

**STRUKTURES: INSTALLAZIONE IMMERSIVA. ESECUZIONE DI STRUTTURE****STRUKTURES: IMMERSIVE INSTALLATION. EXECUTION OF STRUCTURES**

PIETRO NACCA, DAVIDE MAIO, DYLAN IULIANO

**Abstract (IT):** ‘Struktures’ è un lavoro realizzato in due versioni. La prima è una performance eseguita in occasione dell’esame di “Esecuzione ed Interpretazione della Musica Elettroacustica” al Conservatorio N. Sala di Benevento. La seconda è un’installazione esposta al Museo del Sannio dal 25 al 31 Agosto 2022, in occasione di “Benevento Città Spettacolo”. ‘Struktures’ offre un’esperienza immersiva audiovisiva attraverso la quale è possibile generare sia il suono – utilizzando un ascolto in quadrifonia – sia la sua rappresentazione grafica mediante il movimento e l’utilizzo di controller. Il cuore del progetto è un computer che invia le informazioni ad un secondo computer, quest’ultimo incaricato di elaborare i dati e trasformarli in suono e in immagine.

**Abstract (EN):** ‘Struktures’ is a work that existed in two different versions. The first is a livenesset performed for the "Execution and Interpretation of Electroacoustic Music" exam at the N. Sala Conservatory of Benevento. The second is an installation exhibited at the Sannio Museum inside the Santa Sofia church from August 25 to 31 2022, during "Benevento Città Spettacolo". ‘Struktures’ offers an immersive audiovisual experience where it's possible to generate both sound – using quadraphonic listening – and its graphic representation through movement and the use of controllers. The core of the installation is a computer that gives informations to a second computer, the second computer is in charge of processing data and transforming them into sound and images.

**Keywords:** Max-MSP, Arduino, quadrifonia, installazione, performance, OSC.

**STRUKTURES: INSTALLAZIONE IMMERSIVA**  
**ESECUZIONE DI STRUTTURE**

PIETRO NACCA, DAVIDE MAIO, DYLAN IULIANO

L'idea del progetto immersivo è nata dalla necessità di realizzare un'esecuzione live di musica elettronica che comprendesse sia la parte sonora che la sua rappresentazione visiva. L'organizzazione del live è stata pensata per tre o più esecutori, grazie all'adozione di più controller sia già esistenti, sia costruiti appositamente per il progetto. Il nostro intervento, infatti, non è stato volto solo ad un adattamento delle possibilità performative offerte dai controller in commercio, ma anche alla progettazione, attraverso l'elettronica e l'informatica, di un controller *ex novo*. In un solo device abbiamo condensato le possibilità di controllo di parametri indirizzati alla gestione della generazione sonora, e le possibilità di performare questo controllo attraverso gesti che i normali fader o knob non permettono. Tra la vastissima gamma di device MIDI che offre il mercato, abbiamo sentito la pressione asfissiante dell'offerta delle numerose aziende, e ciò ci ha all'inizio spaesato. Non riuscivamo a trovare un nesso tra le tecnologie disponibili e il fine della nostra esecuzione. In un secondo momento siamo arrivati alla conclusione che per sfruttare al meglio le possibilità performative avevamo bisogno di pochi ingredienti, ma che ci rendessero coscienti delle caratteristiche da impiegare e controllare. Costruire ciò di cui avevamo bisogno è stato il passo decisivo per poter plasmare alle nostre esigenze la tecnologia, diventando sia artefici che interpreti dei device. «La padronanza dei propri mezzi; chiedere ad essi - come succedeva ai maestri del passato - di occuparsene in “prima persona”; spingere il musicista ad acquisire una “competenza [...] delle proprie tecnologie”, che gli consenta di diventare “*interprete* della tecnologia in senso

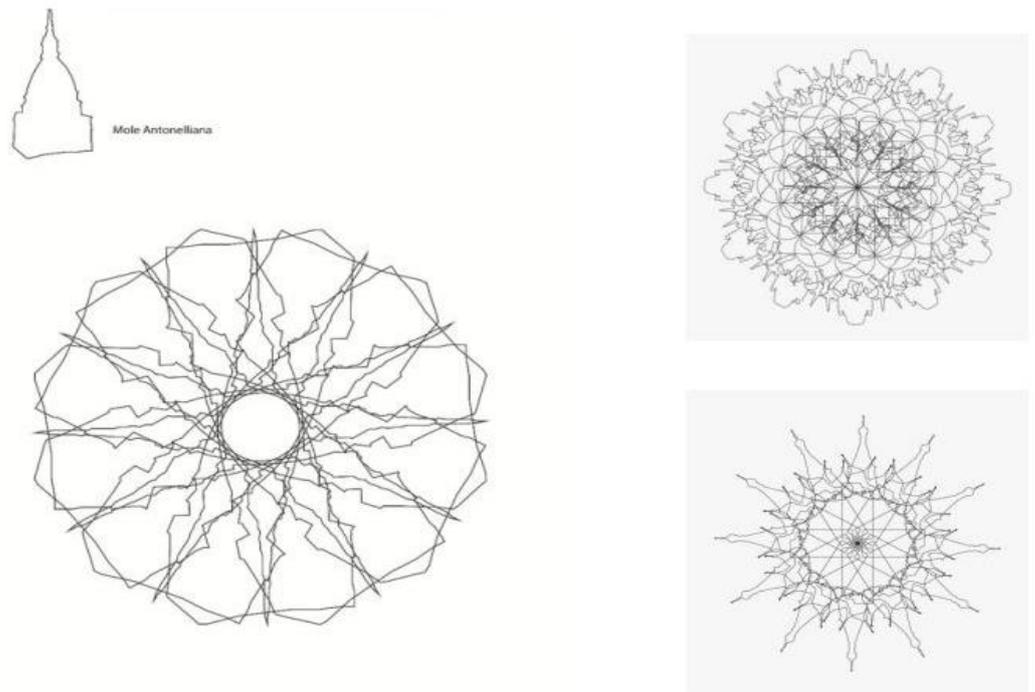
**[divulgazione audiotestuale]**

**STRUKTURES: INSTALLAZIONE IMMERSIVA**

generale”» (Di Scipio, 2012). Il device che abbiamo costruito è fornito di 4 fader e due sensori di prossimità. Per poter convertire i segnali analogici che questi componenti elettronici generano, abbiamo usato Arduino, che tramite porta seriale comunica con il computer. In più abbiamo utilizzato sia i nunchuck<sup>1</sup> sia un mixer multicanale. I nunchuck comunicano con il computer attraverso il bluetooth, mentre il mixer multicanale viene adoperato sia per il missaggio dei suoni, sia per la spazializzazione quadrifonica nell’ambiente. Il flusso che porta al risultato sonoro finale parte dagli esecutori che utilizzano i vari device, i quali comunicano con il computer di controllo (CdC); poi i dati ricevuti vengono indirizzati tramite il protocollo OSC al computer di generazione (CdG), il quale smista i dati per controllare video e audio parallelamente (sia per una questione di gestione delle risorse più efficiente, sia per una questione logistica); infine i segnali sonori prodotti dal CdG vengono missati dall’esecutore che controlla il mixer e inviati ai quattro speaker per ottenere la quadrifonia, mentre il segnale video viene proiettato da uno schermo. Gli esecutori si dispongono all’interno dell’area formata dai quattro speaker, rivolti di fronte lo schermo. La performance, quindi le modalità di utilizzo dei controller, viene suggerita da una partitura grafica, la quale si basa sui disegni di Graziana Zampetti e Gianna Casiello, artiste digitali Beneventane. Queste tavole sono rappresentazioni geometriche e bidimensionali di monumenti da tutto il mondo. La legenda della partitura è semplicemente un insieme di suggestioni utili agli esecutori e basate su vari effetti visivi che si susseguono durante lo score come il tremolio, il motion blur, la rotazione, la velocità e i colori.

<sup>1</sup> I nunchuck sono dei controller senza fili creati per la Nintendo Wii.

## STRUKTURES: INSTALLAZIONE IMMERSIVA

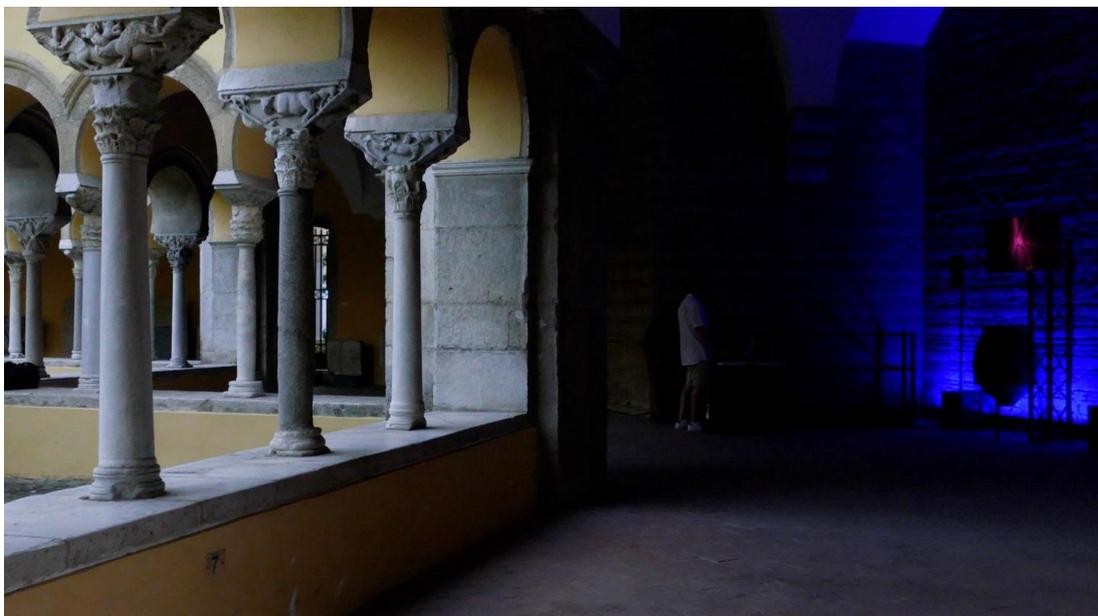


[Fig. A. Esempi tratti dalla partitura grafica.]

Questo principio generativo di gesti esecutivi è stato suggerito durante le lezioni del corso di Esecuzione ed Interpretazione della Musica Elettroacustica, nelle quali abbiamo affrontato varie tipologie di live electronics e in generale di esecuzioni musicali, tra cui l'approccio adoperato da Karlheinz Stockhausen per l'improvvisazione di gruppo. «Ho scritto dei testi per dirigere l'improvvisazione, perché ho trovato che qualche volta con poche parole si stimolano i musicisti a fare qualcosa di molto puro, che contiene pochissimi cliché; suonando *Aus den Sieben Tagen* abbiamo trovato che esiste un feedback tra i musicisti, è come una spirale» (Mariotti, 2013). La Musica Intuitiva di Stockhausen è la base fondante della scelta di più interpreti, i quali sono chiamati a performare la stessa partitura in maniera personale, navigando nel margine generato dalle indicazioni aperte. Partendo dall'impostazione creata per la performance live per tre esecutori, abbiamo ottenuto una versione che rientrasse nei parametri dell'installazione vera e propria, la quale è

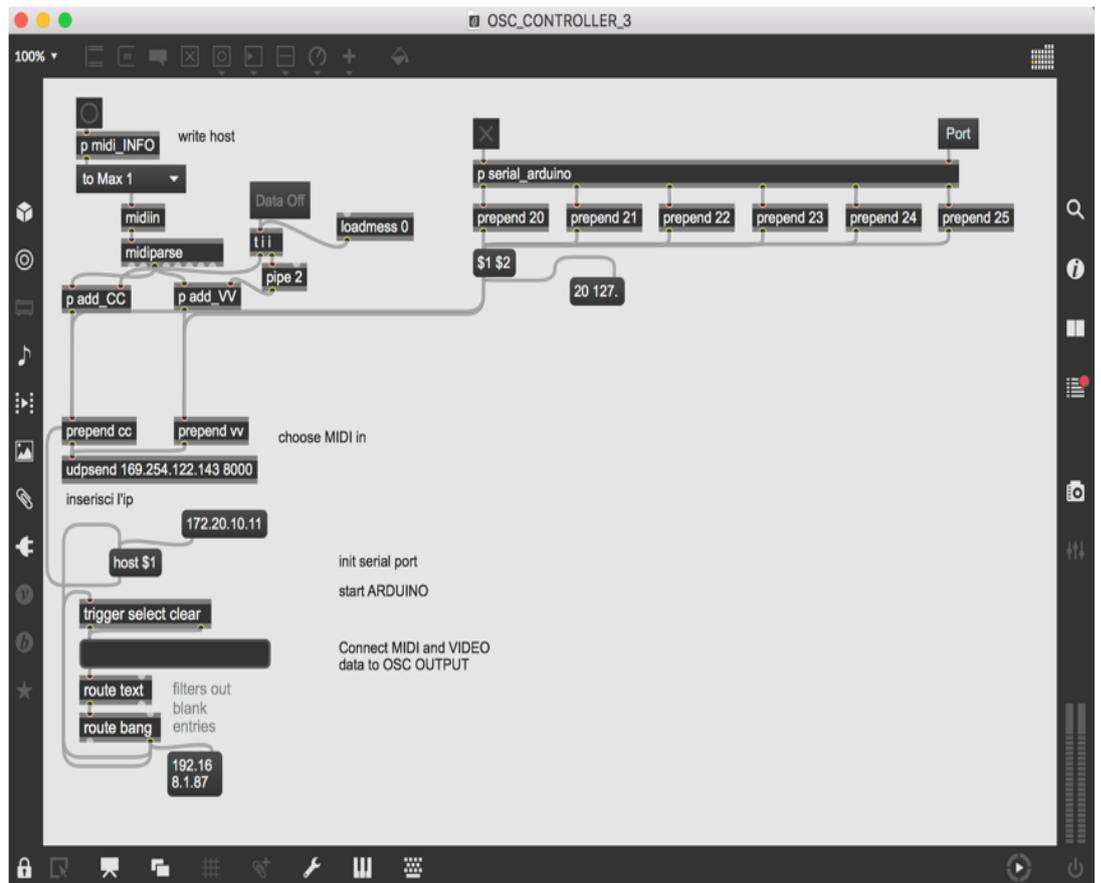
**STRUKTURES: INSTALLAZIONE IMMERSIVA**

stata esposta al Museo del Sannio dal 25 al 31 Agosto 2022, in occasione di “Benevento Città Spettacolo”.



[Fig.1] Corridoio del chiostro nel Museo del Sannio, sulla destra l’installazione Struktures.

Questa versione prevedeva l’utilizzo dei controller e del mixer da parte del pubblico (i visitatori del museo), invitato ad interagire. Il loro intervento era libero, senza l’utilizzo della partitura usata per l’esame, ed era volto ad esplorare le possibilità generative audiovisive offerte dall’installazione. L’interpretare le tecnologie, come suggerisce Di Scipio, non si esaurisce in questo progetto con la creazione di device usando l’elettronica, ma si esprime anche mediante l’utilizzo di software di programmazione che processano i dati dei controller.

STRUKTURES: *INSTALLAZIONE IMMERSIVA*

[Fig.2. Patch di Max-MSP che controlla il flusso di dati sul CdC.]

Tramite Max-MSP<sup>2</sup> abbiamo creato degli algoritmi che ci hanno permesso di leggere i dati provenienti dalla porta seriale di Arduino, smistarli in due categorie di parametri (video e MIDI) per poi inviarli tramite il protocollo OSC<sup>3</sup> al CdG. CdC e CdG

<sup>2</sup> Max-MSP è un ambiente di sviluppo grafico per la progettazione a oggetti di software dedicati ad applicazioni musicali e multimediali in tempo reale. È stato progettato e realizzato all'Ircam di Parigi da Miller Puckette. L'autore sottolinea lui stesso che Max nasce tra il 1980 e il 1990 al termine di una lunga serie di esperienze nell'ambito della progettazione informatica.

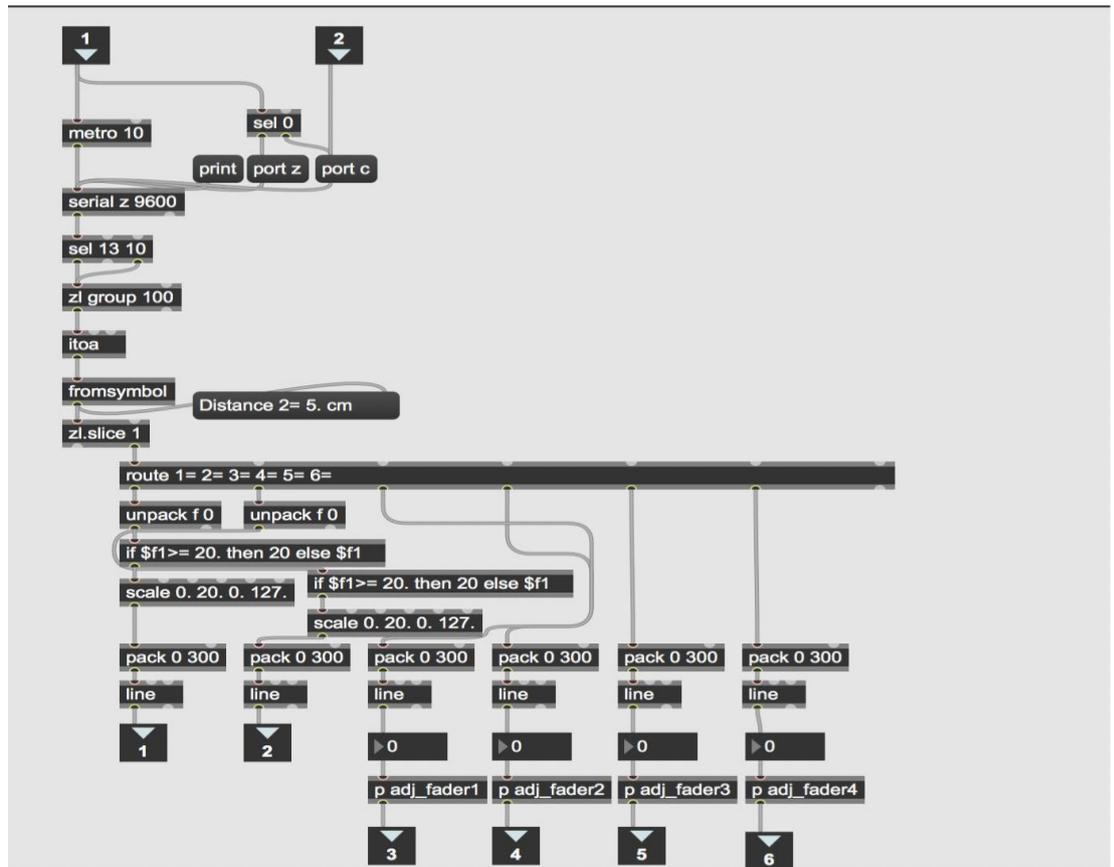
<sup>3</sup> Open Sound Control ("OSC") è stato creato dal CNMAT (Berkeley University) nel 1997 ed è un protocollo "open source" per la comunicazione tra computer, sintetizzatori audio e altri dispositivi

**STRUKTURES: INSTALLAZIONE IMMERSIVA**

comunicano tramite due patch create con Max-MSP. La *figura 1* mostra la patch eseguita dal CdC. Si possono individuare tre gruppi di oggetti. Il primo è composto da *midi\_INFO*, *midiparse* e *upsend*. La subpatch *midi\_INFO* permette la visualizzazione in un menu a tendina delle possibili scelte di routing dei messaggi MIDI, in modo tale da poter indirizzare, tramite *midiparse*, i CC provenienti da Osculator. Osculator è un software che permette di convertire i messaggi OSC, trasmessi da controller collegati via bluetooth al CdC, in messaggi MIDI, tramite un output MIDI dedicato. I Control Change (CC) verranno smistati in due subpatch che si occuperanno di indicizzare i CC per il MIDI (*add\_CC+prepend cc*) e per i controlli video (*add\_VV + prepend vv*); Infine i dati verranno inviati tramite *upsend* via protocollo UDP al secondo computer connesso alla stessa rete. Lo User Datagram Protocol (UDP), nelle telecomunicazioni, è uno dei principali protocolli di rete della suite di protocolli Internet. È un protocollo di livello di trasporto a pacchetto, usato di solito in combinazione con il protocollo di livello di rete IP. Il secondo gruppo di oggetti è composto da *textedit* e *host port*. L'oggetto *textedit* permette di scrivere l'IP della rete e inviarlo all'*udpsend*, la porta 8000 è fissata come default. Il terzo gruppo contiene gli oggetti dedicati alla decodifica dei dati sulla porta seriale inviati da Arduino nella subpatch *serial\_arduino*.

multimediali. OSC è progettato per supportare un'architettura client/server. I dati OSC sono trasmessi in unità denominate pacchetti. OSC è un protocollo di alto livello: non si occupa del meccanismo di trasmissione dei dati. La rete che trasporta un pacchetto OSC è responsabile della consegna sia del contenuto che del dato riguardante la sua grandezza. Un pacchetto OSC può essere rappresentato in un datagramma come un protocollo di rete UDP.

## STRUKTURES: INSTALLAZIONE IMMERSIVA

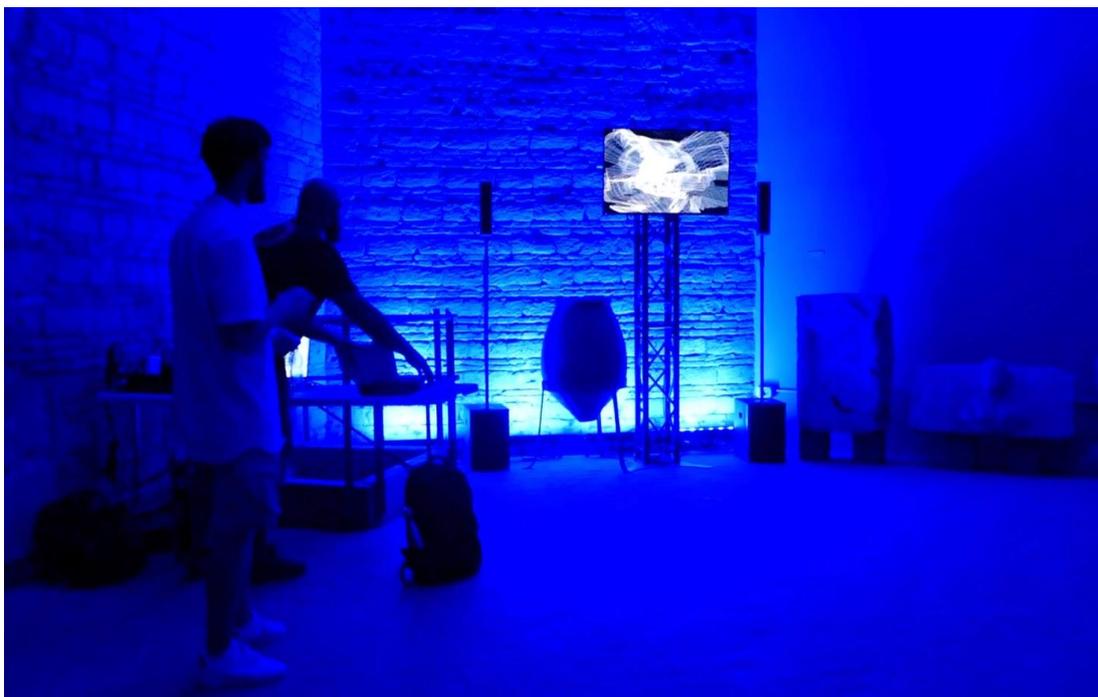
[Fig.3. Contenuto della subpatch *serial\_arduino*.]

La *figura 2* presenta il diagramma di flusso che va dall'acquisizione dei dati della porta seriale del CdC allo smistamento dei dati. L'oggetto *serial* ha bisogno di un bang<sup>4</sup> per poter mandare i dati in uscita, una volta impostato un *metro* a 10 ms, ogni 100 elementi ricevuti da *serial*, *zl.group* li manda in output come lista, tranne se riceve 13, in questo caso il *sel* manda un bang a *zl.group* e manda in output qualsiasi cosa abbia immagazzinato fino a quel momento.<sup>5</sup>

<sup>4</sup> Un "impulso" digitale.

<sup>5</sup> Un reset del processo di acquisizione dei dati, dovuta alle caratteristiche della trasmissione di Arduino.

## STRUKTURES: INSTALLAZIONE IMMERSIVA



[Fig.4. Utilizzo del pubblico dei controller (nunchuck utilizzati dall'esecutore all'estrema sinistra; sensori ad ultrasuoni innescati dalle mani del secondo esecutore) che cambiano forma alla sfera nello schermo.]

Poi i dati vengono convertiti in simboli e smistati in base agli indici prestabiliti. Infine abbiamo l'ottimizzazione del flusso di variabili ricevute tramite porta seriale. La subpatch *serial\_arduino* ci permette di leggere i dati stampati tramite Arduino IDE<sup>6</sup> nel Serial Monitor, di smistarli e ottimizzarli per poi inviarli tramite *udpsend* come messaggi MIDI e come controlli per il video; stesso procedimento usato per i dati di *midiparse*. Nello specifico abbiamo l'oggetto *serial* che riceve dati dalla porta seriale connessa in questo caso ad Arduino tramite USB. Le prime schede utilizzavano una porta seriale di quelle con il connettore a nove pin, una volta abbastanza diffusa su tutti i PC, ma ormai introvabile. La porta seriale è stata sostituita da una porta USB,

<sup>6</sup> Software dedicato per la programmazione del microcontrollore in dotazione con i vari device Arduino.

**STRUKTURES: INSTALLAZIONE IMMERSIVA**

che richiede un chip specializzato (FT232), impiegato nei primi modelli di Arduino e poi anch'esso rimpiazzato da un semplice microcontrollore ATMEL ATmega8U2, programmato opportunamente. Ricordiamo che stamperemo i valori dei fader e dei sensori tramite Serial Monitor per poter essere letti da Max e poter inviare i dati dei controlli tramite OSC al CdG. Per quanto riguarda la configurazione di Arduino come controller esterno abbiamo creato un apposito sketch, dal software dedicato Arduino IDE, che ci ha permesso di collegare 4 fader e 2 sensori di prossimità ad ultrasuoni. Nello specifico abbiamo usato per l'installazione Arduino Mega, una breadboard, diversi jumper, 4 fader, 2 sensori ad ultrasuoni, stagno, saldatore, case di plastica. Per utilizzare i due sensori ad ultrasuoni abbiamo incluso nello sketch la libreria dedicata NewPing. Il sensore HC-SR04<sup>7</sup> offre un range di misurazione da 2 cm a 400 cm (circa), con una accuratezza di 3mm. Il modulo comprende dei trasmettitori ad ultrasuoni, un ricevitore e un circuito di controllo.



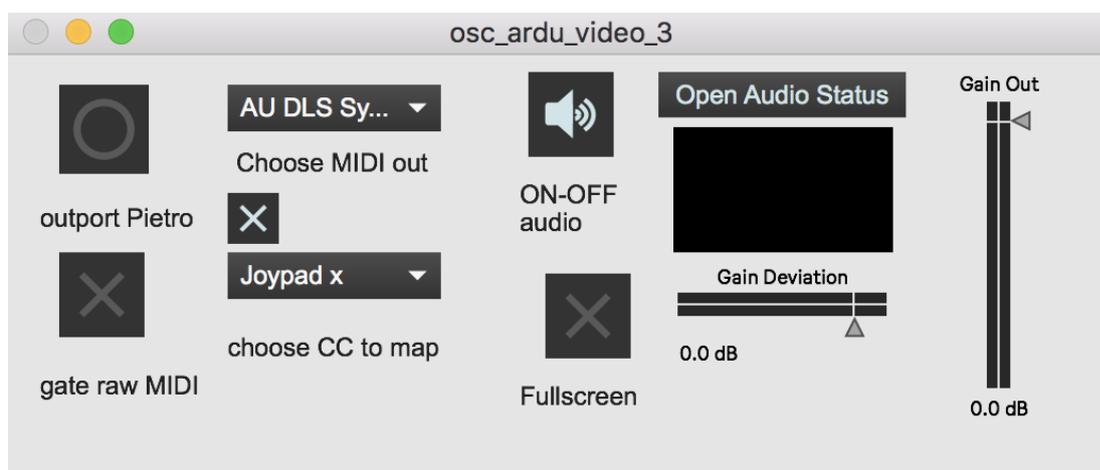
[Fig. 5 Sensore ad ultrasuoni.]

I dati riguardanti Arduino sono inviati verso il computer e si possono leggere aprendo il «Serial Monitor» di Arduino, cioè una finestra che visualizza i messaggi ricevuti (e

<sup>7</sup> Il sensore automaticamente invia otto onde di 40 kHz e rileva se ci sono impulsi di ritorno. La formula per calcolare la distanza in base al ritardo è: Test distance = (high level time×velocity of sound (340M/S) / 2.

## STRUKTURES: INSTALLAZIONE IMMERSIVA

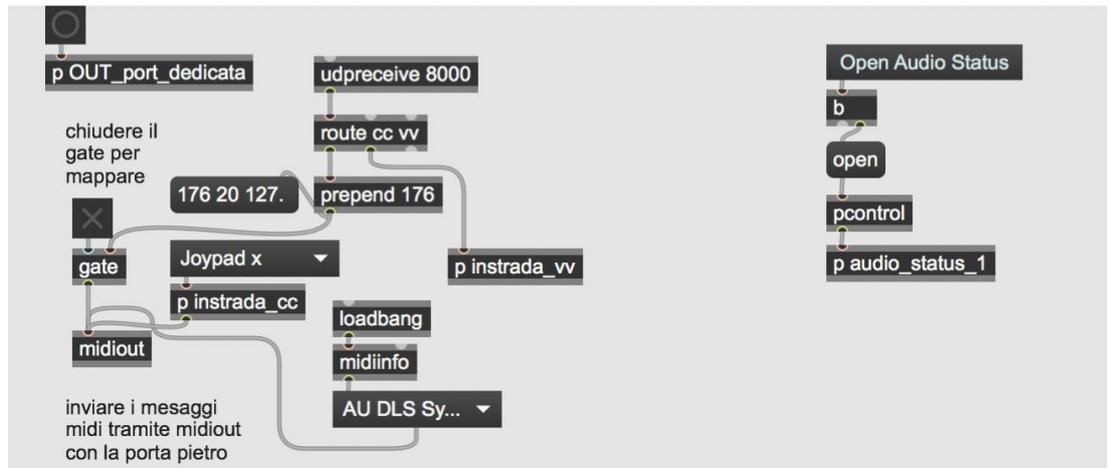
può anche trasmetterli). Una volta che i dati vengono organizzati e inviati dal CdC, il CdG riceve i messaggi, li smista e li invia alla DAW Ableton Live, tutto questo grazie alla patch di Max-MSP che vediamo in modalità presentazione in *figura 6*.



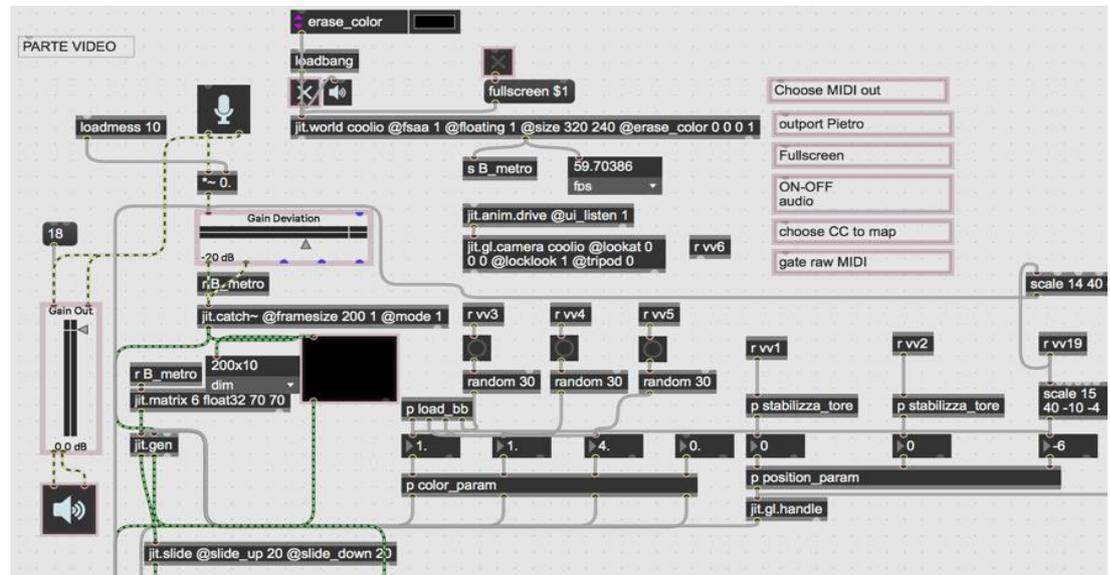
[Fig. 6. Patch di generazione video e invio di messaggi MIDI.]

Si può suddividere in due macro sezioni, quella di gestione dei dati e quella della generazione video. Dall'oggetto *udpreceive* passano tutti i dati inviati dal primo computer. Una volta arrivati, i dati prendono due strade diverse. Di nuovo, il flusso di dati si biforca in CC e VV, rispettivamente Control Change e parametri per controllare il video. L'oggetto *prepend* antepone 176 ai messaggi, in questo modo si indica al *midout* che deve inviare dei messaggi di Control Change al canale MIDI 1. Il flusso dei dati provenienti da Osculator, quindi dai sensori dei nunchuck, sono continui. Per poter essere in grado di "mappare" su Ableton ogni singolo parametro senza accavallamenti di CC sono stati immagazzinati in un oggetto *coll* (nella subpatch *instrada\_cc*) i CC disponibili, i quali sono richiamabili chiudendo il gate e selezionando il controllo desiderato dal menu a tendina.

STRUKTURES: INSTALLAZIONE IMMERSIVA



[Fig. 7. Sezione della gestione dati dal CdG.]



[Fig. 8. Sezione video della patch su CdG.]

È possibile selezionare l'uscita MIDI grazie all'oggetto *midiinfo* e il corrispettivo menu. In questo caso abbiamo creato un'interfaccia MIDI virtuale (nella subpatch *OUT\_port\_dedicata*), un routing interno del MIDI, grazie agli SM, ovvero i messaggi di sistema che Max ci permette di inviare. I CC vengono inviati ad Ableton Live che riceve i controlli, li applica alle clip MIDI, alle Drum Machine precedentemente

STRUKTURES: *INSTALLAZIONE IMMERSIVA*

organizzate e il segnale sonoro risultante viene smistato. Una copia del segnale rientra nella patch per modulare l'interferenza del video generato, mentre l'altra copia del segnale tramite la scheda audio collegata al CdG viene indirizzato al mixer multicanale per la spazializzazione. Il routing dei controlli per il video, invece, sono dei semplici *send* e *receive*. È possibile scegliere il driver audio, l'input e l'output device tramite la subpatch *audio\_status\_1*, cliccando su "Open Audio Status" si apre una finestra con i menu a tendina dedicati. La seconda macro sezione riceve i dati dal CdC e li usa per automatizzare la generazione del video. La parte video presenta come fulcro dello spazio di lavoro l'oggetto *jit.world* che è incaricato di creare l'ambiente per programmare video attraverso Jitter;<sup>8</sup> il segnale audio proveniente da Ableton Live viene convertito in matrice attraverso l'oggetto *jit.catch~*, che viene utilizzata per generare distorsioni. La matrice viene gestita dall'oggetto *jit.matrix*. L'oggetto *jit.gen*, invece, si occupa di creare la forma sferica di default, le distorsioni, modulazioni e i movimenti applicati alla sfera stessa. Per creare le distorsioni vengono moltiplicate due matrici tra di loro, tramite l'oggetto *jit.op*, per ottenere determinati effetti generati attraverso delle operazioni algebriche. Tutti i parametri video utilizzati per la rappresentazione visuale dell'esecuzione ricevono i dati dagli oggetti *receive* precedentemente menzionati. L'esecuzione prevede un primo interprete che utilizza i *nunchuck* che invia dati al CdC per poi essere rielaborati sia per controlli MIDI che per controlli video. Il secondo interprete utilizza il device costruito per l'installazione adoperando i fader e i sensori di prossimità. I sensori vengono innescati attraverso le mani dell'esecutore. Anche in questo caso i dati vengono utilizzati sia per il video che per il MIDI. Infine un terzo interprete utilizza il mixer per la quadrifonia, gestendo segnale pulito e segnale riverberato provenienti da Ableton Live. La rappresentazione

<sup>8</sup> In Jitter, sezione relativa ai video in Max-MSP, troviamo funzioni che consentono di eseguire sintesi, analisi e modifica delle immagini.

STRUKTURES: *INSTALLAZIONE IMMERSIVA*

visuale degli eventi prevede la creazione di una sfera che può assumere diverse forme e diversi colori a seconda di come si utilizzano i controller.



[Fig. 9. La sfera in posizione di equilibrio.]



[Fig. 10. Controller costruito per Strucktures.]