

RAE

1. **TIPO DE DOCUMENTO:** TRABAJO DE GRADO
2. **TITULO:** DESARROLLO DE UN CONTROLADOR MIDI NO CONVENCIONAL, IMPLEMENTADO EN UN SISTEMA EMBEBIDO, UTILIZANDO EL KINECT.
3. **AUTOR:** ANDRÉS CABALLERO PAZ
4. **LUGAR:** BOGOTÁ
5. **FECHA:** ENERO 17 DE 2013
6. **PALABRAS CLAVES:** SISTEMAS EMBEBIDOS, DSP, LINUX, UBUNTU, PURE DATA, KINECT, INNOVACIÓN.
7. **DESCRIPCION DEL TRABAJO:** El documento presenta el desarrollo de un controlador MIDI no convencional implementado en un Sistema Embebido utilizando el Kinect. El dispositivo permite utilizar el protocolo MIDI para controlar hardware y/o software remoto vía WiFi por medio de la captura de movimientos. Con la utilización de este tipo herramientas tecnológicas se puede construir controladores no convencionales aplicados a la ingeniería de sonido, integrables a diferentes redes de comunicación de forma práctica, eficaz e innovadora.
8. **LINEA DE INVESTIGACION:** ANÁLISIS Y PROCESAMIENTO DE SEÑALES.
9. **FUENTES CONSULTADAS:**
 1. Abbott, Doug. Linux for Embedded and Real-Time Application. Newnes. 2003.
 2. Webb, Jarrett- Ashley, James. Beginning Kinect Programming with the Microsoft Kinect SDK. Apress, 2012
10. **CONTENIDOS:** INTRODUCCIÓN, PLANTEAMIENTO DEL PROBLEMA, METODOLOGÍA, LÍNEA DE INVESTIGACIÓN, MARCO TEÓRICO, DESARROLLO INGENIERIL, ANÁLISIS DE RESULTADOS, CONCLUSIONES, PRODUCTOS, RECOMENDACIONES, BIBLIOGRAFÍA, GLOSARIO, ANEXOS.
11. **METODOLOGIA:** EMPÍRICO-ANALÍTICA
12. **CONCLUSIONES:** El documento presenta el diseño y desarrollo de un controlador MIDI no convencional utilizando el Kinect, implementado en el Sistema Embebido Pandaboard ES. En el S.E. se implementó la distribución de Ubuntu, la cual permite la instalación y configuración de los programas y servicios con los cuales se realiza la transmisión de los datos en protocolo MIDI a un computador fijo adquiriendo los datos a partir de movimientos capturados por el Kinect.

DESARROLLO DE UN CONTROLADOR MIDI NO CONVENCIONAL,
IMPLEMENTADO EN UN SISTEMA EMBEBIDO, UTILIZANDO EL KINECT.

ANDRÉS CABALLERO PAZ

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SONIDO
BOGOTÁ
2012

DESARROLLO DE UN CONTROLADOR MIDI NO CONVENCIONAL,
IMPLEMENTADO EN UN SISTEMA EMBEBIDO, UTILIZANDO EL KINECT.

ANDRÉS CABALLERO PAZ

Trabajo de Grado para optar al título de Ingeniero de Sonido

Director

Ing. NELSON FELIPE ROSAS JIMÉNEZ., MSc.

UNIVERSIDAD DE SAN BUENAVENTURA
FACULTAD DE INGENIERÍA
INGENIERÍA DE SONIDO
BOGOTÁ
2012

Nota de aceptación:

Presidente del Jurado

Jurado

Jurado

Ciudad y fecha (dd/mes/año) _____, ____/____/____

"A mis padres y a mi hermano, Víctor Andrés, Adriana y Felipe"

"A mi novia, Kelly"

Este trabajo ha sido posible gracias a ellos

Agradecimientos

A mi profesor y director de tesis Nelson Rosas Jiménez, por su apoyo incondicional, confianza y orientación a lo largo del proyecto.

A mis amigos Andrés Alonso y Juan Carlos Sánchez, quienes aportaron directa o indirectamente en este proceso y estuvieron presentes a lo largo del mismo.

A mis compañeros de trabajo y amigos Katherine García, Nicolás Calderón, Johanna Gaitán, María Fernanda Boscán y Andrea Gualtero, por ser un apoyo a lo largo de la carrera.

Por último pero a las personas más importantes en mi vida, mis padres Víctor Andrés y Adriana, mi hermano Felipe y mi novia Kelly, gracias por su amor, comprensión y paciencia en los momentos más complicados, por hacer parte de mi vida en esta etapa que finaliza.

Tabla de Contenido

INTRODUCCIÓN	12
1. PLANTEAMIENTO DEL PROBLEMA	14
1.1. ANTECEDENTES.....	14
1.2. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA	20
1.3. JUSTIFICACIÓN.....	20
1.4. OBJETIVOS.....	21
1.4.1. Objetivo General	21
1.4.2. Objetivos Específicos	21
1.5. ALCANCES Y LIMITACIONES	21
1.5.1. Alcances.....	21
1.5.2. Limitaciones	21
2. METODOLOGÍA.....	22
2.1. ENFOQUE DE LA INVESTIGACIÓN	22
2.2. HIPÓTESIS.....	22
2.3. VARIABLES	22
2.3.1. Independientes.....	22
2.3.2. Dependientes	23
3. LÍNEA DE INVESTIGACIÓN DE LA UNIVERSIDAD/ SUB-LÍNEA DE LA FACULTAD/ CAMPO TEMÁTICO DEL PROGRAMA.....	23
4. MARCO TEÓRICO.....	24
4.1. Controlador MIDI.....	24
4.1.1. Bytes MIDI.....	26
4.1.2. Mensaje de Nota On/Off.....	27
4.1.3. Mensaje de Cambio de Programa.....	27
4.1.4. Mensaje de Cambio de Control.....	27
4.2. Kinect.....	28
4.2.1. Componentes y Funcionamiento.....	29
4.2.2. Controladores.....	31

4.2.2.1. Open Kinect	31
4.3. Sistemas de Transmisión Inalámbrica	32
4.3.1. Redes Inalámbricas.....	32
4.3.2. Transmisor MIDI: QmidiNet.....	34
4.4. Sistemas Embebidos	35
4.4.1. Software	36
4.5. Pure Data.....	37
5. DESARROLLO INGENIERIL.....	39
5.1. Adecuación S.E.	39
5.1.1 Compilación Sistema Operativo	41
5.1.1.1. Compilación Paso-a-Paso	42
5.1.1.2. Imagen Pre-Compilada.....	44
5.1.1 Interfaces Graficas de Ubuntu.....	45
5.1.1.1 XFCE	46
5.1.1.2 KDE	46
5.1.1.3 Gnome	47
5.1.1.4 LXDE	48
5.1.1.5 Unity	49
5.1.2 Sistemas Embebidos Implementados	50
5.1.2.1 Beagleboard C3.....	50
5.1.2.2 Beagleboard xM.....	52
5.1.2.3 Pandaboard ES	54
5.2. Transmisión mensajes MIDI por WLAN	56
5.2.1. Configuración Router para una Red Local Inalámbrica (WLAN)	57
5.2.2. Configuración de Red en Ubuntu	58
5.2.3. Codificación y envío de MIDI por UDP/IP Multicast.....	60
5.2.4. Implementación y Pruebas Transmisor MIDI QmidiNet.....	60
5.3. Integración Kinect	62
5.3.1. Compilación Pure Data Extended	62
5.3.2. Integración Controladores Kinect	65

5.3.2.1. Instalación Controladores Freenect	66
5.3.3. Integración Pure Data-Kinect	69
5.4. Implementación Controlador MIDI No Convencional	73
5.4.1. Recepción y Decodificación	73
5.4.2. Comprobación	74
5.4.3. Implementación	75
6. ANÁLISIS DE RESULTADOS	79
7. CONCLUSIONES.....	80
8. PRODUCTOS	81
9. RECOMENDACIONES	82
BIBLIOGRAFÍA	83
GLOSARIO	89
ANEXO A: Fotos e Imágenes Complementarias	91
ANEXO B: Anexos digitales.....	95

LISTA DE FIGURAS

Fig. 1 Controlador MIDI. Fuente: Propia	24
Fig. 2 Conexiones protocolo MIDI. Fuente: Propia	25
Fig. 3 Microsoft Kinect [32].....	28
Fig. 4 Kinect sin carcasa exterior [32]	29
Fig. 5 Fotografía de grilla de puntos de luz infrarroja emitidos por el Kinect e imagen de profundidad. Fuente: Propia.....	30
Fig. 6 Fotografía de la cámara RGB e imagen de profundidad del Kinect. Fuente: Propia	31
Fig. 7 Conexión modo Ad-Hoc [33]	33
Fig. 8 Conexión modo infraestructura [33]	34
Fig. 9 Acerca de QmidiNet. Fuente: Propia.....	35
Fig. 10 Sistema Embebido Raspberry Pi [35]	36
Fig. 11 Sección de documentación e información de Pure Data. Fuente: Propia ..	38
Fig. 12 Diagrama de bloques Adecuación S.E.....	39
Fig. 13 Descripción Desarrollo Controlador MIDI con Convencional	40
Fig. 14 Descripción proceso instalación Ubuntu en sistemas embebidos.....	41
Fig. 15 Arquitectura general de un Sistema de Linux Embebido [39]	42
Fig. 16 Ventana compilación Ubuntu minimal. Fuente: Propia.....	43
Fig. 17 Ventana creación imagen pre compilada. Fuente: Propia.....	45
Fig. 18 Interfaz Gráfica Xfce. Fuente: Propia	46
Fig. 19 Interfaz Gráfica Kde. Fuente: Propia.....	47
Fig. 20 Interfaz Gráfica Gnome. Fuente: Propia	48
Fig. 21 Interfaz Gráfica Lxde. Fuente: Propia	49
Fig. 22 Interfaz gráfica Unity. Fuente: Propia.....	50
Fig. 23 Beagleboard C3. Fuente: Propia.....	52
Fig. 24 Beagleboard xM. Fuente: Propia.....	54
Fig. 25 Pandaboard ES. Fuente: Propia	55
Fig. 26 Transmisión Mensajes MIDI.....	57
Fig. 27 Implementación escritorio remoto desde una Tablet. Fuente: Propia	59
Fig. 28 Diagrama de flujo codificación y envío MIDI	60
Fig. 29 Diagrama conexión QmidiNet	60
Fig. 30 Interfaz de configuración QmidiNet. Fuente: Propia	61
Fig. 31 Entradas y Salidas MIDI de Ubuntu. Fuente: Propia.....	61
Fig. 32 Integración Kinect	62
Fig. 33 Descripción proceso compilación Pure Data Extended.....	63
Fig. 34 Interfaz gráfica principal de Pd Extended. Fuente: Propia	65

Fig. 35 Ventana de Comando de Compilación Freenect, con el Kinect conectado a la Pandaboard ES. Fuente: Propia	66
Fig. 36 Ejemplo imagen generada por los controladores Freenect. Fuente: Propia	67
Fig. 37 Diagrama de control de PrimeSense [40]	68
Fig. 38 Ensamble implementado en Pd para la captura de datos en el eje Z y X ..	70
Fig. 39 Sección principal ensamble	70
Fig. 40 Sección activación Gem.....	71
Fig. 41 Sección obtención datos X y Z.....	71
Fig. 42 Configuración MIDICTL.....	72
Fig. 43 Configuración MIDINOTE	72
Fig. 44 Descripción proceso de implementación y prueba del sistema.....	73
Fig. 45 Diagrama de bloques recepción y decodificación	74
Fig. 46 Recepción de datos MIDI vía WiFi a MIDI-OX. Fuente: Propia.....	74
Fig. 47 Recepción de datos MIDI desde VMPK al Kmidimon. Fuente: Propia	75
Fig. 48 Implementación Controlador MIDI no Convencional en Pandaboard ES. Fuente: Propia	76
Fig. 49 Interfaz principal FL Studio. Fuente: Propia	77
Fig. 50 Captura de pantalla de la implementación en el FL Studio. Fuente: Propia	77
Fig. 51 Interfaz de captura de los datos del Kinect implementado en Pure Data. Fuente: Propia	78
Fig. 52 Pandaboard ES conectada remotamente a Samsung Galaxy Tab 10.1. Fuente: Propia	91
Fig. 53 Interfaz del editor conexiones de red. Fuente: Propia.....	91
Fig. 54 Configuración de control remoto MIDI de FL Studio. Fuente: Propia	92
Fig. 55 Conexión Pandaboard ES con el Kinect. Fuente: Propia.....	92
Fig. 56 Interfaz de usuario de TouchOSC para Samsung Galaxy Tab 10.1. Fuente: Propia	93
Fig. 57 Controlador MIDI Virtual VMPK. Fuente: Propia	93
Fig. 58 Ensamble de recepción OSC desde la aplicación TouchOSC. Fuente: Propia	94

LISTA DE TABLAS

Tabla 1 Mensajes MIDI	26
Tabla 2 Comandos Bytes de Estado.....	26
Tabla 3 Comparación especificaciones sistemas embebidos	56

RESUMEN

El documento presenta el desarrollo de un controlador MIDI no convencional implementado en un Sistema Embebido utilizando el Kinect. El dispositivo permite utilizar el protocolo MIDI para controlar hardware y/o software remoto vía WiFi por medio de la captura de movimientos. Con la utilización de este tipo herramientas tecnológicas se puede construir controladores no convencionales aplicados a la ingeniería de sonido, integrables a diferentes redes de comunicación de forma práctica, eficaz e innovadora.

ABSTRACT

The document presents the development of a non-conventional MIDI controller implemented into an embedded system using the Kinect. The device uses the MIDI protocol to control software and/or hardware via WiFi through movement detection. With the utilization of this kind of technological tools its possible development non-conventional controllers to Sound Engineer, connecting to different network environment and practical and Kinect allows developing such applications in a practical and effective way.

INTRODUCCIÓN

Los sistemas embebidos son herramientas prácticas y versátiles para la implementación de proyectos relacionados con la ingeniería de sonido, los cuales incorporan nuevas opciones para el desarrollo de procesos control y producción audiovisual.

En este campo se han desarrollado aplicaciones, tales como un compresor de audio y video en tiempo real utilizando un Sistema Embebido, en el cual se obtuvo una alta velocidad y calidad en el procesamiento y una disminución en costos [38]. A su vez este tipo de sistemas permiten realizar procesos de transferencia tecnológica y de conocimientos, con herramientas económicas y de fácil acceso, gracias a sus dimensiones y a las prestaciones o características de Hardware/Software que soportan. [4]

En comparación con los computadores de propósito general, los sistemas embebidos son diseñados e implementados para ejecutar aplicaciones específicas y previamente definidos por parte del diseñador. [4][39]

Al ser sistemas programables se pueden implementar diferentes sistemas de comunicación, entre los cuales se encuentra el protocolo MIDI, creado para unificar los criterios en cuanto a comunicación de instrumentos digitales. Este protocolo se puede implementar en automatizaciones, desarrollo de sistemas de control de luces, automatización de secuencias musicales y sistemas domóticos.

A partir de herramientas tecnológicas modernas como el Kinect, se puede implementar sistemas versátiles e interactivos para el usuario con el objetivo de desarrollar controladores no convencionales para diferentes aplicaciones. El Kinect es capaz de cambiar su espacio de funcionamiento, disminuyendo el área de captura a un rango determinado por el usuario gracias a su proyector y sensor infrarrojos.

Los controladores MIDI son dispositivos utilizados para realizar ediciones, producciones y diferentes tipos de control, los cuales trabajan con el protocolo MIDI que al ser universal permite la conexión a diferentes dispositivos de forma

práctica. Estos dispositivos, normalmente, están compuestos por teclas, potenciómetros y botones, los cuales son el medio de interacción con el usuario.

El dispositivo implementado en el proyecto de grado se enfoca en el desarrollo de una herramienta versátil e interactiva, en la cual el usuario es capaz de generar mensajes MIDI por medio de detección de movimientos, para luego enviarlos vía WiFi y previamente procesados por un computador que realiza control MIDI de forma remota.

1. PLANTEAMIENTO DEL PROBLEMA

1.1. ANTECEDENTES

- **“Future human interfaces to computer controlled sound systems”[28]**

Este artículo plantea nuevas posibilidades de interacción hombre-máquina por medio de seguimiento de posición especial, de ojos, de cabeza y de entradas táctiles. Presenta algunas tecnologías de interfaces humanas, las cuales pueden ser implementadas en los sistemas de sonido.

Los avances en la interacción hombre-máquina se divide en 3 secciones; ingreso y reconocimiento de datos por la computadora, sistemas visuales y de escucha y análisis de espacialidad.

Los problemas asociados a estos diseños suelen afectar la usabilidad de los sistemas que están conectados a este, implementar un dispositivo óptico-eléctrico conlleva diferentes problemas.

Con el sentido de la vista, el sentido de la audición es la forma principal en la que los humanos recolectan información de su entorno, como la localización espacial. [28]

- **"Gesture to Operate DAW Software" [29]**

Hoy en día existen diferentes formas de capturar gestos, como en un mouse o en una pantalla multi-touch, para su aplicación a las estaciones de trabajo de audio digital, un ejemplo es el movimiento de un dedo para el cambio de un valor de un parámetro específico. La pregunta es ¿en qué medida el software de audio es beneficiado de esto? Se describe el diseño de estrategias que consisten en integrar gestos determinados asociados a comandos. El artículo plantea el uso de 22 funciones DAW. En una tabla indican los gestos y la función asociada a este, y la reacción de las personas que hicieron las respectivas pruebas.

Al final se concluye que los participantes escogieron algunas funciones por encima de otras. Sobre todo porque esto requiere cierto tipo de entrenamiento para que puedan explotar todas las posibilidades de gestos, y la habilidad de creación de gestos por parte de los usuarios.

- **"Interactive Mixing Using Wii Controller" [30]**

El artículo describe el diseño, construcción y análisis de un control interactivo por medio de gestos para la mezcla de audio, utilizando un control inalámbrico de video juegos. El concepto está basado en la idea que el ingeniero de mezcla no esté cerca la consola y se convierta en parte de la interpretación de la pista de audio. El sistema permite un control total y en vivo de los controles de ganancia, paneo, ecualización, rango dinámico de la compresión y otra variedad de efectos multi canal.

El control del Wii posee varios sensores de movimiento, que permiten al usuario ubicarse en 3 dimensiones gracias a su acelerómetro. También cuenta con un sensor infrarrojo que determina la posición en el eje X y Y, y posee un dispositivo bluetooth para su conexión.

La construcción de un control de movimiento para la mezcla de audio muestra una forma de innovar y mejorar la mezcla en tiempo real. Permite al ingeniero liberarse de la forma convencional de mezcla, dándole libertad de movimiento.

- **"Embedded Controllers For Audio Devices" [31]**

Los sistemas embebidos para el audio son presentados como un controlador basado en el protocolo MIDI, son fácilmente adaptados para el audio, el artículo muestra el desarrollo y las herramientas utilizadas para programar un Sistema Embebido para el control de dispositivos de audio.

Los sistemas embebidos abren un gran número de posibilidades en la implementación de aplicaciones audio. El control requiere un Sistema Embebido el cual será programado según sea necesario. Este se divide en cuatro secciones: memoria, unidad lógica, unidad de control y entradas y salidas. [31]

- Memoria: Se almacena y se lee la información del sistema.
- Unidad Lógica: Realiza las operaciones lógicas y matemáticas de la información.

- Unidad De Control: Dirige las actividades de la computadora, siguiendo las instrucciones almacenadas en la memoria para cumplir una tarea específica. Aquí se encuentra el procesador y microprocesadores.
- Entradas y Salidas: En los dispositivos de entrada hay interacción con el usuario de forma física para comunicarse con el computador. Los de salida se utilizan para que el computador se comunique con el usuario.
- **Using Xbox 360 Kinect Games on Enhancing Visual Performance Skills on Institutionalized Older Adults with Wheelchairs” [76]**

El objetivo de este estudio es explorar los beneficios de los videojuegos en personas de la tercera edad, el Kinect pueden llegar a ser una buena herramienta para población de la tercera edad con silla de ruedas. El estudio se realizó con 53 participantes divididos en dos grupos, un grupo experimental de 22 personas de unos 78 años, y un grupo de control de 31 personas de unos 79 años, el grupo experimental accedió voluntariamente a practicar 3 sesiones de entrenamiento de Xbox 360 con el Kinect por una semana, con una duración de 30 minutos cada vez. El Vienna Test System y el test de Soda Pop fueron usados para medir la reacción y la coordinación entre ojos y manos de las habilidades de interpretación visual, cuyos resultados favorables indicaron que las personas de la tercera edad mejoran este tipo de habilidades gracias a Xbox 360 Kinect. [76]

- **“Kinect-based obstacle detection for manipulator” [77]**

Este artículo presenta un método para distinguir obstáculos y poder manipularlos cuando se encuentran en el mismo espacio, para esto se utiliza el Kinect como dispositivo de captura, además de la explicación del método de calibración de Kinect, se describe el método de calibración interactivo de mínimos cuadrados entre el Kinect y el manipulador. El modelo del manipulador es generado usando la librería OpenGL. Finalmente, la superficie del manipulador es eliminada de la escena utilizando el sensor infrarrojos del Kinect. [77]

- **“Human motion tracking of mobile robot with Kinect 3D sensor” [78]**

Este artículo presenta un sistema de control de seguimiento del movimiento con el Kinect, implementado en robot móvil. En este sistema, la información obtenida con el Kinect de la posición en tres dimensiones de la persona, permite controlar directamente la velocidad y los movimientos del robot.

Para reducir el ruido, se utilizó el efecto de oclusión incluida en los datos adquiridos por el Kinect y asu estimar el estado del movimiento, se usa el filtro Kalman, el cual también es descrito en este artículo. [78]

- **“The design and initial calibration of an optical tracking system using the Microsoft Kinect” [79]**

La corrección de movimientos requiere un confiable y minucioso seguimiento, el cual es llevado a cabo por sistemas que usan dispositivos como el sensor de movimiento Polaris Vicra¹ para monitorear la posición y orientación de una herramienta. Los sistemas de seguimiento del movimiento se fijan directamente en el objeto sin necesidad de usar marcadores, este método fue desarrollado gracias al Kinect, que posee una cámara RGB y un sensor infrarrojo de bajo costo el cual puede capturar movimientos rígidos del cuerpo. El algoritmo del proyecto abierto llamado “PointCloudLibrary” fue implementado para registrar la grilla de puntos generados por el Kinect. [79]

- **“Static Hand Gesture Recognition Based on HOG with Kinect” [80]**

En este artículo se implementa un método de reconocimiento estático de gestos de la mano usando información de profundidad del Kinect. Debido a que la mano es relativamente un objeto pequeño con articulaciones complejas, es difícil lograr reconocer sus movimientos. Por tal razón, se empleó la función del momento geométrico invariante adaptado a los cambios de luz, mediante el análisis de las características de la mano. Gracias al algoritmo de entrenamiento “Adaboost” se obtuvieron los modelos gestuales de prueba, con el fin de construir un sistema

¹ Es un dispositivo de medición óptico que mide la posición en 3D de marcadores activos o pasivos que son capturados por una aplicación específica

preciso y eficiente de reconocimiento de gestos manuales usando el sensor del Kinect. [80]

- **“The Kinect up close: Adaptations for short-range imaging” [81]**

En este artículo se describe la calibración y funcionamiento de captura de imágenes en rangos específicos utilizando el Kinect, con la calibración adecuada de su cámara RGB y de profundidad, el Kinect puede capturar detalladamente nubes de puntos hasta 30 fotogramas por segundo. Esta capacidad convierte al Kinect en una herramienta ideal para su uso en la robótica como un sensor de navegación de bajo costo. Por lo tanto, las técnicas de calibración de la cámara de profundidad Kinect permiten determinar un rango de funcionamiento específico. Para realizar la calibración de profundidad, es necesario tener una plataforma de calibración y software, que fueron desarrollados para asignar automáticamente valores brutos de profundidad para objetos. La plataforma de calibración consistió en un tablero de ajedrez blanco facilitado al software asignar los valores de profundidad. Para modificar la óptica del Kinect y mejorar la imagen de corto alcance, se utilizó el adaptador “Nyko Zoom”, aunque es eficaz en la reducción de la distancia mínima de Kinect, este introduce distorsión al sistema, para esto se desarrolló un modelo para corregir dicha distorsión mediante software. Con esto se logró disminuir el alcance mínimo del Kinect en aproximadamente 30%. [81]

- **“A new embedded image information processing system design” [82]**

Este artículo presenta un sistema embebido de adquisición de información de imagen, el cual incluye la codificación y almacenamiento de imágenes capturadas y otros métodos de tratamiento que cumplen diferentes secciones; la primera es investigar acerca de una pantalla en tiempo real de la información digital de imagen a través de un LCD, y segundo lograr el diseño de una parte de almacenamiento de información de imagen en el sistema a través de la codificación de imagen BMP. [82]

- **“Mobile Robots Navigation in Indoor Environments Using Kinect Sensor” [83]**

Este artículo presenta el desarrollo de un sistema de percepción para ambientes de interior con el propósito de permitir la navegación autónoma de robots móviles de vigilancia. El sistema se compone de dos partes, la primera es un sistema de navegación reactiva en el cual el robot móvil se mueve evadiendo obstáculos, usando el Kinect, y la segunda parte usa una red neural artificial (ANN) para reconocer diferentes configuraciones del entorno. El ANN está entrenado usando información capturada desde el Kinect en interiores, de este modo, el robot es capaz de realizar una navegación topológica combinando un comportamiento reactivo interno. El mapa topológico es representado por una gráfica que representa la configuración del ambiente.

El sistema también trabaja en la oscuridad, lo cual es una gran ventaja para sistemas de vigilancia. El experimento fue llevado a cabo con un robot Pioneer P3-AT equipado con un sensor Kinect con el fin de validar y evaluar este enfoque. [83]

- **“A Multi-User Interaction System Based on Kinect and Wii Remote” [84]**

El sistema de interacción multi-usuario que implementa el Kinect y un control remoto Wii, fue desarrollado para manipular el escritorio de Windows. Este sistema combina la información gestual capturada desde estos sensores brindando una experiencia más real y natural al usuario. [84]

1.2. DESCRIPCIÓN Y FORMULACIÓN DEL PROBLEMA

Hoy en día se ve como los avances tecnológicos simplifican procesos e interactúan de forma más amigable con el usuario, brindándole una experiencia más realista, hasta el punto de evitar en lo posible el contacto físico con los objetos para poder controlarlos, ya sea por sensores o por detección de movimiento.

Por eso se quiere llevar esta tecnología al campo de la ingeniería de sonido, para ofrecer en el mercado formas innovadoras para realizar grabaciones, mezclas, montajes de sonido en vivo, etc. Con el fin de captar la atención de las personas que aún no conocen la carrera.

Al ser software poco común, se buscan formas prácticas y sencillas para crear proyectos caseros de sonido de forma interactiva, evitando memorizar por ejemplo métodos abreviados de teclado o tener que abrir pestaña por pestaña, buscando la opción que requiera.

¿Cómo brindar una experiencia más realista a los usuarios de programas de sonido, por medio de la implementación de sistemas de control MIDI no convencionales?

1.3. JUSTIFICACIÓN

Hoy en día se busca que todos los procesos se apliquen de una forma sencilla, y además se cuente con una interfaz amigable con el usuario, por eso se requiere que la comunicación con un computador sea lo más interactiva posible, evitando la instalación de cualquier tipo de controladores para que un dispositivo externo a él funcione, por lo cual se utilizan avances tecnológicos, diseñados para tal fin.

La implementación de sistemas embebidos para este fin, aún no ha sido aplicada a la ingeniería de sonido, por lo cual se quiere promover su uso mediante la transmisión de MIDI vía WiFi para control de diferentes dispositivos aplicados a la Ingeniería de Sonido.

1.4. OBJETIVOS

1.4.1. Objetivo General

Desarrollar un controlador MIDI interactivo, implementado en un Sistema Embebido que envíe mensajes MIDI, utilizando el KINECT.

1.4.2. Objetivos Específicos

- Integrar el KINECT con un sistema embebido.
- Desarrollar interfaz gráfica para la interacción con el usuario, que pueda interpretar mensajes MIDI.
- Desarrollar sistema de control de usuario, que permita el control por medio de 3 tipos de mensaje MIDI con el sistema embebido

1.5. ALCANCES Y LIMITACIONES

1.5.1. Alcances

- Se podrá controlar software compatible con el protocolo MIDI, con diferentes funciones según movimientos asignados por el usuario.

1.5.2. Limitaciones

- Se programarán determinados mensajes MIDI, para evitar que el KINECT confunda algunos movimientos que podrían llegar a ser muy similares, y envíe mensajes erróneos al sistema.

2. METODOLOGÍA

2.1. ENFOQUE DE LA INVESTIGACIÓN

El enfoque de este proyecto es empírico-analítico, debido a que se están aplicando nuevas formas de procesamiento y manejo de mensajes MIDI a partir de controladores no convencionales, a través de la aplicación de procesos de transferencia tecnológica.

Los sistemas embebidos se han convertido en una fuente de desarrollo práctico, este tipo de herramientas permite crear aplicaciones específicas de software y hardware para el desarrollo de dispositivos, gracias a sus dimensiones y a las prestaciones.

Con el desarrollo de este dispositivo pueden comprobarse la efectividad y versatilidad de la aplicación de sistemas embebidos para el procesamiento de señales.

2.2. HIPÓTESIS

De acuerdo al comportamiento del Kinect en un sistema embebido, se busca que la comunicación del usuario con el sistema sea más agradable e interactiva, tomando los movimientos capturados para ejecutar acciones específicas con ayuda de protocolo MIDI, y se tendrá en cuenta que la latencia de captura de imagen no afecte la interacción con el sistema.

2.3. VARIABLES

2.3.1. Independientes

Asignar movimientos a los siguientes mensajes MIDI:

- Cambio de Control
- Cambio de Programa

- Nota MIDI

2.3.2. Dependientes

Comportamiento del Kinect en un Sistema Embebido y latencia del mensaje al llegar a algún dispositivo

3. LÍNEA DE INVESTIGACIÓN DE LA UNIVERSIDAD/ SUB-LÍNEA DE LA FACULTAD/ CAMPO TEMÁTICO DEL PROGRAMA

Línea de investigación: análisis y procesamiento de señales.

Núcleo Problémico: Acústica y audio.

4. MARCO TEÓRICO

En el desarrollo de un controlador MIDI no convencional, utilizando el Kinect, es necesario identificar las siguientes temáticas: Controladores MIDI, Kinect, Sistemas de Transmisión Inalámbrica, Sistemas Embebidos y Pure Data

4.1. Controlador MIDI

Dispositivo capaz de interpretar y reproducir mensajes MIDI. En sus modelos más simples cuentan únicamente con teclas, entrada y salida MIDI, con lo que cualquier cambio que se desee hacer en el programa debe hacerse directamente desde un software específico. Los más completos disponen de más opciones para el manejo de bancos, cambio de control o de programa.

Normalmente el controlador tiene la apariencia de un instrumento tradicional, rediseñado para funcionar bajo el protocolo MIDI. Los controladores comunes son teclados como el que se observa en la Figura 1, adicionalmente pueden modificar la modulación, afinación, etc... .Pero no produce sonidos por sí mismos.[2][3][5]



Fig. 1 Controlador MIDI. Fuente: Propia

Para la interconexión de dos o más interfaces se utilizan tres diferentes puertos y un cable con conectores DIN de 5 pines. Los tres puertos son IN, OUT y THRU los cuáles cumplen las siguientes funciones:

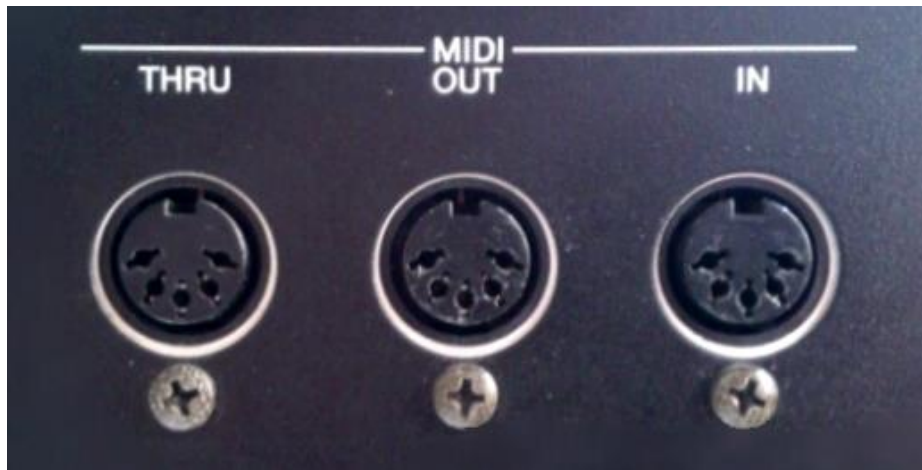


Fig. 2 Conexiones protocolo MIDI. Fuente: Propia

MIDI IN: Es la entrada del dispositivo, captura la información proveniente desde otro dispositivo MIDI.

MIDI OUT: Es la salida del dispositivo, por donde se envía la información generada por el dispositivo MIDI.

MIDI THRU: Es un puerto que se utiliza para la interconexión de otro dispositivos MIDI. Este es una copia exacta de la información proveniente del puerto MIDI IN y se utiliza en el caso de tener un dispositivo maestro que envíe información a más de un dispositivo a la vez.[7]

4.1.1. Bytes MIDI

Hay dos clases de bytes: De estado y de datos. Principalmente se diferencian por el primer bit, que puede ser 0 o 1, siendo el primero de datos y el segundo de estado, al enviar un mensaje MIDI se envía un byte de estado, que puede estar seguido de determinada cantidad de bytes de datos, a su vez, los mensajes de estado se dividen en dos: mensajes de canal y mensajes de sistema., los mensajes de canal se envían a un dispositivo específico configurado en ese canal, mientras que los mensajes de sistema son recibidos por todos los equipos.[2](Ver Tabla 1, Tabla 2)

Byte Estado	Descripción
1000cccc	Desactivación de nota
1001cccc	Activación de nota
1010cccc	Post-pulsación polifónica
1011cccc	Cambio de control
1100cccc	Cambio de programa
1101cccc	Post-pulsación monofónica de canal
1110cccc	Pitch
11110000	Mensaje exclusivo del fabricante
11110001	Mensaje de trama temporal
11110010	Puntero posición de canción
11110011	Selección de canción
11110110	Requerimiento de entonación
11110111	Fin de mensaje exclusivo
11111000	Reloj de temporización
11111010	Inicio
11111011	Continuación
11111100	Parada
11111110	Espera activa
11111111	Reseteo del sistema

Tabla 1 Mensajes MIDI

Valor (Decimal)	Valor (Hexadecimal)	Comando	Byte de dato
128-143	80-8F	Nota Off	2 (nota, Velocidad)
144-159	90-9F	Nota ON	2 (nota, Velocidad)
160-175	A0-AF	Presión de tecla	2 (nota, Presión)
176-191	B0-BF	Cambio de Control	2 (# de Control, Valor)

Tabla 2 Comandos Bytes de Estado

4.1.2. Mensaje de Nota On/Off

Al momento de pulsar una tecla el controlador emite un mensaje MIDI de Nota On, el byte de estado indicara por cual canal se emitirá el mensaje, seguido por dos bytes que especifican el número de tecla y el valor de la velocidad de la pulsación el cual determina que tan fuerte fue pulsada.[2]

Al soltar la tecla, el controlador emite un mensaje MIDI de Nota Off, este también lleva bytes de datos para determinar la nota y la velocidad con que se ha soltado la tecla, normalmente el byte de datos que determina la velocidad no es tomado en cuenta, ya que al apaga la nota por medio del byte de estado.

4.1.3. Mensaje de Cambio de Programa

Es el parámetro que modifica los elementos que generan el sonido dentro de un sintetizador, básicamente un cambio de programa que se refiere a cambiar el instrumento del programa activo. Consiste en un byte de encabezamiento (1100) seguido por un byte de datos que indica el número de programa, al recibir este mensaje cambiará inmediatamente el timbre con el que se está tocando las notas. [3]

4.1.4. Mensaje de Cambio de Control

Se usan para controlar una gran variedad de funciones de un sintetizador, en los mensajes de cambio de control se incluye la mayoría de los controles de expresión de sonido, los cuales se transmiten en mensajes independientes, estos controles tienen asignado un número que está establecido en la norma MIDI.

En los mensajes de control, el primer byte de datos indica el número de controlador o número de control y el segundo indica el valor que se ha variado a dicho control. En el caso de los controles que solo pueden tomar dos estados únicamente (On/Off), como el sostenido, el segundo byte de datos toma el valor 0 cuando está apagado y 127 cuando está activo.

4.2. Kinect²

Fue desarrollado con la idea de crear una interacción juego-humano más realista, con lo que se logró obtener la capacidad de entender las dimensiones del espacio y tener reconocimiento de voz. Gracias a esto identifica e interpreta sus movimientos y los traduce en un formato en el que los desarrolladores puedan usar para implementar nuevas aplicaciones y mejorar la experiencia del usuario. [6]

Este combina una cámara RGB³, una cámara infrarroja, un sensor infrarrojo y cuatro micrófonos de desarrollo propio. Estos componentes brindan nuevas experiencias a través de cualquier consola Xbox 360, Gracias a esto se logra capturar movimientos en tres dimensiones, reconocimiento facial y de voz.



Fig. 3 Microsoft Kinect [32]

²Controlador de juego libre desarrollado por Alex Kipman y Microsoft para la consola Xbox 360.

³Del inglés *Red, Green, Blue*; "rojo, verde, azul" modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla los tres colores luz primarios.

4.2.1. Componentes y Funcionamiento

La cámara RGB percibe la luz que es reflejada por los objetos hacia el lente de la cámara, luego la imagen es reconstruida simulando al ojo humano. El Kinect obtiene datos de distancia en 3D de los objetos mediante la emisión de puntos de luz infrarroja que proyecta el sensor infrarrojo, el cual permite al Kinect percibir su posición en el espacio sin importar su forma física.

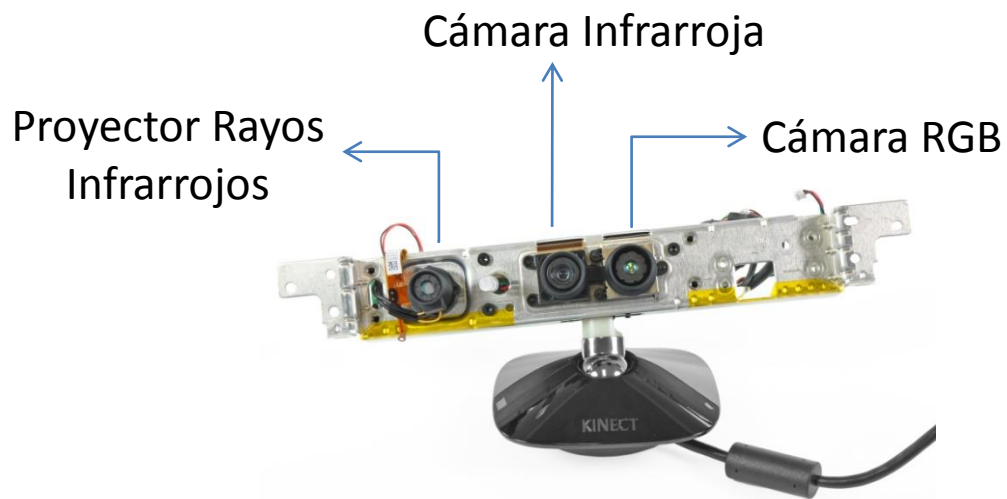


Fig. 4 Kinect sin carcasa exterior [32]

El proyector de rayos infrarrojos se encuentra ubicado a la izquierda del Kinect como se puede ver en la Figura 4, este emite una grilla de puntos infrarrojos, por el rango de frecuencia que maneja se hacen imperceptibles para el ojo humano y solo la cámara infrarroja es capaz de procesarlos, esta se encuentra ubicada a la derecha del Kinect.



Fig. 5 Fotografía de grilla de puntos de luz infrarroja emitidos por el Kinect e imagen de profundidad. Fuente: Propia.

Es posible determinar la distancia de un objeto a la cámara debido a que cada Kinect está calibrado a determinado rango de operación, así cuando un objeto se encuentra fuera de ese rango, los puntos generados por el sensor infrarrojo cambian de dirección, de esta manera a partir de la posición de la grilla de puntos se puede conocer las distancias de cada objeto al Kinect.

Una imagen de profundidad es más fácil de entender por un computador que una imagen convencional, ya que cada pixel muestra que tal lejos está de la cámara, lo cual ayuda a determinar en qué punto inicia y termina un objeto. También contiene información sobre el espacio en tres dimensiones, esto con el fin de identificar y seguir la trayectoria de los movimientos de cada individuo, inclusive localiza las articulaciones del cuerpo.

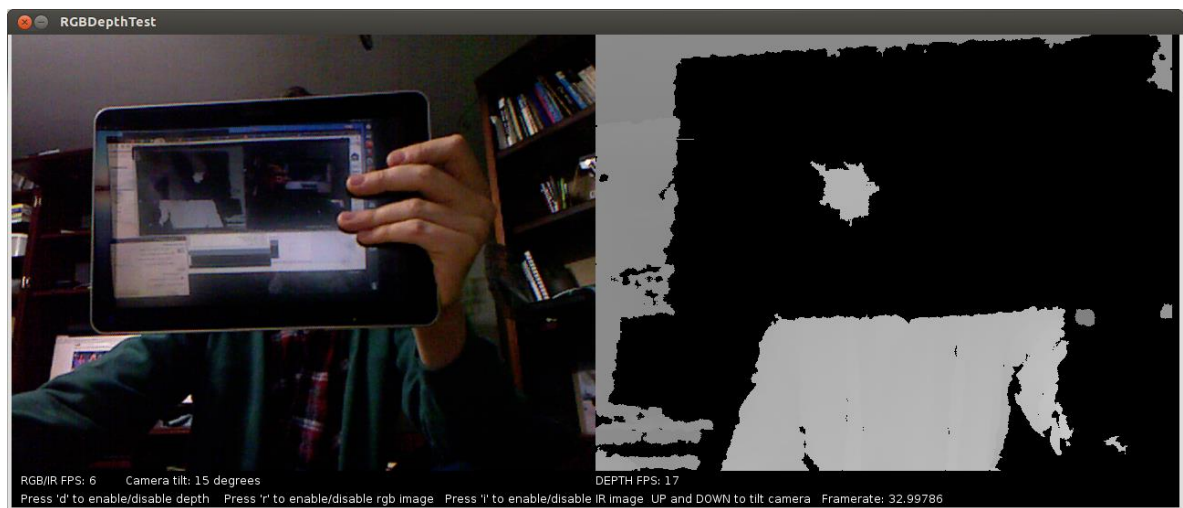


Fig. 6 Fotografía de la cámara RGB e imagen de profundidad del Kinect. Fuente: Propia

La cámara RGB muestra una imagen común sin ninguna información aparentemente útil, pero al alinearla con la imagen de profundidad, se puede alterar la imagen de la cámara RGB, creando un espacio en tres dimensiones o un entorno virtual a color, con la capacidad de manipular el espacio gracias a los datos proporcionados por el sensor infrarrojo. [7]

4.2.2. Controladores

Son los encargados de conectar el Kinect con el computador o sistema embebido, tienen la capacidad de traducir los datos capturados Kinect para que puedan ser entendidos por el ordenador.

4.2.2.1. Open Kinect

Es una comunidad de personas que desarrolla librerías para el Kinect, principalmente se enfocan en el libfreenect⁴, que es el controlador que utiliza esta comunidad. Maneja una licencia de código completamente abierto, siendo este gratuito las personas interesadas puede contribuir en el mejoramiento del mismo. [9][7]

⁴ Enlace de la comunidad OpenKinect: http://openkinect.org/wiki/Main_Page

El libreenect soporta la cámara RGB, cámara infrarroja, motor, acelerómetro, led y micrófonos, brindando un control completo sobre todos sus componentes, este maneja la información de posición en los ejes X, Y y Z.

4.3. Sistemas de Transmisión Inalámbrica

Son los encargados de interconectar dos o más dispositivos por medio de la transmisión de señales de un lugar a otro utilizando la modulación de ondas electromagnéticas. Los dispositivos físicos son utilizados como emisor y receptor de señal. [11]

4.3.1. Redes Inalámbricas

Es un mecanismo de conexión en red de dos o más dispositivos electrónicos, sin la necesidad de una conexión cableada, este tipo de conexión esta presente en los dispositivos habilitados con WiFi⁵, tales como: ordenadores portátiles, consolas de videojuegos, teléfonos inteligentes o un reproductor de audio digital.[12][15]

Existen diferentes categorías de redes inalámbricas y dependen del área de cobertura:

1. Red de área personal inalámbrica (WPAN). [72]
2. Red de área local inalámbrica (WLAN). [73]
3. Red de área metropolitana inalámbrica (WMAN). [74]
4. Red de área extendida inalámbrica (WWAN). [75]

La Red de área local inalámbrica (WLAN) es la de interés, siendo esta la utilizando en hogares, empresas y sitios públicos. Tiene un alcance de unos 20 metros en interiores, se pueden cubrir grandes áreas con la superposición de múltiples puntos de acceso. [12][10]

Los receptores inalámbricos en muchos casos se encuentran integrados dentro de los mismos dispositivos, o son implementados como periféricos USB⁶, en su

⁵Marca de la *Wi-Fi Alliance*. Estándar de comunicación inalámbrica.

⁶Universal Serial Bus. "Bus Universal en Serie". Es un estándar de conexión para dispositivos y periféricos electrónicos.

mayoría PnP⁷. Los transmisores utilizados son los enrutadores que proporcionan conectividad de nivel de red de área local, su función es reenviar y enrutar paquetes de datos a diferentes receptores conectados, los dispositivos son identificados por medio de direcciones IP⁸. [10]

Trabaja bajo el estándar IEEE denominado 802.11x, abarca la capa física y una subcapa de acceso al medio de la capa de enlace del modelo OSI⁹. Define el concepto de Conjunto Básico de Servicio (BSS), consiste en el reconocimiento de dos o más nodos inalámbricos con la capacidad de transmitir información entre ellos. [13] Pueden intercambiar información de dos maneras distintas:

1. Directa: Cada nodo se comunica sin necesidad de una interfaz que se encargue del intercambio de datos, creando una red inalámbrica descentralizada. Este modo es llamado Ad-Hoc, solo permite la transmisión de datos entre dispositivos inalámbricos.[13]

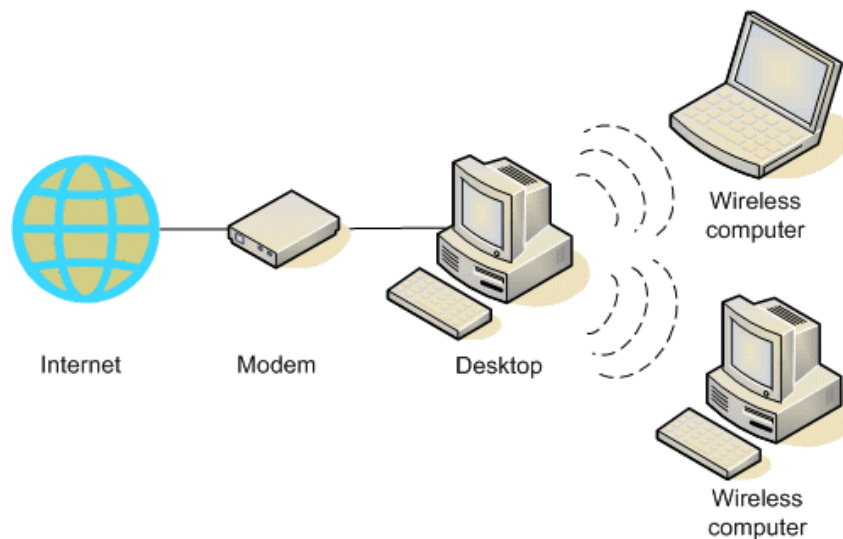


Fig. 7 Conexión modo Ad-Hoc [33]

2. Por Interfaz: Cada nodo se comunica con una interfaz de comunicación encargada de gestionar la transmisión de datos y la creación de una red local, pueden ser Enrutadores o Puntos de Acceso. Este es llamado

⁷Plug-and-Play, "Conectar y utilizar". es la tecnología que permite a un dispositivo ser conectado a una computadora sin necesidad de controladores.

⁸ Del Ingles *Internet Protocol*. "Protocolo de Internet" Protocolo de comunicación de datos digitales.

⁹Open systeminterconnection. "Modelo de interconexión de sistemas abiertos" Marco de referencia para la definición de arquitecturas de interconexión de sistemas de comunicación.

modo Infraestructura y permite vincular red inalámbrica con red cableada, ya que la interfaz funciona como puente de comunicación entre estos dos.[13]

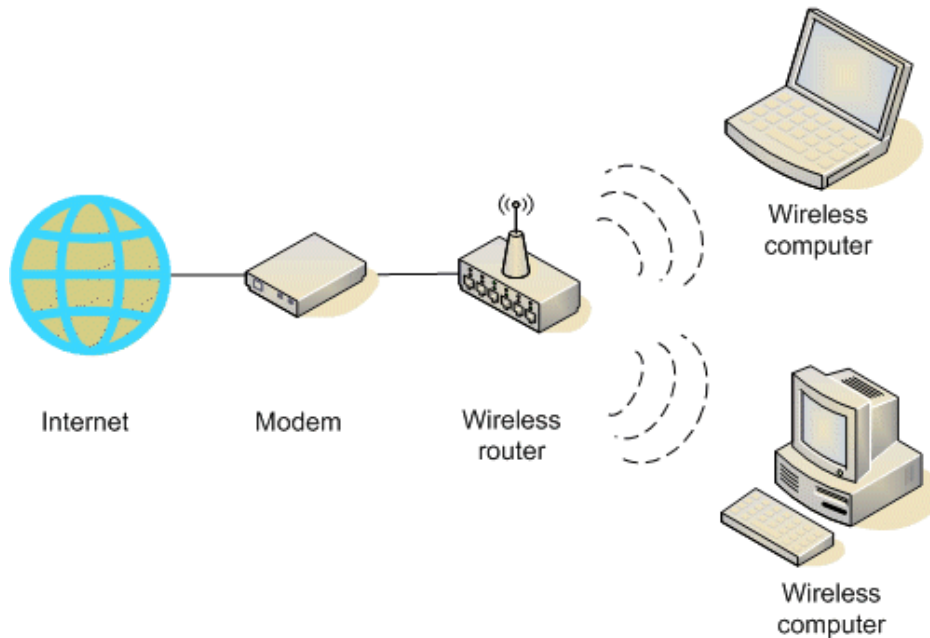


Fig. 8 Conexión modo infraestructura [33]

4.3.2. Transmisor MIDI: QmidiNet

Software encargado de codificar y decodificar mensajes MIDI para su envío por medio de redes locales. Se encarga de abrir un enlace de red para la transmisión de mensajes MIDI vía UDP/IP Multicast. Recibe y envía datos MIDI utilizando el secuenciador de ALSA o el puerto MIDI de JACK a través de redes, alámbricas e inalámbricas.

Este crea puertos MIDI virtuales que son reconocidos por JACK, que facilita su interconexión con diferentes aplicaciones dentro de LINUX. Para su implementación se utiliza un controlador MIDI para Linux llamado VMPK, el cual genera y recibe mensajes MIDI. [14] Es software libre, distribuido como GNU¹⁰ Licencia Publica General versión 2 o posterior.

¹⁰Proyecto iniciado por Richard Stallman con el objetivo de crear un sistema operativo completamente libre.

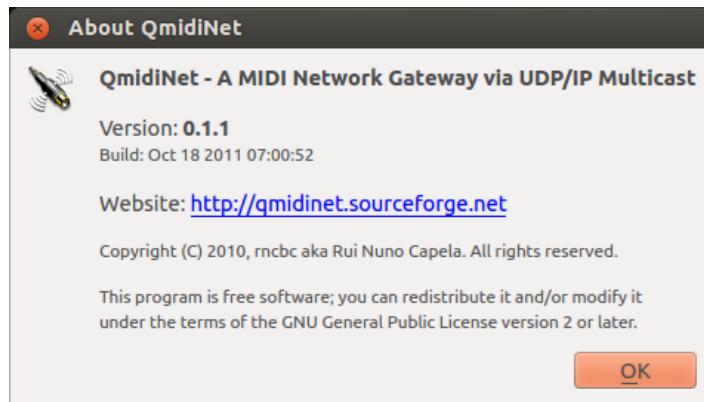


Fig. 9 Acerca de QmidiNet. Fuente: Propia

4.4. Sistemas Embebidos

Son sistemas programables, diseñados para realizar tareas específicas determinadas por el usuario, con el fin de optimizar los procesos para mejorar su desempeño y eficiencia, reduciendo tamaño y costos de producción. Los sistemas embebidos se convierten en herramientas de desarrollo útiles dentro de diversos ámbitos de la ingeniería de sonido.[18]

A diferencia de los computadores convencionales se caracterizan por el bajo consumo de energía, son económicos, poseen periféricos limitados, implementación de diferentes sistemas operativos y algunos son diseñados para realizar tareas específicas.

Están compuestos por tres componentes principales: Procesador, Dispositivos de Almacenamiento y Periféricos. Los cuales varían según los parámetros de diseño requeridos. Para sistemas embebidos pre diseñados o de uso general como lo son la BeagleBoard¹¹, PandaBoard¹² o Raspberry Pi¹³. Utilizan arquitecturas de

¹¹ Sistema embebido de medianas prestaciones que permite a novatos, innovadores e ingenieros la implementación de proyectos.

¹² Sistema embebido de altas prestaciones con una mayor cantidad de periféricos diseñada para ser correr varias aplicaciones simultáneamente.

¹³ Sistema embebido de muy bajo costo diseñado para correr aplicaciones cotidianas de un computador de bajas prestaciones.

procesador similares, todas son arquitectura ARM¹⁴, utilizan una memoria SD/MicroSD¹⁵ como dispositivo de almacenamiento y memoria de trabajo RAM¹⁶. [35][36][37]



Fig. 10 Sistema Embebido Raspberry Pi [35]

4.4.1. Software

Los sistemas embebidos requieren la instalación de un sistema operativo el cual está diseñado para administrar de forma eficiente los recursos del dispositivo, este permite la comunicación hardware/software con el fin de ejecutar las tareas para las que fue programado.[18] Algunos S.O. Embebidos que se trabajan actualmente son:

- Android: Plataforma móvil desarrollada en base Linux creada por Google, junto con aplicaciones middleware está diseñada para ser utilizada en teléfonos inteligentes, tablets, televisores, etc. Diseñado principalmente para la arquitectura ARM.

¹⁴es una familia de procesadores diseñado por AcornComputers y desarrollado por Advanced RISC Machines Ltd.

¹⁵Secure Digital. Dispositivo digital de almacenamiento.

¹⁶del Ingles *random-accessmemory*. "Memoria de Acceso Aleatorio"

Tiene una comunidad de desarrolladores grande al ser una plataforma de software libre y de código abierto. [19][20][21]

- Linux: Sistema operativo multiplataforma, desarrollado para las arquitecturas x86, x86-64 y ARM. Por lo cual puede ser ejecutado tanto en computadores convencionales como en sistemas embebidos, aunque muchas de las aplicaciones para Ubuntu están para las tres arquitecturas hay herramientas de compilación cruzada gratuitas para llevar las aplicaciones de una arquitectura a otra.[22][23][24][26]

4.5. Pure Data

Es un entorno de programación por medio de bloques para audio, video y procesamiento de imagen. Es un software libre desarrollado por Miller Puckette el cual tiene una gran cantidad de desarrolladores trabajando en nuevas librerías, por lo que todo el paquete es en gran medida un esfuerzo de la comunidad.¹⁷

Es un sistema multiplataforma en software, por lo que es muy portátil, existen versiones para Win32, IRIX, GNU / Linux, BSD, Mac OSX y se ejecuta en cualquier equipo.

Fue creado con el fin de explorar maneras alternativas de programación por bloques, y para que la implementación de aplicaciones en audio y MIDI se extendiera a gráficos y vídeo.

Pure Data cuenta con librerías externas que son escritas en el lenguaje de programación C++, estas son desarrolladas por los mismos usuarios y con el tiempo se incluyen en las librerías estándar de Pure Data.[27]

¹⁷La página web de la comunidad de Pure Data es: <http://puredata.info/>

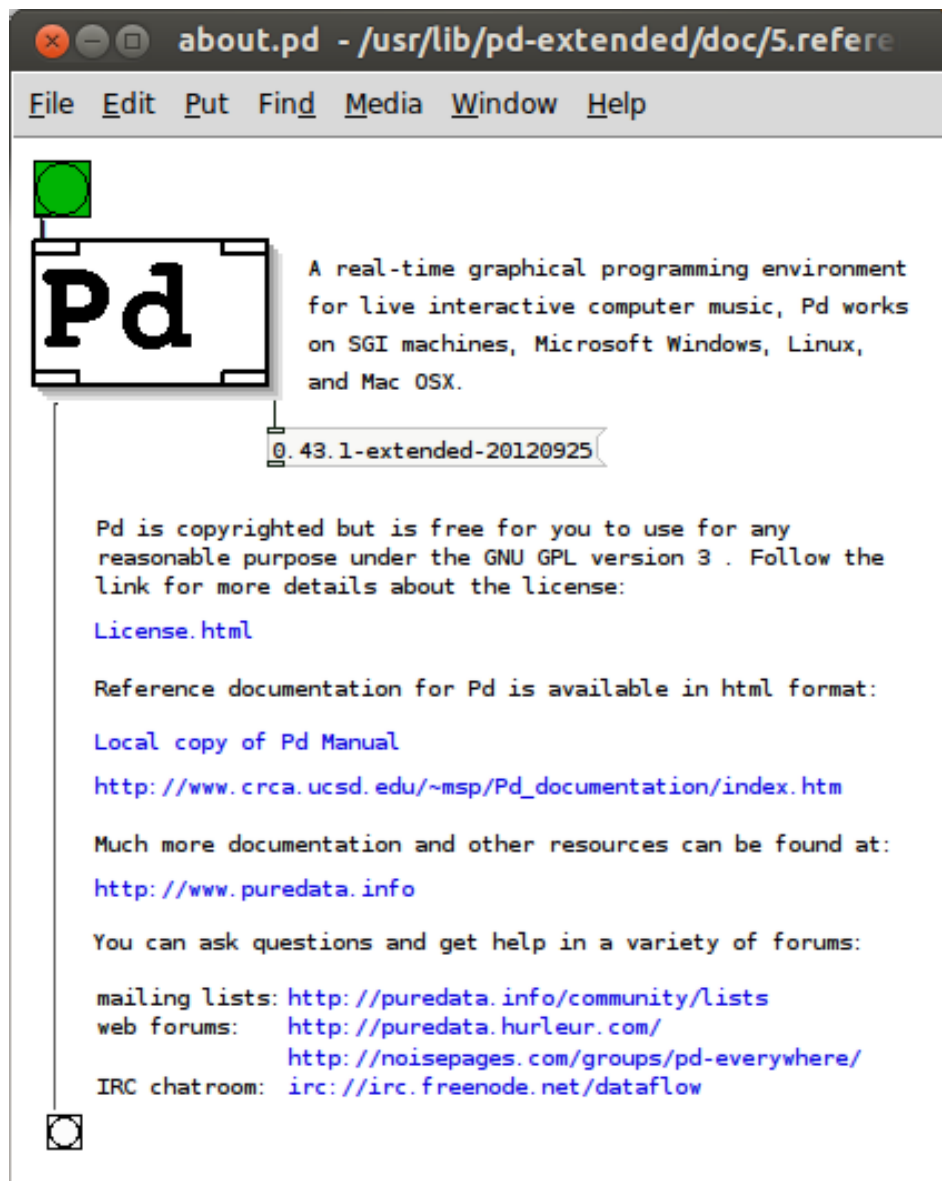


Fig. 11 Sección de documentación e información de Pure Data. Fuente: Propia

5. DESARROLLO INGENIERIL

En el proyecto de grado se desarrolló un controlador MIDI no convencional implementado en un sistema embebido, utilizando el Kinect. El sistema utiliza el protocolo MIDI para controlar hardware o software remoto vía WiFi por medio de la captura de movimientos. El desarrollo se describe en el diagrama de bloques presentado en la Figura 13. A continuación se describen las etapas de desarrollo del proyecto:

5.1. Adecuación S.E.

Los S.E. son dispositivos desarrollados a partir de un co-diseño hardware/software, el cual se basa en la interacción de los dos componentes de una forma eficiente, con el fin de mejorar los procesos del sistema.

En el proyecto se implementaron sistemas embebidos comerciales, por lo que fue necesario determinar cuál era el sistema operativo óptimo para trabajar, se utilizó Linux la distribución de Ubuntu, puesto que es un S.O. versátil, tiene un gran soporte, es libre, adaptable y tiene fuentes de desarrollo completas. [18] Estas ventajas son las principales motivaciones por las cuales se emplea el Linux en el desarrollo de la tesis. Existen diferentes versiones las cuales están diseñadas específicamente para cada procesador o memoria, con el fin de obtener un mejor rendimiento dependiendo del hardware, algunas son pre-cargadas con módulos innecesarios y pueden afectar el rendimiento del mismo.

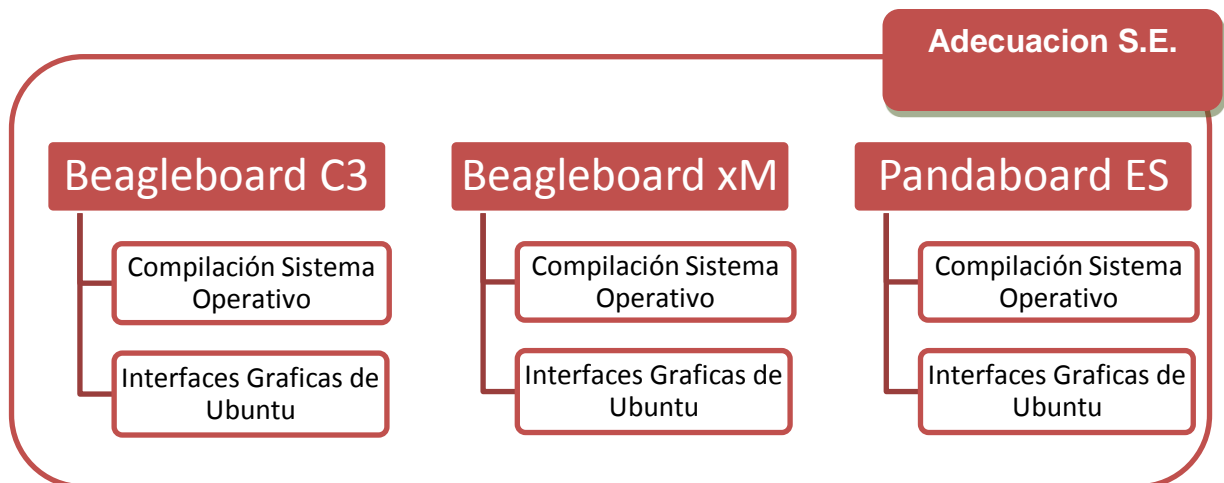


Fig. 12 Diagrama de bloques Adecuación S.E.

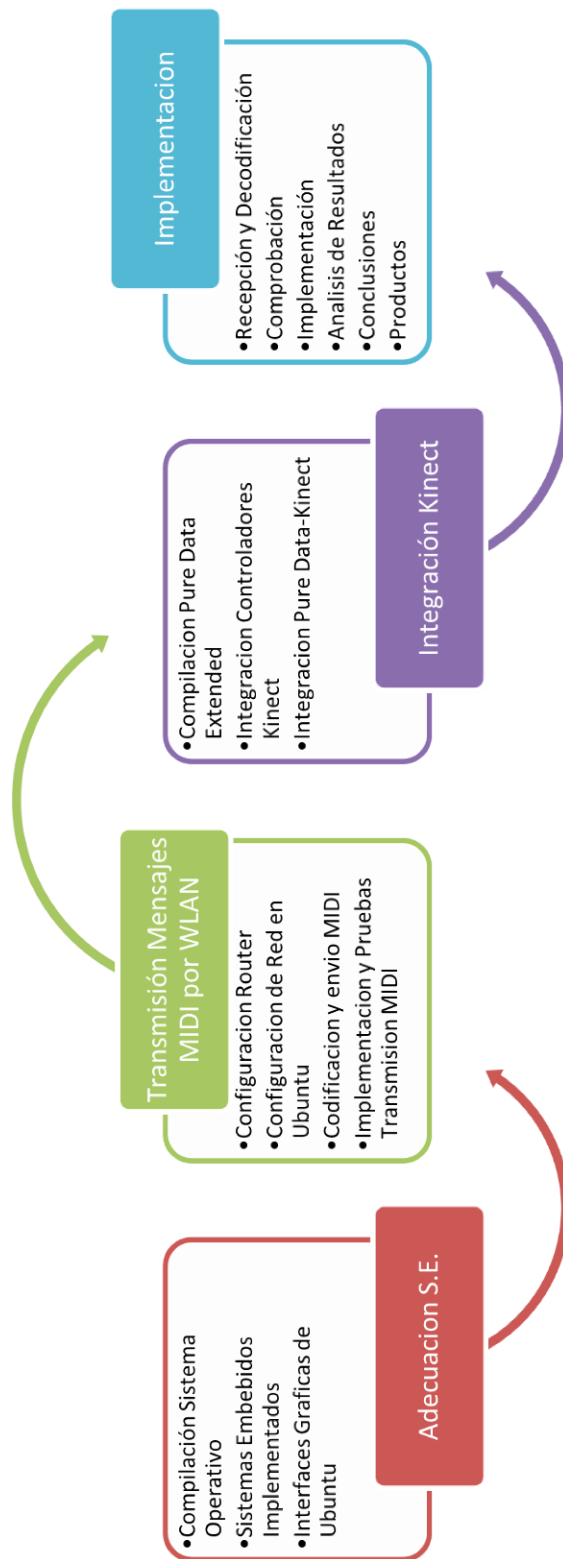


Fig. 13 Descripción Desarrollo Controlador MIDI con Convencional

5.1.1 Compilación Sistema Operativo

Se efectuó el análisis de varias versiones de Ubuntu en tres sistemas embebidos diferentes, con imágenes pre compiladas e imágenes compiladas manualmente, el proceso del desarrollo general para obtener la versión óptima se describe en el diagrama de bloque de la Figura 14.

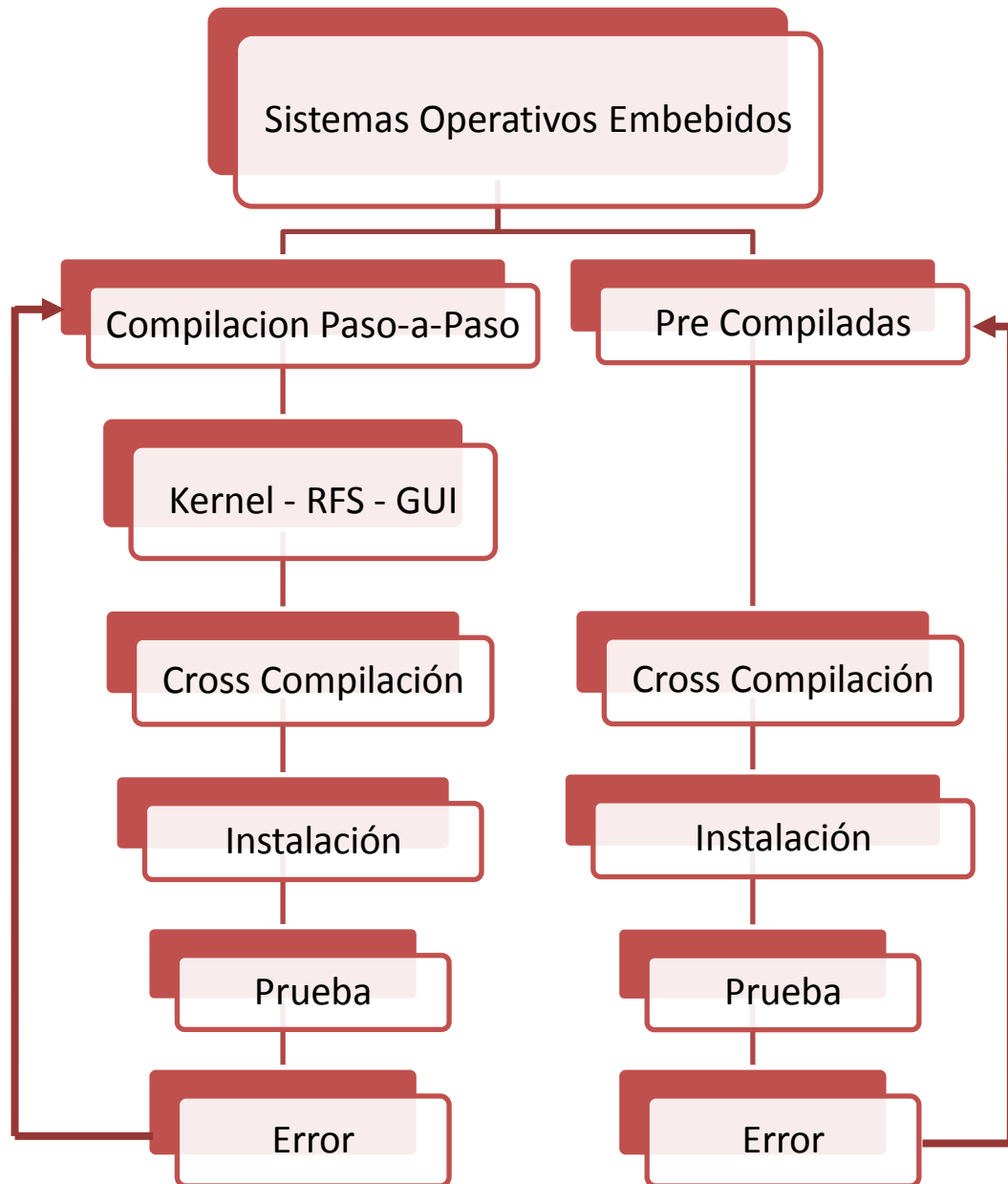


Fig. 14 Descripción proceso instalación Ubuntu en sistemas embebidos

5.1.1.1. Compilación Paso-a-Paso

El Kernel es el encargado de gestionar y controlar directamente los recursos de hardware, este fue escogido en base a las recomendaciones dadas por las versiones de Ubuntu. El RFS o sistema de Archivos Raíz son los archivos encargados del arranque del sistema operativo, poseen una estructura de directorios jerárquicos en forma de árbol, donde se almacenan aplicaciones, librerías y archivos de trabajo. La GUI o interfaz gráfica de usuario es la que se encarga de brindar un entorno grafico amigable con el usuario, permitiendo su configuración por medio de ventanas, menús, explorador de archivos. [18]

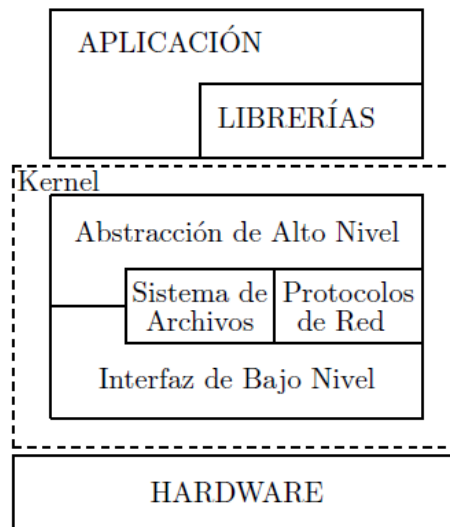


Fig. 15 Arquitectura general de un Sistema de Linux Embebido [39]

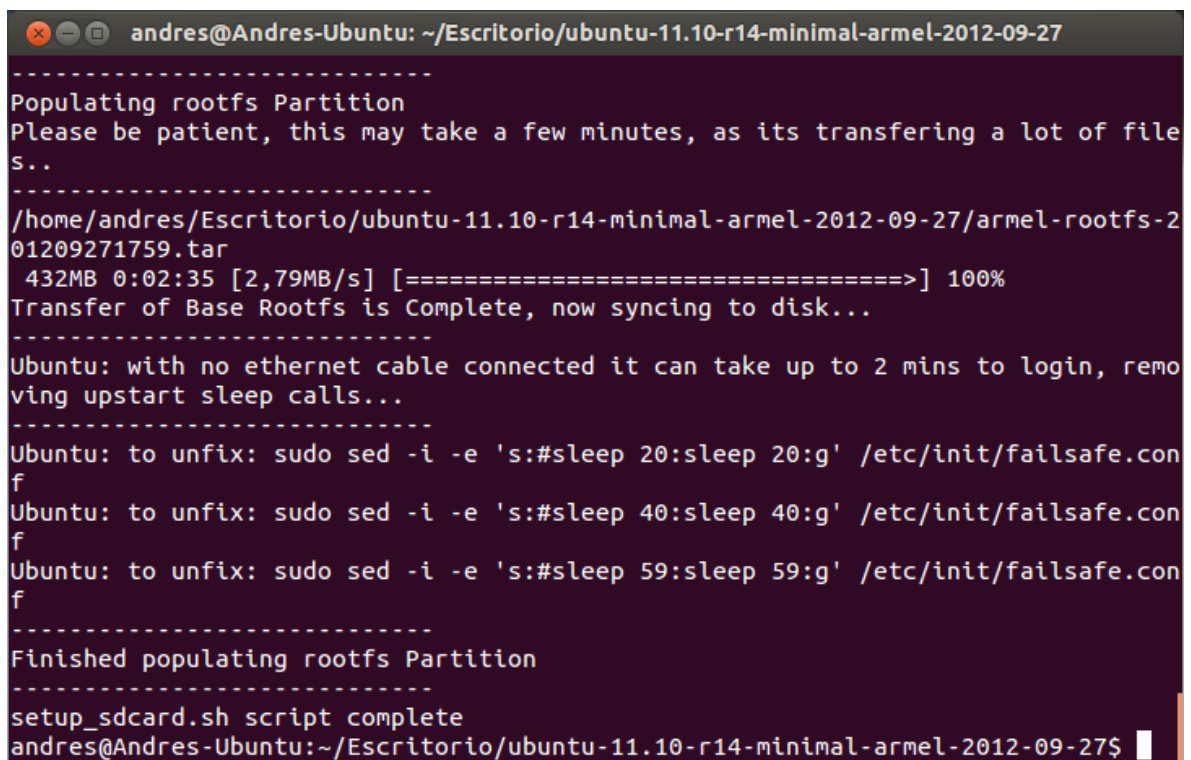
En la Figura 15 se presenta la estructura del Kernel, el cual se compone de: la Interfaz de Bajo Nivel y la Abstracción de Alto Nivel, generalmente es independiente del hardware y provee abstracciones comunes a todos los sistemas Unix, incluyendo procesos, archivos, sockets y señales. Entre estos dos niveles se encuentran los Componentes de interpretación, interactúan con las estructuras de datos recibidas o transmitidas por los dispositivos. Los tipos de sistemas de archivos (FAT32, NTFS, EXT2, EXT3, EXT4) y los protocolos de red son ejemplos de estructuras de datos.

Cuando el Kernel se encuentra operando, requiere manejar una estructura adecuada del sistema de archivo, que se denomina RFS el cual es cargado al momento del arranque del sistema operativo, en este caso solo carga librerías

necesarias para su ejecución, por lo que si se requieren más deberán ser instaladas manualmente. Una vez se tienen determinados estos parámetros se enlazan gracias al script¹⁸ de cross compilación: [18]

Setup_sdcard.sh

El comando se encarga de preparar la memoria para que inicie desde un Sistema Embebido como se observa en la Figura 16, el script crea la partición de arranque donde se encuentra el Kernel del sistema y la partición de archivos de ejecución RFS.



```
andres@Andres-Ubuntu: ~/Escritorio/ubuntu-11.10-r14-minimal-armel-2012-09-27
-----
Populating rootfs Partition
Please be patient, this may take a few minutes, as its transferring a lot of file
s..
-----
/home/andres/Escritorio/ubuntu-11.10-r14-minimal-armel-2012-09-27/armel-rootfs-2
01209271759.tar
 432MB 0:02:35 [2,79MB/s] [=====>] 100%
Transfer of Base Rootfs is Complete, now syncing to disk...
-----
Ubuntu: with no ethernet cable connected it can take up to 2 mins to login, remo
ving upstart sleep calls...
-----
Ubuntu: to unfix: sudo sed -i -e 's:#sleep 20:sleep 20:g' /etc/init/failsafe.con
f
Ubuntu: to unfix: sudo sed -i -e 's:#sleep 40:sleep 40:g' /etc/init/failsafe.con
f
Ubuntu: to unfix: sudo sed -i -e 's:#sleep 59:sleep 59:g' /etc/init/failsafe.con
f
-----
Finished populating rootfs Partition
-----
setup_sdcard.sh script complete
andres@Andres-Ubuntu:~/Escritorio/ubuntu-11.10-r14-minimal-armel-2012-09-27$
```

Fig. 16 Ventana compilación Ubuntu minimal. Fuente: Propia

Al finalizar se inserta la memoria en el Sistema Embebido y se corren los comandos necesario para instalar la GUI previamente escogida dependiendo del mismo.

¹⁸ Conjunto de instrucciones generalmente almacenadas en un archivo de texto que deben ser interpretados línea a línea en tiempo real para su ejecución

Se realizan las pruebas del sistema operativo, en las que se analiza velocidad de inicio, si el sistema es estable, si soporta las librerías necesarias, etc. Esto esperando encontrar el sistema ideal con la interfaz gráfica adecuada para su óptimo funcionamiento.

5.1.1.2. Imagen Pre-Compilada

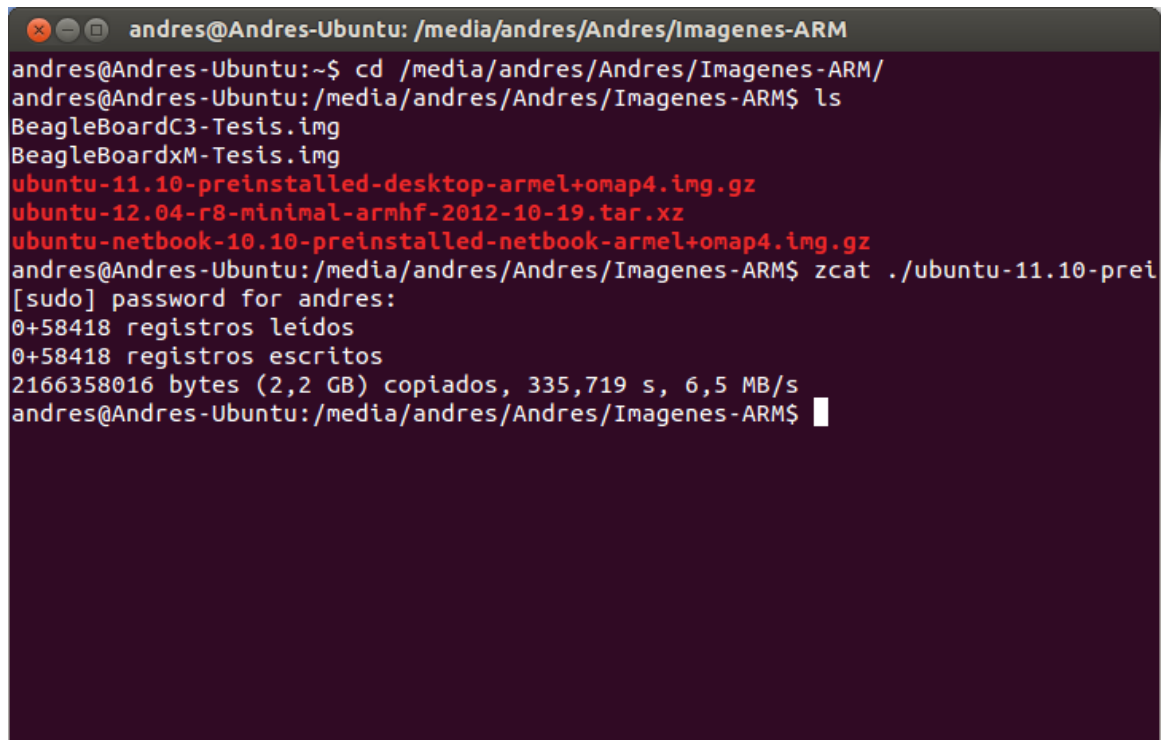
Por este método contamos con imágenes desarrolladas oficialmente por Ubuntu en las que se encuentra un Kernel, RFS y GUI definidos, existen diferentes versiones para cada tipo de Sistemas Embebidos.

La imagen pre-compilada contiene un paquete de librerías más amplio lo cual las hace más densas que la versión minimal, pero ahorran mucho tiempo a la hora de descargar librerías extra o activar módulos del Kernel. Se cross-compilan con el comando:

```
$ zcat ./"Nombre de la imagen" |sudo dd bs=4M  
of=/dev/"dirección dispositivo SD" ; sudo sync
```

Al finalizar el script como se muestra en la Figura 17 se inserta la memoria en el sistema embebido, y a comparación con en el método Paso-a-Paso, este va a iniciar una interfaz gráfica automáticamente, la cual guiará al usuario a través de la instalación del Sistema Operativo, similar al proceso que se lleva en una computadora común.

Una vez instalado se hacen las pruebas necesarias para determinar el sistema operativo ideal para el sistema, siguiendo los mismos parámetros analizados en el primer método: velocidad de inicio, si el sistema es estable o si soporta las librerías necesarias.

A terminal window titled 'andres@Andres-Ubuntu: /media/andres/Andres/Imagenes-ARM'. The user enters 'cd /media/andres/Andres/Imagenes-ARM/' and 'ls', listing files: 'BeagleBoardC3-Tesis.img', 'BeagleBoardXM-Tesis.img', 'ubuntu-11.10-preinstalled-desktop-armel+omap4.img.gz', 'ubuntu-12.04-r8-minimal-armhf-2012-10-19.tar.xz', and 'ubuntu-netbook-10.10-preinstalled-netbook-armel+omap4.img.gz'. Then, the user runs 'zcat ./ubuntu-11.10-prei', prompting for a password. The output shows file statistics: '0+58418 registros leídos', '0+58418 registros escritos', and '2166358016 bytes (2,2 GB) copiados, 335,719 s, 6,5 MB/s'.

```
andres@Andres-Ubuntu: /media/andres/Andres/Imagenes-ARM
andres@Andres-Ubuntu:~$ cd /media/andres/Andres/Imagenes-ARM/
andres@Andres-Ubuntu:/media/andres/Andres/Imagenes-ARM$ ls
BeagleBoardC3-Tesis.img
BeagleBoardXM-Tesis.img
ubuntu-11.10-preinstalled-desktop-armel+omap4.img.gz
ubuntu-12.04-r8-minimal-armhf-2012-10-19.tar.xz
ubuntu-netbook-10.10-preinstalled-netbook-armel+omap4.img.gz
andres@Andres-Ubuntu:/media/andres/Andres/Imagenes-ARM$ zcat ./ubuntu-11.10-prei
[sudo] password for andres:
0+58418 registros leídos
0+58418 registros escritos
2166358016 bytes (2,2 GB) copiados, 335,719 s, 6,5 MB/s
andres@Andres-Ubuntu:/media/andres/Andres/Imagenes-ARM$
```

Fig. 17 Ventana creación imagen pre compilada. Fuente: Propia

5.1.1 Interfaces Graficas de Ubuntu

Existen varias interfaces gráficas que se pueden implementar en Ubuntu y son variaciones del mismo, algunas de ellas son Kubuntu, Lubuntu y Xubuntu. Estas distribuciones incluyen algunos de los paquetes esenciales pero con interfaces gráficas ligeras.

Las cinco interfaces utilizadas fueron KDE, LXDE, XFCE, Gnome y Unity. Gnome solía ser el estándar en Ubuntu hasta la llegada de Unity una interfaz más completa pero a su vez más pesada, está incluida desde la versión 11.04 de Ubuntu.

5.1.1.1 XFCE

La interfaz XFCE fue la primera en ser implementada, con esta se obtuvieron mejoras en cuanto a rendimiento, permitiendo al usuario poder ingresar a los diferentes menús, instalar software complementario y configurar sus periféricos desde la interfaz gráfica. El problema surge al instalar programas de procesamiento digital de señales, el monitor de memoria alcanza el 80% de la memoria por ende disminuye el rendimiento del sistema.

```
$ sudo apt-get install xfce
```

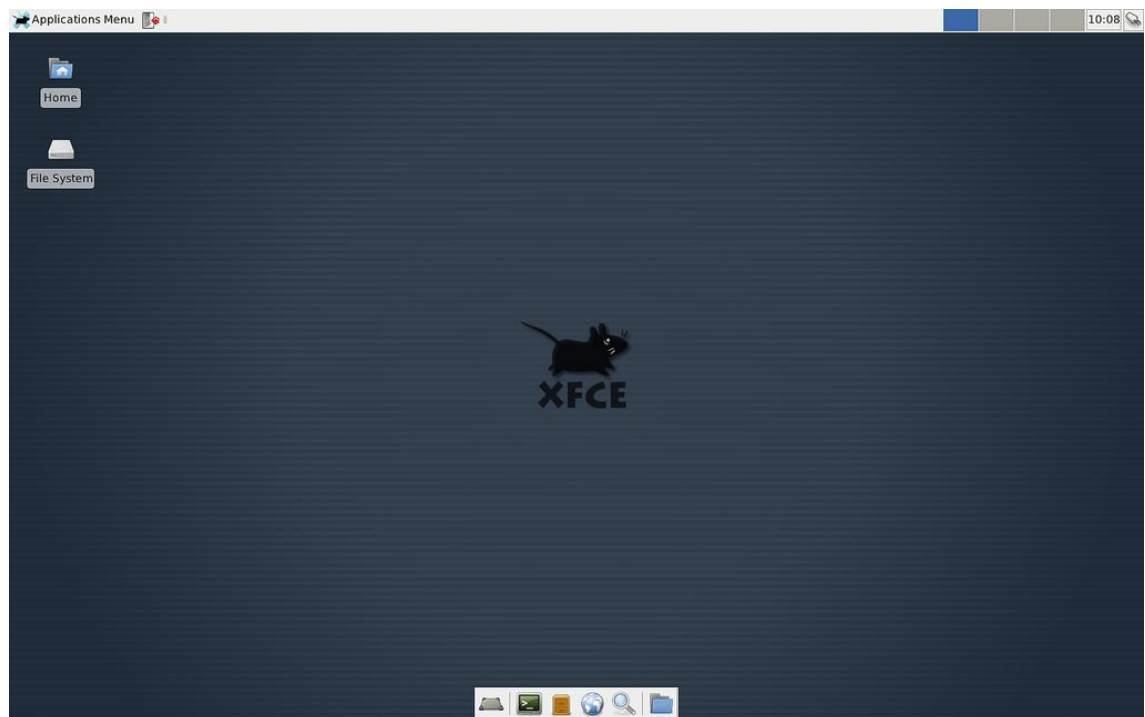


Fig. 18 Interfaz Gráfica Xfce. Fuente: Propia

5.1.1.2 KDE

La interfaz KDE surge de la necesidad de buscar un mejor rendimiento de la Beagleboard y aunque el rendimiento aumentó con respecto al Unity, no superaba al XFCE ya que tan solo ejecutar la terminal e instalar paquetes por más sencillos que fueran, el monitor de memoria alcanzo el 100% por ello se descartó su implementación.

```
$ sudo apt-get install kde
```

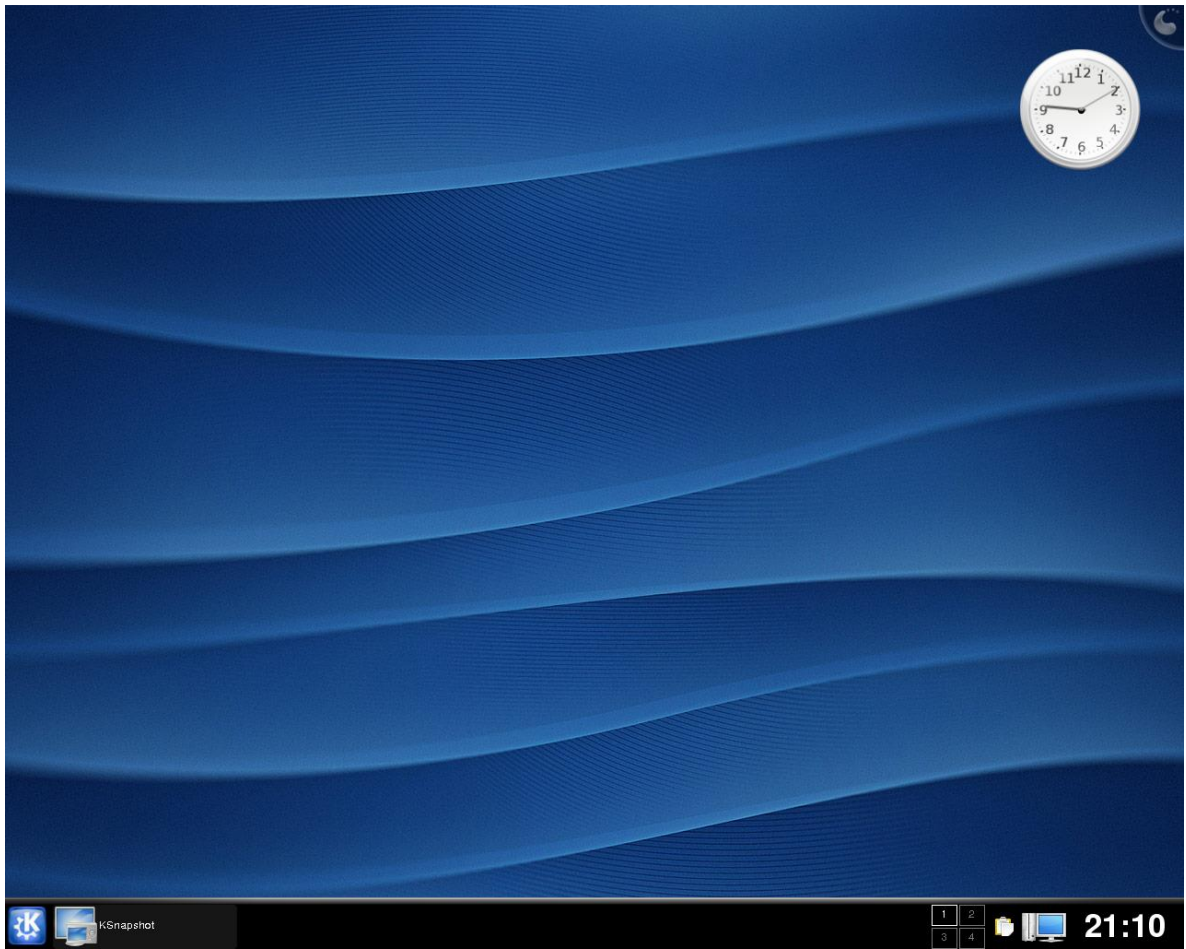


Fig. 19 Interfaz Gráfica Kde. Fuente: Propia

5.1.1.3 Gnome

La interfaz Gnome es una interfaz eficiente para trabajar, las distribuciones de Ubuntu que incluye Gnome podían ser instaladas en computadoras con procesadores mono núcleo y con poca memoria RAM, se concluye que puede ser una solución óptima. Al hacer el proceso de cross-compilación en la Beagleboard se encuentra una mejora con respecto a KDE pero aun así no superaba a XFCE el cual era el objetivo. Es una interfaz muy amigable y muy conocida por los usuarios antiguos de Ubuntu entonces esa es una ventaja sobre el XFCE a pesar de su rendimiento.

```
$ sudo apt-get install gnome-panel
```




Fig. 20 Interfaz Gráfica Gnome. Fuente: Propia

5.1.1.4 LXDE

La interfaz LXDE es una interfaz ligera donde programas de procesamiento digital de señales tienen un mejor rendimiento y el sistema tiene un comportamiento más estable, teniendo en cuenta lo anterior se llevaron a cabo las primeras pruebas de envío MIDI a través de red.

```
$ sudo apt-get install lxde
```



Fig. 21 Interfaz Gráfica Lxde. Fuente: Propia

5.1.1.5 Unity

La interfaz gráfica Unity fue implementada únicamente en la Pandaboard ES debido a su procesador doble núcleo, es una interfaz más amigable con el usuario pero a su vez consume una cantidad considerable de recursos del sistema. Se instala con el comando:

```
$ sudo apt-get install ubuntu-desktop
```

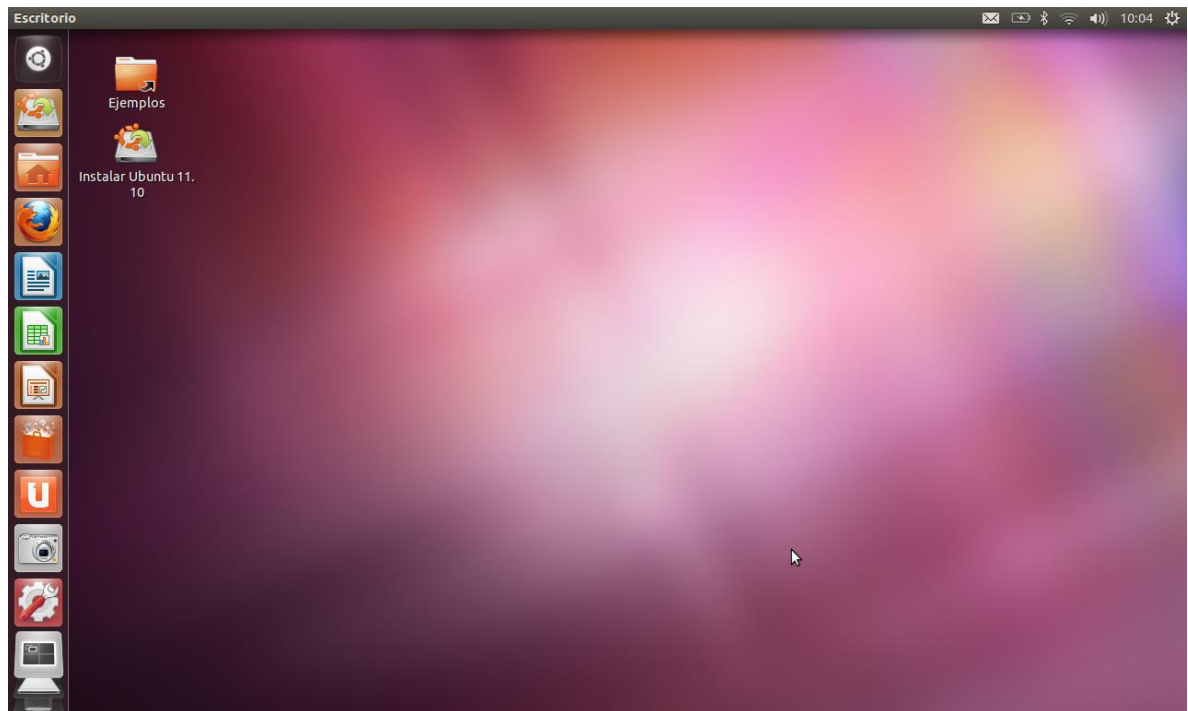


Fig. 22 Interfaz gráfica Unity. Fuente: Propia

5.1.2 Sistemas Embebidos Implementados

Estos procesos se llevaron a cabo en los siguientes sistemas embebidos: Beagleboard C3, Beagleboard xM y Pandaboard ES. A continuación se describe el procedimiento específico para cada uno:

5.1.2.1 Beagleboard C3

Para el desarrollo en este Sistema Embebido es importante tener un buen manejo de herramientas de cross-compilación, debido a sus limitados recursos de hardware es necesario conocer como optimizar el sistema, los resultados obtenidos fueron los siguientes:

Las pruebas iniciales se llevaron a cabo en la versión 11.04 de Ubuntu, se trabajó la cross-compilación de Kernel y Root File System pre-compilados por el grupo de desarrollo de Ubuntu, para esto se ejecuta el comando:

```
$ zcat ./ubuntu-11.04-preinstalled-headless-armel+omap.img.gz |sudo dd bs=4M of=/dev/sdX ; sudo sync
```

Al compilar la imagen en la tarjeta SD se instalaron paquetes de inicio que hicieron tardar más de lo normal el inicio del sistema, por lo que la descompresión del sistema tarda más de lo esperado. Se intentó arreglar el problema desinstalando manualmente los paquetes pero sin obtener un mejor rendimiento del S.E.

El siguiente sistema operativo fue Ubuntu 11.10, aunque las mejoras obtenidas no fueron significativas para el funcionamiento de varios programas simultáneamente. La ventaja en cuanto a la versión 11.04 fue la implementación de controladores de video diseñados para este sistema embebido. Se compila en la SD con el siguiente comando:

```
$ zcat ./ubuntu-11.10-preinstalled-desktop-armel+omap.img.gz |sudo dd bs=4M of=/dev/sdX ; sudo sync
```

Por último se implementó una compilación paso-a-paso, utilizando un Kernel con únicamente los módulos necesarios para el arranque del sistema y de los periféricos. Se diseñó una imagen optimizada para el funcionamiento exclusivo en la Beagleboard C3.

Estas imágenes son llamadas Ubuntu minimal y no incluyen interfaz gráfica, el proceso de instalación debe empezar desde la configuración de la tarjeta de red hasta la instalación de paquetes por medio de la terminal utilizando comandos shell. El Primer paso a seguir es descargar y se descomprimir la imagen:

```
$ wget http://rcn-ee.net/deb/rootfs/natty/ubuntu-11.04-r7-minimal-armel.tar.xz
$ tar xJf ubuntu-11.04-r3-minimal-armel.tar.xz
```

En esta carpeta se encuentra el script que ejecuta todo el proceso, en este punto se puede seleccionar Kernel y Root File System, pero es recomendable utilizar los que están incluidos en la imagen:

```
$ sudo ./setup_sdcard.sh --mmc /dev/sdX -uboot beagle_cx
```

Al iniciar la Beagleboard se debe configura la red para poder descargar librerías, controladores de video y paquetes de desarrollo.

```
$ sudo ifconfig -a
$ sudo dhclient eth0 -v
$ sudo apt-get update
$ sudo apt-get upgrade
$ sudo apt-get install xserver-xorg-video-omap3 network-
manager build-essential libavformat-dev ffmpeg cmake
```

Al realizar las pruebas con el Kinect se presenta una limitante de procesamiento, debido a que este requiere un mejor procesador, por lo cual este Sistema Embebido puede funcionar como un controlador MIDI inalámbrico, pero la recepción de datos se debe hacer por medio de interfaces que no realicen procesamiento de imágenes, por su limitación de hardware.



Fig. 23 Beagleboard C3. Fuente: Propia

5.1.2.2 Beagleboard xM

Debido a las limitaciones de hardware de la Beagleboard C3, surge la necesidad de adquirir un nuevo sistema embebido con un mejor procesador y memoria para obtener mejores resultados.

Se llevó un proceso de compilación similar al de la Beagleboard C3 con la diferencia que este disminuye el tiempo de instalación y el procesamiento en general es más eficiente por lo que se pudo probar más distribuciones de Ubuntu con diferentes interfaces gráficas.

Se trabajó con la imagen pre compilada de Ubuntu 12.04, esta se tomó alrededor de diez minutos en su descompresión e instalación, se obtuvo un Ubuntu similar al de escritorio, exitosamente se pudo enviar mensajes MIDI por WiFi y el procesamiento de señales fue aceptable, pero surgen problemas con los controladores del Kinect, no funcionaban de la forma esperada, por lo que este Sistema Operativo Embebido no posee controladores optimizados para el chip de video, obteniendo latencia al momento de adquirir y procesar los datos.

Fue necesario volver a la versión 11.10 de Ubuntu, su descompresión e instalación fue exitosa y rápida, su instalación tardó alrededor de diez minutos, se descargaron los controladores y paquetes necesarios para hacer pruebas con el Kinect. Los resultados obtenidos no fueron óptimos a pesar de contar con controladores de video optimizados para la Beagleboard xM y se determinó realizar un cambio de GUI, por lo que se crea la imagen específica con el comando:

```
$ sudo ./setup_sdcard.sh --mmc /dev/sdX --uboot beagle_xm
```

Al incluir controladores de video se escoge la versión 11.10 de Ubuntu con el fin de optimizar el sistema, se descarga la versión mínima de esta versión y se implementó un Kernel optimizado, al contar con mejor procesador se decide instalar no solo la interfaz gráfica si no todo el entorno de desarrollo, en este caso al obtener el mejor rendimiento con LXDE en la Beagleboard C3, se escoge el entorno de desarrollo Lubuntu, con esto se obtiene un completo funcionamiento de los servicios, y no es necesario activar los módulos MIDI ni de red manualmente.

```
$ sudo apt-get install lubuntu-desktop
```

Se logró optimizar el sistema obteniendo mejores resultados del procesamiento de imagen y sonido, su latencia fue mínima con lo que la adquisición de datos se pudo hacer en tiempo real, se probaron los ensambles de Pure Data con los drivers Freenect del Kinect y el resultado fue exitoso, se logró la captura de datos de profundidad convirtiéndolos en mensajes MIDI para posteriormente enviarlos por medio de WiFi.

Buscando implementar la última generación de sistemas embebidos se encuentra uno con mejor procesamiento a un costo considerablemente bajo teniendo en cuenta su mejora en cuanto a hardware.



Fig. 24 Beagleboard xM. Fuente: Propia

5.1.2.3 Pandaboard ES

Las características de hardware de la Pandaboard son superiores a las de la Beagleboard, proporcionando un mejor rendimiento gracias al procesador doble núcleo, debido a este no existe la necesidad de utilizar imágenes compiladas manualmente, la idea era minimizar el proceso teniendo en cuenta todo el desarrollo ingenieril realizado en las dos Beagleboard son la base para poder implementar y entender otros tipos de sistemas más completos.

Se utilizan imágenes pre-compiladas, la ventaja es que los paquetes innecesarios no afectan al sistema, si se quisiera compilar una interfaz gráfica más ligera para optimizar más el sistema se podría realizar el proceso manual de compilación con los mismos comandos utilizados en los otros S.E.

Para la instalación y cross-compilación del Ubuntu 12.10, se propone un nuevo sistema de instalación que funciona exclusivamente para sistemas OMAP 4,

donde la creación de la SD se hace desde el mismo S.E., utilizando una memoria USB para iniciar el sistema y continuar el proceso de instalación tal cual se haría en su versión para escritorio, al parecer es sencillo pero al ser tan reciente no es compatible con muchas librerías ya probados anteriormente, esta imagen solo se crea con el fin de conocer el nuevo proceso y ver el rendimiento de esta S.O.



Fig. 25 Pandaboard ES. Fuente: Propia

Después de varias pruebas se decide implementar la versión 12.04 de Ubuntu con interfaz gráfica Gnome y Unity, se logra ejecutar el tiempo real el Kinect y los cambios son notorios con respecto a la Beagleboard, obteniendo un procesamiento de señales más eficiente. [Ver Tabla 3]

	Beagleboard C3	Beagleboard xM	Pandaboard ES
Procesador	720 MHz	1 GHz	1.2 GHz
Tipo de Procesador	ARMCortexA8	ARMCortexA8	Dual Core ARM Cortex-A9
Memoria RAM	256 MB	512 MB	1 GB
GPU	PowerVR SGX530	PowerVR SGX2D/3D	PowerVR SGX540@384 MHz
Salida de Video	DVI-D, S Video	DVI-D, S Video	DVI-D, HDMI (Full HD)
Tipo de Memoria	SD	MicroSD	SD
Ethernet	No	Si	Si
WiFi	No	No	Si
Bluetooth	No	No	Si
Power Supply	5V	5V	5V
Puertos de Sonido	Conector 3.5 mm I/O	Conector 3.5 mm I/O	Conector 3.5 mm I/O
Puertos USB	1	4	2
Costo	US\$ 125	US\$ 149	US\$ 162
Dimensiones	7.62 cm x 7.62 cm	8.25 cm x 8.25 cm	10 cm x 11 cm

Tabla 3 Comparación especificaciones sistemas embebidos [36][37]

Después de las diferentes pruebas se llega a la conclusión que la Pandaboard ES es la mejor opción por la relación de costo-beneficio como se observa en la tabla 3, a pesar de eso no se descarta la implementación de más proyectos en la Beagleboard, a pesar de sus prestación de hardware también demuestra una gran capacidad de desarrollo en el campo de la ingeniería de sonido.

5.2. Transmisión mensajes MIDI por WLAN

En este proceso es necesario conocer el funcionamiento y configuración de los routers, de la red con IP estática y la cross-compilación del software para transmitir MIDI por medio del protocolo UDP, como se presenta en la Figura 26.

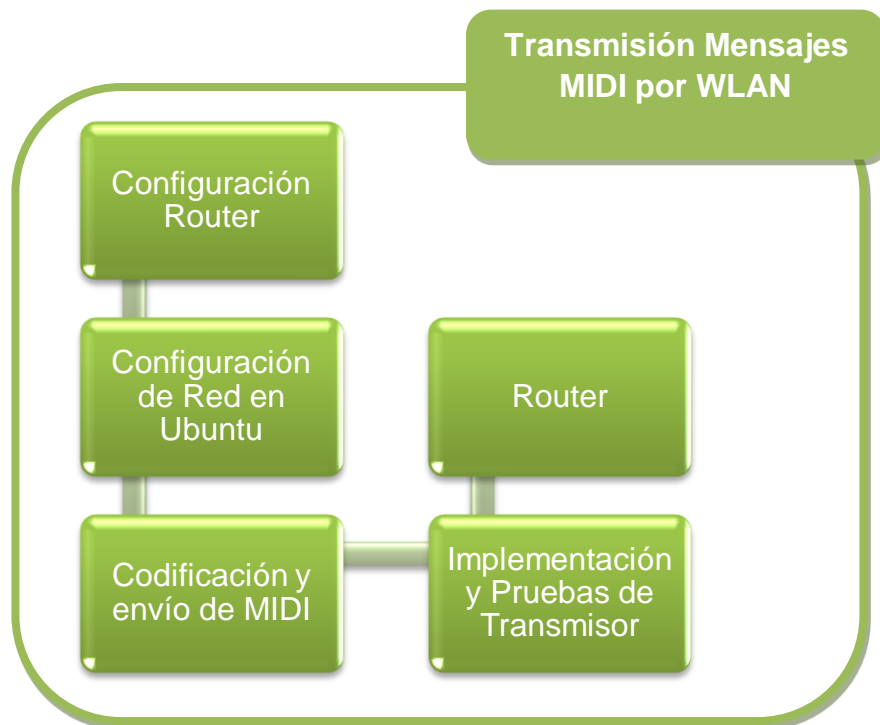


Fig. 26 Transmisión Mensajes MIDI

5.2.1. Configuración Router para una Red Local Inalámbrica (WLAN)

Los routers actuales son diseñados para ser configurados por usuarios finales con o sin experiencia para la creación de una red local inalámbrica (WLAN), el primer paso es determinar si se desean direcciones IP estáticas o dinámicas, las IPs dinámicas brindan la posibilidad de tener un rango amplio de direcciones disponibles para la conexión de dispositivo WiFi, este asigna automáticamente estas direcciones para no tener conflicto de conexión.

Las IPs estáticas están limitadas a las IPs configuradas, esta configuración también debe ser aplicada en el dispositivo asignándole una de las IPs configuradas previamente, provee mayor seguridad pero es menos práctica y más complicada para un usuario final sin experiencia.

5.2.2. Configuración de Red en Ubuntu

La configuración se realiza con direcciones IPs dinámicas en el router, se le asigna una dirección fija al Sistema Embebido con el fin de implementar la función de escritorio remoto como se muestra en la Figura 27 de forma rápida y sencilla para evitar conectar una pantalla al puerto HDMI.

Primero se debe conocer el rango el cual fue configurado el router y la puerta de enlace, luego se modifican las configuraciones de red del embebido, de forma gráfica o por línea de comando.

En la ventana de conexiones se encuentran las pestañas de Cableada o Inalámbrica como se muestra en la Figura 53, dependiendo de la conexión del dispositivo se elegirá uno de los dos, en cualquiera de los dos casos se selecciona la conexión activa y se edita IPv4, dentro de esta pestaña se modifica el método de *Automático* a *Manual* y se agregan las opciones de: Dirección, Máscara de red y Puerta de enlace, se ingresa preferiblemente una de las ultimas direcciones de red del rango disponible para no tener conflictos con la dirección de los demás dispositivo; en Máscara de red se ingresa: "255.255.255.0" y al final la puerta de enlace que es la dirección IP del router.

Para hacer la configuración por línea de comando es necesario modificar el archivo *interfaces*

```
$ sudo vim /etc/network/interfaces
```

Si dentro del archivo existen comandos como: *iface*, *eth0*, *inet* o *dhcp*, se deben borrar para evitar conflictos y se agregara las configuraciones de red, para esto se debe determinar la interfaz de red que se quiera utilizar, ya sea *eth0* para la interfaz cableada, o *wlan0* para la inalámbrica, dependiendo de eso el archivo cambiara, a continuaciones se muestra un ejemplo para la interfaz cableada:

```
# Configuración IP estática en eth0
auto eth0
iface eth0 inet static
address x.x.x.x
gateway x.x.x.x
netmask 255.255.255.0
```

Se cambian los valores deseados, donde *Address* es la dirección IP que se le quiere designar al dispositivo y *Gateway* es la puerta de enlace del router, una vez configurado se deben reiniciar los servicios de red para que surta efecto con el siguiente comando:

```
$ sudo /etc/init.d/networking restart
```

Con esto se finaliza la configuración de red, para comprobar que todo haya quedado correctamente configurado se ejecuta *ifconfig* para conocer el estado de las interfaces de red activas, en el cual se presentan los parámetros modificados previamente.



Fig. 27 Implementación escritorio remoto desde una Tablet. Fuente: Propia

5.2.3. Codificación y envío de MIDI

El protocolo de Datagrama de Usuario se implementa sin la necesidad de una conexión, funciona a través de direcciones IP y ofrece un enlace directo para enviar y recibir datagramas, por lo que se utiliza para enviar audio a través de redes. La ventaja que ofrece el empaquetamiento de datos por medio de datagramas es que se maneja de forma independiente; y Al no requerir conexión se pueden enviar datos sin la necesidad de proporcionar dirección IP o nombre del equipo receptor, esté posee la facultad de ser multicast, mientras el receptor posea el decodificador podrá recibir los datos MIDI, solo se debe asignar un puerto para conocer qué puerto debe ser habilitado por el receptor, cada puerto cuenta con los 16 Canales MIDI predeterminados del protocolo, por lo que se cuenta con un gran número de canales MIDI para enviar datos independientes.



Fig. 28 Diagrama de flujo codificación y envío MIDI

5.2.4. Implementación y Pruebas Transmisor MIDI QmidiNet

La compilación de este software se realiza utilizando la herramienta GCC, al tener acceso al código fuente se puede portar la aplicación a diferentes plataformas, se compila utilizando los comandos:

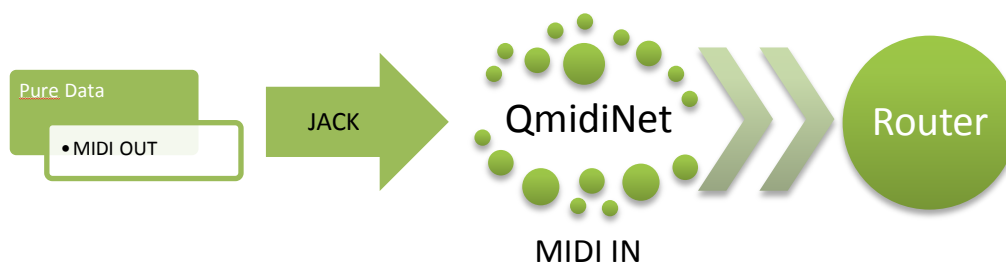


Fig. 29 Diagrama conexión QmidiNet

```
$ ./configure  
$ make
```

La interfaz gráfica permite escoger interfaz de transmisión: red alámbrica o inalámbrica, puerto UDP, número de puertos y secuenciador MIDI, es recomendado utilizar ALSA.

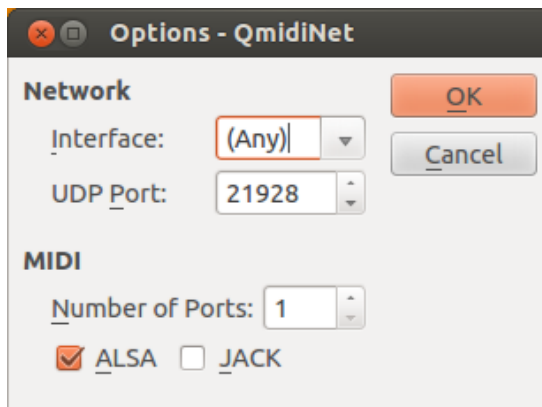


Fig. 30 Interfaz de configuración QmidiNet. Fuente: Propia

Una vez configurado se ingresa a JACK donde aparecen las conexiones MIDI de Sistema Embebido como lo muestra la Figura 26, en la interfaz se pueden interconectar los software que posean entradas y salidas MIDI, como ejemplo se conecta la salida MIDI 1 de Pd a la entrada del puerto 0 de QmidiNet para enviar los datos generados de Pd hacia otro dispositivo.

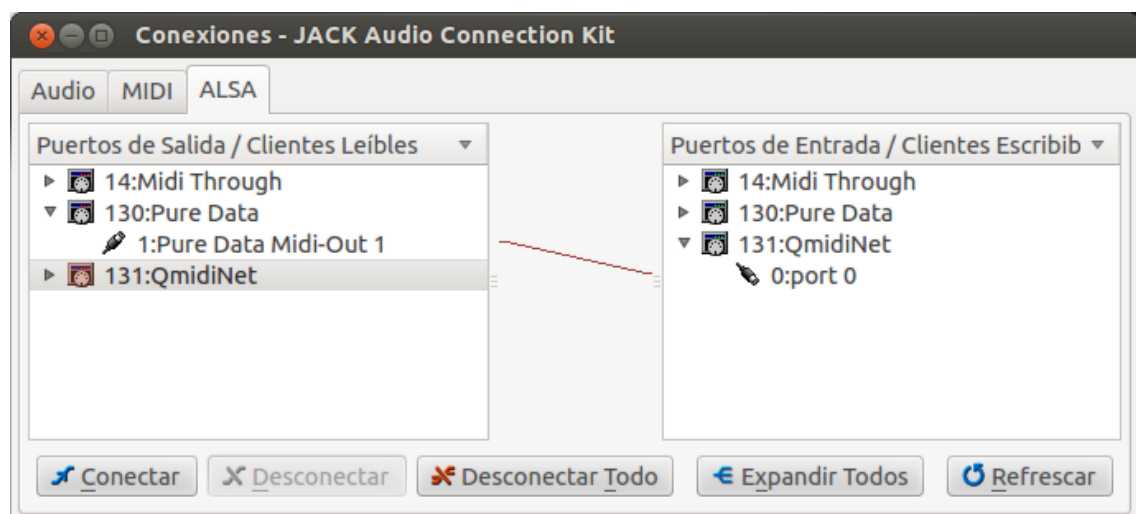


Fig. 31 Entradas y Salidas MIDI de Ubuntu. Fuente: Propia

5.3. Integración Kinect

Esta fase se divide en 3 partes las cuales son cross-compilación de Pure Data Extended, integración de los controladores del Kinect e integración de Pure Data-Kinect, como lo describe la Figura 32.

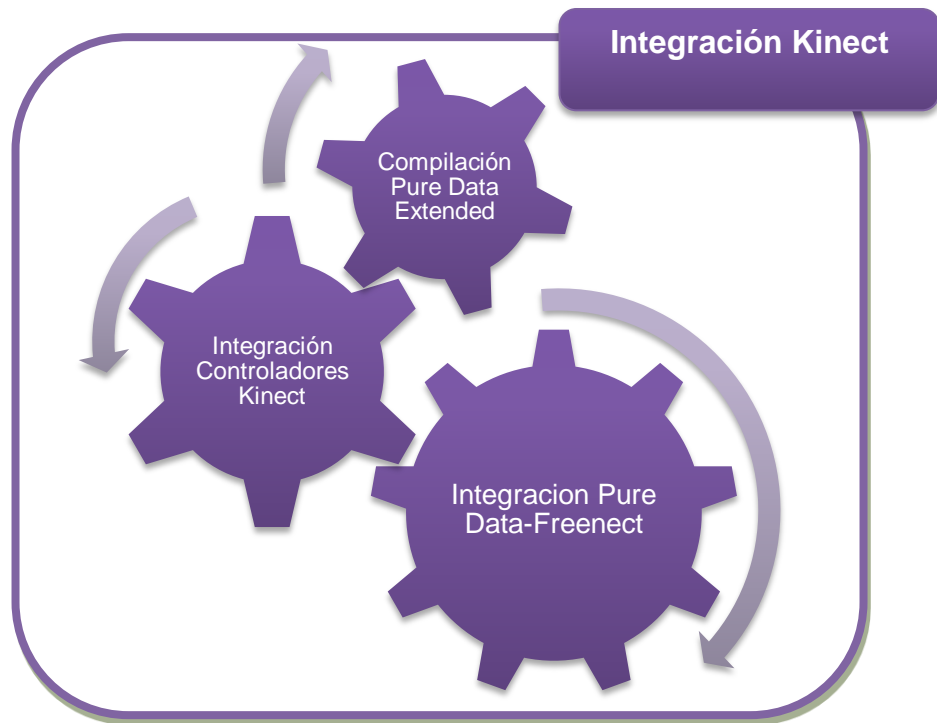


Fig. 32 Integración Kinect

5.3.1. Compilación Pure Data Extended

Pure Data tiene versiones disponibles para diferentes arquitecturas y Sistemas Operativos, en Ubuntu está incluido dentro de sus repositorios pero no la versión extended, la cual se debe implementar debido a sus diferentes “*Externals*”, El diagrama de bloques de la Figura 33 describe el proceso de cross-compilación:

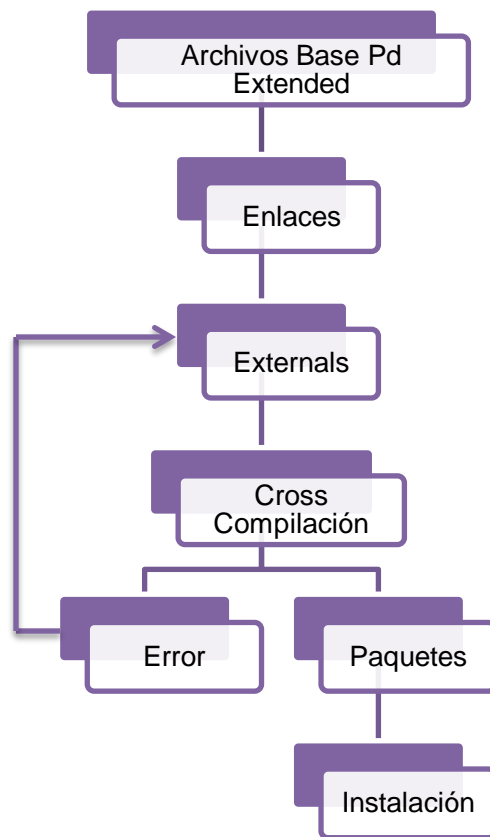


Fig. 33 Descripción proceso compilación Pure Data Extended

El primer paso es instalar todos los paquetes necesarios para poder compilar Pure Data Extended los cuales son:

```
$ sudo apt-get install build-essential git subversion
libmp3lame0 libmp3lame-dev ladspa-sdk dssi-dev libimlib2-dev
libtheora-dev libjack0 libjack-dev fakeroot puredata gem
ttf-dejavu libfftw3-dev libspeex-dev libvorbis-dev libflac-
dev libsndfile1-dev libgavl-dev libmad0-dev dssi-dev liblo-
dev flite1-dev libtk-img tcllib libmp3lame0 libmp3lame-dev
libimlib2-dev libtheora-dev libusb-0.1-4 libusb-dev
liblua5.1-0-dev lua5.1 tcl8.5-dev swig libnetpbm10-dev ruby
nasm libcv-dev libdc1394-22-dev
```

Se debe descargar los códigos fuente de Pd Extended, “Externals” de Pd Extended y Gem de la siguiente manera:


```
git clone git://pure-data.git.sourceforge.net/gitroot/pure-  
data/pd-extended.git  
  
svn co https://pure-data.svn.sourceforge.net/svnroot/pure-  
data/trunk pd-svn  
  
git clone git://pd-gem.git.sourceforge.net/gitroot/pd-  
gem/Gem gem.git
```

A partir de estas carpetas se debe crear una carpeta que enlace todo utilizando accesos directos de la siguiente manera:

```
Cd pd-svn/  
mv pd pd-original  
ln -s ../pd-extended.git pd  
  
ln -s ../gem.git Gem  
cd externals  
ln -s ../../gem.git Gem
```

Una vez enlazados el archivo Make para realizar la compilación se encuentra en la siguiente carpeta:

```
cd pure-data/packages/linux_make  
make install && make package
```

La compilación puede tardar dependiendo del Sistema Embebido entre dos y cinco horas, al final se obtendrá el instalador así que el proceso solo debe ser ejecutado una vez mientras no se cambie de arquitectura.

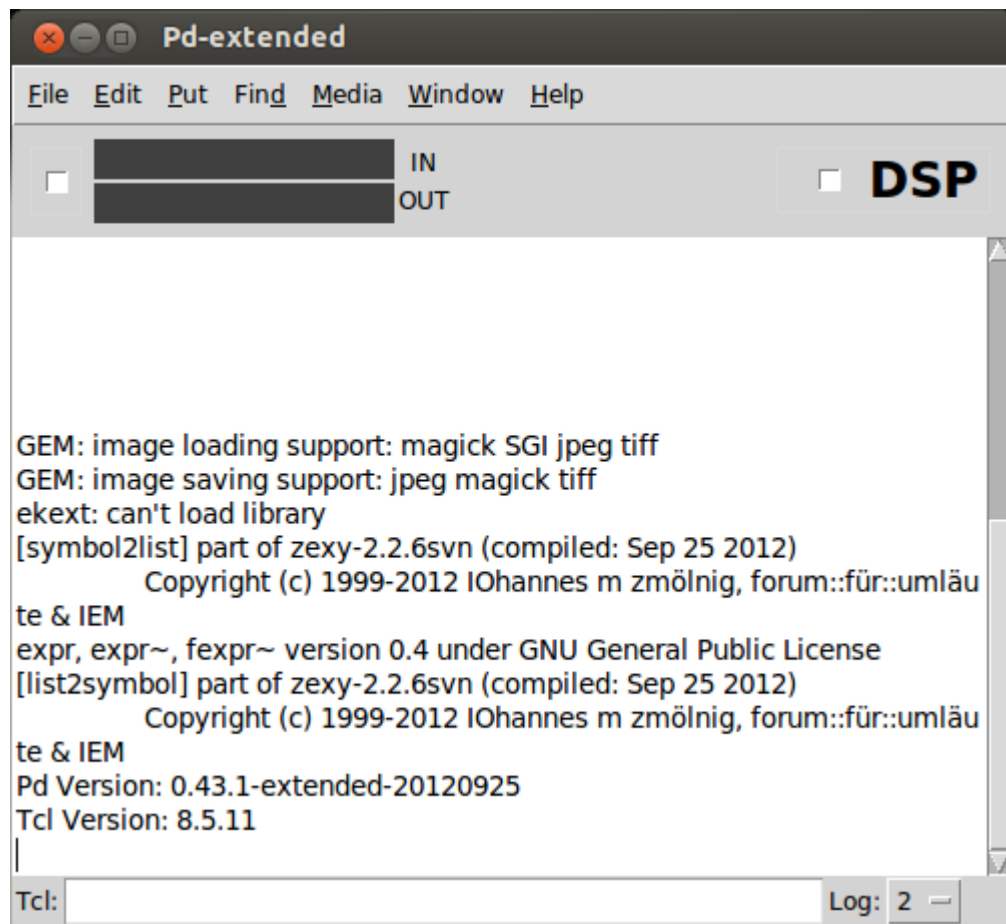


Fig. 34 Interfaz gráfica principal de Pd Extended. Fuente: Propia

5.3.2. Integración Controladores Kinect

Estos controladores se implementan en dos sistemas embebidos por cuestión de procesamiento, el Freenect puede ser ejecutado desde la Beagleboard xM y la Pandaboard ES.

Los paquetes necesarios para compilar los dos controladores son los siguientes:

```

$ sudo apt-get install default-jre git-core cmake freeglut3-
dev pkg-config libxmu-dev libxi-dev libusb-1.0-0-dev doxygen
graphviz mono-complete g++python gcc-multilib default-jdk
libcv-dev

```

5.3.2.1. Instalación Controladores Freenect

Los controladores freenect se compilan con la herramienta cmake¹⁹ previamente instalada, es importante conceder los permisos necesarios para que funcionen correctamente, para esto se implementan reglas de ejecución.

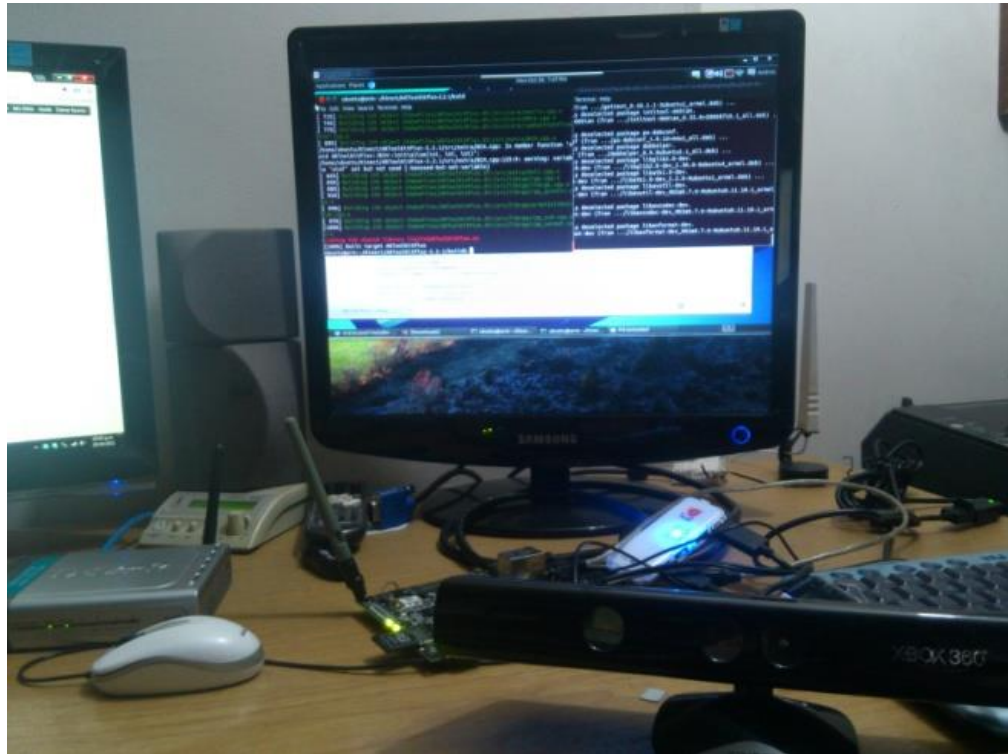


Fig. 35 Ventana de Comando de Compilación Freenect, con el Kinect conectado a la Pandaboard ES. Fuente: Propia

Dentro de la carpeta Kinect se inicia la descarga y la compilación:

```
git clone https://github.com/OpenKinect/libfreenect.git
cd libfreenect
mkdir build && cd build
cmake ..
make
sudo make install
```

¹⁹ Es una herramienta de compilación multiplataforma de código abierto, diseñada para compilar, comprobar y empaquetar software

Al finalizar se agrega al usuario de Ubuntu a los siguientes grupos:

```
$ sudo add user $USER video
$ sudo add user $USER plugdev
```

Se añade una regla de ejecución para no tener la necesidad de ejecutarlo como súper usuario, para esto se edita el archivo “51-kinect.rules” se le agrega lo siguiente:

```
$ sudo gedit /etc/udev/rules.d/51-kinect.rules
```

```
#Rules for Kinect#####
SYSFS{idVendor}=="045e", SYSFS{idProduct}=="02ae",
MODE="0660",GROUP="video"

SYSFS{idVendor}=="045e", SYSFS{idProduct}=="02ad",
MODE="0660",GROUP="video"

SYSFS{idVendor}=="045e", SYSFS{idProduct}=="02b0",
MODE="0660",GROUP="video"
### END#####
```

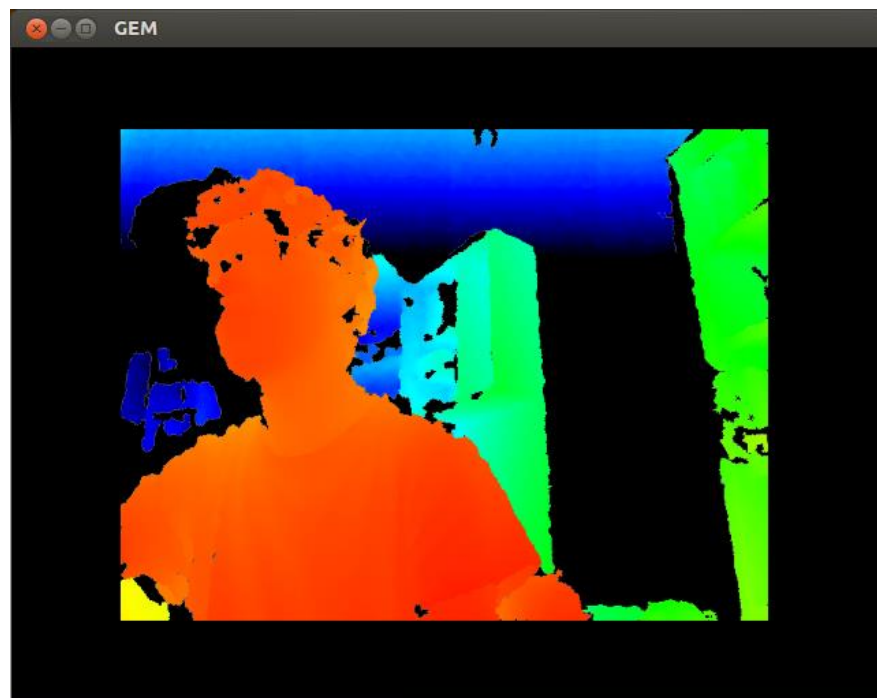


Fig. 36 Ejemplo imagen generada por los controladores Freenect. Fuente: Propia

El controlador Freenect permite acceder al sistema multi-sensor de PrimeSense²⁰, instala nuevos módulos en el Kernel y agrega librerías al sistema.

La Figura 37 muestra como es implementado el chip PS1080 SoC²¹, este es sistema multi-sensor, capaz de sincronizar la imagen de profundidad con la de color y a la vez capturar sonido. Todos los algoritmos de adquisición de datos de profundidad se ejecutan dentro de este SoC, y con la facultad de enviarlos a otros sistemas por medio de un cable USB, gracias a esto puede ser implementado en dispositivos con capacidad limitada. [40]

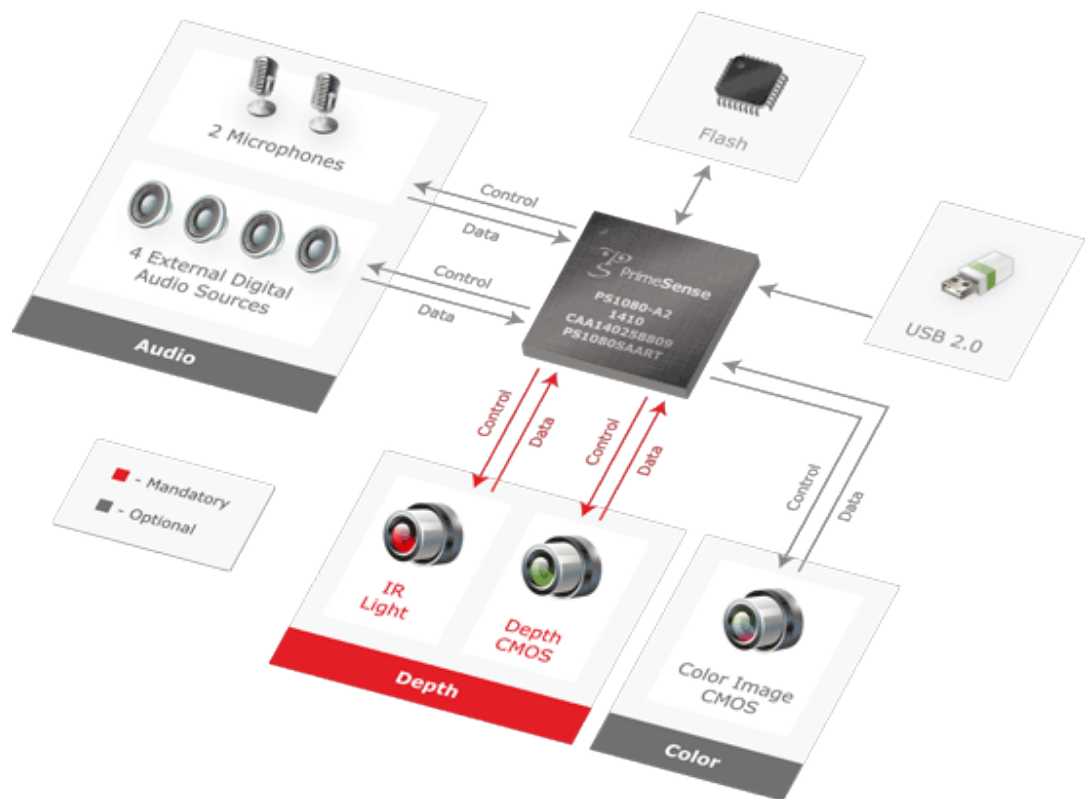


Fig. 37 Diagrama de control de PrimeSense [40]

²⁰ Compañía encargada de fabricar el sistema multi-sensor para el Kinect

²¹ Tecnologías de fabricación que integran todos o gran parte de los módulos de un sistema electrónico en un único circuito integrado o chip.

5.3.3. Integración Pure Data-Kinect

Para ejecutar los ejemplos de Pure Data diseñados para estos controladores es necesario compilar dos “*Externals*” fundamentales: *pix_freenect* y *pix_depth2rgba*, los cuales se compilan utilizando los paquetes descargados de Pure Data y los módulos instalados por el *freenect*.

El primero en compilar es el *pix_freenect*, al descargarlo hay que tener en cuenta que hay dos archivos *Make* el de interés es el *Makefile_linux*, los otros se eliminan y a éste se le renombra como *Makefile*, a continuación se edita y se le da la ruta de ubicación de la carpeta de Pure Data Extended, Gem y Freenect. Al terminar se cierra y desde la terminal se ejecuta el comando:

```
$ make
```

Se copia el “*External*” creado y el archivo de ayuda del Pd con los siguientes comandos:

```
$ sudo cp *.pd_linux /usr/lib/pd-extended/extra  
$ sudo cp *.pd /usr/lib/pd-extended/doc/5.reference
```

Se compila el *pix_depth2rgb* de la misma manera, se modifica el archivo *Make* y se compila, al finalizar se copian las librerías y la ayuda de Pd:

```
$ sudo cp *.pd_linux /usr/lib/pd-extended/extra  
$ sudo cp *.pd /usr/lib/pd-extended/doc/5.reference
```

Ahora ya se pueden ver la imagen del Kinect a través de Pd y su complemento gráfico Gem.

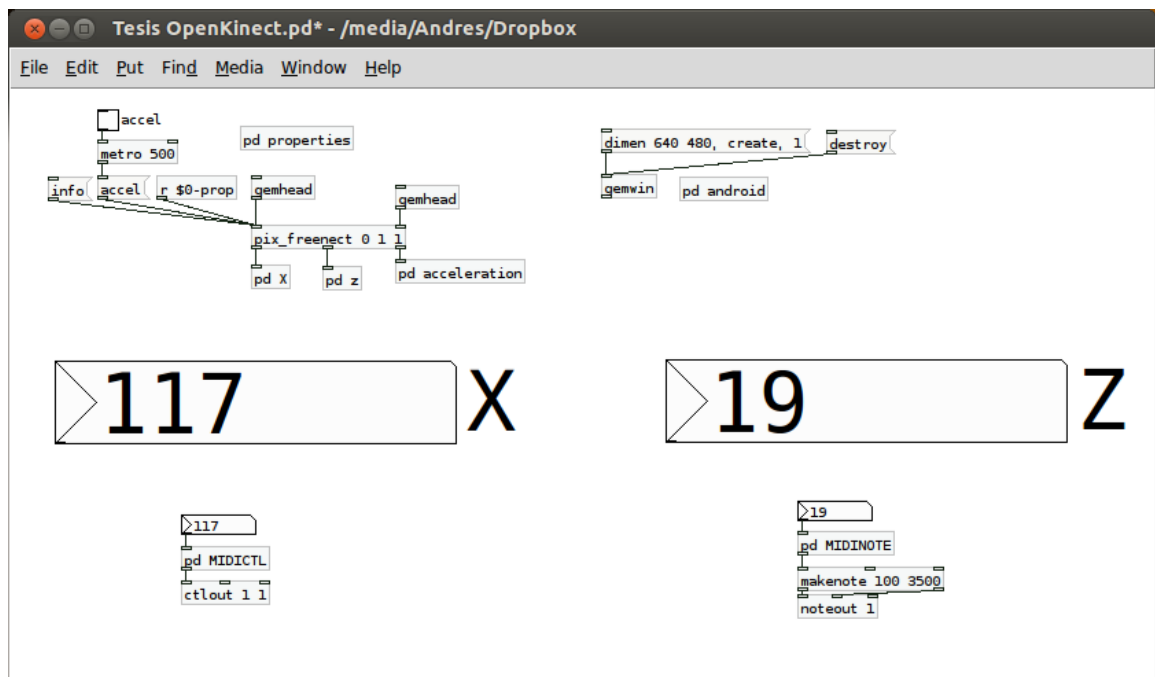


Fig. 38 Ensamble implementado en Pd para la captura de datos en el eje Z y X

En la primera sección se encuentra la caja principal del programa *pix_freect* la cual se muestra en la Figura 39, esta cuenta con entradas y salidas, en las entradas se tienen propiedades, acelerómetro y Gem. *pd properties* maneja qué tipo de imagen se desea mostrar, calidad de imagen, control de led, control del motor, etc. A la salida se encuentran los datos RGB e infrarrojos, adicional se puede adquirir los datos del acelerómetro en 3 ejes.

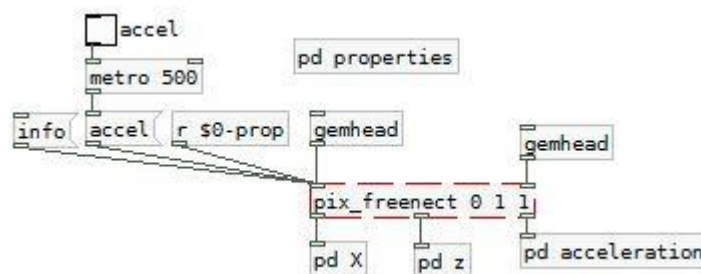


Fig. 39 Sección principal ensamble

En la segunda sección del ensamble la cual se encuentra en la Figura 39, se hace la activación de la interfaz de Gem, se crea a partir de la caja de mensaje “dimen 640 480, create, 1” donde se puede modificar la resolución de la ventana de Gem;

con la caja de mensajes “destroy” se elimina la interfaz creada, adicionalmente esta la caja de *pd android*, la cual da acceso al TouchOSC para controlar el ensamble a partir de dispositivos Android.



Fig. 40 Sección activación Gem

La tercera sección se divide entre los datos obtenidos en el eje X y los datos de profundidad en el eje Z como se muestra en la Figura 41.

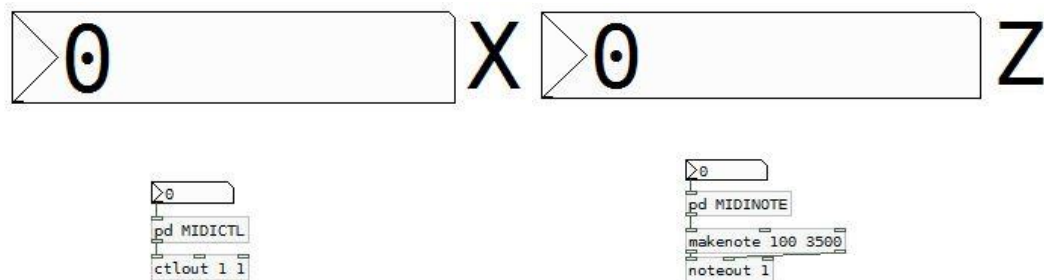


Fig. 41 Sección obtención datos X y Z

Las cajas de número *X* y *Z* se utilizan para mostrar los valores capturados en esos dos ejes respectivamente, los valores son enviados a las cajas *pd MIDICTL* y *pd MIDINOTE* respectivamente, las que se muestran en detalle en las figuras 42 y 43

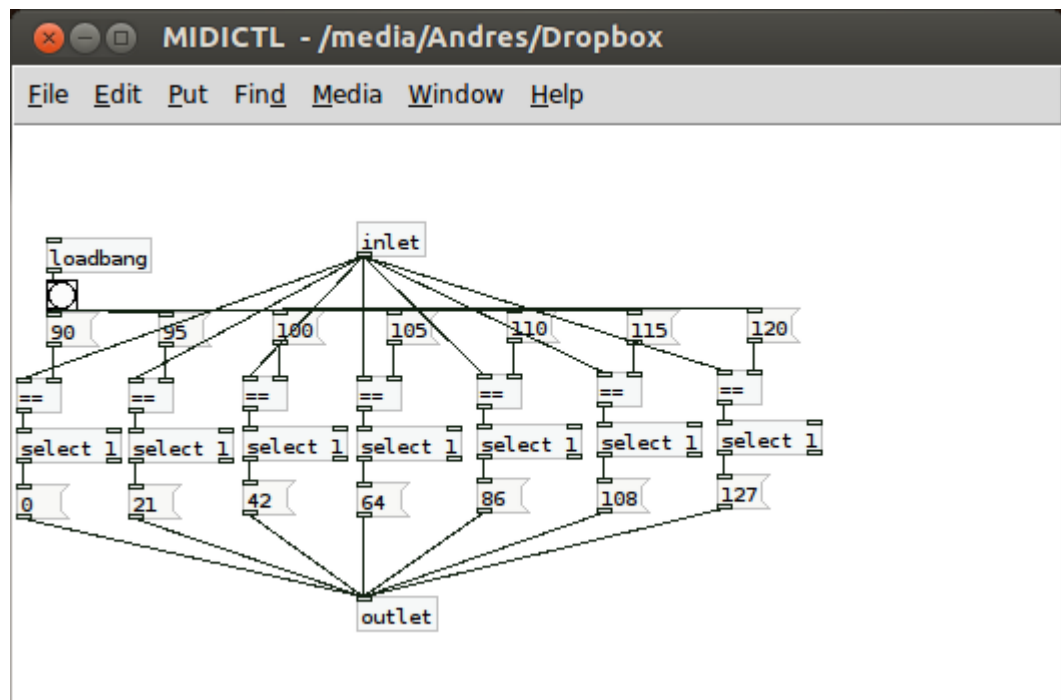


Fig. 42 Configuración MIDICTL

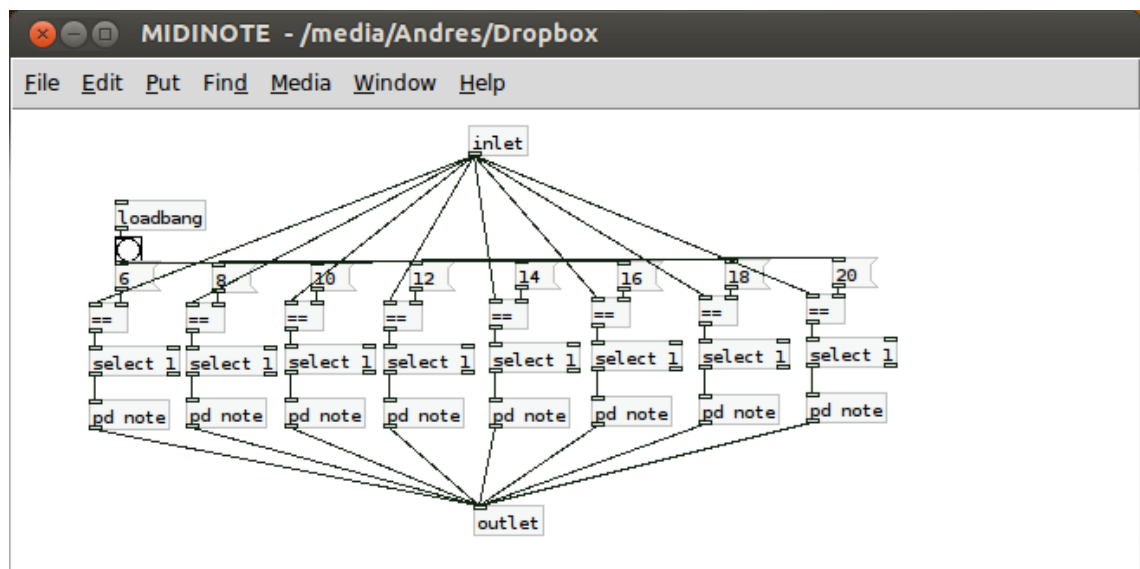


Fig. 43 Configuración MIDINOTE

5.4. Implementación Controlador MIDI No Convencional

La última fase donde se llevan a cabo pruebas y se analizan los resultados para sacar conclusiones acerca del funcionamiento del Kinect sobre Sistemas Embebidos, se describe en la Figura 44.

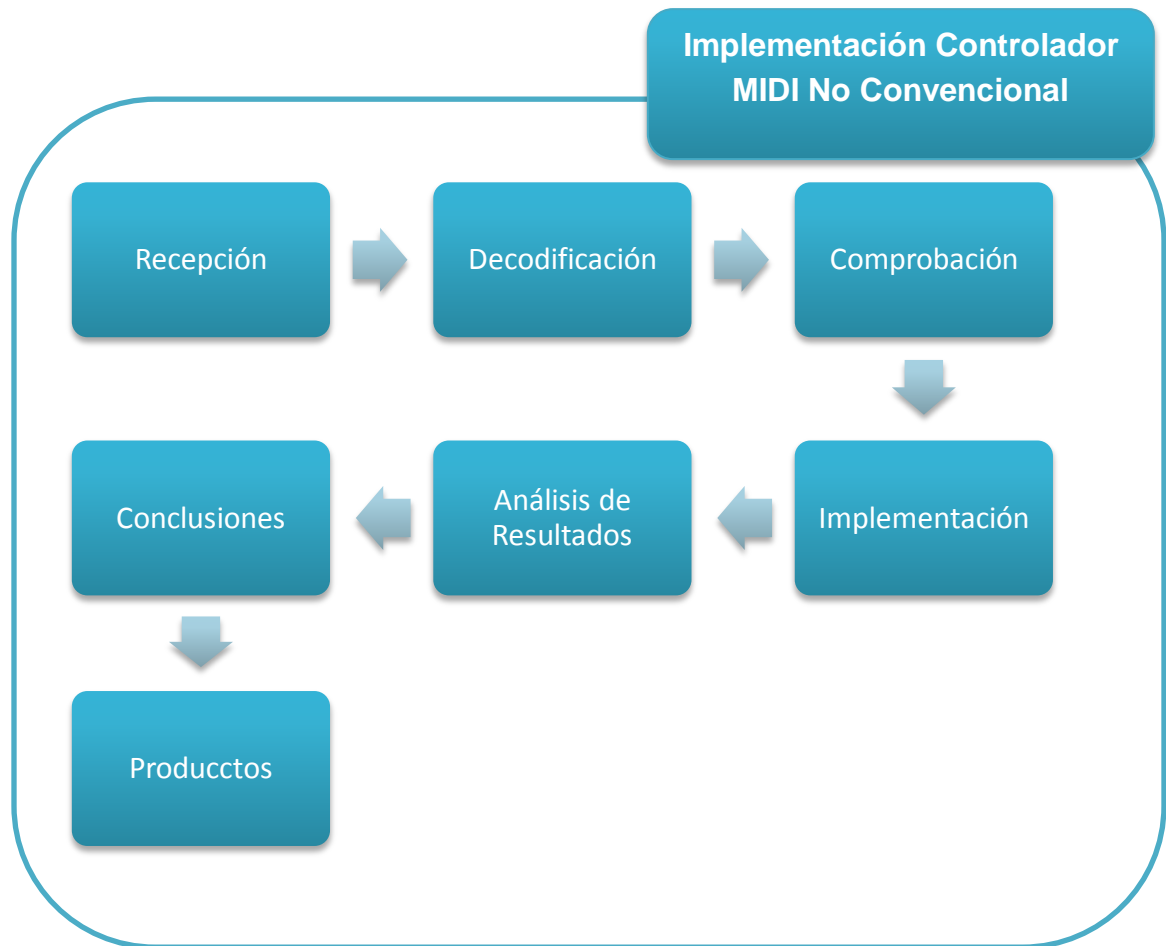


Fig. 44 Descripción proceso de implementación y prueba del sistema

5.4.1. Recepción y Decodificación

Para la recepción de los datos MIDI se utilizan el software ipMIDI, el cual es una versión de prueba para Windows, es un receptor de datagramas por medio del protocolo UDP, se encarga de decodificar los datagramas generando nuevamente mensajes MIDI, éste cuenta con 20 puertos y cada puerto con 16 canales MIDI, con lo que se tiene un total de 320 canales MIDI independientes.



Fig. 45 Diagrama de bloques recepción y decodificación

5.4.2. Comprobación

La recepción es monitorizada con el software MIDI-OX, el cual se encarga de analizar el puerto de entrada MIDI deseado y mostrar la información de los datos MIDI recibidos, como tipo de mensajes o velocidad, la interfaz de usuario se muestra en la Figura 46.

Para comprobar que el receptor está funcionando de forma correcta se utiliza un celular con sistema operativo Android para enviar mensajes MIDI desde la aplicación TouchOSC, esto con el fin de asegurarse que las conexiones y los puertos de recepción sean los indicados.

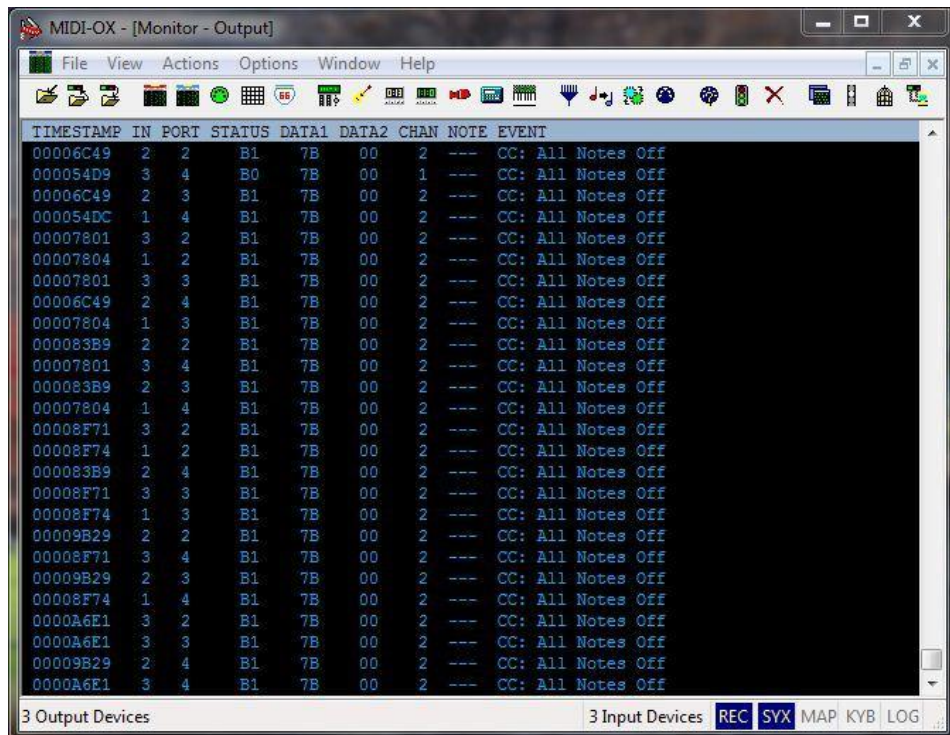


Fig. 46 Recepción de datos MIDI vía WiFi a MIDI-OX. Fuente: Propia

Una vez se comprueba que el receptor está en funcionamiento se utiliza el Kmidimon para poder comprobar los tipos de mensajes que se están enviando desde el Pure Data o desde el VMPK que es un controlador MIDI virtual el cual se observa en la figura 57. Este es software similar a MIDI-OX pero para Linux.



The screenshot shows the Kmidimon application window titled "(ningún archivo) [grabando] - Kmidimon". It has a menu bar with "New", "Open", "Save As", and "Quit", and playback controls "Retroceder" and "Reproducir". Below the menu is a "Pista: 1" tab. The main area contains a table with the following data:

Tics	Tiempo	Origen	Tipo de evento	Canal	Datos 1	Datos 2	Datos 3
47696	99.3672	129:0	Nota desactivada	1	60	0	
49263	102.6325	129:0	Cambio de programa	1	28		
53804	112.0929	129:0	Cambio de programa	1	32		
57714	120.2383	129:0	Cambio de controlador	1	2	23	
57719	120.2558	129:0	Cambio de controlador	1	2	21	
57725	120.2716	129:0	Cambio de controlador	1	2	18	
57731	120.2754	129:0	Cambio de controlador	1	2	16	
59940	124.8756	129:0	Nota activada	1	73	127	
60005	125.0129	129:0	Nota desactivada	1	73	0	

Fig. 47 Recepción de datos MIDI desde VMPK al Kmidimon. Fuente: Propia

5.4.3. Implementación

La implementación se hace con el software FL Studio™²², el cual es un software amigable para la configuración de los mensajes MIDI de control, se realiza por medio de detección con el fin de optimizar la conexión.

Se utiliza un sintetizador para la recepción de mensajes MIDI de Nota On/Off y el control de *Faders* con los mensajes de cambio de control.

Las Figuras 48 muestran el funcionamiento desde la Pandaboard ES conectada remotamente a un computador de escritorio y a una Tablet

²² Es un software de producción capaz de grabar en múltiples pistas, secuenciador y mezclador para producciones profesionales de alta calidad.



Fig. 48 Implementación Controlador MIDI no Convencional en Pandaboard ES. Fuente: Propia

La Figura 49 muestra la pantalla principal del FL Studio y la configuración del puerto de entrada MIDI por Ethernet

La transmisión MIDI se lleva acabo exitosamente, con lo cual se logra la recepción de mensajes MIDI desde un Sistema Embebido a un computador utilizando los datos del Kinect para el control del sistema.

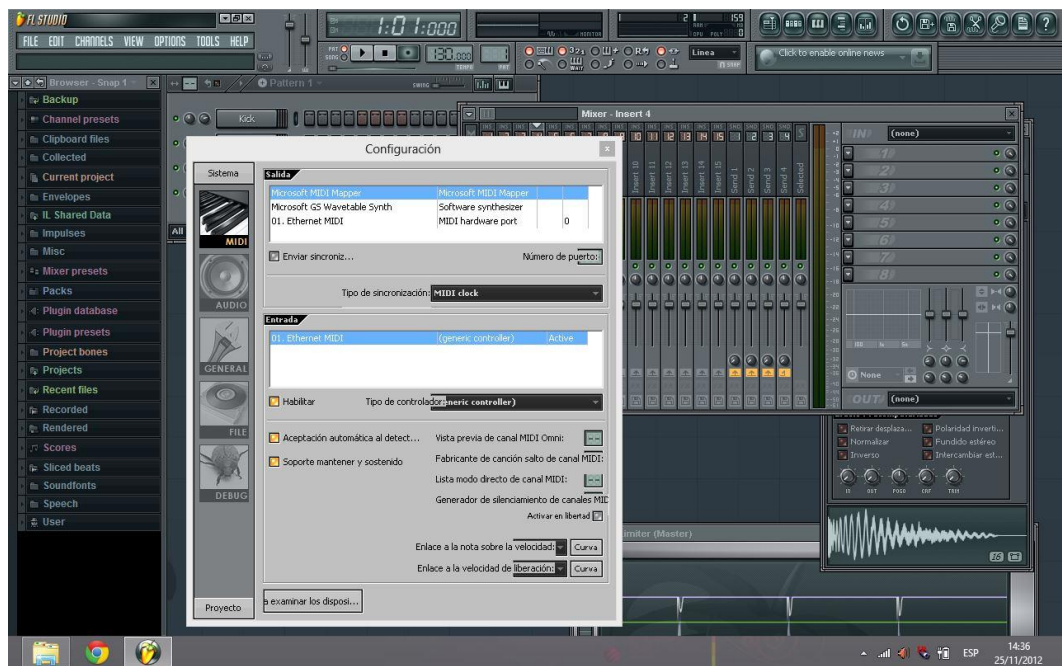


Fig. 49 Interfaz principal FL Studio. Fuente: Propia



Fig. 50 Captura de pantalla de la implementación en el FL Studio. Fuente: Propia

La adquisición de datos del Pure Data se puede observar en la Figura 51, donde a partir del desplazamiento de la persona en los ejes X y Z se generan mensajes MIDI de control y de nota On/Off que son adquiridos por el FL Studio como se observa en la Figura 50, en la que el cambio de control es asignado al control principal de volumen y las notas On/Off al sintetizador POIZONE²³.

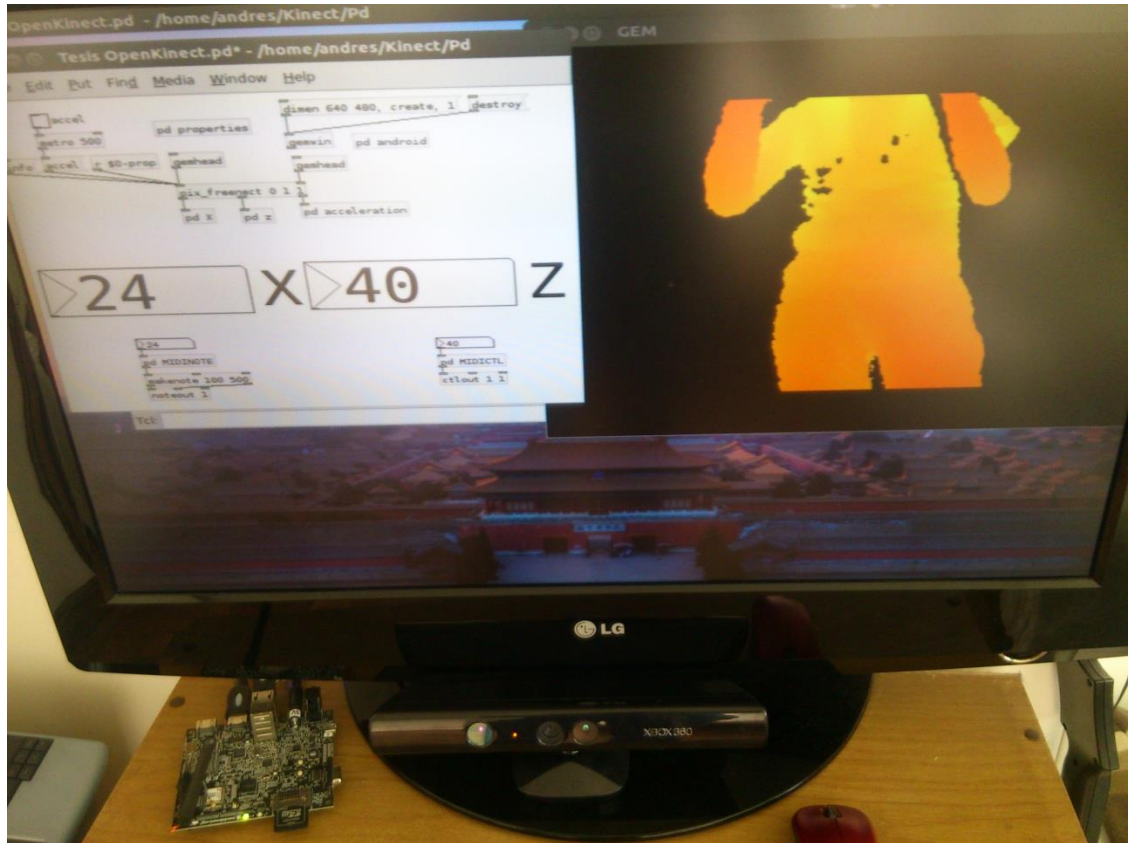


Fig. 51 Interfaz de captura de los datos del Kinect implementado en Pure Data. Fuente: Propia

²³ Sintetizador substractivo diseñado para realizar producciones profesionales, se puede implementar como complemento VST o como aplicación Standalone.

6. ANÁLISIS DE RESULTADOS

Resultados obtenidos implementando el Sistema Embebido Pandaboard ES con sistema operativo Ubuntu 12.04 se presentan a continuación:

- Los datos obtenidos por el computador receptor corresponden a los enviados a través de Pure Data como efectivamente se comprobó con el MIDI-OX y Kmidimon, esta precisión se logró gracias a la segmentación de imagen generada por el Kinect, el cual permite crear límites de captura, en este caso, una distancia máxima de 1.75m y una distancia mínima de 0.5m.
- El protocolo UDP es confiable y rápido para su implementación como transmisor MIDI, esto se observó en los resultados obtenidos donde el tiempo de respuesta fue instantánea.
- La renderización de imagen realizada por la Pandaboard ES es óptima y se muestra de una forma fluida, el usuario puede realizar movimientos que se verán proyectados en la pantalla o en la tablet de manera instantánea por lo que el usuario puede monitorizar los datos en tiempo real.
- Con este tipo de sistemas de captura no convencionales se pueden tener imprecisión en los datos que el usuario desea enviar, esto debido a que el sistema de captura por medio de posición no es 100% preciso, a pesar de estos gracias a la habilidad del Kinect para crear límites de captura, se puede asegurar que otros objetos no interfieran dentro del sistema.

7. CONCLUSIONES

- El documento presenta el diseño y desarrollo de un controlador MIDI no convencional utilizando el Kinect, implementado en el Sistema Embebido Pandaboard ES. En el S.E. se implementó la distribución de Ubuntu, la cual permite la instalación y configuración de los programas y servicios con los cuales se realiza la transmisión de los datos en protocolo MIDI a un computador fijo adquiriendo los datos a partir de movimientos capturados por el Kinect.
- El desarrollo de la interfaz gráfica para la interacción con el usuario fue un apoyo visual para monitorear los datos generados y capturados por el Kinect, el Pure Data fue una herramienta esencial para presentarle al usuario los datos de una forma ordenada y coherente de tal forma que pudiera interpretarlos.
- La portabilidad de hardware y software demuestra de los sistemas embebidos son una herramienta de desarrollo eficiente, y el sistema operativo Ubuntu se encuentra en la capacidad de brindar las herramientas suficientes para el desarrollo de nuevas tecnologías dentro de los Sistemas Embebidos aplicados a la Ingeniería de Sonido
- La implementación de herramientas tecnológica como los Sistemas Embebidos o el Kinect, permiten el desarrollo de una gran variedad de dispositivos no disponibles en el mercado, adaptables a hardware o software existente.

8. PRODUCTOS

A partir de la implementación del proyecto se desarrollaron los siguientes productos:

ARTICULOS EN EDICION

- **Desarrollo de un Controlador MIDI no Convencional, Implementado en un Sistema Embebido, Utilizando el Kinect.**

El artículo presenta el desarrollo de un controlador MIDI no convencional implementado en un Sistema Embebido utilizando el Kinect. El dispositivo permite utilizar el protocolo MIDI para controlar hardware y/o software remoto vía WiFi por medio de la captura de movimientos. Con la utilización de este tipo herramientas tecnológicas se puede construir controladores no convencionales aplicados a la ingeniería de sonido, integrables a diferentes redes de comunicación de forma práctica, eficaz e innovadora.

- **Controlador MIDI WiFi Implementado en un Sistema Embebido.**

El artículo presenta el desarrollo del proyecto de Investigación, en el cual se diseñó un sistema para realizar la transmisión por WiFi del protocolo MIDI, el cual utiliza herramientas de software libre y sistemas embebidos para su implementación.

- **Compresor de Formato de Audio Desarrollado en Pure Data, Implementado en un Sistema Embebido**

GRUPO DE INVESTIGACIÓN: Acústica Aplicada

En el grupo de investigación ING 006A-028 se desarrolló un Sistema De Transmisión Inalámbrica Para El Protocolo De Comunicación Midi, Utilizando Sistemas Embebidos

El proyecto consiste en diseñar un sistema para realizar la transmisión por WiFi del protocolo MIDI, el cual utilice herramientas de software libre y sistemas embebidos.

9. RECOMENDACIONES

- Crear interfaz de usuario física para generar cambios del ensamble de Pure Data en de forma externa al sistema para tener un control específico, ya fuera por medio de micro-controladores o por sensores inalámbricos.
- Implementar cámaras alternativas al Kinect como el ASUS Xtion PRO LIVE el cual fue desarrollado por el mismo grupo que desarrolló el Kinect, pero fue creado específicamente para desarrollo y no para una consola de juegos.
- Buscar Sistema Embebidos y Sistema Operativos alternativos que pudieran ser óptimos para la implementación como controladores MIDI no convencionales.

BIBLIOGRAFÍA

- [1] Abbott, Doug. *Linux for Embedded and Real-Time Application*. Newnes. 2003
- [2] Rona, Jeffrey. *The MIDI Companion: The Ins, Outs and Throughs*. Hal Leonard, 1994
- [3] Miles Huber, David. *The MIDI Manual: A Practical Guide to MIDI in the Project Studio*. 2012
- [4] Camargo Bareño, Carlos Iván. "*Transferencia tecnológica y de conocimientos en el diseño de sistemas embebidos*". Universidad Nacional de Colombia. Bogotá, Colombia. 2011
- [5] Milano, Dominic. *Mind Over MIDI*. H. Leonard Books. 1987
- [6] Zhang, Zhengyou. "*Microsoft Kinect Sensor and Its Effect*". University of Missouri. Misuri, Estados Unidos.
- [7] Fry, Ben. Reas, Casey. *Getting Started with Processing*. O'reilly.
- [8] PrimeSense. Willow Garage. Side-Kick. ASUS. AppSide. <<OpenNI organization>><http://openni.org/>
- [9] Martin, Hector. Blake, Josh. Machulis, Kyle. <<OpenKinect Community>>
http://openkinect.org/wiki/Main_Page
- [10] Gralla, Preston. *Cómo funcionan las redes inalámbricas*. Anaya Multimedia. 2007
- [11] Engst, Adam. *Introducción a las redes inalámbricas*. Anaya Multimedia. 2005
- [12] Varela, Carlos. Dominguez, Luis. *Redes Inalámbricas*. Universidad de Valladolid. Valladolid, España. 2002
- [13] Werchau, Pablo Jara. Nazar, Patricia. *Estándar IEEE 802.11 X de las WLAN*. Universidad Tecnológica Nacional. Buenos Aires, Argentina.

- [14] Capela, Rui Nuno. QmidiNet A MIDI Network Gateway via UDP/IP Multicast. <http://qmidinet.sourceforge.net/qmidinet-index.html>
- [15] Dhawan, S. *Analogy of Promising Wireless Technologies on Different Frequencies: Bluetooth, WiFi, and WiMAX*, 2007.
- [16] Steve Heath. *Embedded Systems Design*. Second Edition. Newnes,,2002.
- [17] P.Raghavan, Amol Lad, Sriram, Neelakandan. *Embedded Linux System Design and Development*. Auerbach Publications, 2005.
- [18] Rosas Jiménez, Nelson Felipe. *Diseño E Implementación De Un Sistema Embebido Para La Adquisición Y Transmisión De Señales Biomédicas A Través De La Red Celular*. Universidad Nacional De Colombia. Bogotá, Colombia. 2011
- [19] Amaro Soriano, José Enrique. *Android: Programación de dispositivos móviles a través de ejemplos*. Marcombo, 2011
- [20] Meier, Reto. *Professional Android 4 Application Development*. John Wiley & Sons, 2012
- [21] Gargenta, Marko. *Learning Android*. O'Reilly Media, Inc., 2011
- [22] Thomas, Keir. *Ubuntu Pocket Guide and Reference*. Ubuntu Pocket Guide, 2009
- [23] Helmke, Matthew. Graner, Amber. Rankin, Kyle. Mako Hill, Benjamin. Bacon, Jono. *The Official Ubuntu Book*. Prentice Hall, 2012
- [24] Helmke, Matthew. *Ubuntu Unleashed 2012 Edition: Covering 11.10 and 12.04*. Sams Publishing, 2012
- [25] Miller, Charlie. Blazakis, Dion. DaiZovi, Dino. Esser, Stefan. Lozzo, Vincenzo. Weinmann, Ralf Philip. *iOS Hacker's Handbook*. John Wiley & Sons, 2012
- [26] Ray, Deborah S. Ray, Eric J. *Unix and Linux*. Peachpit Press, 2009
- [27] Brinkmann, peter. *Making Musical Apps: Real-time audio synthesis on Android and iOS*. O'Reilly Media, Inc., 2012
- [28] Rosenberg Craig, Moses Bob. *Future Human Interfaces to Computer Controlled Sound Systems*. University of Washington, Seattle. Rane Corporation, Mukilteo. 1993

- [29] Balind Wincent, Lovis cach Jorn. *Gestures to Operate DAW Software*. University Oldenburg, Oldenburg. Fachhochschule Bielefeld, Bielefeld. 2011
- [30] Selfridge Rod, Reiss Josh. *Interactive Mixing Using Wii Controller*. Queen Mary University of London. Londres. Inglaterra. 2011
- [31] Bartlett Greg. *Embedded Controllers for Audio Devices*. Pavo, Philadelphia. 1994
- [32] Luke. Kyle. <<iFixit>>. <http://www.ifixit.com/Teardown/Microsoft+Kinect+Teardown/4066/3>
- [33] Azhar. <<How to Set Up an AdHoc Wireless Network in Windows XP>>. <http://www.sizledcore.com/2010/01/how-to-set-up-an-adhoc-wireless-network-in-windows-xp/>
- [34] Fry, Ben.Reas, Casey.<<Processing>>. <http://processing.org/>
- [35] Upton, Eben. Mullins, Rob. Lang, Jack. Mycroft, Alan. <<Raspberry Pi>> <http://raspberrypi.org/>
- [36] <<Pandaboard>><http://pandaboard.org/>
- [37] Kridner, Jason. Coley, Gerald. Osier-Mixon, Jeffery. <<Beagleboard>> <http://beagleboard.org/>
- [38] Zhang, Yongqiang. Zhang, Handan Bing. *Design of Embedded Audio and Video Compression System*. Hebei University of Engineering. Hadan, China IEEE Explorer
- [39] Yaghmour, Karim. *Building Embedded Linux Systems*. O'Reilly Media, Inc., 2003
- [40] Maizels, Aviad. <<PrimeSense Natural Interaction>> <http://www.primesense.com>
- [41] Kamal, Raj. *Embedded Systems 2E*. Tata McGraw-Hill Education, 2008
- [42] Ganssle, Jack G. *Embedded Systems: World Class Design*. Newnes, 2008

- [43] Collins, Christopher Michael. *An Evaluation of Embedded System Behavior Using Full-System Software Emulation*. University of Maryland. California, Estados Unidos. 2000
- [44] Yu, Tingting. *Testing Embedded System Applications*. University of Nebraska, Lincoln, Estados Unidos. 2010
- [45] Huang, Fay. Wang, Reen-Cheng. *Arts and Technology*. Springer, 2010
- [46] Sean, Jared St. *Kinect Hacks: Tips & Tools for Motion and Pattern Detection*. O'Reilly Media, 2012
- [47] Helmke, Matthew- Hudson, Andrew. *Ubuntu Unleashed 2012: Covering 11.10 and 12.04*. Sams Publishing, 2012
- [48] Borenstein, Greg. *Making Things See: 3D vision with Kinect, Processing, Arduino, and MakerBot*. O'Reilly Media, Inc., 2012
- [49] Webb, Jarrett- Ashley, James. *Beginning Kinect Programming with the Microsoft Kinect SDK*. Apress, 2012
- [50] Kean, Sean-Hall, Jonathan-Perry, Phoenix. *Meet the Kinect: An Introduction to Programming Natural User Interfaces*. Apress, 2011
- [51] Raggi, Emilio-Thomas, Keir-Van Vugt, Sander. *Beginning Ubuntu Linux: Natty Narwhal Edition*. Apress, 2011
- [52] Nixon, Robert. *Ubuntu: Up and Running: Up and Running*. O'Reilly Media, Inc., 2010.
- [53] Books Llc. *Audio Programming Languages: Mathematica, Csound, Max, Pure Data, Supercollider, Comparison of Audio Synthesis Environments*. General Books LLC, 2010
- [54] Noble, Joshua. *Programming Interactivity*. O'Reilly Media, Inc., 2012
- [55] Igoe, Tom. *Making Things Talk: Practical Methods for Connecting Physical Objects*. O'Reilly Media, Inc., 2007
- [56] Harke, Werner. *Domótica para Viviendas y Edificios*. Marcombo, 2010
- [57] Huidobro Moya, José Manuel. Millán Tejedor, Ramón Jesús. *Domótica: edificios inteligentes*. Creaciones Copyright, 2004

- [58] Shneiderman, Ben. Plaisant, Catherine. *Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva*. Pearson Educación, 2005
- [59] Raskin, Jef. *The Human Interface: New Directions for Designing Interactive Systems*. Addison-Wesley Professional, 2000
- [60] Wong, Ronnie Anthony. *Assessment of Open Source Business Strategies in the Domain of Embedded Systems*. University of Oulu. Oulu, Finlandia. 2011
- [61] Granlund, Gösta H. Knutsson, Hans. *Signal Processing for Computer Vision*. Springer, 1994
- [62] Ngan, King Ngi. Li, Hongliang. *Video Segmentation and Its Applications*. Springer, 2011
- [63] Zhang, Yu-Jin. *Advances in Image And Video Segmentation*. Idea Group Inc (IGI), 2006
- [64] Belbachir, Ahmed Nabil. *Smart Cameras*. Springer, 2009
- [65] Bradski, Gary. Kaehler, Adrian. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc., 2008
- [66] Kisacanin, Branislav, Bhattacharyya, Shuvra S. Chail, Sek. *Embedded Computer Vision*. Springer, 2008
- [67] Hammoud, Riad I. *Augmented Vision Perception in Infrared: Algorithms and Applied Systems*. Springer, 2008
- [68] Kisacanin, Branislav. Pavlovic, Vladimir. Huang, Thomas S. *Real-Time Vision for Human-Computer Interaction*. Springer, 2005
- [69] Dietrich, Dietmar. Garn, Heinrich. Kebschull, Udo. *Embedded Vision System*. Lightning Source Incorporated, 2006
- [70] Bigun, Josef. *Vision with Direction: A Systematic Introduction to Image Processing and Computer Vision*. Springer, 2006
- [71] Dal Mutto, Carlo. Zanuttigh, Pietro. Cortelazzo, Guido M. *Time-of-Flight Cameras and Microsoft Kinect™*. Springer, 2012
- [72] Fisher, R. *60 GHz WPAN Standardization within IEEE 802.15.3c*. 2007

- [73] Kim, Yong Soo. Kim, Juho. *Security enhanced IEEE 802.1x authentication method for WLAN mobile router*. 2012
- [74] Chung, Yun Won. Yun, Mi Young. Kim, Yeong-Jin. *An efficient Diameter-based accounting scheme for wireless metropolitan area network (WMAN)*. 2004
- [75] Guo, Chuanxiong. Guo, Zihua. Zhu, Wenwu. *Efficient mobility management for vertical handoff between WWAN and WLAN*. 2003
- [76] I-Tsun Chiang. *Performance Skills on Institutionalized Older Adults with Wheelchairs*. Grad. Inst. of Sports & Health, Nat. Changhua Univ. of Educ., Changhua, Taiwan, 2012
- [77] Rakprayoon, P. *Kinect-based obstacle detection for manipulator*. Fac. of Eng., Kasetsart Univ., Bangkok, Thailand, 2011
- [78] Machida E. *Human motion tracking of mobile robot with Kinect 3D sensor*. Tokyo Metropolitan Coll. of Ind. Technol., Tokyo, Japan, 2012.
- [79] Noonan, P.J.-Wolfson Mol. *The design and initial calibration of an optical tracking system using the Microsoft Kinect*. Univ. of Manchester, Manchester, UK, 2011
- [80] Hui Li. *Static Hand Gesture Recognition Based on HOG with Kinect*. Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference. Nanchang, Jiangxi, 2012
- [81] Qiu-hong, Chen. *A new embedded image information processing system design*. Dept. of Comput. Sci. & Eng., Henan Univ. of Urban Constr., Pingdingshan, China. 2010
- [82] Chen Qiu-hong. *A new embedded image information processing system design*. Dept. of Comput. Sci. & Eng., Henan Univ. of Urban Constr., Pingdingshan, China, 2010.
- [83] Correa, D.S.O. *Mobile Robots Navigation in Indoor Environments Using Kinect Sensor*. ICMC - SSC - LRM (Mobile Robots Lab.), USP - Univ. of Sao Paulo, Sao Carlos, Brazil, 2012
- [84] Yihua Lou. *A Multi-User Interaction System Based on Kinect and Wii Remote*. State Key Lab. of Software Dev. Environ., Beihang Univ., Beijing, China, 2012.

GLOSARIO

TERMINAL: Es una forma de acceder al sistema sin utilizar la interfaz gráfica, realizar todo tipo de tareas en formato texto.

EXTERNAL: Complementos de librerías de Pure Data.

COMPILACIÓN: Proceso de traducción de un código a datos que puedan ser ejecutados por una computadora.

CROSS COMPILACIÓN: Proceso de traducción de un código de datos de un sistema a otro.

MULTICAST: En español "Multidifusión", es el envío de la información en una red a múltiples destinos simultáneamente.

COMANDOS SHELL: Sistemas de comandos utilizados por el sistema UNIX que permite administrar el sistema sin entorno gráfico.

KERNEL: o Núcleo, es el encargado de conectar el hardware con el software y gestionar los recursos de manera adecuada.

ROOT FILE SYSTEM (RFS): Es el conjunto de archivos fundamentales para ejecutar un sistema operativo.

S.E.: Sistema Embebido

S.O.E.: Sistema Operativo Embebido.

FREENECT: Controlador para Kinect desarrollado por la comunidad Open Kinect

TOUCHOSC: Superficie de control MIDI y OSC que utiliza el protocolo UDP para el envío de datos por medio de WiFi, desarrollado para Android e iOS

SOCKETS: Mecanismo que permite la conexión entre distintos procesos

OMAP: Procesador multinúcleo ARM desarrollado para diferentes aplicaciones embebidas, fabricado por Texas Instruments

HDMI (High-Definition Multimedia Interface): Conector de audio y vídeo digital cifrado sin compresión de alta definición.

IPV4: Cuarta versión del protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red.

UDP (User Datagram Protocol): Protocolo basado en el intercambio de datagramas a través de la red, sin necesidad de establecer previamente una conexión.

SISTEMA DOMOTICO: Sistemas aplicados al control y la automatización inteligente de una vivienda, aportando servicios de gestión energética, seguridad, bienestar o comunicación.

ANEXO A: Fotos e Imágenes Complementarias

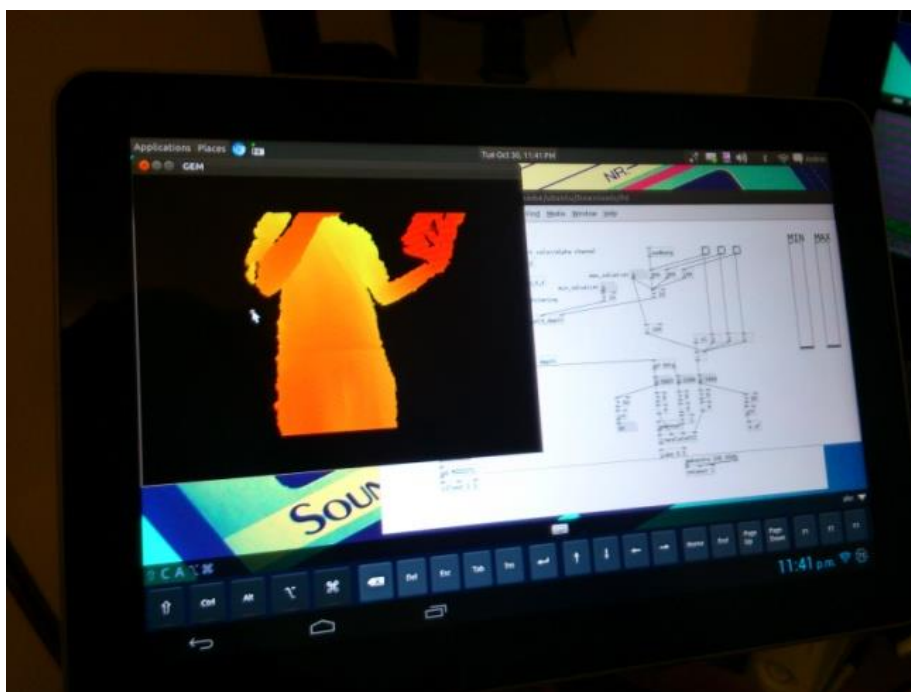


Fig. 52 Pandaboard ES conectada remotamente a Samsung Galaxy Tab 10.1. Fuente: Propia

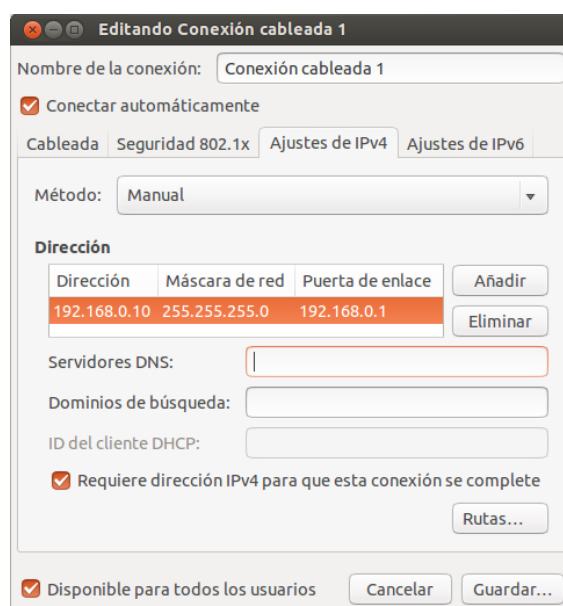


Fig. 53 Interfaz del editor conexiones de red. Fuente: Propia



Fig. 54 Configuración de control remoto MIDI de FL Studio. Fuente: Propia



Fig. 55 Conexión Pandaboard ES con el Kinect. Fuente: Propia

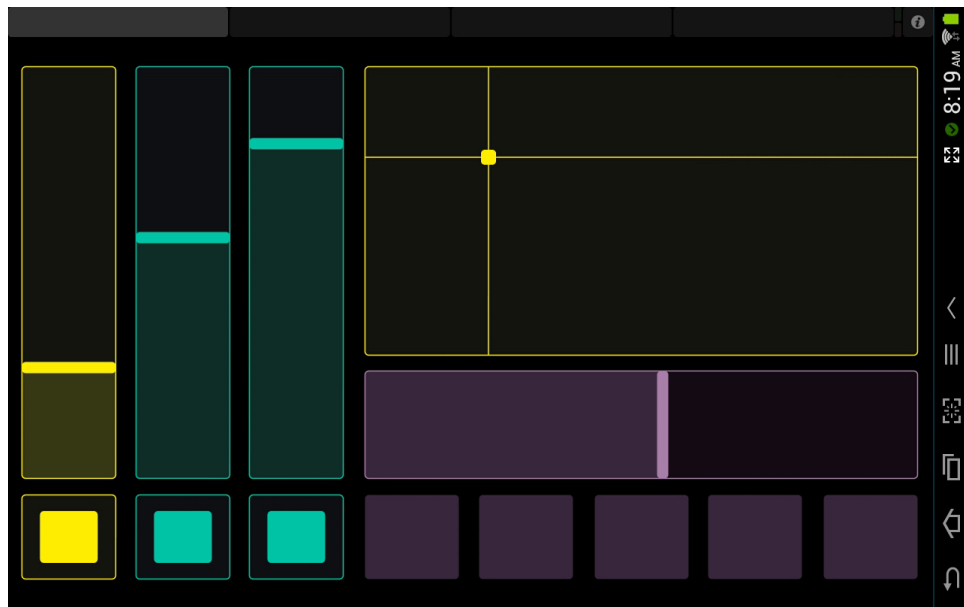


Fig. 56 Interfaz de usuario de TouchOSC para Samsung Galaxy Tab 10.1. Fuente: Propia

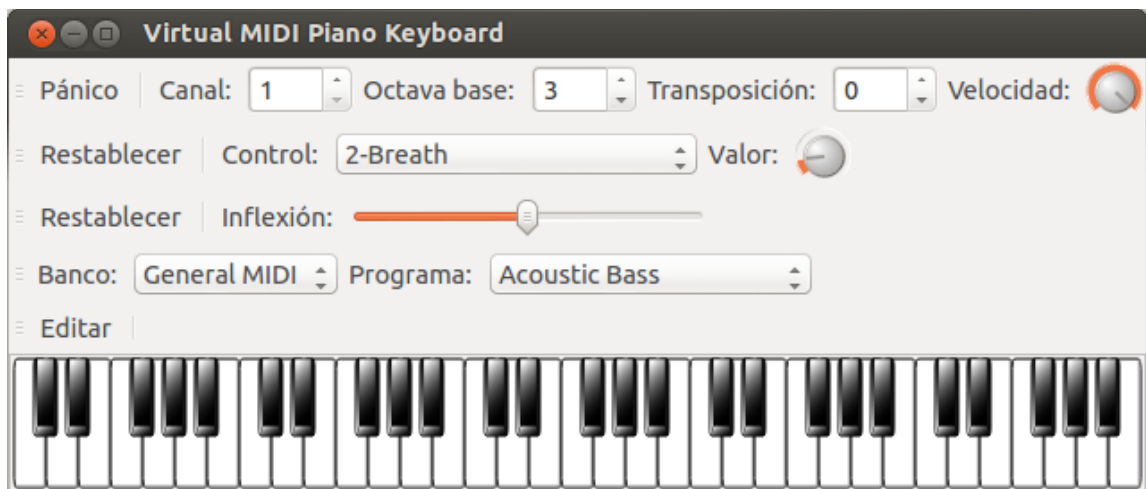


Fig. 57 Controlador MIDI Virtual VMPK. Fuente: Propia

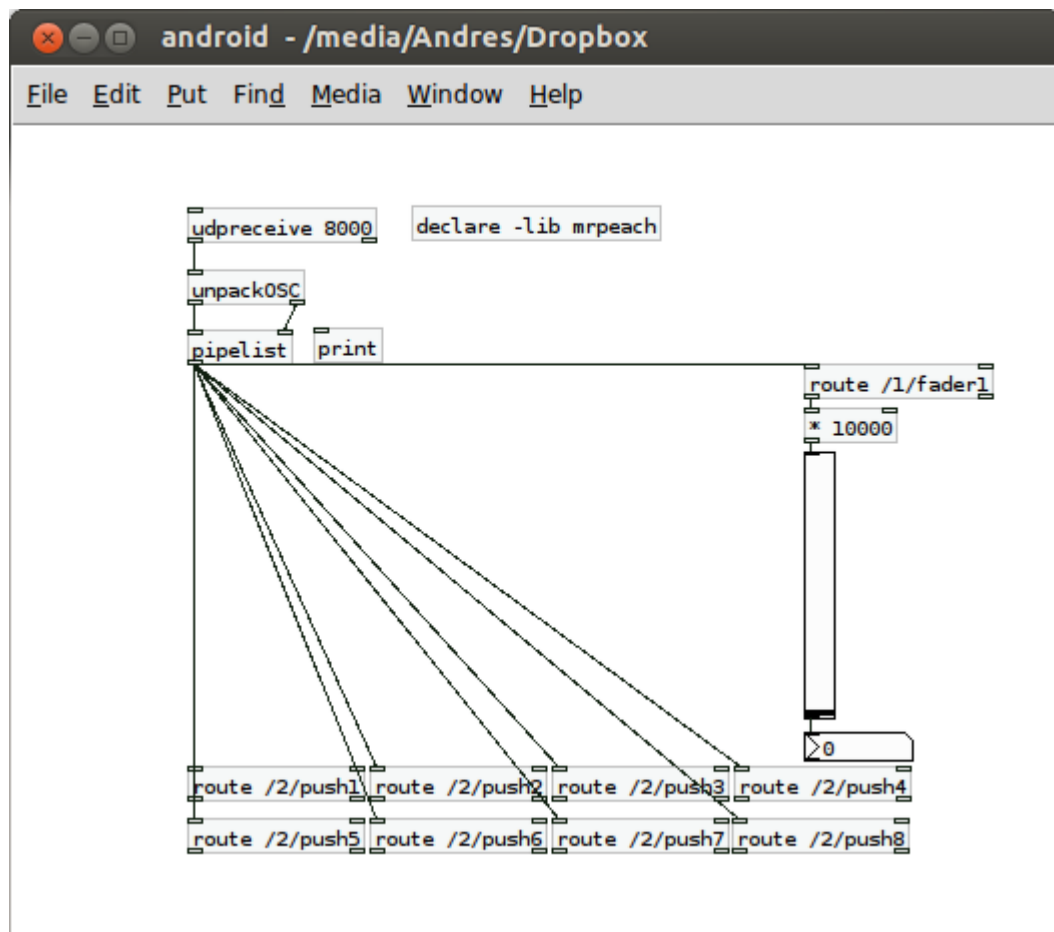


Fig. 58 Ensamble de recepción OSC desde la aplicación TouchOSC. Fuente: Propia

ANEXO B: Anexos digitales

Los siguientes archivos se encuentran en el CD anexo al documento:

- A1. Video prueba del Controlador MIDI no Convencional. "Tesis.mp4"
- A2. Videos demostrativos diferentes fases de desarrollo y productos realizados en el proceso. "Compresor.mp4" y "ControladorMidi.mp4"