



UNIVERSIDAD DE BELGRANO

Las tesis de Belgrano

Facultad de Ingeniería y Tecnología Informática
Carrera Ingeniería Electrónica

Sistemas AoE
Audio over Ethernet
Distribución de Audio en sistemas pre instalado

N° 578 Fernando Nicolás Ficoseco Leis

Tutor: Ing. Eduardo Alberto Martínez

Departamento de Investigaciones
2013

Universidad de Belgrano
Zabala 1837 (C1426DQ6)
Ciudad Autónoma de Buenos Aires - Argentina
Tel.: 011-4788-5400 int. 2533
e-mail: invest@ub.edu.ar
url: <http://www.ub.edu.ar/investigaciones>

Índice

Tema	5
Título	5
Tutor	5
Introducción	5
Objetivo	5
Alcance	5
Hipótesis	5
Estado del Arte	6
Marco Teórico	7
Ethernet.....	7
¿Cómo Funciona Ethernet?	9
mBed	9
¿Cómo Funciona el modulo mBed?	10
ADC - DAC	11
ADC → Conversor Analógico a Digital.-	12
DAC → Conversor Digital a Analógico	14
C / C++	15
Escribiendo el programa en C	15
Descripción general de la idea	16
Descripción general del Modulo	18
Construcción del dispositivo AoE	22
Problemas de Diseño	23
Errores y Contratiempos	24
Conclusiones y Comparaciones con productos actuales	25
A Futuro	26
Glosario	26
Bibliografía	27

Tema

Voz sobre Ethernet, AoE, transmisión e interpretación de voz incorporada en paquetes de datos Ethernet.

Titulo

Distribución de audio a través de sistemas de redes pre-instaladas.

Tutor

Ing. Eduardo Alberto Martínez.

Introducción

Aplicando e interactuando conceptos de diferentes disciplinas de comunicación, transmisión de datos, programación y digitalización de audio, hemos arribado a la conclusión que es posible distribuir audio digitalizado de calidad a través de redes Ethernet.

De esta manera se presenta como un medio de economizar en recursos económicos y temporales; en relación a las instalaciones de complicadas y particulares redes, para distribución y transmisión de audio en residencias, estadios, centros de conferencia, aeropuertos, etc... Es decir para uso particular como así también comercial, obteniendo en calidad y performance el mismo resultado que con los sistemas dedicados, aprovechando las instalaciones pre-existentes y modificando el sistema de recepción y envío de los paquetes de datos.

Objetivo

El objetivo de la presente es desarrollar y demostrar un nuevo producto, que permite modificar la forma e idea en la transmisión de audio convencional. El nuevo perfil de transmisión utilizara los canales conocidos y pre instalados de conexiones de redes, aprovechando los beneficios de su gran rendimiento, ancho de banda y eficiencia, transfiriendo audio en redes LAN.

Alcance

Se pretende llegar a construir un prototipo del sistema planteado, que demuestre, que es posible transmitir audio de alta fidelidad en cualquier red LAN pre instalada.

El sistema consta de distintas estaciones o "peers" de recepción y envío de información, para demostrar y no confundirlo con un sistema p2p (peer to peer).

Se trata de obtener un producto final que brinde la solución mencionada y sea factible de su inserción en un mercado real.

Hipótesis

Demostrar que es posible comunicar dos micro controladores, a través de un vinculo Ethernet, permitiendo así distribuir y transmitir paquetes de audio digitalizados por el mismo.

Estado del Arte

En la actualidad no existen sistemas que operen de la manera que se plantea en el presente proyecto, o no son comercialmente conocidos. Es probable que la mayoría sean desarrollos privados o de índole académico.

Lo que se intenta presentar en este trabajo es la comparación de este nuevo sistema con los actuales y comercialmente conocidos, es decir un sistemas de VoIP versus nuestro desarrollo de AoE, analizando y contraponiendo los puntos positivos y negativos de ambos desarrollos por lo que sigue.

Marco Teórico

Ethernet



Es uno de los estándares más utilizados para las redes de computación pequeñas o en espacios reducidos como LAN, *Local Área Network*. Aparece comercialmente en 1980 y se estandariza cinco años más tarde como red oficial en la IEEE¹ bajo el protocolo 802.3 o "Ethernet", haciéndose la red más utilizada en ámbitos LAN, hasta el día de la fecha.

La historia de Ethernet se remonta desde 1973, cuando *Robert Metcalfe*, empezó sus investigaciones en el centro de XEROX, dentro del departamento de PARC² ubicado en la localidad de Palo Alto, CA. Donde describió cual era su idea de funcionamiento sobre lo que sería Ethernet y nombrándolo así por primera vez.

La idea surgió de la responsabilidad que se le había asignado de conectar un conjunto computadoras personales en XEROX y de esta manera lograr conseguir compartir recursos.

Con la experiencia del proyecto ARPANET³ (sería luego la base de Internet) y analizando profundamente el documento del proyecto ALOHA PACKET NETWORK, con la intensa colaboración de *Dave Boggs*⁴, se instaló la primera red Ethernet. Su objetivo fue conectar una impresora laser a un conjunto de terminales o PCs.

Siendo las impresoras laser un recurso escaso, por su alto valor, este nuevo desarrollo lograba que dicho recurso se compartiera; y a su vez permitiría también la conectividad acceso a Internet de alta velocidad a todas las computadoras de la compañía.

La primera instalación funcionaba a través de un grueso cable coaxial, instalado en el medio de los pasillos el cual funcionaba como bus principal al que se conectaban las PC's de cada oficina. A este cable principal se lo llamo "The Ether"... "los paquetes subían al Ether y los obteníamos del Ether"⁵, expresó Robert en una entrevista, el nombre se basa en el concepto físico del elemento Ether; argumentando que en el conductor los paquetes de datos viajaban "como en un vacío" tanto por la eficiencia como por la velocidad, que habían logrado.

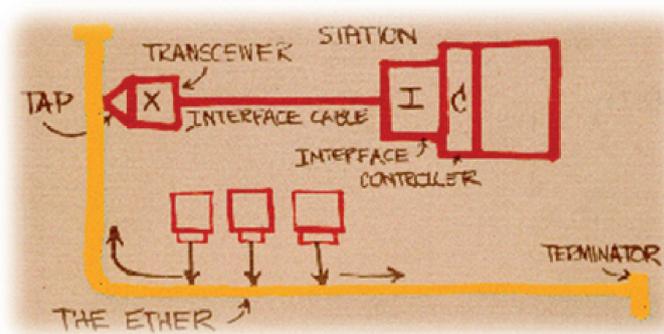


Imagen 1.0 – Bosquejo de Ethernet.

¹ IEEE: Instituto de Ingenieros Eléctricos y Electrónicos.-

² PARC: Xerox PARC (Palo Alto Research Center, 'centro de investigación de Palo Alto') era una división de investigación de Xerox Corporation, con sede en Palo Alto (California, EE.UU.).-

³ ARPANET: La red de computadoras Advanced Research Projects Agency Network (ARPANET) fue creada por encargo del Departamento de Defensa de los Estados Unidos ("DOD" por sus siglas en inglés) como medio de comunicación para los diferentes organismos del país.-

⁴ David Reeves Boggs (nacido en 1950, EEUU) Ingeniero de radiocomunicación y electricidad desarrollador de los primeros protocolos de internet, gateways, tarjetas de interfaz de redes y co-inventor de Ethernet.

⁵ Minuto 2,00; Metcalfe, Bob, perf. The History of Ethernet. 2006. Filmstrip. 19 Feb 2013.

<<http://www.youtube.com/watch?v=g5MezxMcRmk>>.

⁶ Minuto 2,00; Metcalfe, Bob, perf. The History of Ethernet. 2006. Filmstrip. 19 Feb 2013.

<<http://www.youtube.com/watch?v=g5MezxMcRmk>>.

Luego de estas primeras aplicaciones, Robert, pasó a trabajar para una tercera compañía conocida en aquel momento como 3COM, la cual se especializaría en producir productos para Ethernet. Esto se pudo concretar porque para 1979, tanto XEROX como INTEL habían decidido trabajar con Ethernet como un estándar de interconexión de sus productos.

3COM fue el encargado de vender las primeras tarjetas Ethernet, para que todo poseedor de computadoras personales pudiera conectarlas entre si y compartir recursos. Dio inicio así, a una de las capacidades y características más importantes de la red "The Sharing Era"⁸.

Uno de los mayores peligros en la historia de Ethernet apareció en su pelea contra el protocolo "Token Ring"⁹ desarrollado por IBM, Ethernet pudo salir adelante venciendo en la carrera tecnológica a TR porque pudo adaptarse perfectamente a las realidades del mercado, y actualizarse pasando de el cable coaxial a utilizar un par trenzado (UTP), aumentando la velocidad de transmisión y convirtiéndose en el dominante de todas las redes LAN.

Metcalf menciona, que hasta 1994 cuando aparece WWW (World Wide Web), Ethernet fue el protocolo de conexiones de computadoras personales por excelencia. Posterior a ello, se planteo el futuro de Ethernet en cuatro direcciones¹⁰:

- a) "Creciendo", porque sus velocidades son cada vez mayores, evoluciono de 2,94 Mbps hacia 10 Gbps y continúa aumentando.
- b) Ethernet crece "a través" es decir se lo utilizo para interconectar distintas redes WAN, remplazando el equipamiento antiguo por "Ethernet Compatible Equipment".
- c) Ethernet crece sobre aplicándolo a redes WIFI / WiMAX.
- d) "Desciende" Ethernet, se refiere que el protocolo es compatible con micro controlador o sistemas de micro-control, los cuales son cada vez más comunes en nuestra vida diaria, evolucionando con la capacidad de introducir redes más fácilmente.

Ethernet está ubicado en la capa física del modelo OSI¹¹. El concepto de "Frame" se origina debido a que la información y los datos para poder ser enviados deben dividirse de alguna forma; es necesario hablar en algún lenguaje para poder expresarse e interpretar los datos correctamente, es decir esto es en varios "frames" o *paquetes de datos*, que a su vez contienen la información de donde vienen y hacia donde van las denominadas "MAC Address"¹², y surge una codificación conocida como EtherType¹³.

Esas son las tres partes más relevantes del paquete. A su vez dependiendo de la configuración de la red, poseen un sistema de detección de errores.

Lo interesante de esta tecnología es que el formato original del paquete Ethernet, se mantiene constante desde su creación y no sufrió grandes cambios a pesar de la evolución de velocidades y de modificaciones de tecnologías.

802.3 Ethernet frame structure								
Preamble	Start of frame delimiter	MAC destination	MAC source	802.1Q tag (optional)	Ethertype (Ethernet II) or length (IEEE 802.3)	Payload	Frame check sequence (32-bit CRC)	Interframe gap
7 octets	1 octet	6 octets	6 octets	(4 octets)	2 octets	42 ^{note 2} –1500 octets	4 octets	12 octets
						64–1522 octets		
						72–1530 octets		
						84–1542 octets		

Imagen 1.1 – División de Paquete Ethernet.

⁷ 3Com NASDAQ: COMS fue uno de los líderes en fabricación de equipos para infraestructura de Redes Informáticas. La compañía fue fundada por Robert Metcalfe y otros socios en 1979 y su sede social está en Marlborough, Massachusetts. El nombre 3Com hace referencia a que los intereses de la compañía son Computadoras, Comunicaciones y Compatibilidad.

⁸ The Sharing Era, hace referencia al inicio de la era de internet en donde se simplifica la manera de compartir recursos e información increíblemente, logrando una interrelacion a través de un nuevo medio de comunicación, que cambio el mundo.

⁹ Token Ring es una arquitectura de red desarrollada por IBM en los años 1970 con topología física en anillo y técnica de acceso de paso de testigo, usando un frame de 3 bytes llamado token que viaja alrededor del anillo. Token Ring se recoge en el estándar IEEE 802.5. En desuso por la popularización de Ethernet; actualmente no es empleada en diseños de redes.

¹⁰ Minuto 5,15; Metcalfe, Bob, perf. *The History of Ethernet*. 2006. Filmstrip. 19 Feb 2013.

<<http://www.youtube.com/watch?v=g5MezxMcRmk>>.

¹¹ El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (en inglés open system interconnection) es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1984. Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.

¹² En las redes de computadoras, la dirección MAC (siglas en inglés de media access control; en español "control de acceso al medio") es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red. Se conoce también como dirección física, y es única para cada dispositivo.

¹³ EtherType es un campo de dos valores octales en un frame Ethernet. Es utilizado para identificar la codificación utilizada en el payload.-

¿Cómo Funciona Ethernet?

Cada estación (computadora o dispositivo) abre una conexión (socket) en el cual el frame Ethernet va a ser procesado e identificado. Este socket va a rechazar todo paquete que venga con un "Destination MAC Address" que no sea el esperado y va a permitir el paso solo al que sea el indicado, luego dentro del frame se procesara el Ethertype, para poder conocer con que protocolo de demodulación se deberá procesar el payload, que contendrá los datos en sí.

Como cada frame se puede auto identificar de cualquier otro, le permite al programador poder mandar paquetes Ethernet con distinto Ethertype e información en la misma red LAN y así aprovechar al máximo su ancho de banda, para transferir información de distintas aplicaciones.

Lógicamente como una gran variedad de computadoras están conectadas a la misma red a través del mismo cable, la información (frames) llegan a todas estén destinadas o no, por lo tanto las placas de red están configuradas para interrumpir al CPU o micro controlador constantemente si el frame no es destinado para ellas.

Actualmente esta tecnología se está trasladando y es responsabilidad del Switch realizarla, disponiendo la tabla de direccionamiento para evitar mandar el paquete a todas por igual y no solo a la destinataria.

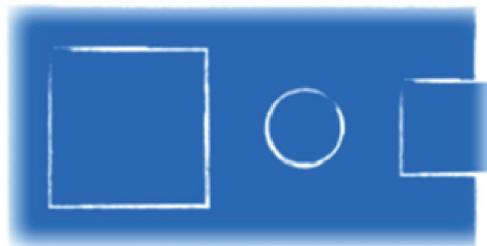
Así como se comparte la información a todos, también se reparte el ancho de banda, por ejemplo si dos maquinas están conectadas a la misma red el ancho de banda de la misma se divide a la mitad, para cada una, y así sucesivamente.

Repetidores/Hubs/Switchs

Repetidores, su función es aumentar el nivel de señal cuando la misma llega muy degradada. Los primeros eran solo de dos bocas y servían como divisores. Cuando los mismos crecieron en tamaño se empezó a realizar configuraciones estrella de redes u otras topologías con la ayuda de estos componentes, incrementando la eficiencia de las mismas.

A diferencia de los anteriores los Switchs, son dispositivos intercomunicadores de Ethernet, cuya función es chequear el paquete completo (no solo los primeros bytes) y corroborar así a quien va dirigido este paquete. Para esto el paquete necesariamente debe cambiar de nivel en el modelo OSI, y se pierde el concepto de RAW Ethernet. Si el paquete es apto, entonces el Switch lo deja pasar y sabe hacia dónde, ya que compone una tabla de MAC Address asociadas a cada una de las bocas, para que luego de realizar el flooding inicial el Switch conoce que dispositivo está conectado a cada boca.

mBed



Los micros controladores de la familia mBed son un modelo de plaquetas de desarrollo, diseñados para realizar prototipos de proyectos digitales o electrónicos utilizando micro controladores de la familia CORTEX ARM¹⁴, se caracterizan por ser flexibles, de bajo riesgo y profesionales.

Se embeben en una placa de desarrollo pequeña de 40-pines que lo hace más conveniente para prototipos con soldadura en protoboard y PCB¹⁵.

¹⁴ ARM es una arquitectura RISC (Reduced Instruction Set Computer=Ordenador con Conjunto de Instrucciones Reducidas) de 32 bits desarrollada por ARM Holdings. Se llamó Advanced RISC Machine, y anteriormente Acorn RISC Machine. La arquitectura ARM es el conjunto de instrucciones de 32 bits más ampliamente utilizado en unidades producidas.

¹⁵ En electrónica, un circuito impreso, tarjeta de circuito impreso o PCB (del inglés printed circuit board), es una superficie constituida por caminos o pistas de material conductor laminadas sobre un sustrato no conductor

Poseen una interfaz USB integrada que simplifica la conexión con la PC, mostrando el mBed como un "flash drive" fácilmente configurable, donde con un sistema de "drag and drop" se transfieren programas ya compilados al micro controlador.

Así también es utilizable como puerto COM Serie con cualquier programa que admita esta configuración, para lograr realizar una comunicación mBed-PC, a través de una terminal cruda.

El compilador para el mismo es una aplicación online que es 100% "user-friendly" y a su vez es muy poderoso ya que, se compatibiliza con distintos de los compiladores más utilizados mundialmente de lenguajes de programación como pueden ser Keil uVision, Code Red, GCC.

El lenguaje de programación más aconsejable para trabajar en los desarrollos mBed son C y C++, tienen una serie de librerías configurables que manejan la mayoría de los periféricos de I/O que poseen escritas en C++.-

¿Cómo Funciona el modulo mBed?

Se desarrolla el programa utilizando un editor de texto, es recomendable escribir el mismo en C para luego asegurar la compatibilidad con el compilador online.

Uno de los editores de texto más conocidos y populares para programar en C es VIM, Vi Improved, son editores freeware que permite configurar variables, comandos, tabulaciones para el mejor estilo de programación.

Otra de las razones más determinantes a la hora de decidir escribir el proyecto en C es que, es el punto medio perfecto entre el hardware y el software, hará que la tarea del programador sea mucho más simple, sin tener que bajar a un lenguaje de muy bajo nivel como el ASM a nivel de bits: y a su vez tampoco trabajar en un nivel tan alto como el java sin entender el hardware con el que trabajaremos.

La tarea más importante del ingeniero electrónico en este caso será desarrollar un software que funcione y esté totalmente interconectado con el hardware que diseñaremos. No solo es la programación lo importante sino la compatibilidad de la misma con el hardware.

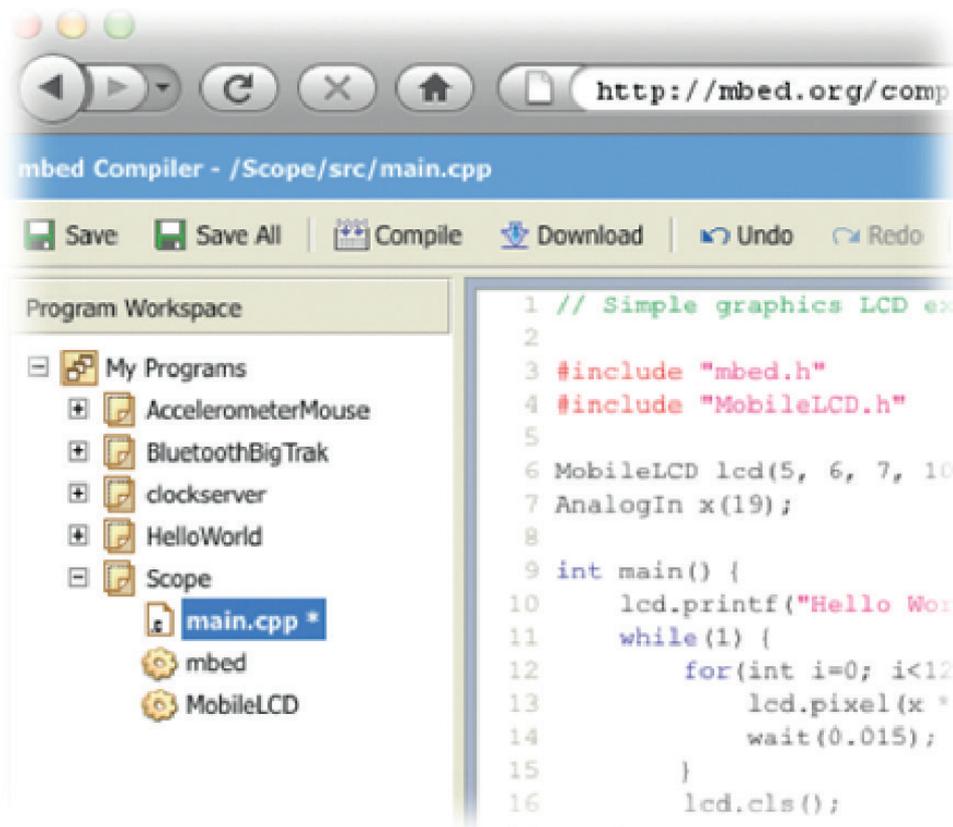


Imagen 1.2 – Compilador mBed.

Para ayudarse durante el desarrollo del programa, se debe conocer las librerías que posee el micro controlador, las mismas están disponibles en la página web de los desarrolladores del kit. La facilidad a la hora del desarrollo será directamente proporcional y dependerá del conocimiento de las librerías, ya que es probable que esto simplifique el trabajo evitando la escritura de muchos códigos fuentes que están en las librerías.

También existen librerías del tipo beta, desarrolladas por programadores personales y usuarios de mBed a lo largo del mundo, si bien no están oficialmente verificadas por el team de mBed, algunas veces son muy útiles y poderosas que las oficiales, pero solo hay un foro de reclamo/consulta, y existe deslinde de responsabilidad ante cualquier inconveniente, es uno de los riesgos.

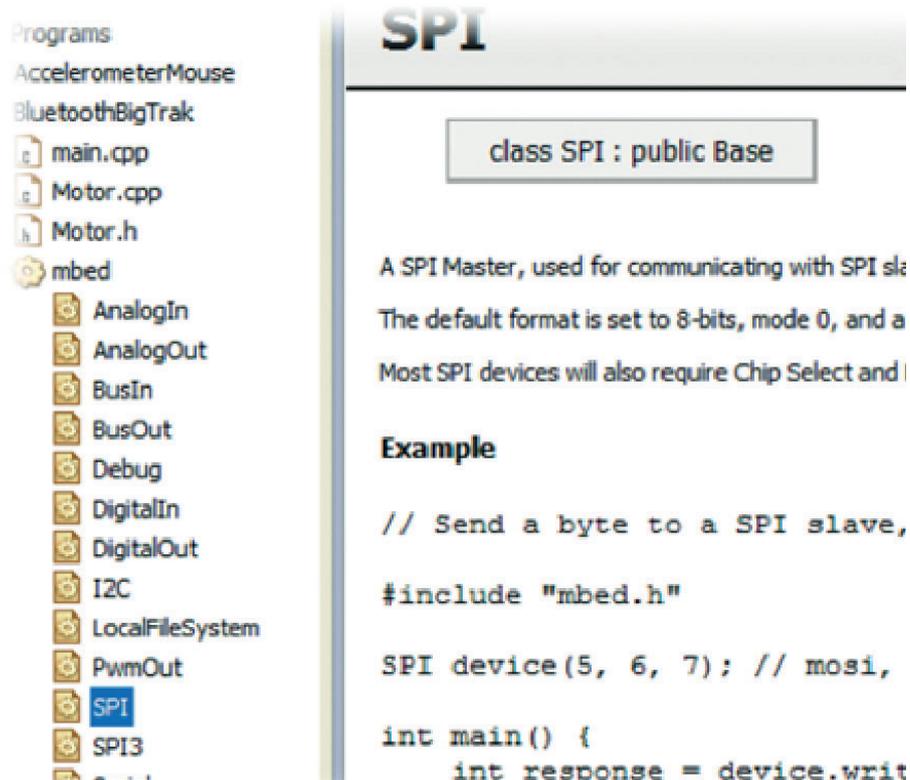


Imagen 1.3 – Librerías mBed.

Una vez que la escritura del código fuente finaliza, se debe compilar en el compilador online y verificar la correcta escritura. Si es así el mismo compilara y generara un archivo ejecutable que se descargara automáticamente a la computadora. Si existe algún error en la sintaxis del programa el compilador entregará el error conjuntamente con la descripción y el número de línea donde se produjo.

Luego se debe copiar el archivo al flash drive que representa el mBed conectado a través del cable USB y presionar el botón RESET en el mBed. Automáticamente el mismo comenzara a funcionar ejecutando el programa.

ADC - DAC

En esta sección se desarrolla los siguientes temas:

- Introducciones a señales y datos analógicos en micro controladores.
- Conceptos de conversión de señales de analógico a digital.
- Pines disponibles en mBed de entradas/salidas analógicas.
- Ejemplos simples de input/output de datos analógicos.

Los micro controladores generalmente están obligados a trabajar e interactuar con señales analógicas. Los mismos son capaces de convertir datos analógicos de entradas, que pueden proveer un sensor de temperatura, un micrófono, etc. a datos del formato digital (ceros y unos).

Asimismo son capaces de convertir sus datos procesados digitalmente a señales analógicas que pueden ser muy útiles a la hora de excitar un parlante, por ejemplo.

ADC → Conversor Analógico a Digital.-

Es un circuito electrónico, que tiene una salida en formato digital directamente proporcional a su entrada en formato analógico. De esta forma logra efectivamente “medir” la entrada analógica y sacar un dato binario proporcional a su tamaño. El rango de trabajo de un ADC es generalmente determinado por su tensión de referencia.

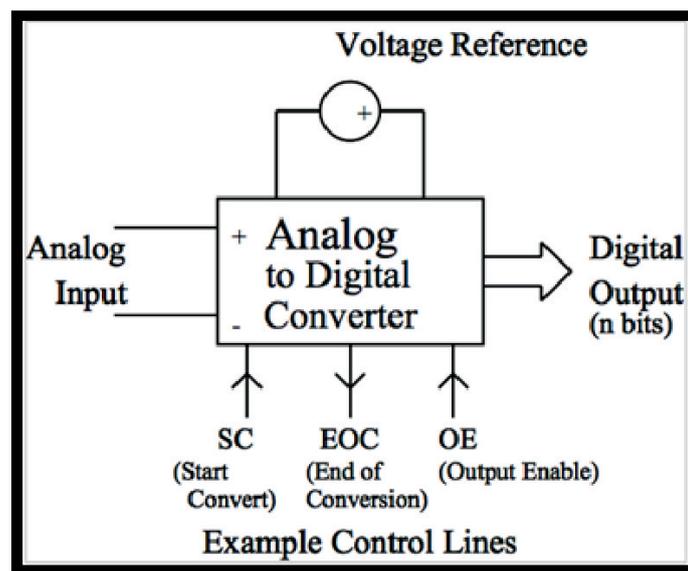


Imagen 1.4 – ADC.

En la imagen superior se puede ver que la conversión puede empezar por la señal SC, *Start Conversion*. El mismo tarda un tiempo finito (estamos trabajando sobre circuitos determinísticos) en convertir la información y sale directamente una vez autorizada por la línea OE, *Output Enable*, al bus de datos si se desea del micro controlador.

Cuando se convierte una señal analógica a una digital, repetidamente tomamos ejemplos o “samples” de la señal y se cuantifica la misma en formato digital.

Estos samples son tomados cada un determinado período, que definen el termino de frecuencia de muestreo.

La frecuencia de muestreo.-

La misma depende en gran medida de las señales que se desea muestrear. Si la misma es una señal con cambios muy rápidos, se necesitara una frecuencia de muestreo elevada para que estos cambios repentinos sean identificados en el muestreo de la misma.

El teorema del muestreo de Nyquist¹⁶ aconseja utilizar por lo menos una frecuencia de muestreo que sea el doble (como mínimo) de la frecuencia de internet más alta de la señal de datos.

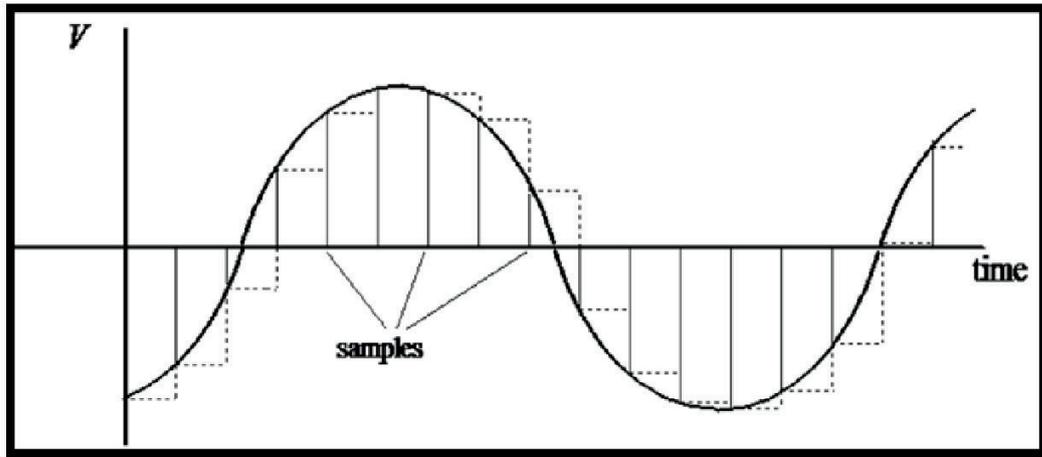


Imagen 1.5 – Sampling Frequency

En mBed poseemos seis entradas de señal que permiten la inyección de señales analógicas, las mismas se encuentran del pin 15 al 20.

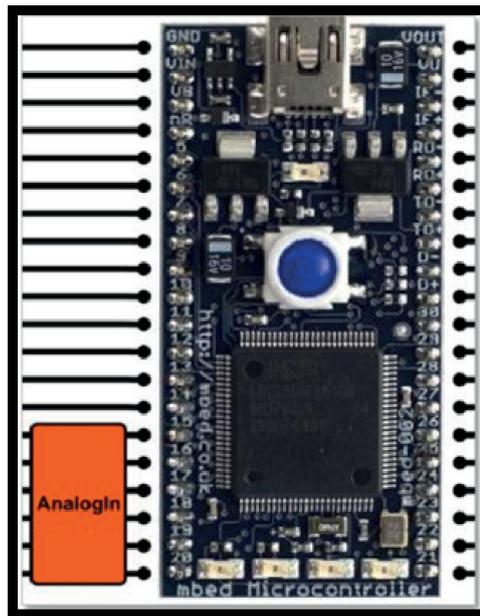


Imagen 1.6 – Analogue Input mBed.

¹⁶ El teorema de muestreo de Nyquist-Shannon, también conocido como teorema de muestreo de Whittaker-Nyquist-Kotelnikov-Shannon, criterio de Nyquist o teorema de Nyquist, es un teorema fundamental de la teoría de la información, de especial interés en las telecomunicaciones.

DAC → Conversor Digital a Analógico

La conversión digital-analógica, se puede representar como un diagrama con datos de entrada digitales, D , y una salida analógica V_o . El rango de salida de tensión de un DAC es una diferencia entre $V_{max}-V_{min}$, particularmente definido por la tensión de alimentación que recibe el modulo DAC.

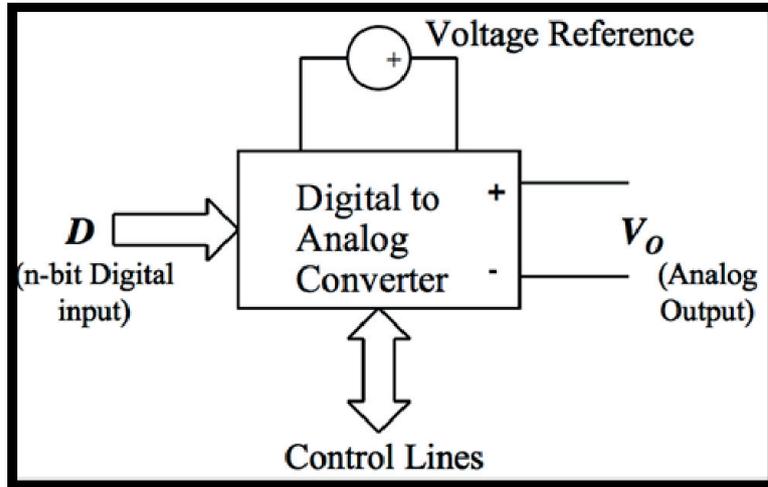


Imagen 1.7 – DAC

Para cada valor de entrada al DAC corresponde un valor de salida analógico dado por:

$$V_o = \frac{D}{2^n} * V_r$$

$$V_r = V_{max} - V_{min}$$

El micro controlador mBed adoptado posee un DAC de 10 bits, y un V_r de 3,3Volts.

Por lo tanto de este punto se puede deducir el rango de trabajo, o resolución, del DAC del mBed, el mismo es $3,3/1024$ (2 a la 10) = 3,2mV per bit de diferencia, entre cada nivel de salida.

En mBed tiene un solo puerto de salida analógica en el siguiente pin:

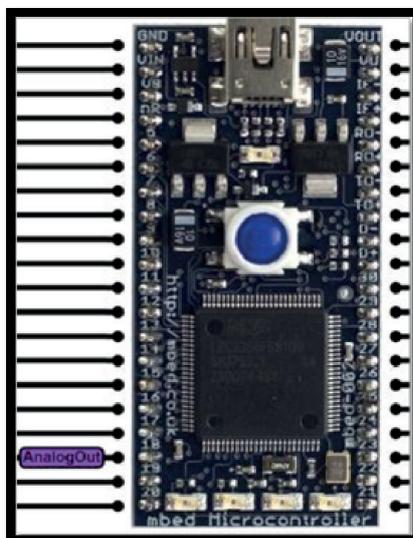


Imagen 1.8 – Analogue Output

C / C++

C es un lenguaje de programación creado en 1972 por *Dennis M. Ritchie*¹⁷ en los Laboratorios Bell como evolución del anterior lenguaje B, a su vez basado en BCPL.

Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para la creación de aplicaciones.

Se trata de un lenguaje tipificado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, se ayuda de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portátil entre plataformas y/o arquitecturas.

Escribiendo el programa en C

Una de las actividades primordiales antes de empezar a escribir un programa en C, es pensar la lógica del mismo, definir a grandes rasgos las distintas etapas que va a poseer el programa y tratar de analizar todos los estados y variables que van a afectar al mismo.

Sin si quiera pensar aun en que lenguaje lo codificaremos, esto es aplicable para cualquier lenguaje de programación. Una vez que poseemos el diagrama de estados, y flujograma detallado, podremos realizar la separación en bloques del proyecto y será más fácil realizar la codificación.

Para facilitar esta tarea construimos un diagrama de estados, FSM¹⁸, en donde se describan las acciones y los estados por los que el programador tiene seguridad, que el programa va a tener que pasar para poder funcionar correctamente.

Es necesario realizar este diagrama lo más detalladamente posible para facilitar la tarea que tendrá el programador ya que el mismo va a definir el éxito que va a tener el programador a la hora de escribir las funciones e interconectarlas. Una vez que empieza a codificarlo, deberá incluir en el diagrama de estado todas las acciones y posibles situaciones que se pueden presentar en el problema que deseamos solucionar con el programa.

Es muy probable que un diagrama empiece con 20 estados luego pase a tener solo 7, se analiza y determina que son innecesarios y luego el proyecto final termine funcionando con 15 que se tuvieron que incorporar sobre la marcha. No existe un método predefinido de hacerlo, la técnica está en tener en cuenta todas las variables que entran en juego en el problema y ver cuál es la manera más óptima de aprovecharlas.

Una vez que se desarrolla satisfactoriamente el diagrama de estados del sistema a desarrollar, es simple escribir el programa y se minimiza la cantidad de errores lógicos que se pueden producir.

Cada estado va a tener excitaciones, que lo lleven a un estado diferente, y que también hayan producido la llegada al mismo, o que simplemente lo mantengan en el mismo. Cada excitación viene acompañada de una serie de acciones que se desean ejecutar a la hora de realizar el cambio de estado.

El formato de la tabla que se debe armar para poder manejarse entre estados, debería seguir el siguiente flujo:

Excitación→Próximo Estado→Acciones del Cambio

En donde la excitación es la iniciadora del proceso de cambio de estado, la máquina de estados, a partir de esta excitación revisará cual será el estado destino y de una tabla obtendrá las acciones a realizar. Básicamente ese es el funcionamiento de una máquina de estados, aunque no es la única manera de implementarla.

¹⁷ Dennis MacAlistair Ritchie (9 de septiembre de 1941 - 12 de octubre de 2011 EEUU) fue un científico computacional. Colaboró en el diseño y desarrollo de los sistemas operativos Multics y Unix, así como el desarrollo de varios lenguajes de programación como el C, tema sobre el cual escribió un célebre clásico de las ciencias de la computación junto a Brian Wilson Kernighan: El lenguaje de programación C.

¹⁸ Final state machine, o maquina de estados.-

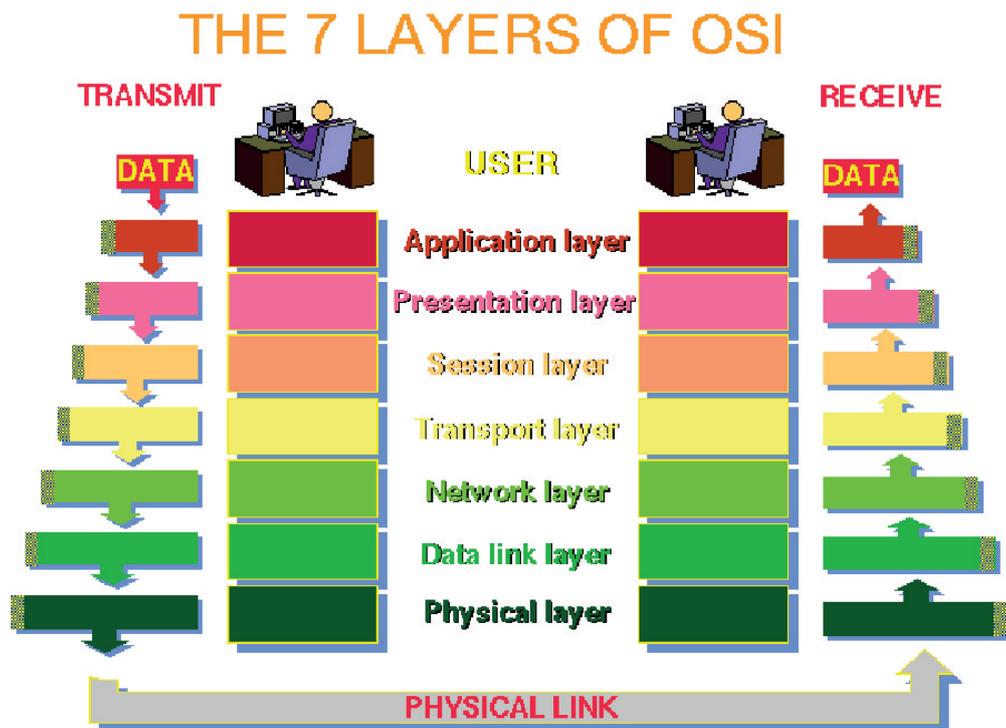
Por otro lado a su vez, un programa corriendo a la par de la máquina de estados estará encargado de obtener novedades para poder saber de qué forma y cuál será la excitación que le pasará, de alguna forma será el que “tome novedades, e informe”. Estas novedades pueden proveer de distintos lugares, ya sea periféricos externos, timers internos, interrupciones etc.

Descripción general de la idea

La idea del proyecto es demostrar la habilidad de transmitir audio en calidad audible, utilizando procesadores poco poderosos y por lo tanto muy económicos, tratando de sustituir la utilización de la comunicación tradicional de línea telefónica en lugares reducidos.

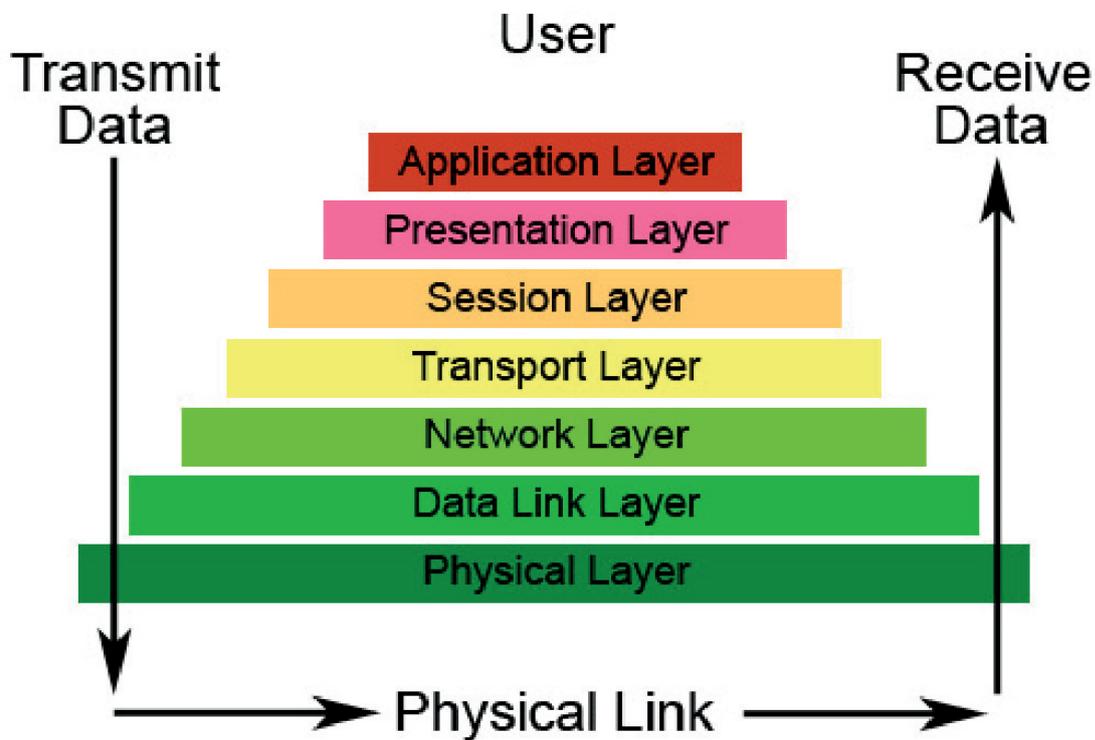
Para lograrlo se necesitan mezclar conocimientos de distintas ramas de la electrónica y así poder desarrollar un producto que transmita voz a través de paquetes crudos en la capa Ethernet (link físico); evitando toda la codificación y preparación necesaria que tendría que realizarse si el mismo es enviado en otra capa de comunicación del modelo OSI.

Como por ejemplo evitando la codificación del stack TCP-IP, lo que no solo simplifica la codificación sino que también nos permite montar el sistema en procesadores menos poderosos.



La imagen superior representa el Modelo ISO-OSI, cada una de las capas posee su propia codificación, para enviar un paquete el mismo baja desde la capa de aplicación hasta la capa física en donde se encuentra el link físico (cable coaxial, fibra óptica, antena de radiocomunicación; al mismo se le agregan campos de cabecera y secciones de control y codificación que lo convierten en un paquete más pesado y por lo tanto con más bits a transmitir, aunque el “peso” neto de la información que estamos enviando no cambia, el “peso” total del paquete sí.

The Seven Layers of OSI



La idea que se propone es realizar un dispositivo que trabaje directamente en las capas inferiores a la altura del protocolo Ethernet crudo, y así evitar la necesidad de agregar todo este “peso” extra al mismo haciendo el paquete mucho más liviano y eficiente a lo que se refiere a relación *peso paquete/datos útiles*. Y lograr enviarlo directamente a través de la capa física, sin necesidad de utilizar las capas superiores.

Una tabla de las diferencias más significativas de AoF Vs VoIP:

VoIP ¹⁹	AoE
higher layer (internet protocol) Utilizable vía internet	Lower level (capa física) Solo disponible en redes locales y LAN de espacios reducidos
mucho menos eficiente y más lento	instantáneo a tiempo real, mayor eficiencia
Necesita cargar un stack IP para configuración	No necesita Stack es transmisión cruda, más simple en procesadores pequeños
Admite compresión de audio (es necesaria)	No es necesaria la compresión pero si recomendable para aprovechar mejor la red Ethernet.

¹⁹ Página 16, Fernández Crocco, Gabriel. Voz sobre IP. MS thesis. Tesinas de Ingeniería y Tecnología Informática, Universidad de Belgrano, 2002. 2002. Web. <<http://184.168.109.199:8080/jspui/handle/123456789/302>>.

Descripción general del Módulo

El módulo está dividido en bloques para facilitar su entendimiento y así simplificar el trabajo, estos bloques están entrelazados entre sí por las señales que reciben y que producen para poder excitar el bloque siguiente.

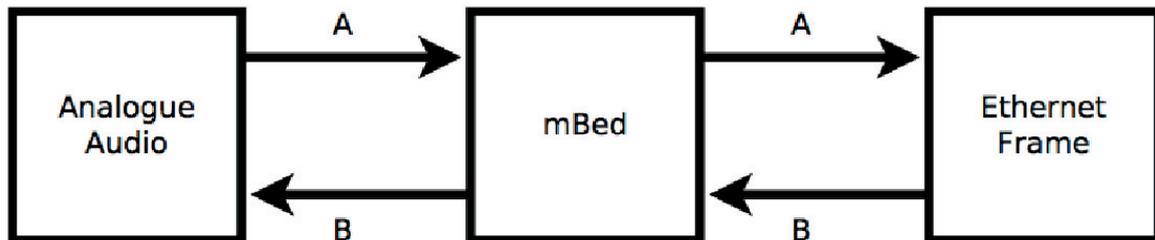


Imagen 1.9 – Module diagrama

Se hace el acercamiento a este proyecto de una forma piramidal de arriba hacia abajo en donde, en lo más alto tendremos la idea general y en la base los detalles técnicos – lógicos de codificación y cálculos teóricos. De esta forma se dividirá el proyecto en módulos simples y de fácil entendimiento.

Por ahora solo debemos seguir el camino de la señal de audio desde A hasta la finalización de B.

Empieza en forma de voz analógica la fuente que nos entrega audio a transmitir, debe ser capaz de brindar una señal analógica de baja potencia al equipo AoE. El mismo la codificará con la ayuda del ADC, (su funcionamiento fue detallado en el marco teórico) y la introduce en distintos paquetes Ethernet, es decir la “empaqueta”, luego este queda a la espera del usuario que deberá informarle que hacer con esos datos. Este proceso será explicado en detalle con la exposición de la máquina de estados. Para comprender como el sistema toma los datos del usuario, una vez conocido el destino introduce el paquete en “la nube” Ethernet, en donde será escuchada por un diferente y específico módulo, que ya en el camino B (de nuestro diagrama de bloques) decodificará esta señal y la sacará por una línea analógica para que pueda ser fácilmente amplificada, con la ayuda del DAC. (detallado en el marco teórico).

Lo más importante a tener en cuenta en este punto, es el camino de la señal y como la misma va cambiando de estados para simplificar su transmisión pero nunca pierde información.

El camino es:

Audio Analógico → ADC → Ethernet Frame → DAC → Amplificador → Audio.

En donde la zona roja será soportada y ejecutada por el mBed y la azul por los módulos soportes que se colocarán en cada AoE device.

Ethernet: se sub-dividió su utilización en dos partes, una será la encargada de comunicar los distintos módulos con el objetivo de “controlar” hacer pedidos de comunicación, avisar el estado de una comunicación, cortar, llamar, y por precaución mandar señales de estados; la otra parte va a ser la encargada de transferir la información el payload de audio codificado dentro de sus frames.

Si bien la manera de comunicarse por Ethernet será la misma, se divide para que se simplifique y sea fácil depurar errores y la construcción del proyecto por partes.

En lo que respecta a la programación del mBed el mismo se realizara en C y C++. (en el apéndice se pueden ver los programas detallados por nombres y documentados).

Para esto se utiliza un software freeware llamado Doxygen. Es un generador automático de documentación desde el código fuente. Al programar se utilizaron ciertas “keys” que este software interpreta automáticamente y genera toda la documentación del módulo desde los comentarios que se van definiendo, para explicar el funcionamiento de cada modulo del programa en C.

Es una buena práctica en la programación utilizar comentarios de cada función con la información específica de lo que hace, que datos recibe y que datos envía para que sea más fácil entenderla y poder relacionarla con otras funciones.

Si algún programador que no está relacionado con el proyecto posee esta documentación se le simplificará el trabajo, en el caso de querer utilizar la función.

La parte más complicada, y extensiva, del proyecto es la programación del módulo que necesita la interrelación con el ser humano a la hora de codificar los números ingresados por teclados o mostrar información en el display. Para esto se utiliza una máquina de estados, haciendo correctamente el diagrama de estados de la misma.

Como se expuso previamente, la misma empieza de una forma y va evolucionando mientras que se avanza en el proyecto, sea buscando su simplificación o agregando estados.

A continuación se exponen las utilizadas en el proyecto:

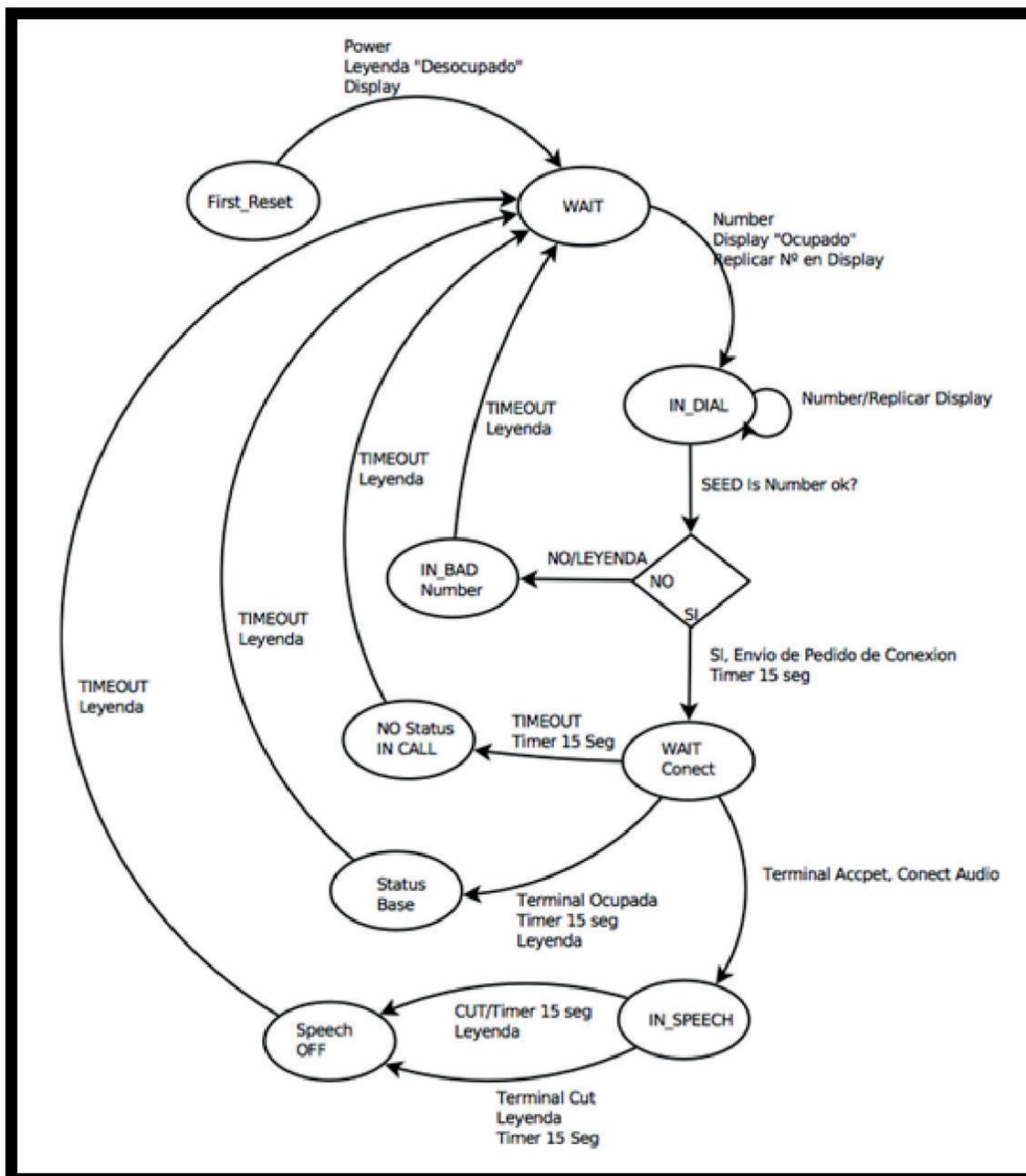


Imagen 2.0 – FSM 1V

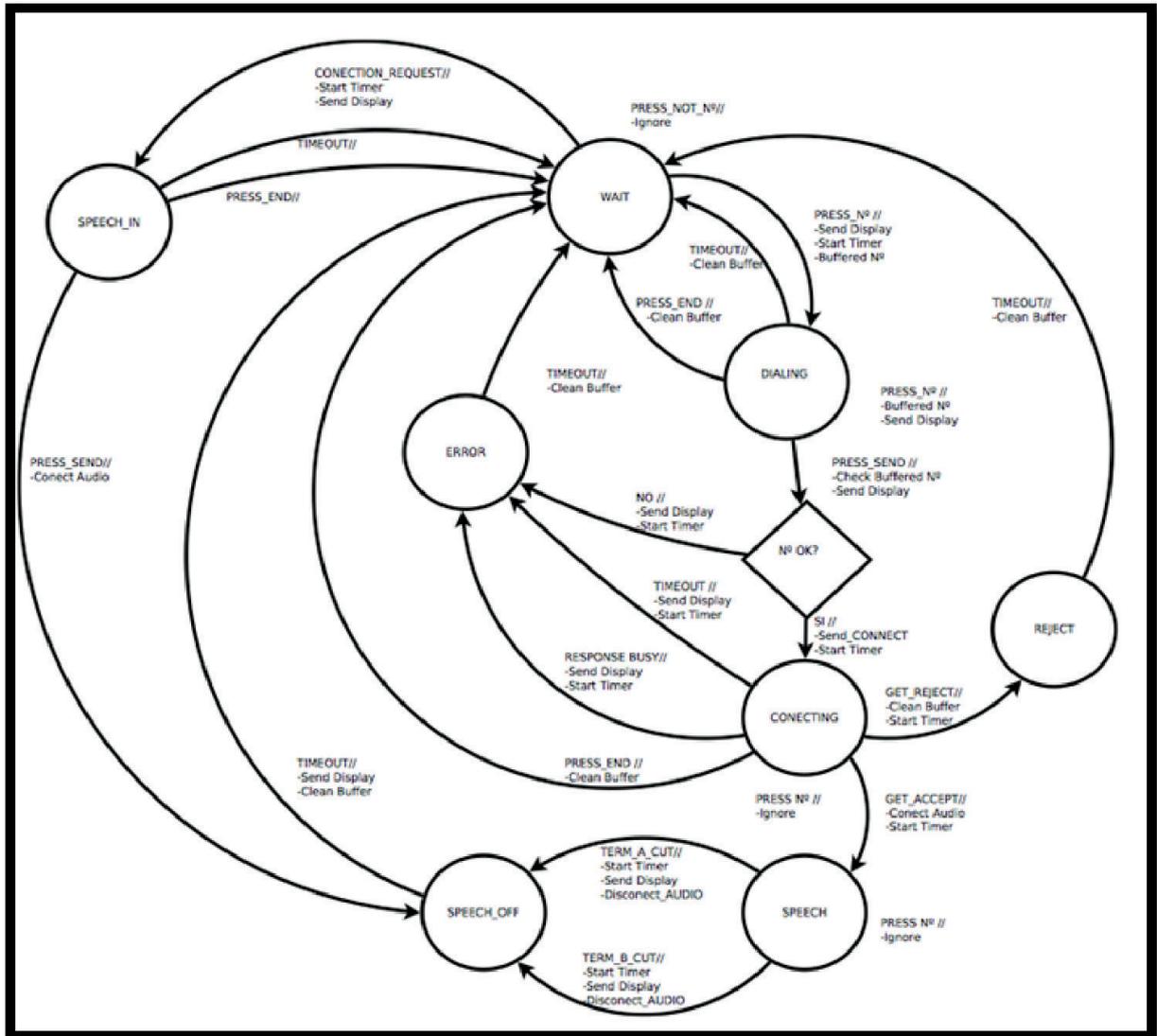


Imagen 2.1 – FSM 2V

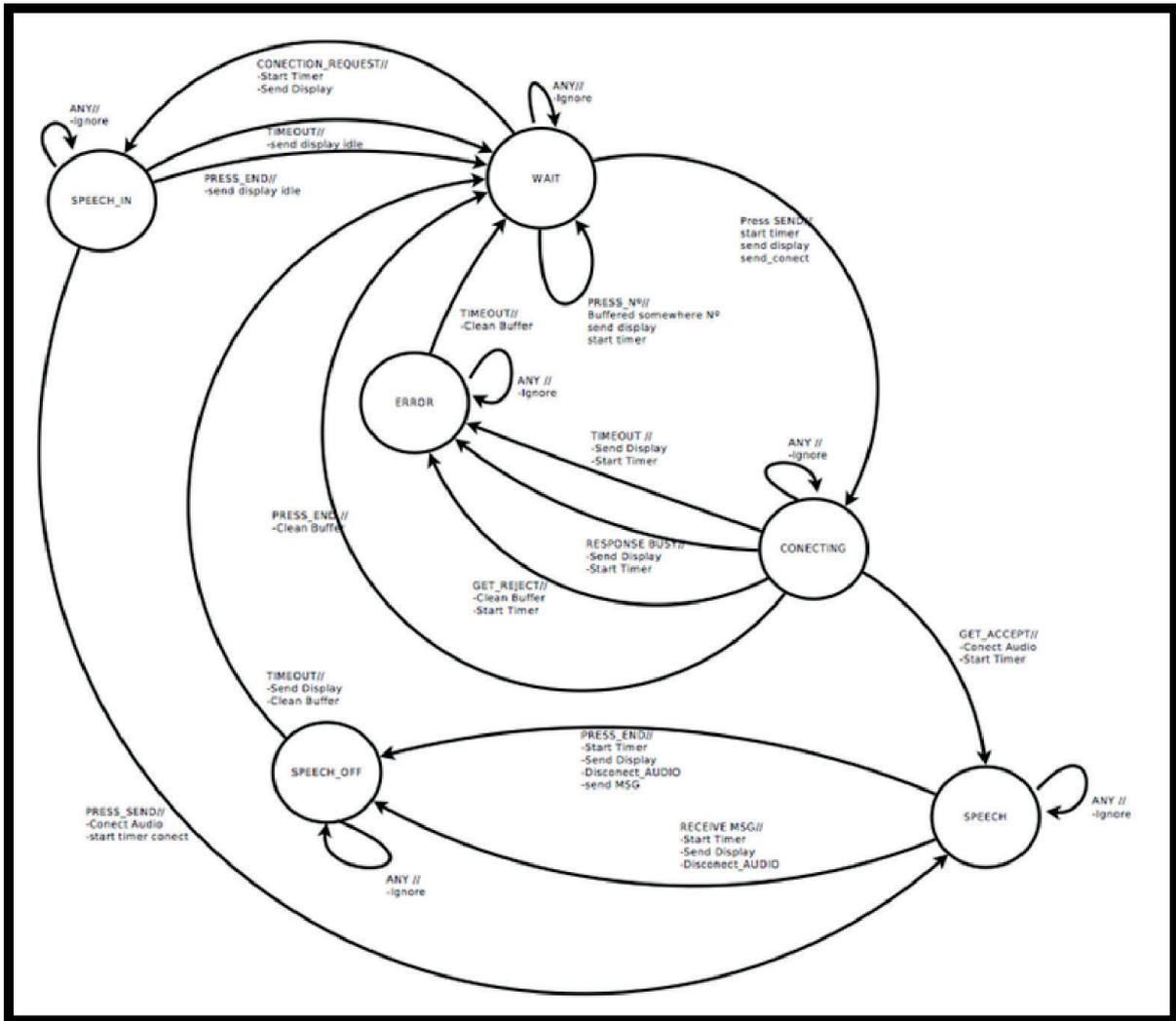


Imagen 2.2 – FSM 3V

Se puede ver como la teoría se cumple y a medida que el proyecto va evolucionando el diagrama de estados por el que se inicio cambia radicalmente, aunque no así el objetivo del proyecto.

Construcción del dispositivo AoE

Para llegar al producto final, se utilizaran distintos módulos obtenidos por separado, estos son:

→ Placa de desarrollo mBed LPC1768

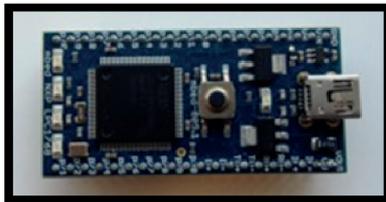


Imagen 2.3

→ Placa de Desarrollo Ethernet de CoolComponents



Imagen 2.4

→ Display LCD 16x2

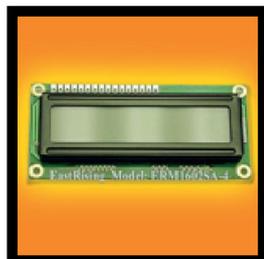


Imagen 2.5

→ Teclado Matricial 4x3

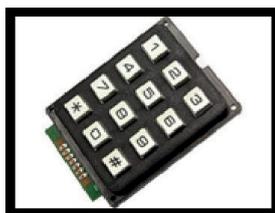


Imagen 2.6

→Cables del tipo WireRap y conectores varios.

Utilizando en conjunto los componentes descriptos se llega al producto final. Se agregaron además distintos conectores como por ejemplo Jacks de audio y conectores de tensión para adaptar el módulo. Los mismos fueron conectados y soldados conjuntamente y armados en un módulo de acrílico dado a conocer como producto final bajo el nombre "Módulo AoE".

Problemas de Diseño

Uno de los problemas presentados en el armado del HW fue la compatibilización de las masas, ya que estamos trabajando con distintas referencias a tierra, analógicas y digitales. Si estas no están correctamente separadas, se produce una interferencia entre las mismas. Su instalación tiene que ser diseñada para que el ruido que genera la corriente que drena por masa, la parte digital, no se introduzca en los filtros analógicos produciendo ruido audible.

La resolución fue, en parte, realizando un diagrama de masa, en donde nos aseguramos que todo esté conectado al mismo punto y los restos capacitivos e inductivos sean lo menos molestos posibles, las capacidades e inductancias impuras que no se pudieron eliminar se mejoran introduciendo resistencias y capacidades análogas para contrarrestarlas y disminuirlas.-

Otro problema, pero más simple, fue la adaptación de la caja de acrílico con los módulos, ya que no se consiguen los conectores necesarios para la correcta instalación y se debe adaptar cables y utilizar cable del tipo plano (o IDE de PC) y no WireRap, para que pueda ser de mas fácil de manejo.

Errores y Contratiempos

Errores e inconvenientes a la hora del desarrollo y el armado del producto.-

→ Durante el desarrollo no se produjeron errores críticos, si existió un contratiempo a la hora de la programación del módulo ADC. El mismo no llegaba a muestrear 44Khz por si solo, en su "Sampling Frequency". Para ser capaz de transmitir música en modo Hi-Fi en las frecuencias de 80hz – 22Khz, no tenía una frecuencia de muestreo tan alta y deformaba la señal a altas frecuencias.

La solución que se encontró, fue la de modificar la librería que posee nuestro micro controlador para manejar el módulo ADC, por una librería cruda, no la oficial, que si era capaz de configurar el módulo del ADC para que trabaje a esta velocidad. Esto, en la práctica, exigía muchísimo más al micro controlador. Asunto que no era necesario ya que nuestro objetivo es demostrar la capacidad de transmitir voz del estilo de canal telefónico y con 10Khz de frecuencia de muestreo se podrá transmitir fácilmente hasta 5Khz de información que es suficiente para la voz humana. Por lo tanto se volvió atrás a utilizar la librería original del mBed.

→ Otro inconveniente a la hora del desarrollo fue la utilización de librerías pre-diseñadas, al no ser propias y conocerlas el programador no puede saber al 100% cómo funcionan y debe estudiarlas por dentro para poder aprovecharlas al máximo y explotar su rendimiento. No todas tenían su documentación actualizada y completa.

→ Durante el armado el error más crítico que se superó, fue conseguir los componentes necesarios, la falta de elementos electrónicos en el país dificulta disponer del material necesario para armar el módulo final AoE. Se tuvo que importar la mayor parte de componentes, que provocó una demora no prevista en el armado del mismo.

Conclusiones y Comparaciones con productos actuales

Para concluir expresamos que el proyecto funciona como se esperaba y de la forma que se pensó y diseñó en un principio.

Puntos Positivos

- Es un módulo económico y potencialmente poderoso, teniendo la capacidad de reemplazar los intercomunicadores y telefonía interna en un lugar de tamaño regular, siempre trabajando dentro de redes LAN.
- Es más eficiente que VoIP a la hora de trabajar en cortas distancias, ya que no necesita subir al protocolo de internet (TCP/IP) del modelo OSI para poder transferir los paquetes, sino que permanece en los dos layer inferiores. Este punto fue estudiado en un principio y fue uno de los principales objetivos a lograr, haciendo que no se necesite un procesador tan poderoso para codificar y transmitir, haciendo muchísimo más simple la programación del CODEC, utilizando con mayor eficiencia la cantidad de información neta que se introduce en cada paquete.
- La información en VoIP se comprime para que sea más eficiente y poder enviar más en cada paquete, en AoE esto no sucede se envían los bits crudos en los paquetes sin utilizar compresión, para aprovechar mejor el ancho de banda de Ethernet, es recomendable en AoE trabajar también con compresión, de la misma manera que se haría si se quisiera enviar información por IP, pero para mantener la demostración simple, se decidió no realizar la compresión para ser capaz de mostrar los paquetes y apreciar el paquete crudo de Ethernet armado.

Puntos Negativos

- Solo se pueden interconectar módulos a través de Switch o Hubs y no se podrá usar un router. El mismo no interpretará los paquetes al no tener una dirección IP. Esto hace que sea un sistema local que se puede aplicar a pequeños lugares. Si bien es cierto que esto limita muchísimo a donde se podrá conectar el mismo está diseñado apuntando a lugares con redes LAN moderadas y la existencia de separadores de señales y nodos con HUBs y Switch, en donde el dispositivo es más eficiente que un sistema de VoIP.

Concluimos diciendo que AoE contiene una semejanza a la Voz sobre IP (VoIP),. AoE va a ser utilizado para transmitir audio de alta fidelidad, profesional de latencia. AoE va a ser un sistema que trabajara a típicamente 1 Mbit por canal y de latencia muy inferior (típicamente menos de 10 milisegundos) que VoIP.

Si bien son dos productos que trabajan de una forma muy similar, cada uno va a ser mejor que su compañero dependiendo de donde se quiera aplicar el sistema.

Para lograr *Audio over Ethernet* es condicionante una red de alto rendimiento. Las exigencias de funcionamiento pueden ser encontradas en una red local dedicada (el LAN) o el LAN virtual (VLAN), donde la misma siempre se ve sobre provisionada y de buena calidad.

A Futuro...

Para el trabajo planteado anteriormente, existen centenares de modificaciones y mejoras. Trataremos de plantear aunque sea un par de las mejoras más importantes que se realizarán al mismo de nuestra parte.

- a) Se agregará la compresión de audio, para poder utilizar el producto dentro de una red Ethernet más “cargada” y con más servicios compartiendo el ancho de banda. Así se logrará aprovechar al máximo el ancho de banda de Ethernet.
- b) Se aplicará el producto al servicio de “Public Address”, esto consiste en una modificación del software del producto para que el mismo cambie su topología a 1 AoE Master y X AoE Slaves, en donde el Master enviará el mensaje y los Slaves solo lo reproducirían.
- c) Se mejorará el diseño de la caja del producto importando cajas para placas electrónicas a medida, en donde los conectores quedarán mucho más profesional y pulcros.
- d) Se mejorará, diseñará y construirá a medida el breakboard necesario para el mBed, de forma de no tener conectores en desuso y mejorar el plano de tierras del producto.

Glosario

Peer to peer: Una red peer-to-peer, red de pares, red entre iguales, red entre pares o red punto a punto (P2P, por sus siglas en inglés) es una red de computadoras en la que todos o algunos aspectos funcionan sin clientes ni servidores fijos, sino una serie de nodos que se comportan como iguales entre sí. Es decir, actúan simultáneamente como clientes y servidores respecto a los demás nodos de la red. Las redes P2P permiten el intercambio directo de información, en cualquier formato, entre los ordenadores interconectados.

Public Address: Redes transmiten mensajes verbales y música dentro de escuelas u hospitales, y sistemas de emisión de baja potencia, que transmiten desde estaciones de radio o TV a pequeñas áreas.

AoE: Modulo construido para transmitir audio a través de canales Ethernet.

OSI: El modelo de interconexión de sistemas abiertos (ISO/IEC 7498-1), también llamado OSI (en inglés open system interconnection) es el modelo de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO) en el año 1984. Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.

Bibliografía

- Kernighan Y Ritchie (1988). *The C Programming Language*. United States: Prentice Hall, Inc.
- Keith Haviland (1998). *UNIX System Programming*. 2nd ed. United States: Addison-Wesley.
- Toulson, Rob. *Fast and Effective Embedded Systems Design: Applying the ARM mbed*. 1er. London: Newnes, 2012. Print.
- Fernández Crocco, Gabriel. *Voz sobre IP*. MS thesis. Tesisnas de Ingeniería y Tecnología Informática, Universidad de Belgrano, 2002. 2002. Web. <<http://184.168.109.199:8080/jspui/handle/123456789/302>>.
- NXP, . "UM10360." *LPC17xx User manual*. NXP, 19 Agosto 2010. Web. 19 Feb 2013. <http://www.nxp.com/documents/user_manual/UM10360.pdf>.
- NXP, . "LPC1769/68/67/66/65/64/63." *32-bit ARM Cortex-M3 microcontroller; up to 512 kB flash and 64 kB SRAM with Ethernet, USB 2.0 Host/Device/OTG, CAN*. NXP, 10 Agosto 2012. Web. 19 Feb 2013. <http://www.nxp.com/documents/data_sheet/LPC1769_68_67_66_65_64_63.pdf>.

Referencias de la Web

- <http://mbed.org> (Última revisión 19 de Febrero 2013, 8:49Hrs)
- <http://www.wikipedia.org> (Última revisión 19 de Febrero 2013, 8:49Hrs)
- <http://www.comsoc.org> (Última revisión 19 de Febrero 2013, 8:49Hrs)
- Metcalfe, Bob, perf. *The History of Ethernet* . 2006. Filmstrip. 19 Feb 2013. <<http://www.youtube.com/watch?v=g5MezxMcRmk>>.
- <http://mbed.org/handbook/mbed-Compiler> (Última revisión 19 de Febrero 2013, 8:49Hrs)

