



Universidad de Santiago de Chile
Facultad de Ingeniería
Departamento de Ingeniería Mecánica.
Automatización y Robótica.



Proyecto Robótica

Informe Avance Proyecto N°3

Profesor: Michael Miranda.

Alumnos: Iván Fernández

Claudio Canales D.



Índice

| | |
|-------------------------|---|
| Objetivo..... | 3 |
| Metas específicas | 3 |
| Procedimiento | 3 |
| Introducción | 4 |
| Desarrollo..... | 5 |
| Conclusión..... | 9 |
| Bibliografía | 9 |

Objetivo

- Estudiar acerca de los métodos de comunicación entre dispositivos o aplicaciones para en envío y recibo de datos

Metas específicas

- Analizar y comprender el protocolo OSC
- Aprender a utilizar el protocolo OSC en conjunto con el programa Processing
- Utilizar el protocolo OSC con Processing para una determinada aplicación (Programa musical Ableton Live)

Procedimiento

- Se estudia acerca de la historia del protocolo OSC y su inicial aplicación.
- En la página Github se buscan tutoriales y códigos ya hechos por estudiantes en Processing y se analiza el código modificándolo continuamente para obtener un mayor aprendizaje.



Introducción

Open Sound Control (OSC) es un protocolo para la comunicación entre computadoras, sintetizadores de sonido y otros dispositivos multimedia que está optimizado para la tecnología de red moderna. Trayendo los beneficios de la tecnología de red moderna al mundo de los instrumentos musicales electrónicos, las ventajas de OSC incluyen interoperabilidad, precisión, flexibilidad y una mejor organización y documentación.

La gran mayoría del software y dispositivo OSC utiliza la red Ethernet o WiFi para transmitir datos. Eso significa que los mensajes OSC se pueden enviar fácilmente de una máquina a otra, o en la misma máquina en todas las aplicaciones.

Para que esto funcione, uno necesita un cliente OSC y un servidor OSC. Un cliente OSC es un dispositivo que envía datos OSC a un servidor OSC. Un servidor OSC recibe datos entrantes en un puerto que posee. Un puerto está simplemente representado por un número. La mayoría del software compatible con OSC es tanto un servidor como un cliente, lo que significa que puede enviar datos y recibir datos.

Un mensaje OSC está compuesto de dos partes: la dirección y los argumentos. La dirección es el nombre del mensaje OSC (por ejemplo, "BRAZO DERECHO"), y los argumentos son la lista de valores suministrados con este mensaje (por ejemplo, COORDENADAS DEL BRAZO DERECHO X Y Z).

Características:

- Esquema de nomenclatura simbólica de estilo URL abierto y dinámico
- Datos de argumento numéricos simbólicos y de alta resolución
- Patrón de coincidencia de idioma para especificar múltiples destinatarios de un solo mensaje
- Etiquetas de tiempo de alta resolución
- "Bundles" de mensajes cuyos efectos deben ocurrir simultáneamente
- Sistema de consulta para conocer dinámicamente las capacidades de un servidor OSC y obtener documentación

Áreas de aplicación:

- Instrumentos musicales electrónicos basados en sensores / gestos
- Asignación de datos no musicales al sonido
- Control musical compartido de múltiples usuarios
- Interfaces web
- Rendimiento musical LAN en red
- Rendimiento de WAN y Telepresence
- Realidad virtual
- Envolviendo otros protocolos dentro de OSC

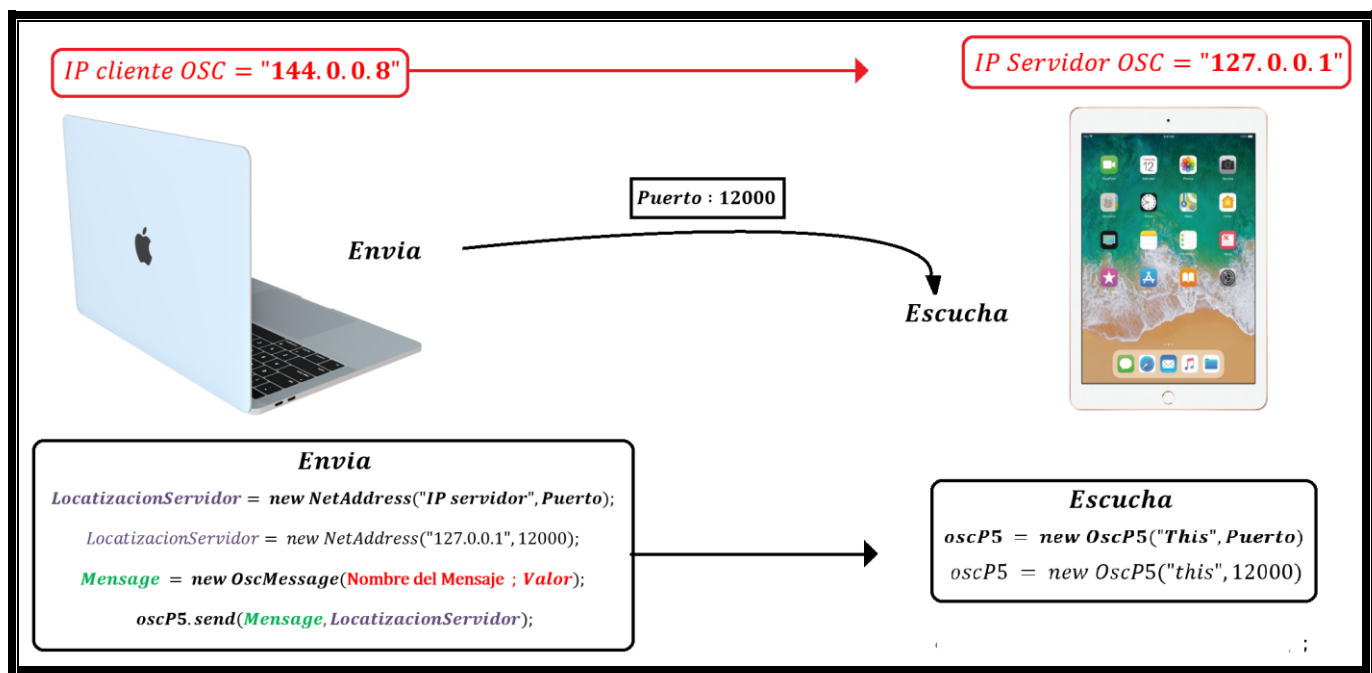
Desarrollo

Parte 1: Mensajes OSC solo con Processing

Para enviar un mensaje desde aplicaciones o dispositivos se deben realizar los siguientes pasos:

- Conocer la IP del servidor y el puerto por donde se enviara el mensaje
- Crear el mensaje con su Nombre y Valor (array, entero, string, etc)
- Enviar el mensaje creado a la localización del servidor
- Para que el servidor lo lea debe conocer la IP del Cliente y el puerto por donde envía el mensaje

A continuación se presenta un esquema representativo del protocolo OSC para enviar mensajes desde un computador a una Tablet:



Código ejemplo para enviar y recibir mensajes OSC al mismo computador al hacer clic con el mouse en una interfaz gráfica en Processing:

```
1  /**
2  * El ejemplo muestra cómo enviar y recibir mensajes de osc.
3  */
4  import oscP5.*;
5  import netP5.*;
6  OscP5 oscP5;
7  NetAddress myRemoteLocation;
8
9  void setup() {
10     size(400,400);
11     frameRate(25);
12     // * inicia oscP5, escuchando los mensajes entrantes en el puerto 12000 * /
13     oscP5 = new OscP5(this,12000);
14
15     /* myRemoteLocation es una dirección de red. una NetAddress toma 2 parámetros,
16     * una dirección IP y un número de puerto. myRemoteLocation se usa como parámetro en
17     * oscP5.send () cuando se envían paquetes de osc a otra computadora, dispositivo,
18     * solicitud. uso ver a continuación. para fines de prueba, el puerto de escucha
19     * y el puerto de la dirección de ubicación remota son iguales, por lo tanto,
20     * enviar mensajes de vuelta a este boceto.
21     */
22     myRemoteLocation = new NetAddress("127.0.0.1",12000);
23 }
24
25 void draw() {
26     background(0);
27 }
28
29 void mousePressed() {
30     /* en las siguientes formas diferentes de crear mensajes osc se muestran con el ejemplo */
31     OscMessage myMessage = new OscMessage("/test");
32     myMessage.add(123); /* agrega un int al mensaje osc */
33
34     /* Envía el mensaje */
35     oscP5.send(myMessage, myRemoteLocation);
36 }
37
38 /* incoming osc message se reenvía al método oscEvent. */
39 void oscEvent(OscMessage theOscMessage) {
40     /* imprime el patrón de dirección y la etiqueta de tipo del OscMessage */
41     print("### received an osc message.");
42     print("  addrpattern: "+theOscMessage.addrPattern());
43     println("  typetag: "+theOscMessage.typetag());
44 }
```

Parte 2: Mensajes OSC aplicados a programa Ableton Live más plugin Live Grabber

Ableton Live:

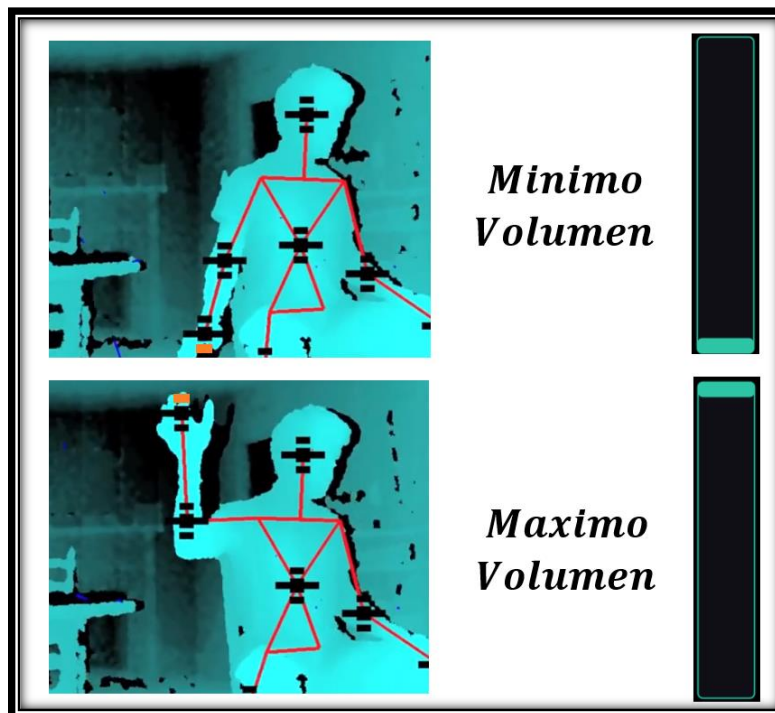
Ableton Live es un secuenciador de audio y MIDI, aplicación también conocida como DAW (Digital Audio Workstation) para el sistema operativo Windows y macOS. Live es un software rápido, fluido y flexible para la creación y el rendimiento musical. Viene con efectos, instrumentos, sonidos y todo tipo de características creativas, todo lo que necesitas para hacer cualquier tipo de música.

Live Grabber:

Los plugin de Live Grabber son un conjunto de complementos gratuitos de Max For Live que envían acciones desde Ableton Live a cualquier dispositivo de la red que admita Open Sound Control (OSC). A partir de la versión 3, Live Grabber también permite lo contrario: controlar Ableton Live con OSC.

Metodología:

En esta segunda parte del laboratorio se controla el volumen de una pista musical reproducida por Ableton Live. Para controlar el volumen se envían datos de números enteros de 0 a 100, donde con 0 no se escucha la pista y con 100 esta al máximo volumen. Estos datos son enviados por el protocolo OSC a un puerto en específico que lo lee Ableton a través de su plugin Live Grabber. Del programa Processing se envían los datos (0 a 100) a el puerto correspondiente, donde estos son recolectados de la posición de la mano derecha del esqueleto de un individuo. Este esqueleto es obtenido gracias al dispositivo Kinect y la librería SimpleOpenNI utilizada en Processing. El punto más alto donde pueda estar la mano se mandara por OSC el valor 100 y el más bajo será 0.



Ableton Live recibiendo los datos:

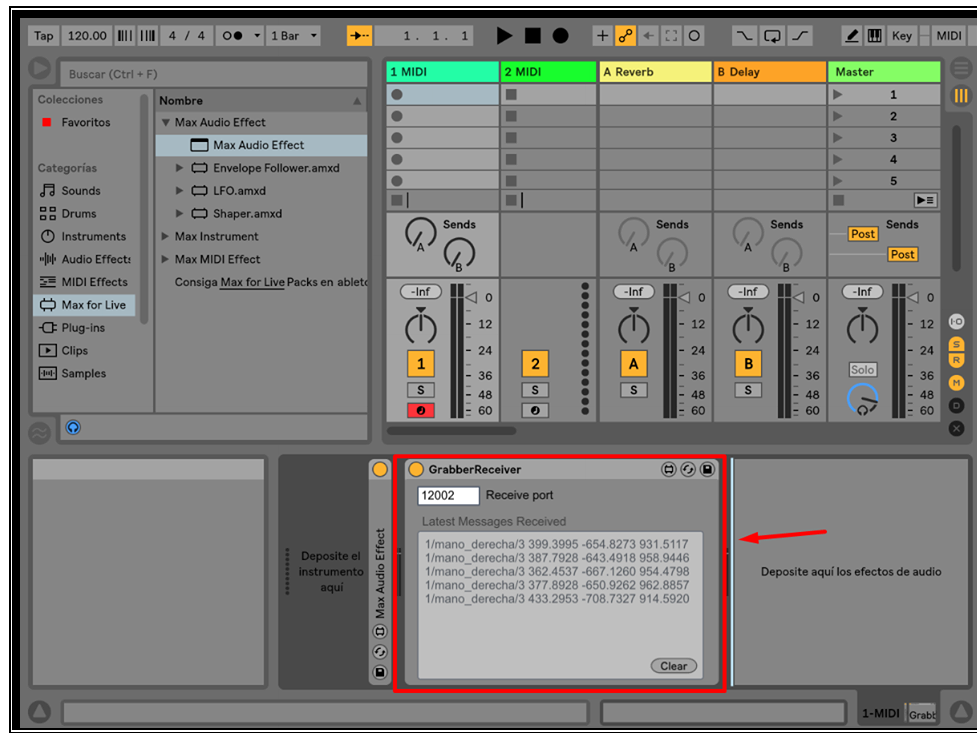
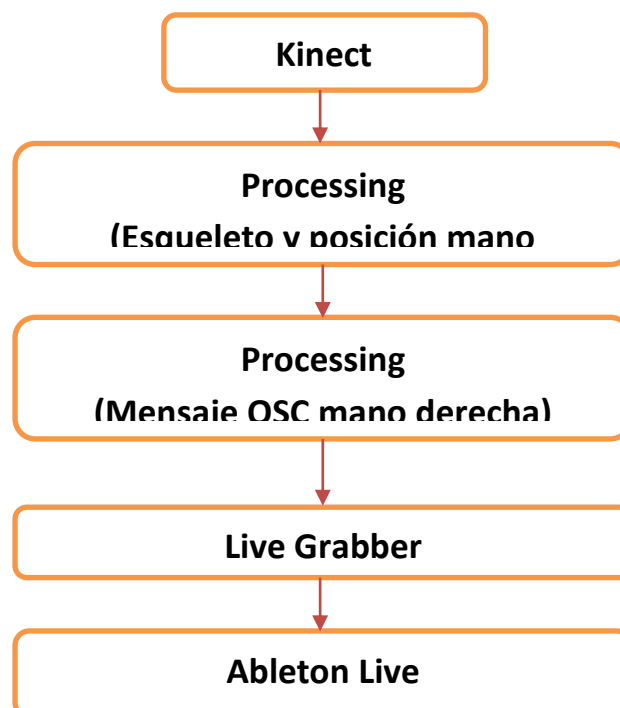


Diagrama de bloques del procesamiento:



Conclusión

Bibliografía

- Página oficial OSC: <http://opensoundcontrol.org/introduction-osc>
- Código de OSC ejemplo: [https://github.com/truthlabs/unity-osc-processing/blob/master/Hello%20World/OSC Processing/OSC Processing.pde](https://github.com/truthlabs/unity-osc-processing/blob/master/Hello%20World/OSC%20Processing/OSC%20Processing.pde)
- Ejemplo aplicaciones OSC: <http://www.sojamo.de/libraries/oscP5/#examples>
- Live Grabber: <https://sonicbloom.net/en/livegrabber-to-sendreceive-osc-in-ableton-live/>
- Ableton: <https://www.ableton.com>