavoro necessaria alla realizzazione di una semplice idea, sia per acquisire/inventare le nuove tecniche di realizzazione sia per la quantità di dati necessari da montare. D'altra parte si pose immediata la comparazione con le attuali tecniche di *computer-graphic*, che con enormi budget e migliaia di ore lavorative consente all'industria cinematografica di far rivivere gli splendori di quello che fino agli anni 90 pareva essere ormai un genere in estinzione: i film di animazione.

Come è noto le tecniche di composizione prevedono un disegno generale, una preliminare idea formale alla quale in corso d'opera si consente di deviare lungo strade ignote. Ma in questo caso si affronta un mezzo tecnico che consente una ridiscussione completa delle gerarchie percettive e mnemoniche, un possibile fare musica dove non più l'arte del calibrare e del sottrarre è modello comportamentale, dove i modelli stessi mancano perché è una musica che si fa immagine che si fa forma che si rifà suono e deve prevedere l'imprevedibile dell'esecuzione e non solo, come sempre, l'imprevedibilità della sua percezione.

Approntata la macchina, calibrato il sistema, si è fatta la scelta di *narrare* la stessa esplorazione come se fosse un'impresa epica, anzi di più, un'ennesima "Genesi".

Il brano, il cui titolo sarebbe dovuto essere "**La vera storia del dio geometra**" e che è diventato invece un più appropriato "**L.194/78**" immagina un'impossibile creazione dal nulla di un piccolo universo domestico ad opera di un dio geometra.

Più volte in filosofia ci si è chiesti se Dio conosca<sup>29</sup> la geometria, intendendo dire se la natura che ha creato con le sue complicatissime leggi fisiche è stata da lui creata con tutte le conoscenze necessarie a determinare i giusti pesi atomici e l'equilibrio delle forze deboli e forti o se è solo la nostra miopia che ci spinge a leggere il creato nelle formule matematiche, fisiche e chimiche e l'atto divino sia stato effettivamente come dice la Torah un *semplice*: "*Sia la luce*" e *luce fu*.

Il mio ateismo sarcastico e miscredente mi ha suggerito l'immagine di un povero dio del primo tipo, e quindi Geometra, ovviamente *Honoris Causa*, che crea una pallina, la fa muovere un po' troppo e questa esplode in altre palline che rimbalzano

---

<sup>29</sup> "Conoscesse" per i nichiani e per quelli che pensano che l'azione sia avvenuta un tempo remoto e un pochino ne dubitano.

fino a formare una linea. Che questa linea inizi a vibrare e formi un'onda, che quest'onda si sdoppi creando un elastico o uno scacciapensieri bidimensionale, che rompendosi disegna figure geometriche, che ruotando diventano solidi, che esplodano in forme libere ... fino a causare il suo licenziamento e relativa sostituzione con un dio architetto o ingegnere ma molto meno dotato, cui solo imputare le attuali brutture del mondo.

Questa la storia, che forse andrebbe bene per un cartone animato ... e qui mi sono reso conto, mentre la realizzavo, che le associazioni audio-grafiche potevano rifarsi, sia compositivamente che percettivamente, solo all'iconografia dei cartoni animati.

Intendo dire che nella ricerca ossessiva di una relazione sincronica tra visto ed ascoltato, due erano le strade possibili: l'associazione tra evento grafico e un suo plausibile suono oppure l'incongruenza. Se disegno una corda che vibra lentamente il suono che posso affiancargli non può essere molto diverso da quello di una modulazione sincrona d'ampiezza in bassa frequenza pena la percezione di due eventi (l'audio e quello grafico) separati e appiccicati. Ma questa che ne risulta non è altro che l'estetica dei cartoni animati, dove la porta fa slam, il tuffo fa splash e la campana fa dong. E pur amando i cartoni animati per la loro possibilità di sfidare tutte le leggi fisiche, logiche e razionali, detesto la loro verosimiglianza, che ne è invece l'indispensabile supporto.

Il mio desiderio è quello di disegnare la musica non di musicare i disegni.

Ma utilizzando uno spazio di riferimento cartesiano bidimensionale, pur forzandolo alla tridimensionalità prospettica, qualunque segno informale che possa depositare sulla tela di pixel sarà a destra di un altro, sopra o sotto di dove era prima, acquisirà immediatamente cioè una sua fisicità spazio-temporale che lo farà percepire come "oggetto". E un oggetto è condannato ad assumere una forma conosciuta, come le nuvole o le onde del mare o le stelle nel cielo.

Anche se già Kandinski aveva trovato la soluzione ed inventato la tecnica per eludere le gabbie formali, sfumando i contorni e contraddicendo le forme nascenti con linee che ne spezzano le ambizioni geometriche, si doveva arrivare a Mondrian e a Pollock per negare maggiormente la forma, entrambi ripetendola all'infinito con infinite sfumature, il primo con intersezioni perpendicolari, il secondo con schizzi su schizzi.

Per dare luce e colore al suono era indispensabile liberarsi della forma e per farlo si doveva rinunciare alla possibilità della sua percezione dimensionale. La soluzione trovata<sup>30</sup> era totalmente incompatibile con il già fatto.

Nel frattempo la gravidanza era andata avanti, e il feto pareva dare segni di vita: era necessario un aborto terapeutico, per salvare la vita dell'idea. Ma era anche necessario salvare la vita dello sfortunato feto. Ci si appellò alla legge 194 del 1978 sull'interruzione di gravidanza.

---

<sup>30</sup> Alla quale sto lavorando e di cui preferisco non scrivere

## 5.2.3 Algoritmi

Di seguito sono esposti gli algoritmi di Max e le ActionScript di Flash utilizzate per la creazione della successione di scene. In Max le patches sono state numerate P001-P002 etc. in Flash hanno i nomi dei files (\*.swf).

### 5.2.3.1 Prima scena “Click”

Il primo filmato di Flash (00ClickInit.swf) consiste in un punto bianco che compare per un frame ad ogni contatto del sensore. In audio (P001) è sincronizzato ad un click.

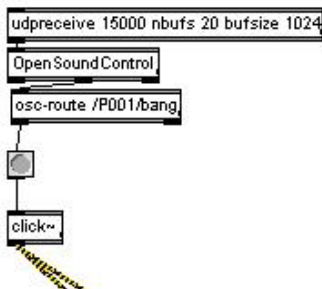
#### Codice ActionScript

```
function buildOscMsgAmpBang(){//costruisce il bundle e lo invia a livello 0 per
                                //inviarlo a MAX
    _level0.msgNameAmp = "";
    _level0.msgArgAmp = 0;
    _level0.msgNameBang = "P001/bang"; //indirizzo del messaggio
    _level0.msgArgBang = "bang";      //argomento del msg
    _level0.sendOSCIterAmpBang();     //interfaccia tra il filmato principale di
                                //connessione e questo
}
var shot:Boolean = false;

onEnterFrame = function (){
    puls._visible = false;
    if( _global.address0 == "/analogin/0/value" )//se arriva msg da sensore 1
    {
        if (shot == false){           //e la variabile di controllo
            if (_global.trimpot0 != 0){ //è falsa cioè non attivata
                puls._visible = true;
                buildOscMsgAmpBang();
                shot = true;           //attiva la var ctrl
            }
        }
    }
}
puls._visible = false;
```

Riceve i dati dei sensori attraverso l’algoritmo principale di Max, visualizza una pallina e invia un bang a Max per il Click. Il solo sensore S1 (/analogin/0/value) è utilizzato, indifferente dalla posizione, come semplice interruttore.

In Max un generatore di Click emette il proprio click.



Il server OSC di Max e il filtro per l’indirizzamento dei messaggi a “/P001/bang”  
Il conseguente bang ad un generatore di click.

### 5.2.3.2 Seconda scena “PulseTrain”

Il filmato si apre con una scena determinata sia in Max che in Flash dove il puntino ha inizialmente dimensioni tali da rendere bianco lo schermo ed in un secondo rimpicciolisce e scompare. In Max la semplice attivazione del secondo algoritmo avvia un involuppo del click da parte di un riverbero (parte cerchiata in blu). A questo punto inizia la sezione nella quale S1 comanda in Max mediante un riscaldamento inverso dei dati, 0-1023 in 1000-15.2, la frequenza di un treno di impulsi e in Flash (`_global.trimpot0`) la posizione della testina di lettura (*cursor*) di un loop di frames (dal frame 27 al 49) che determina un più o meno rapido scomparire ed apparire della pallina.

#### ActionScript “PulseTrain.swf”

frame 27\_49

```
onEnterFrame = function (){
    if( _global.address0 == "/analogin/0/value" )
    {
        cursor = Math.round(Math.sqrt(_global.trimpot0/1020) * 25 + 25);
        if (cursor <= 26){
            gotoAndStop (50);
        }
    }
}
gotoAndPlay (cursor);
```

frame 50

```
onEnterFrame = function (){
    if( _global.address0 == "/analogin/0/value" )
    {
        cursor = Math.round(Math.sqrt(_global.trimpot0/1020) * 25 + 25);
        if (cursor != 25){
            gotoAndPlay (26);
        }
        else{
            gotoAndStop(50);
        }
    }
}
```

Nella patch di Max è presente un generatore di treni di impulsi e un riverbero controllato temporalmente.

In alto sono presenti i ricevitori dei dati degli otto sensori, ma sono collegati solo il primo e terzo sensore della mano destra, che controllano rispettivamente il periodo del treno di impulsi da 1000 a 15 impulsi al secondo e il bilanciamento tra i due canali con una funzione seno.





#### 5.2.3.4 Quarta scena “Biliardo”

Il punto iniziale singolo, raggiunto il limite superiore di vibrato, esplose in P004 dando vita a 5 punti che rimbalzano sui bordi dello schermo.

5 perché è il numero massimo di oggetti singoli in movimento che si possono seguire contemporaneamente. La prima prova è stata effettuata con 20 ai quali si associavano venti oscillatori controllati in frequenza dalla y del punto e in pan dalla x.

Il primo impatto è stato un po' deludente, in quanto la gran mole di informazioni visive e audio era ingestibile cognitivamente, saturando quindi immediatamente le possibilità di discernimento delle individualità associate in posizione schermo/altezza+pan audio.

L'effetto complessivo, richiamante all'inizio un corteo di macchine della polizia, vigili del fuoco, autombuanze e quant'altro, lasciava il posto, con difficoltà, ad una possibile ricerca di coerenze, soprattutto per gli eventuali punti con  $y=0$  (freq. Fissa) o  $x=400$  (pan centrale fisso).

Quindi, con la pazienza che di certo non si può aspettare da un pubblico critico, prevenuto e saturo di immagini computerizzate realistiche e di stupefacente complessità, si potevano notare le congiunzioni centrali o laterali dei punti, che creavano agglomerati abbastanza bene individuabili.

La perfezione dell'automatismo (richiamante per i meno giovani) il secondo gioco per computer dopo il tris, quello del tennis (Pong), e la sua apparente semplicità realizzativa, lo rendevano immediatamente stucchevole e banale. Considerato che le qualità del suono implementate erano le maggiormente informative, in quanto evidenziavano, credo al massimo, i movimenti vettoriali delle palle, escludendo quindi che si sarebbe ottenuta una maggior “comprensibilità” associando altre qualità del suono, si è scelto di ridurre il numero a quello di una mano (o forse meglio solo tre, il numero di voci più facilmente seguibile).

Non volendo comunque rinunciare al lavoro fatto ottenendo 20 palle sonanti, e non potendolo risfruttare in seguito, almeno nello stesso modo di palle su un biliardo, ho preferito immaginare una “sbocciatura” iniziale con conseguente “deperimento/smarrimento spaziale” delle 17 palle eccedenti.

Inoltre, per accrescere l'interesse, inserire il ludico controllo umano nelle traiettorie e nelle dimensioni delle tre superstiti mediante l'affidamento di due qualità di ogni palla ad una coppia di sensori del **Make**.

##### Action Script

```
var xb:Array = new Array();
var yb:Array = new Array();
var xfake:Array = new Array();
var yfake:Array = new Array();
var xbVel:Array = new Array();
var ybVel:Array = new Array();
var contatoreFrames:Number = 0;
var soglia1:Number = 24 ; //numero di frames dopo il quale init snd part
var soglia2:Number = 24 ; //partialTime init = soglia1
var decrementVolte:Number = 20; //per far scomparire le balls 3_19
var quantiSecondi:Number = 1; //inizializza la costante che determina il numero di secondi
    //tra la scomparsa di una ball e l'altra
var secondiPassati:Number = 0; //conta i secondi passati dalla scomparsa della precedente (o init)
var qualeMC:Number = 0; //individua la ball 0_2 influenzata dai sensori
var deepVar:Number = 1; // fattore profondità delle variazioni height e width della ball scelta(0_2)
Mouse.hide();

function buildOscMsgXY(){ //costruisce il bundle e lo invia a livello 0 per inviarlo a MAX
    _level0.msgNameX = "P004/bx"+ i;
    _level0.msgArgX = (xb[i]);
    _level0.msgNameY = "P004/by"+ i;
```

```

    _level0.msgArgY = (yb[i]);
    _level0.sendOSCInterface();
}
function buildOscMsgMute(decrementVolte){ //costruisce il bundle e lo invia a livello 0 per inviarlo a
//MAX
    _level0.msgNameMute = ("b" + decrementVolte + "mute");
    _level0.msgArgMute = 1;
    _level0.sendOSCInterMute();
}

function randRange(min:Number, max:Number):Number {
    var randomNum:Number = Math.floor(Math.random() * (max - min + 1)) + min;
    return randomNum;
}

    for (i = 0; i < 20; i++){ //init 20 balls posizione centrale sovrapposti
        xb[i]=400;
        yb[i]=300;
        buildOscMsgXY(); //invia a level0 i dati per l'invio init a Max
        xfake[i]=400;
        yfake[i]=300;
//chiamata della function "randRange" per l'assegnazione delle 20 coppie di incrementi xy
        xbVel[i] = randRange(-20, 20)
        ybVel[i] = randRange(-20, 20)
    } //fine ciclo for

function ricavaCoordinate(){ //e resetta le velocità negative
    for ( i = 0; i < decrementVolte; i++){
        xb[i] = getProperty ("b" + i), _x); //trova la x attuale delle balls
        if (xb[i] < 0) {
            xbVel[i] = Math.abs(xbVel[i]);
        }
        yb[i] = getProperty ("b" + i), _y); //trova la y attuale delle balls
        if (yb[i] < 0) {
            ybVel[i] = Math.abs(ybVel[i]);
        }
    }
}

function controllaRange(){
//verifica il range utile 1600*1200 che è il doppio
//del formato 800*600, onde ottenere i rimbalzi algebricamente, e richiama
//la funzione di avanzamento con l'indice "i" variabile di riferimento ball[]
    for ( i = 0; i < decrementVolte; i++){
        if ((xbfake[i] < 1600) && (ybfake[i] < 1200) ){
            rotola(i);
        }
        else if (xbfake[i] >= 1600){
            xbfake[i] = xbfake[i] -1600; //azzera la x indice per ripartire da x = 0
            rotola(i);
        }
        else if (ybfake[i] >= 1200) {
            ybfake[i] = ybfake[i] - 1200; //azzera la y indice per ripartire da y = 0
            rotola(i);
        }
    }
}

function controllaRange3(){
//verifica il range utile 1600*1200 che è il doppio
//del formato 800*600, onde ottenere i rimbalzi algebricamente, e richiama
//la funzione di avanzamento con l'indice "i" variabile di riferimento ball[]
    for ( i = 0; i < 3; i++){
        if ((xbfake[i] < 1600) && (ybfake[i] < 1200) ){

```

```

        rotola3(i);
    }
    else if (xbfake[i] >= 1600){
        xbfake[i] = xbfake[i] -1600;    //azzera la x indice per ripartire da x = 0
        rotola3(i);
    }
    else if (ybfake[i] >= 1200) {
        ybfake[i] = ybfake[i] - 1200;    //azzera la y indice per ripartire da y = 0
        rotola3(i);
    }
}
}

```

```

function rotola(i) {
    // funzione di incremento coordinate
    //esamina le condizioni possibili della coppia di coordinate indice (fake)
    //e cambia le coordinate delle balls (b[i]._x e ._y). Questa function viene
    //richiamata tante volte quante il n. di balls. Quindi invia le nuove coordinate
    //al builder OSC per il trattamento MaxMSP
    if ((xbfake[i] < 800) && (ybfake[i] < 600)){
        //x e y positive
        setProperty("b"+i, _x, (xb[i] + xbVel[i])); //incremento positivo
        setProperty("b"+i, _y, (yb[i] + ybVel[i])); //incremento positivo
        xb[i]=(xb[i] + xbVel[i]);
        yb[i]=(yb[i] + ybVel[i]);
        buildOscMsgXY();
    }
    else if ((xbfake[i] < 800) && (ybfake[i] >= 600)){
        //y fuori
        setProperty("b" + i, _x, (xb[i] + xbVel[i])); //incremento positivo
        setProperty("b" + i, _y, (yb[i] - ybVel[i])); //incremento negativo
        xb[i]=(xb[i] + xbVel[i]);
        yb[i]=(yb[i] - ybVel[i]);
        buildOscMsgXY();
    }
    else if ((xbfake[i] >= 800) && (ybfake[i] >= 600)){
        //x e y fuori
        setProperty("b" + i, _x, (xb[i] - xbVel[i])); //incremento negativo
        setProperty("b" + i, _y, (yb[i] - ybVel[i])); //incremento negativo
        xb[i]=(xb[i] - xbVel[i]);
        yb[i]=(yb[i] - ybVel[i]);
        buildOscMsgXY();
    }
    else if ((xbfake[i] >= 800) && (ybfake[i] < 600)){
        //x fuori
        setProperty("b" + i, _x, (xb[i] - xbVel[i])); //incremento negativo
        setProperty("b" + i, _y, (yb[i] + ybVel[i])); //incremento positivo
        xb[i]=(xb[i] - xbVel[i]);
        yb[i]=(yb[i] + ybVel[i]);
        buildOscMsgXY();
    }
    xbfake[i] = xbfake[i] + xbVel[i]; //incremento della variabile indice
    ybfake[i] = ybfake[i] + ybVel[i]; //incremento della variabile indice
}
}

```

```

//nuove variabili sensori per le tre balls residue
function rotola3(i) {
    // funzione di incremento coordinate
    //esamina le condizioni possibili della coppia di coordinate indice (fake)
    //e cambia le coordinate delle balls (b[i]._x e ._y). Questa function viene
    //richiamata tante volte quante il n. di balls. Quindi invia le nuove coordinate
    //al builder OSC per il trattamento MaxMSP

```

```

    if ((xbfake[i] < 800) && (ybfake[i] < 600)){
        //x e y positive
        setProperty("b"+i, _x, (xb[i] + xbVel[i])); //incremento positivo
        setProperty("b"+i, _y, (yb[i] + ybVel[i])); //incremento positivo
        xb[i]=(xb[i] + xbVel[i]);
        yb[i]=(yb[i] + ybVel[i]);
        buildOscMsgXY();
    }

```

```

    }
    else if ((xbfake[i] < 800) && (ybfake[i] >= 600)){           //y fuori
        setProperty("b" + i), _x, (xb[i] + xbVel[i]);         //incremento positivo
        setProperty("b" + i), _y, (yb[i] - ybVel[i]);         //incremento negativo
        xb[i]=(xb[i] + xbVel[i]);
        yb[i]=(yb[i] - ybVel[i]);
        buildOscMsgXY();
    }
    else if ((xbfake[i] >= 800) && (ybfake[i] >= 600)){           //x e y fuori
        setProperty("b" + i), _x, (xb[i] - xbVel[i]);         //incremento negativo
        setProperty("b" + i), _y, (yb[i] - ybVel[i]);         //incremento negativo
        xb[i]=(xb[i] - xbVel[i]);
        yb[i]=(yb[i] - ybVel[i]);
        buildOscMsgXY();
    }
    else if ((xbfake[i] >= 800) && (ybfake[i] < 600)){           //x fuori
        setProperty("b" + i), _x, (xb[i] - xbVel[i]);         //incremento negativo
        setProperty("b" + i), _y, (yb[i] + ybVel[i]);         //incremento positivo
        xb[i]=(xb[i] - xbVel[i]);
        yb[i]=(yb[i] + ybVel[i]);
        buildOscMsgXY();
    }
    xbfake[i] = xbfake[i] + xbVel[i];           //incremento della variabile indice
    ybfake[i] = ybfake[i] + ybVel[i];           //incremento della variabile indice
}

function contaSecondi(){
    secondiPassati = Math.floor(contatoreFrames / 24) - Math.floor(soglia2 / 24); // attenzione!
    if (secondiPassati == quantiSecondi){
        decrementVolte-- //diminuisce e scompare una ball nei cicli for segg.
        setProperty("b" + decrementVolte), _visible, 0);
        buildOscMsgMute(decrementVolte); //manda a Max il segnale di Mute
        soglia2 = contatoreFrames ; // azzerata i secondiPassati
    }
}

onEnterFrame = function() {
    ++contatoreFrames; //main ripetuto 24 volte al sec
    //pre increment to 1
    if (contatoreFrames < soglia1){
        ricavaCoordinate();
        controllaRange();
    }
    else if (contatoreFrames >= soglia1){

        if ((contatoreFrames >= soglia2) && (decrementVolte > 3) ){
            contaSecondi();
            ricavaCoordinate();
            controllaRange();
        }
        else if ((contatoreFrames >= soglia2) && (decrementVolte == 3) ){
            ricavaCoordinate();
            //ricava dati dai sensori solo nella 3a parte (3 balls)
            if( _global.address0 == "/analogin/0/value" ){ //modifica altezza temporaneamente
                setProperty("b" + qualeMC), _height, (_global.trimpot0 /1024 *30 * deepVar + 10));
            }
            if( _global.address2 == "/analogin/2/value" ){ //Punto->Linea temporaneo
                setProperty("b" + qualeMC), _width, (_global.trimpot2 /1024 *40 * deepVar + 10));
            }
            if( _global.address3 == "/analogin/3/value" ){ //velocità x
                xbVel[qualeMC] = 10 + (_global.trimpot3 /1024 *20);
            }
            if( _global.address4 == "/analogin/4/value" ){ // decide quale ball riceve i sensori

```

```

//vars
if (_global.trimpot4 <= 341 ){
    qualeMC = 0;
}
else if (_global.trimpot4 <= 628){
    qualeMC = 1;
}
else {
    qualeMC = 2;
}
}
if( _global.address6 == "/analogin/6/value" ){ //velocità y
    ybVel[qualeMC] = 10 + (_global.trimpot6 /1024 *20);
}
if( _global.address7 == "/analogin/7/value" ){ //profondità variazioni x e y
    deepVar = 1+(_global.trimpot7 / 1024 * 10)
}
controllaRange3();
}
}
}

```

Le palline “perse” rientrano una alla volta nello schermo e quando sono di nuovo 20 si allineano equidistanti con  $y = 300$  (centro).

Questo viene ottenuto tramite un filmato step by step realizzato in flash con l'uso del solo Motion tween, prestando le coordinate x delle 20 balls e ... manualmente inserendo i valori positivi e negativi intorno alla y 300 (centrale).

A questo punto, viene accettato il valore in ingresso di un sensore lineare per controllare l'ampiezza max di un inviluppo delle 20 balls a formare una corda vibrante.

L'ampiezza viene normalizzata a 30, il numero di frames e quindi ripetizioni necessarie ad esaurire la vibrazione smorzata è:

$$nFrames = 24 * 2 ^ (amp/10 - 1)$$

dove amp è l'ampiezza max derivata dal segnale del sensore (scale 0\_1023 in 0\_30).

Si hanno per amp = 30 (max forza dell'eccitazione virtuale) un decadimento smorzato in 4 secondi (96 frames a 24f/s); per amp = 20 2 secondi (48 frames) e un secondo per amp = 10.

La seguente formula empirica serve a produrre una serie di numeri quanti sono gli nFrames (in funzione dell'ampiezza) tali da avere un andamento inversamente esponenziale nei limiti amp e 0.

```

for (i=1;i<nFrames+1;i++){
    trace((Math.exp (Math.LN10- i/amp))*amp/10);
}

```

per amp compreso tra 1 e 30. valori superiori di amp fanno schizzare la curva. LN10 è il logaritmo naturale di 10

### 5.2.3.5 P005 e seguenti

Le palline allineate si trasformano in una linea retta ( gli estremi fuori schermo).  
La linea si anima controllata dalla Make.

*La confusione cresce ed il pezzo muore*

### 5.3 “LittleWings” secondo pezzo

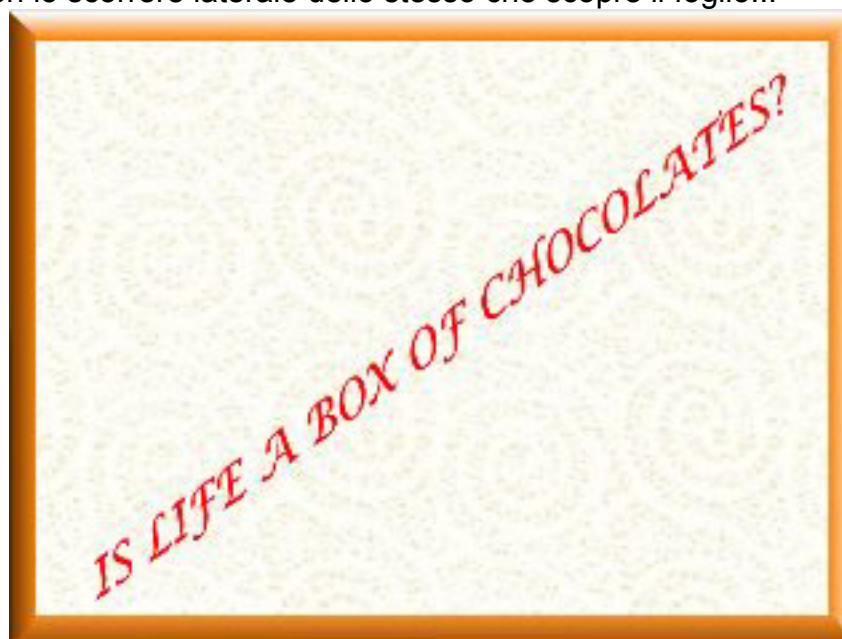
Realizzato per essere rappresentato in una performance durante un festival/expo sulla musica contemporanea, si basa sull'idea che le generazioni più giovani del nostro mondo siano vittime di un pericoloso messaggio: la vita è una scatola di cioccolatini. Intendendo con ciò che sia fatta di facili e molteplici esperienze effimere e saturanti: una successione di *ciliegie* precotte e predigerite, confezionate per attrarre più che per soddisfare il desiderio di nuovo di un giovane di ogni tempo. Non è un discorso sulla caducità delle mode, sempre esistita, ma sulla fascinazione anche tecnologica che porta l'interesse dei giovani a focalizzarsi sempre più brevemente e distrattamente da una distrazione all'altra: una continua distrazione dalla vita stessa.

Suggestionato dal nome dell'interfaccia e dalla musica di Hendrix, ho quindi immaginato una semplicissima e chiara allegoria di un'esistenza spesa nello “scartare cioccolatini”.

Il pezzo inizia con la visione del coperchio di una scatola di cioccolatini ...



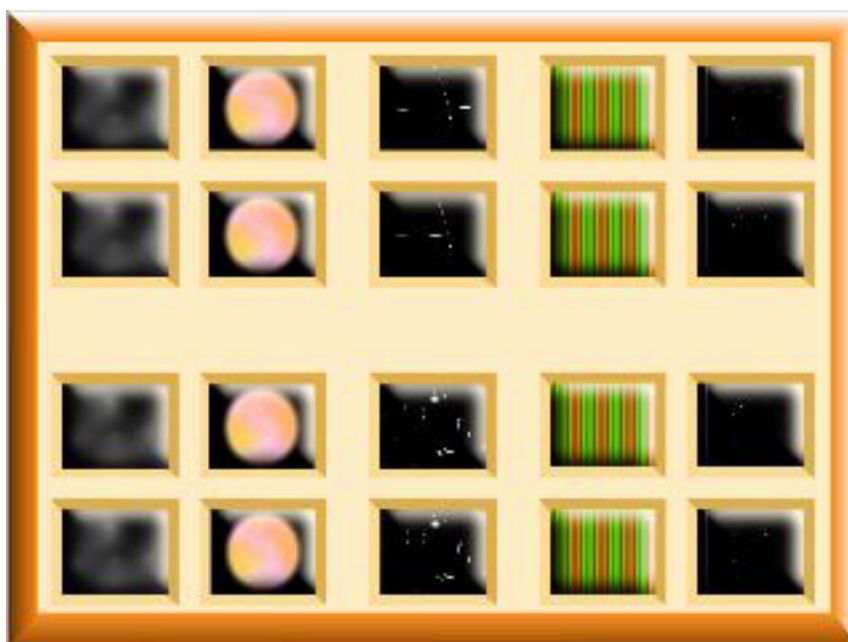
... volendo tradurre: Piccole ali, caramelle grafiche e suoni cioccolatosi. Prosegue con lo scorrere laterale dello stesso che scopre il foglio...



“La vita è una scatola di cioccolatini?”

Questa domanda fa il verso ad una battuta nel film “Forrest Gump” dove il protagonista cita a sua volta la madre che gli diceva che la vita è una scatola di cioccolatini, non sai mai quale ti tocca. Ma il senso che gli attribuisco io è molto diverso.

Anche questo foglio scivola via e scopre la scatola promessa, contenente grafiche dolcezze e suoni di cioccolata, un tripudio sinestetico.



Si è deciso di associare a questo inizio molto semplicemente un brano audio, uno a caso ... “Little Wing” di jimi Hendrix. Ne viene diffusa inalterata la sola introduzione strumentale, usatissima anche in jingles pubblicitari e troppo nota all’ascolto per essere sgradevole.

D’ora in avanti vedremo avvicinarsi a noi un cioccolatino alla volta, che invadendo l’intero schermo introdurrà la scena da cui è tratto l’involucro raffigurato.

Ad ogni intervallo si ripresenterà la scatola con dei vuoti al posto dei cioccolatini “mangiati/visti/ascoltati” ed alcune battute della stessa introduzione di Hendrix, queste volte leggermente modificate nei finali, tutti simili alla fine dell’introduzione (rullante in terzine).

### 5.3.1 Informel

Il primo quadro/cioccolatino è basato sull’associazione rumore (noise) filtrato ed involupato in ampiezza / nuvole di color grigio dinamiche in forma e luminosità/trasparenza.

L’interfaccia fisica **LittleWings** controlla la spazializzazione delle figure grafiche e dei suoni associati, il filtraggio (band-width e Fc e gain) di un generatore di rumore. Per amalgamare le figure e poterle fondere si utilizza anche un filmato di cui si controlla la trasparenza mediante **S4**.

Sulla scena sono presenti 3 filmati “Right” “Center” e “Left” rappresentanti un ovoide bianco sfumato in basso.

ActionScriptCode

```
Mouse.hide();  
var RightFC;  
var AltezzaR;  
var LeftFC;  
var AltezzaL;
```

```

var RightAlpha;
var LeftAlpha;
var RightPan;
var LeftPan;
var SfondoVar;
var test;
onEnterFrame = function (){
    if( _global.address7 == "/analogin/7/value" )
    {
        SfondoVar._alpha = (_global.trimpot7/1023) * 100;    //prima così non flasha
    }
    if( _global.address0 == "/analogin/0/value" )    //Right fc
    {
        if (_global.trimpot0 > 5){
            Right._y = (1-(_global.trimpot0 / 1023))* 600 +300 ;    // altezza del mc R
            AltezzaR = getProperty(Right, _y);
        }
        else if (_global.trimpot0 <= 5){
            var i ;
            for (i = 0; i < 24; i++) {
                Right._y = AltezzaR + i;
            }
        }
    }

    if( _global.address1 == "/analogin/1/value" )    //Right force/gain
    {
        Right._alpha = (_global.trimpot1 / 1023)*100;    //pressione = trasparenza minore
    }
    if( _global.address2 == "/analogin/2/value" )    //Right panpot
    {
        Right._x = ((_global.trimpot2 -2)/700)*400 +200;    //sposta mc dal centro vs ds
    }
    // *****Left*****
    if( _global.address4 == "/analogin/4/value" )    //Left fc
    {
        if (_global.trimpot4 > 5){
            Left._y = (1-(_global.trimpot4 / 1023))* 600 +300 ;    // altezza del mc L
            AltezzaL = getProperty(Left, _y);
        }
        else if (_global.trimpot4 <= 5){
            var i ;
            for (i = 0; i < 24; i++) {
                Left._y = AltezzaL + i;
            }
        }
    }

    if( _global.address5 == "/analogin/5/value" )    //left force/gain
    {
        Left._alpha = (_global.trimpot5 / 1023)*100;    //pressione = trasparenza minore
    }
    if( _global.address6 == "/analogin/6/value" )    //Left panpot
    {
        Left._x = (1-((_global.trimpot6 -2)/700))*400 + 200;    //sposta mc dal centro vs sn
    }
    // *****Center parameters*****
    RightFC = getProperty(Right, _y);
    LeftFC = getProperty(Left, _y);
    Center._y = ((300-RightFC)+ (300-LeftFC)) / 2 + 600;

    RightAlpha = getProperty(Right, _alpha);

```



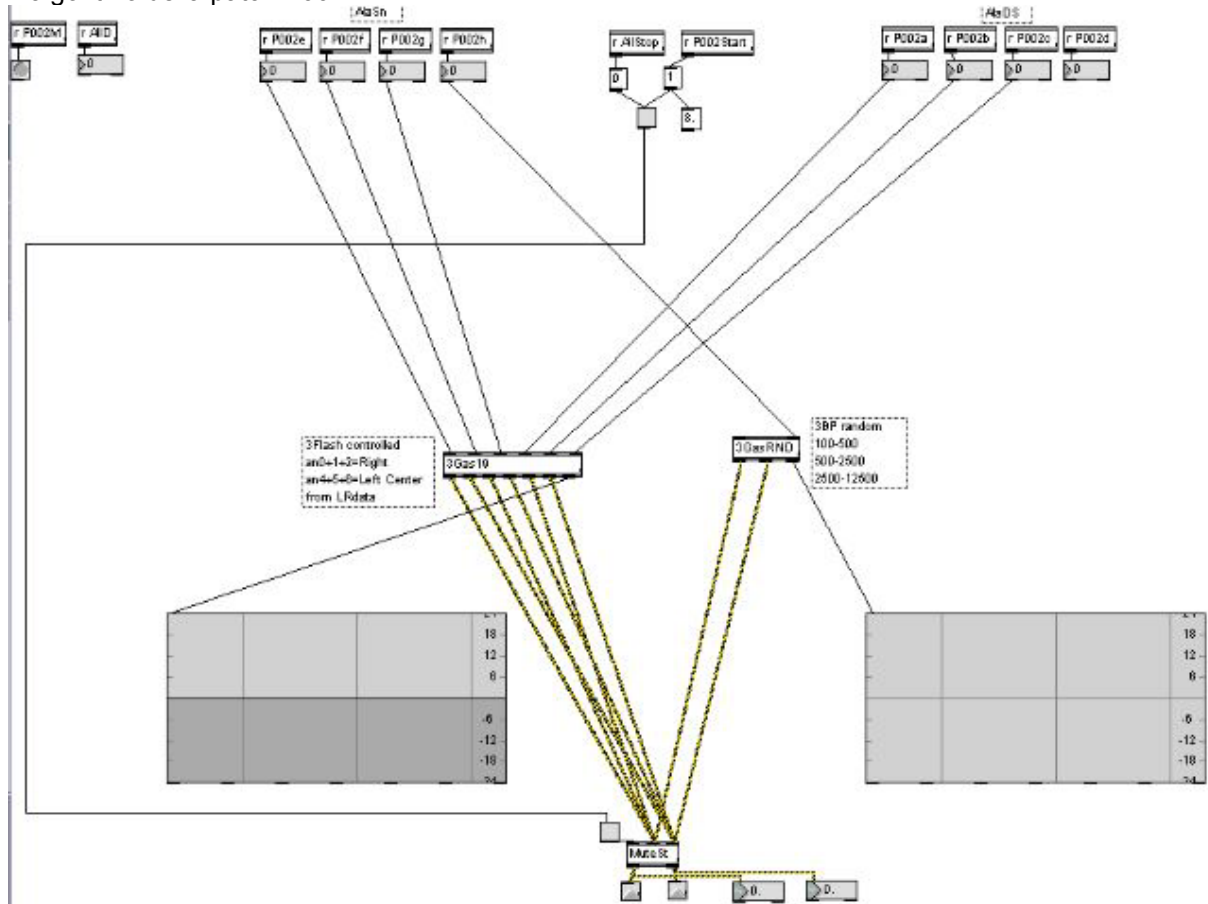
```

LeftAlpha = getProperty(Left, _alpha);
Center_alpha = (RightAlpha + LeftAlpha) / 2;

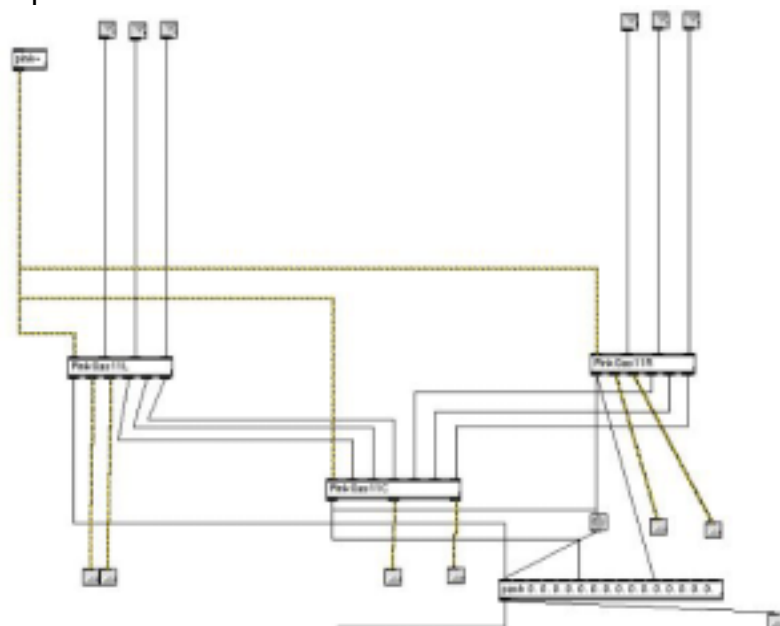
RightPan = getProperty(Right, _x);
LeftPan = getProperty(Left, _x);
Center_x = (LeftPan + RightPan) / 2;
}

```

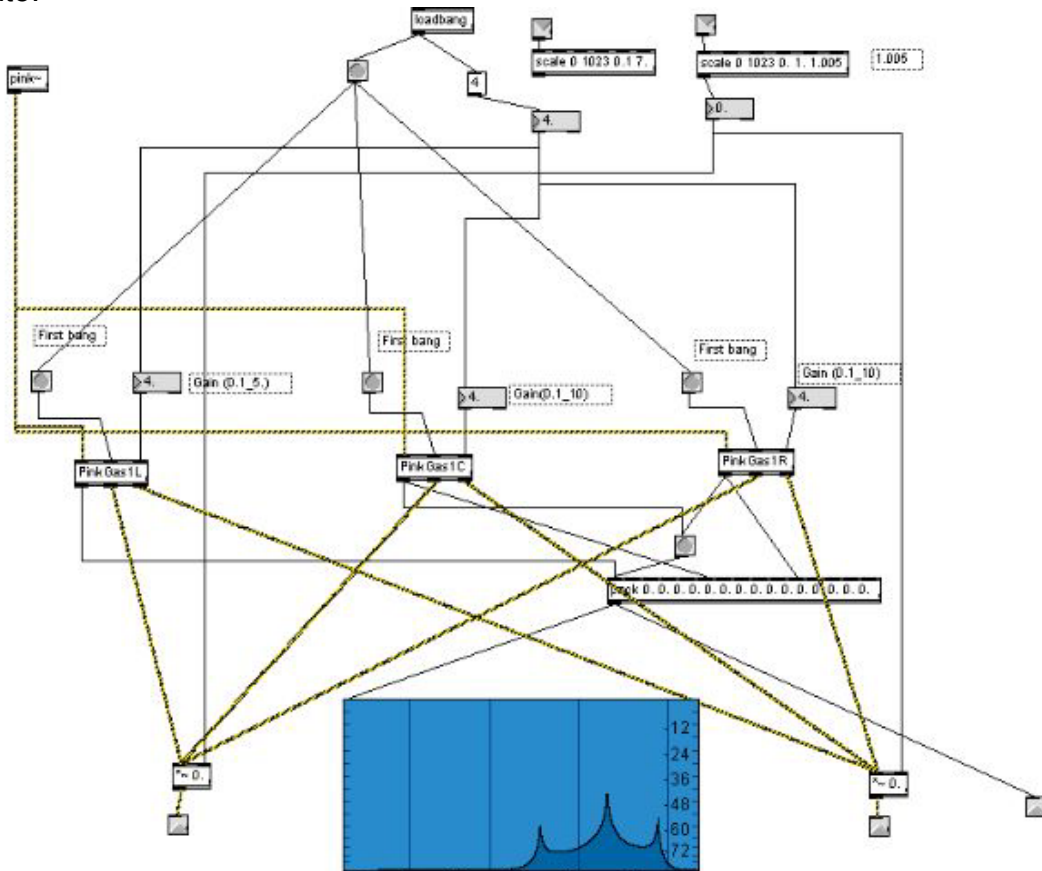
L'algoritmo della patch P002



In alto i ricevitori degli 8 sensori, sei collegati ad una subpatch ed uno ad un'altra. La prima genera i suoni filtrati e controllati da LittleWings, la seconda genera un insieme di rumori filtrati ed involuppati in modo random, per l'associazione al "background" ripetitivo.



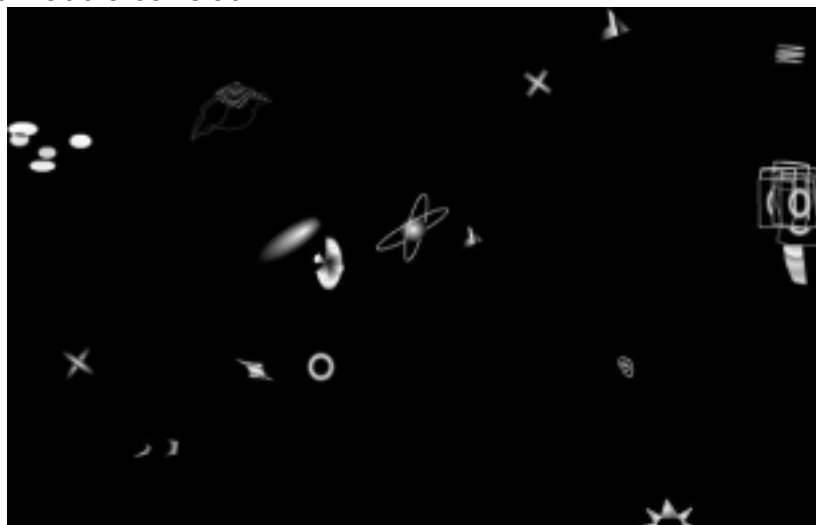
Nella figura precedente il contenuto della prima patch, a sua volta contenente altre tre subpatch, separate per la gestione esterna del bilanciamento stereo. Al loro interno un filtro *biquad* è controllato dal gestore di involuppi precedentemente illustrato.



Questo è invece l'algoritmo random che filtra il noisepink mediante ulteriori subpatch anch'esse basate sul filtraggio e su moduli random.

### 5.3.2 BigBang

Dopo la sparizione del cioccolatino, riappare con le stesse modalità la successiva scena. Questa scena è figlia dell'esplosione di palline bianche che rimbalzano contro i bordi della finestra, se ne differenzia per la diversificazione delle 20 forme grafiche e dei 20 campioni audio correlati.



Gli oggetti sono tutti filmati completi di forme roteanti e dinamiche. Mentre i suoni



### 5.3.3 Scacciapensieri

Una corda ondulante si sdoppia ruotando sugli assi e giocando con una sfera.

Questa è una scena nella quale il controllo dell'algo e del codice è totale, nel senso che non succede niente se non intervengono i sensori.

L'associazione tra corda e suono di scacciapensieri è stata forse ingenua ma efficace alla dimostrazione delle possibilità strumentali dell'interfaccia. Uno strumento in quest'occasione intonato su di una scala minore. La polifonia è a sole tre voci ed il ritmo delle due corde e della sfera è una sovrapposizione di duine e terzine.

L'Action script si snoda sui diversi filmati che la compongono, inviando a Max il segnale di sincronia (bang).

```
Mouse.hide();
var altezza1;
var rotazione1;
var altezza2;
var rotazione2;
var ballalpha;

onEnterFrame = function(){
    if( _global.address1 == "/analogin/1/value" )
    {
        altezza1 = ( _global.trimpot1 - 2 ) / 10.21;        // /1021 * 100;
    }
    if( _global.address2 == "/analogin/2/value" ){
        rotazione1 = ( _global.trimpot2 - 2 ) / 5.672;      // /1021 * 180;
    }
    if( _global.address3 == "/analogin/3/value" )
    {
        ballalpha = ( _global.trimpot3 ) / 5.115; // (x/1023*200) arriva prima al max
    }

    if( _global.address5 == "/analogin/5/value" )
    {
        altezza2 = ( _global.trimpot5 - 2 ) / 10.21;      // /1021 * 100;
    }

    if( _global.address6 == "/analogin/6/value" ){
        rotazione2 = ( _global.trimpot6 - 2 ) / -5.672;   // /1021 * -180;
    }
    setProperty("mc" , _yscale, (altezza1));
    setProperty("mc" , _rotation, (rotazione1));
    setProperty("Line2" , _yscale, (altezza2));
    setProperty("Line2" , _rotation, (rotazione2));
    if(altezza1 < 1){
        sfera._visible = false;
    }
    else {
        sfera._visible = true;
        setProperty("sfera" , _alpha, ballalpha);
    }
}
}
```

codice di una delle due corde

```
function buildOscMsgAmpBang(){ //costruisce il bundle e lo invia a livello 0 per inviarlo a
                                //MAX
    _level0.msgNameAmp = "";
    _level0.msgArgAmp = 0;
    _level0.msgNameBang = "P006/bang2";
```

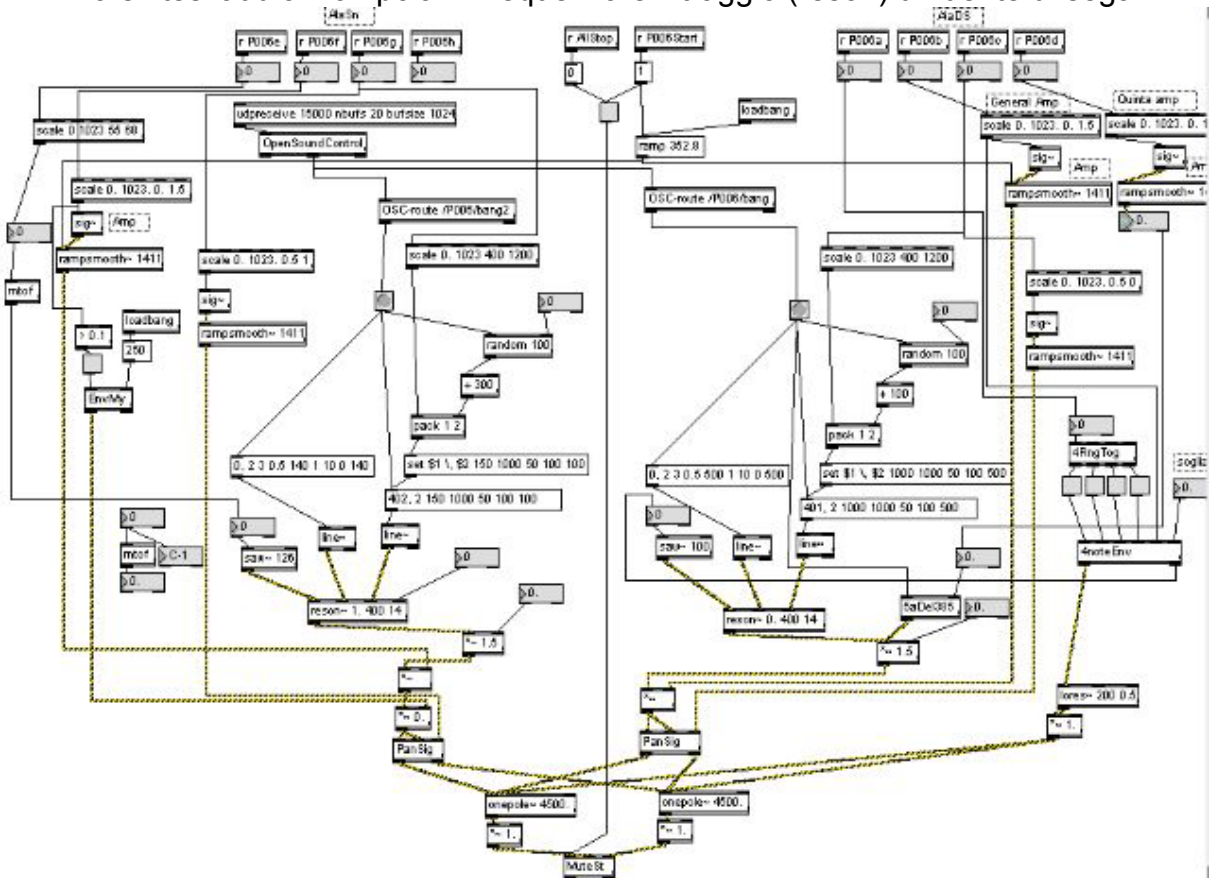
```

_level0.msgArgBang = "bang";
_level0.sendOSCInterAmpBang(); //
}

```

```
buildOscMsgAmpBang();
```

La sintesi audio manipola in frequenza e filtraggio (reson) un dente di sega



e lo intona secondo una scala minore di intervalli.

### 5.3.4 Ruote metalliche

Una sorta di minaccioso giroscopio metallico composto da due cerchi che ruotano variando i propri assi. Ad un certo punto si moltiplicano creando una gabbia sferica virtuale. Sul finire i cerchi iniziano a proiettare una propria fascia luminosa che intensificandosi porta il suono a rasentare la distorsione (riduzione della BW del filtro).

Actionscript

```

var rotazioneP1;
var rotazioneP3;
var DuploP1:Boolean = false;
var RipetizioniP1 = 40;
var n1 = 0;
var temp = 0;
onEnterFrame = function(){

    if( _global.address0 == "/analogin/0/value" ){
        rotazioneP1 = Math.round(_global.trimpot0 / 5.683);// /1023 * 180;
    }

    if( _global.address3 == "/analogin/3/value" ){//duplicazione
        if( _global.trimpot3 > 0 ){//start duplicazione
            DuploP1 = true;
            RipetizioniP1 = Math.round(_global.trimpot3 / 25.575);// /1023 * 40;
        }
    }
}

```

```

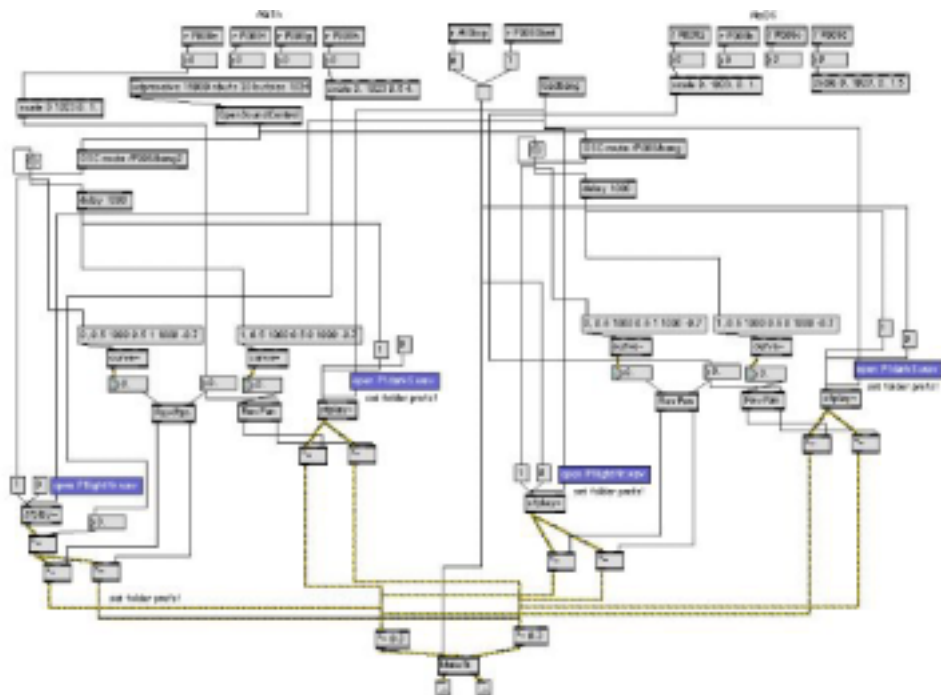
    }
    else {
        DuploP1 = false;
    }
} //fine trimpot3

if( _global.address4 == "/analogin/4/value" ) {
    rotazioneP3 = Math.round(_global.trimpot4 / -5.683); // /1021 * -180;
}
if( _global.address7 == "/analogin/7/value" ) {
    linesAlpha = Math.round(_global.trimpot7 / 10.23); // /1023 * 100;
}

setProperty("P1" , _rotation, (rotazioneP1));
setProperty("L1" , _rotation, (rotazioneP1));
setProperty("P3" , _rotation, (rotazioneP3));
setProperty("L3" , _rotation, (rotazioneP3));
setProperty("L1" , _alpha, linesAlpha);
setProperty("L3" , _alpha, linesAlpha);

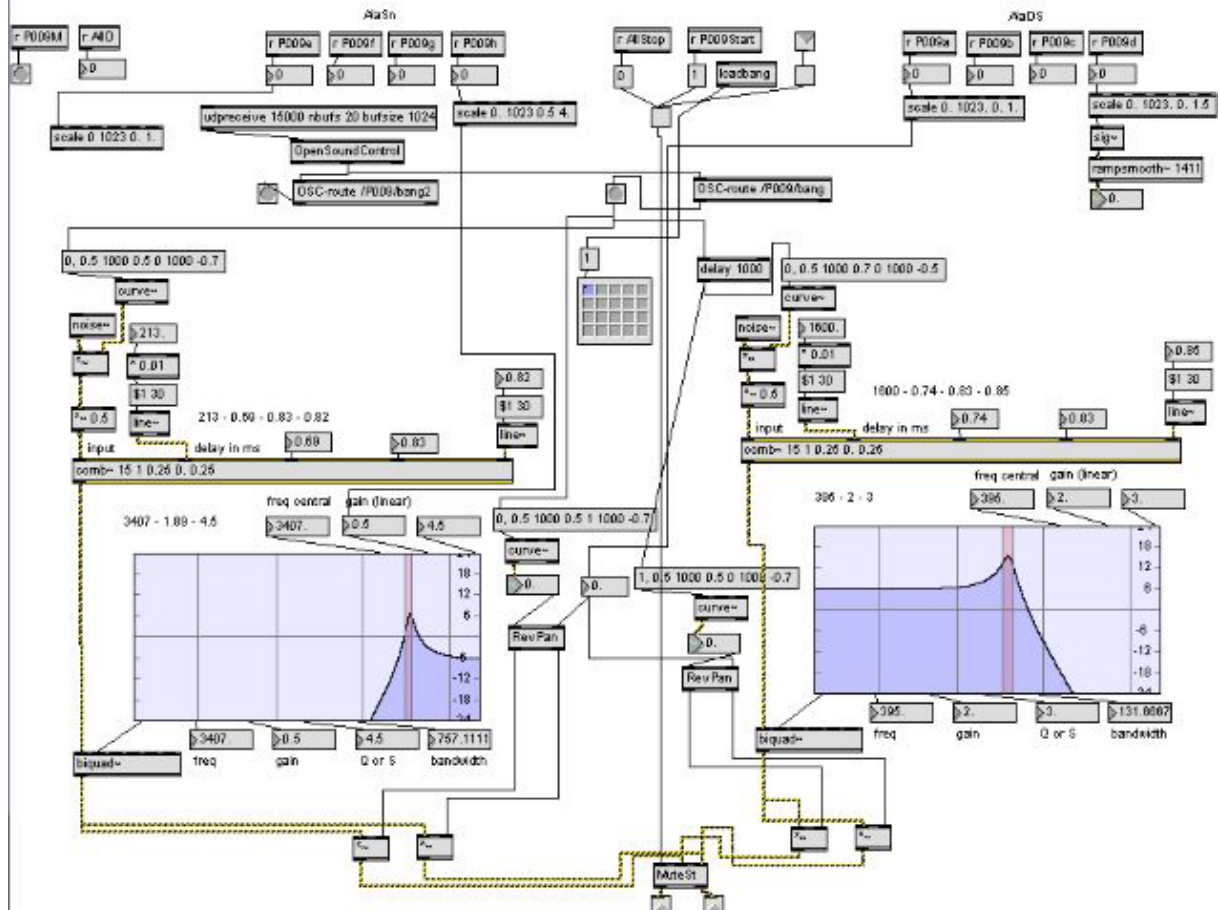
if(DuploP1 == true){
    n1 = n1 + 1; //contatore, n1 = 0
    if ((n1%4)==1) {
        duplicateMovieClip("P1", "p1d" + n1,this.getNextHighestDepth());
        duplicateMovieClip("P3", "p3d" + n1,this.getNextHighestDepth());
    }
    if (n1 >= RipetizioniP1) {
        removeMovieClip("p1d" + n1);
        removeMovieClip("p3d" + n1);
        n1--;
    }
}
else{
    if (n1 > 0){
        if ((n1%4)==1) {
            removeMovieClip("p1d" + n1);
            removeMovieClip("p3d" + n1);
        }
        n1--;
    }
}
}
}

```





L'algorithmo audio utilizza due file audio realizzati con un altro algorithmo



Questo è basato su 2 generatori di noise filtrati con 2 **comb** con tagli diversi (visibili nei **graph**).

I file creati vengono gestiti dal primo algorithmo con degli involucri "curvilinei" **curve** ed i bilanciatori reversibili **revpan** illustrati in precedenza. Il loro triggering è dato dai messaggi bng provenienti dal filmato swf, mentre la spazializzazione direttamente ed in parallelo con Flash da LittleWings. Il Potenziometro S7 (ala sinistra) regola sul finire della scena la trasparenza dei fasci di luce, che in Max vengono realizzati da modifiche sulla BW dei filtri utilizzando files wave precedentemente realizzati.

### 5.3.5 Sfere

Dal buio emergono sfere rotanti con colori cangianti. Le 4 dominanti cromatiche sono associate a 4 gruppi di filtri che agiscono su file realizzati in precedenza con gli stessi filtri.

Dopo innumerevoli tentativi di generare sinteticamente la grafica delle sfere, si è optato per l'utilizzo di una tecnica storica che fa uso di bitmap: una maschera circolare anteposta ad un più grande settore circolare riempito con una texture a simmetria centrale multi colore. LittleWings agisce parametricamente sulla trasparenza di quattro sfere aventi velocità direzioni e texture diverse, rendendo una gassosità delle luci esaltata da un'altra maschera a settore circolare che sfuma l'oggetto sferico verso il nero di sfondo.

## L'ActionScript

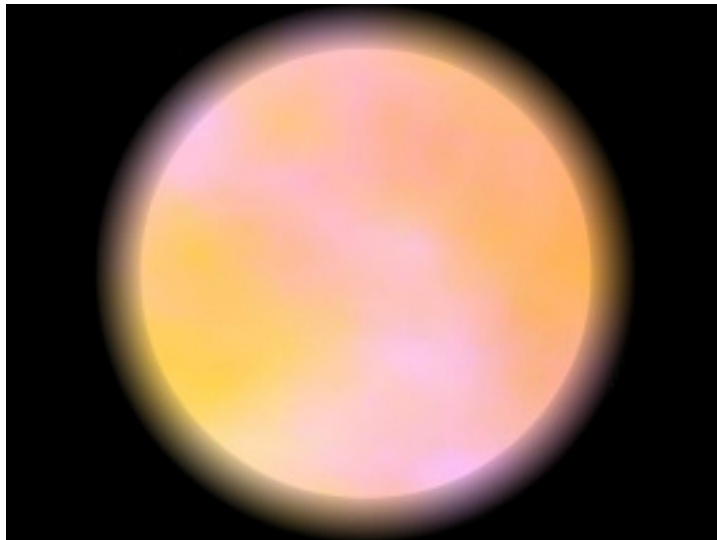
```
import flash.display.*;
//Mouse.hide ();
var alfagreen;
var alfared;
var alfablu;
var alfayellow;
var frame:Number =0;
var testframe:Number;
onEnterFrame = function (){
    if( _global.address0 == "/analogin/0/value" )
        {
            alfagreen = _global.trimpot0 / 1023;
        }
    if( _global.address2 == "/analogin/2/value" ){
        alfared = _global.trimpot2 / 800;//scarsa risposta!
    }
    if( _global.address3 == "/analogin/3/value" )
        {
            alfayellow = _global.trimpot3 / 1023;
        }
    if( _global.address4 == "/analogin/4/value" )
        {
            alfablu = _global.trimpot4 / 1023;
        }

    //green
    setProperty("ring1", _rotation, (getProperty(ring1,_rotation)+0.5));
    setProperty("ring1", _alpha, (alfagreen*100));

    //red
    setProperty("ring2", _rotation, (getProperty(ring2,_rotation)-0.7));
    setProperty("ring2", _alpha, (alfared*100));

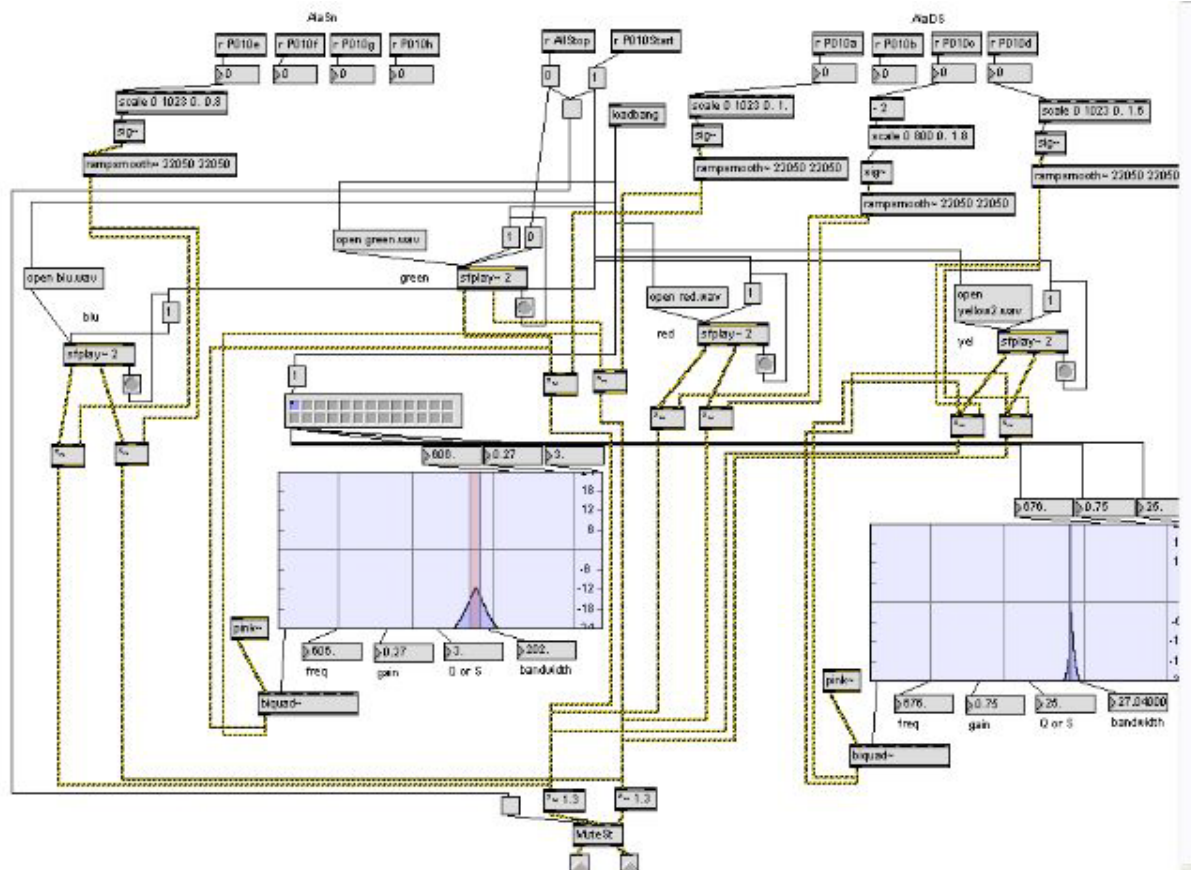
    //blu
    setProperty("ring3", _rotation, (getProperty(ring3,_rotation)-0.2));
    setProperty("ring3", _alpha, (alfablu*100));

    //yellow
    setProperty("ring4", _rotation, (getProperty(ring4,_rotation)+1));
    setProperty("ring4", _alpha, (alfayellow*100));
}
```



In Max la patch si occupa del filtraggio e del missaggio dei 4 formanti, anch'essi realizzati in precedenza come file waves per non sovraccaricare il processore.





### 5.3.6 Matrix 4D

Una matrice grafica di 16x12 istanze (4 colonne per 4 ripetizioni e 4 righe per tre ripetizioni) di un breve filmato di fade-in-out bicromatico riempie lo schermo. Essa a sua volta è gestita da una matrice di 256 permutazioni che consentono di creare varie strutture simmetriche con la rotazione di 90° nei quattro sensi dei quadratini. Un'ulteriore rotazione generale e continua a 360° degli stessi consente una suppletiva risorsa alla performance.

La correlazione audio-grafica si basa sull'associazione percettiva della sfumatura cromatica verde-rosso come involuppo d'ampiezza se orizzontale e di filtraggio se verticale. Ne conseguono ritmi frenetici con pulsazione costante e durate ed attacchi diversi.

L'ActionScript

```

Mouse.hide();
var i=0;
var A:Array = new Array(256);// da 0 a 255
var B:Array = new Array(256);//
var C:Array = new Array(256);//
var D:Array = new Array(256);//
var light;
var ch:Array = new Array(4);//da 0 a 3
ch[0]=0;
ch[1]=90;
ch[2]=180;
ch[3]=270;

```

```

for(a=0;a<4;a++){
    for(b=0;b<4;b++){
        for(c=0;c<4;c++){
            for(d=0;d<4;d++){
                //ciclo della prima colonna
                //ciclo della seconda colonna
                //ciclo della terza colonna
                //ciclo della quarta colonna
            }
        }
    }
}

```

```

        A[i] = ch[a];
        B[i] = ch[b];
        C[i] = ch[c];
        D[i] = ch[d];
        i++;
    }
}
}
}
}
var ap=0;
var alphaA:Array = new Array(256);
var alphaB:Array = new Array(256);
var alphaC:Array = new Array(256);
var alphaD:Array = new Array(256);
var alphaH:Array = new Array(4);

alphaH[0]=25;
alphaH[1]=50;
alphaH[2]=75;
alphaH[3]=100;
for(a=0;a<4;a++){
    for(b=0;b<4;b++){
        for(c=0;c<4;c++){
            for(d=0;d<4;d++){
                alphaA[ap] = alphaH[a];
                alphaB[ap] = alphaH[b];
                alphaC[ap] = alphaH[c];
                alphaD[ap] = alphaH[d];
                ap++;
            }
        }
    }
}
function ABCD(){
    for (column = 1; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, A[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaA[ap]);
    }
    for (column = 2; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, B[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaB[ap]);
    }
    for (column = 3; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, C[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaC[ap]);
    }
    for (column = 4; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, D[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaD[ap]);
    }
}
function DCBA(){
    for (column = 1; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, D[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaD[ap]);
    }
    for (column = 2; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, C[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaC[ap]);
    }
    for (column = 3; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, B[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaB[ap]);
    }
}

```

```

    }
    for (column = 4; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, A[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaA[ap]);
    }
}
function BDAC(){
    for (column = 1; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, B[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaB[ap]);
    }
    for (column = 2; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, D[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaD[ap]);
    }
    for (column = 3; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, A[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaA[ap]);
    }
    for (column = 4; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, C[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaC[ap]);
    }
}
function CBDA(){
    for (column = 1; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, C[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaC[ap]);
    }
    for (column = 2; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, B[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaB[ap]);
    }
    for (column = 3; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, D[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaD[ap]);
    }
    for (column = 4; column < 17; column = column +4){
        setProperty(column + "a" + riga , _rotation, A[i] + pan);
        setProperty(column + "a" + riga , _alpha, alphaA[ap]);
    }
}

onEnterFrame = function(){
    //main routine
    if( _global.address7 == "/analogin/7/value" )
    {
        i = Math.floor(_global.trimpot7 / 1023 * 255);//in Max Scale restituisce int <= (floor)
    }
    if( _global.address4 == "/analogin/4/value" )
    {
        light = Math.round((_global.trimpot4 ) / 1023 * 100);
    }
    if( _global.address6 == "/analogin/6/value" )
    {
        ap = Math.round((_global.trimpot6 - 2) / 800 * 4 + light);
    }
    if( _global.address0 == "/analogin/0/value" )
    {
        pressure = Math.ceil(_global.trimpot0 / 1023 * 4);//int >= x (da 0 a 3)
    }

    if( _global.address3 == "/analogin/3/value" )
    {

```

```

        pan = Math.floor(_global.trimpot3 / 1023 * 90);
    }
    switch (pressure){
    case 0:
        for (riga = 1; riga < 13; riga++){
            ABCD();
        };
        break;

    case 1:
        for (riga = 1; riga < 13; riga = riga +2){
            ABCD();
        }
        for (riga = 2; riga < 13; riga = riga +2){
            DCBA();
        };
        break;

    case 2 :
        for (riga = 1; riga < 13; riga = riga +3){
            ABCD();
        }
        for (riga = 2; riga < 13; riga = riga +3){
            DCBA();
        }
        for (riga = 3; riga < 13; riga = riga +3){
            BDAC();
        };
        break;
    case 3 :
        for (riga = 1; riga < 13; riga = riga +4){
            ABCD();
        }
        for (riga = 2; riga < 13; riga = riga +4){
            DCBA();
        }
        for (riga = 3; riga < 13; riga = riga +4){
            BDAC();
        }
        for (riga = 4; riga < 13; riga = riga +4){
            CBDA();
        };
        break;
    default :
        for (riga = 1; riga < 13; riga = riga +4){
            ABCD();
        }
        for (riga = 2; riga < 13; riga = riga +4){
            DCBA();
        }
        for (riga = 3; riga < 13; riga = riga +4){
            BDAC();
        }
        for (riga = 4; riga < 13; riga = riga +4){
            CBDA();
        };
        break;
    }
}
}

```



### **5.3.7 Finale**

Il finale è rappresentato da una scena del primo tipo, cioè completamente definita e senza intervento di LittleWings, dove il volar via della matrice bicromatica rivela lo svuotamento completo della scatola di cioccolatini, accompagnata dal “solo” di Hendrix alla chitarra che chiude “Little Wing” ed in questo caso anche “Little Wings”.

## **6 – PERFORMANCE**

## 6.1 Boot sequence

E' doveroso rispettare la sequenza sottoindicata, onde garantire il corretto boot del sistema.

1. MIDI pedal
2. AudioBoard
3. MakeCtrl ON [connection LAN]
4. Flosc bat java server
5. Max-MSP load patch
6. SFW init

## 6.2 Setup

### 6.2.1 Setup Laptop

1. Disabilitare stand-by
2. Copiare la cartella "**Butterfly/MaxMSP/LittleWings**" e metterla come prima preferenza nelle impostazioni files di MAX!
3. Creare la cartella **Butterfly/Flash/LittleWings/files**
4. Impostare risoluzione schermo 800\*600

### 6.2.2 Setup iniziale

1. Montaggio Butterfly e connessione mini-XLR 8 sensori
2. Connessione LAN Butterfly-PC
3. Connessione Videoproiettore-PC
4. Connessione PC-card RME \_ RME board Multiface
5. Connessione Optical scheda audio Mixer
6. Connessioni Mixer – Amplificatore – Altoparlanti
7. Connessione RME board Multiface – Pedaliera MIDI
8. Alimentazione ON pedaliera MIDI
9. Pulsante UP Pedaliera MIDI (bank 5\_9)
10. Alimentazione Videoproiettore
11. Alimentazione PC e verifica riconoscimento videoproiettore e RME
12. Mixer RME e Setup RME
13. PC Load Max-MSP GeneralPatch (Runtime e .exe per minimizzare carico CPU)
14. Controllare In Max Dsp Vector Size [64]
15. Avvio FloscRcv35k-snd15k.bat su PC
16. Alimentazione Butterfly – accensione interruttore e controllo avvenuta connessione di rete Make-PC. **Azzerare an 3 e an 7.**
17. Load PatchFlashInit.swf e maximize [**CTRL+F**]
18. Da questo punto tutte le operazioni sono MIDI controlled
19. Push 6 [Metro]
20. Verifica e calibrazione sensori
21. Alimentazione Mixer
22. Alimentazione Amplificatore (altoparlanti autoamp.)
23. Load Max utility Extras/Audiotester e verifica uscite e livelli audio
24. Verifica uscita video



## 6.3 Standby

### 6.3.1 Standby system (30'-60')

1. Stop Max audio [**push 2/7**] e Metro [**1/6**] general patch, non chiudere l'exe.
2. Chiudere swf general 00
3. Chiudere bat Flosc
4. Off Little Wings e staccare alimentatori suo e MIDIpedal dalla ciabatta (disconnessione soft rete LAN)
5. (Mixer ON, Laptop ON, VideoPro ON, AMP ON)

### 6.3.2 Wake-up system

1. **open utility Max** (extras / audio tester) per verificare i canali LR e l'amp e mixer totale. Quindi chiuderla! Se l'audio non esce dagli altoparlanti in modo corretto
  - verificare livello uscita Master Out del Mixer Yamaha, se presente rivolgersi al service
  - se non presente, verificare livello uscita video e audio con cuffia presa cuffia RME dal mixer RME [controllare PAN canali asioPlay 1 e 2]
  - se non presente verificare apertura DAC sul Audiotester di MAX
  - altrimenti chiudere e riavviare il sistema
2. Rimettere gli alimentatori nella ciabatta Little Wings e MIDIpedal
3. Pulsante UP Pedaliera MIDI (bank 5\_9)
4. Little Wings ON (attendere connessione soft LAN )
5. Open bat Flosc (attendere apertutra porte)
6. Max general patch
7. Open swf general 00
8. Push 1/6 [Metro ON]
9. Verificare SWF00 riceve segnali analogici e MIDI e se occorre calibrare an0 e an4
10. Push on MIDIpedal #7 (screen data off)

## 6.4 Problemi e soluzioni

1. Crackle audio
  - verifica vector size del buffer DSP in Max = 64
2. MIDI program change 300 non funziona, ma viene ricevuto da flash e si vede
  - E' rimasta aperta in Max un'altra patch con lo stesso messaggio on/off alternato
3. Flash non riceve i dati sensore
  - Controllare connessione e cavo LAN
  - Controllare connessioni sensori
4. an1 o an5 danno valori incontrollati, come se restasse schiacciato
  - allontanare da sopra **F** con screw linea

## **6.5 Utilizzo di LittleWings**

### **6.5.1 Montaggio**

I due supporti dei sensori vanno inseriti negli otto bulloni sui quali vanno avvitati i dadi ciechi. Quindi vanno inserite le spine dei sensori nelle prese Mni XRL nell'ordine siglato sia sulle spine che sulle prese. Infine va inserita la spina dell'alimentazione e il cavo RJ45.

### **6.5.2 Calibrazione**

Portare in senso antiorario a 0 entrambi i potenziometri.

Con un giravite a croce minuto stringere le viti preposte per diminuire la pressione e allentarle per aumentarla. Detta operazione va effettuata alternando i giri tra la vite destra e sinistra, non collegate tra loro, onde evitare eccessive pendenze della lamina che sorregge F e S1/5.

### **6.5.3 Esecuzione**

I sensori lineari vanno attuati mediante l'uso delle dita pollice ed indice delle mani, principalmente, lasciando le altre libere per agire sulle ali e fletterle.

Passaggi del pollice sono utili per azionare contemporaneamente al sensore lineare anche il potenziometro posto al lato. Nonostante la corretta finestratura, data la possibilità di minimi spostamenti laterali del sensore lineare all'interno della sua gabbia, è preferibile evitare i bordi laterali.

Nelle patches che non prevedono fadeout e grabbing automatici, lasciare ed avvicinare i sensori lineari sempre dal basso.

## **7 – CONCLUSIONI**

## 7.1 Work in progress

La performance di "Lillte Wings", ritardata da un black-out locale, è stata portata a termine senza crash di sistema e con buon gradimento di pubblico.

Essendo un work in progress soprattutto compositivo ci si sente assillati dall'esigenza di sfruttare tutte le potenzialità del sistema ma si è un po' atterriti dall'immane lavoro necessario.

## 7.2 Miglioramenti e futuri sviluppi

L'interfaccia **LittleWings** ha bisogno di essere sistemata in modo da limitare ed eliminare le pressioni parassite sul sensore **S2**.

E' pensabile incrementare le utility con una nuova **MakeControllerKit** cui affidare semplici interruttori con funzioni di ulteriori variazioni interne agli algoritmi o di riscaldamento dei range applicati o anche di diversi indirizzamenti degli stessi parametri.

## 7.3 Mandala o 3D?

Si è già iniziato un lavoro di ideazione su una prospettiva diversa del sistema, basata non sulla creazione di oggetti evocanti spazi tridimensionali, ma su visioni del tipo sinestetico, simili alle realizzazioni elencate in 1.4.1.4 Sinestesia ed informatica.

La figura del mandala con le sue simmetrie radiali si presta ad una concentrazione sulla varianza cromatica e luminosa piuttosto che sull'identificazione di forme realistiche. L'utilizzo delle figure di Chladni con la visualizzazione dei nodi e dei ventri è risultato deludente e poco significativo ed è quindi da considerare impraticabile.

Resta la difficoltà di relazione spettrale audio-grafica ed il suo utilizzo in real-time con le attuali disponibilità di velocità di calcolo.

D'altra parte si indaga anche sul versante opposto che consiste nell'utilizzo di programmi alternativi per la sintesi video e grafica tridimensionale.

Un ulteriore campo d'indagine consiste nell'aprire il sistema all'utilizzo di strumenti e voci con altri interpreti, nonché con altri sistemi-strumenti come il lavoro di Luigino Pizzaleo su arpe di vetro sensorizzate e quello di Silvia Lanzalone, Beatrice Lasio, Teodoro Pace e Luigi Di Giampietro.

Già è in costruzione un progetto di riunione delle forze che si interessano a tali problematiche e che ha preso il nome di **GRAD: Gruppo Ricerche Analogico / Digitali**, e che vede i succitati artisti come co-fondatori.



Insomma lo strumento c'è, le idee e le possibilità non mancano ...

## **8 – RIFERIMENTI**

## 8.1 BIBLIOGRAFIA

I. Newton – Optics (1704)  
S. Freud - Sogno ipnosi e suggestione (vv.1888-1913)  
Mendoza - Audition Colorée (1890)  
C.G.Jung - Psicologia analitica (conferenze 1935 -prima ed. 1969)  
W. Metzger - I fondamenti della psicologia della Gestalt(1941)  
D. Katz - La psicologia della forma (1950)  
K. Lorenz - L'anello di re Salomone (1967)  
A.A.V.V - La comunicazione non verbale nell'uomo ( R. A. Hinde 1972)  
A.A.V.V. - teorie dell'evento (E.Morin 1972)  
S.Rose - Il cervello e la coscienza (1973)  
Lawrence Marks - The Union Of The Senses (1979)  
E. Bouguignon - Antropologia psicologica (1979)  
Cytowic - Synaesthesia, a union of the senses (1989)  
Id - The man who tasted shapes(1988)

U. Eco - il nome della rosa  
U. Galimberti - Il corpo  
E. Fubini - Musica e linguaggio nell'estetica contemporanea(1972)  
J.Blauert - Spatial Hearing(1974-1983-1997)  
P. Boulez - Punti di riferimento (1981)  
E.Zwicker H.Fastl - Psycho-acoustics (1990)  
S. McAdams - Musica una scienza della mente? (Musica e scienza 1991)  
M. Lupone - Ricerca: attitudine musicale - Metodo scientifico (Musica e scienza 1991)  
W.M. Hartmann - Signals, sound and sensation(1998)

## 8.2 LINKS

Lemur, interfaccia virtuale su touchscreen	<a href="http://www.jazzmutant.com">http://www.jazzmutant.com</a>
Eyes Web	<a href="http://www.infomus.dist.unige.it/">http://www.infomus.dist.unige.it/</a>
Studio 1 Licia Galizia, mus. Michelangelo Lupone	<a href="http://www.liciagalizia.it/">http://www.liciagalizia.it/</a>
Matilde Battistini	
Riflessioni su Vasilij Kandinskij e la musica	<a href="http://users.unimi.it/~gpiana/dm3/dm3kanmb.htm">http://users.unimi.it/~gpiana/dm3/dm3kanmb.htm</a>
cinema sperimentale	<a href="http://www.muspe.unibo.it/corso/annate/04-05/cinema/boarini1.rtf">http://www.muspe.unibo.it/corso/annate/04-05/cinema/boarini1.rtf</a>
Viking Eggeling	<a href="http://en.wikipedia.org/wiki/Viking_Eggeling">http://en.wikipedia.org/wiki/Viking_Eggeling</a>
FSR Guide	<a href="http://www.media.mit.edu/resenv/classes/MAS836/fsrguide.pdf">http://www.media.mit.edu/resenv/classes/MAS836/fsrguide.pdf</a>
Touch v1.4	<a href="http://infusionsystems.com/">http://infusionsystems.com/</a>
Bend sensor, potentiometer	<a href="http://www.makingthings.com">http://www.makingthings.com</a>
OSC	<a href="http://cnmat.berkeley.edu/publication/open_sound_control_new_protocol_communicating_sound_synthesizers">http://cnmat.berkeley.edu/publication/open_sound_control_new_protocol_communicating_sound_synthesizers</a>
OSC objects per Max-MSP	<a href="http://cnmat.berkeley.edu/downloads">http://cnmat.berkeley.edu/downloads</a>
FLOSC	<a href="http://www.benchun.net/flosc/">http://www.benchun.net/flosc/</a>