

Tarjeta de audio de procesamiento de voz digital

Vázquez González Erick, ESIME Zacatenco

Abstract

El sonido y la computadora son inseparables, hoy en día en pocas aplicaciones se trabaja audio analógico y casi todas editan usando software y computadoras. Pero el sonido, en su origen continua siendo analógico. La computadora, por lo tanto, tiene que contar con un dispositivo que transforme el audio analógico en digital a este se llama tarjeta de audio o de sonido. En este reporte se describe el proceso de digitalizar una señal analógica de la voz humana, así como la serie de pasos a seguir para lograr este proceso

Keywords

Muestreo — Cuantización — ADC — Acondicionamiento de señal — Microcontrolador — Nyquist

¹ Ingeniería en Comunicaciones y Electrónica, ESIME Zacatenco, México DF, 2014

*Datos del Autor: gzerex@gmail.com, 55 2526 7770

Contents

Introducción	1
0.1 Objetivo General	1
0.2 Nivel usuario	2
0.3 Nivel técnico	2
0.4 Nivel sistema	2
1 Marco Teórico	2
1.1 Teoría de muestreo	2
1.2 Tamaño de muestra	2
1.3 Amplificador no Inversor	2
Capacitor Bypass	
1.4 Filtro de CD	3
1.5 Offset de CD	3
1.6 Convertidor Analógico-Digital	4
1.7 Arreglo R/2R DAC	4
1.8 Filtro Pasa Bajas	4
1.9 Seguidor de Tensión	5
1.10 Un-Offset de CD	5
2 Diseño	5
2.1 Esquemático	5
2.2 Simulación en MATLAB	5
Simulación de señal continua • Simulación de señal discreta	
2.3 Lista de Materiales	6
2.4 Programación ISP	6
2.5 Códigos de programación en C	6
Muestreo a 8kHz en puerto A0 • Muestreo a 8kHz en A0 con salida a DAC de 8 bits • Muestreo a 38.5kHz por el puerto A0 • Muestreo a 38.5kHz en puerto A0 con salida a DAC de 8 bits	
2.6 Circuito Impreso	8
3 Conclusiones y Observaciones	8
4 Bibliografía y referencias	9

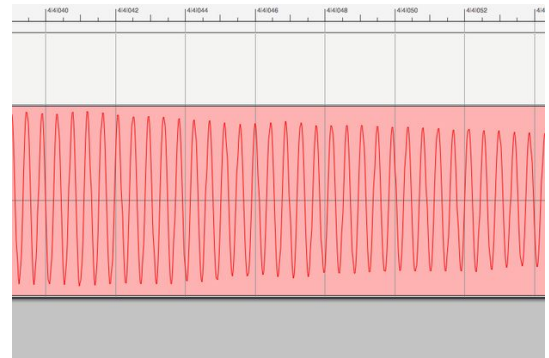


Figure 1. Señal de audio amplificada en tiempo

Introducción

Si alguna vez ha grabado audio en su computadora, es muy probable que haya visto una forma de onda que varía dependiendo de la intensidad del sonido grabado, si amplificamos esta señal en el tiempo (figura 1) podemos observar que esta señal está formada por cientos de miles de oscilaciones que van y vienen, a esto le llamamos señal de audio y cuando tratamos con señales de audio en la electrónica, esas oscilaciones representan voltajes en el tiempo. Una señal de audio típica oscila alrededor de 0V esto es normal para señales de audio, la amplitud de una señal de audio puede ser de 2V, sin embargo esto significa que alcanza su máximo voltaje en 2V y su mínimo en -2V esto es un problema, que debemos resolver, es decir, debemos acondicionar nuestra señal para poder ser leída por el microcontrolador

Objetivos

0.1 Objetivo General

Diseño simulación y construcción de un circuito que permita acondicionar una señal de voz para poder convertirla en una señal digital, por medio de un microcontrolador

0.2 Nivel usuario

- Obtener cualquier una señal de voz
- Convertirla en una señal digital
- Reproducir el sonido digital por medio de altavoces

0.3 Nivel tecnico

- Aplicar el teorema de Nyquist
- Simular la señal en MATLAB (Generar tonos de prueba)
- Identificar cada uno de los circuitos del sistema
- Probar cada circuito independientemente
- Integración del sistema

0.4 Nivel sistema

- Identificar cada uno de los sistemas que harán funcionar a nuestro circuito
- Amplificar la señal a 5v
- Aplicar un Offset de DC para que el microcontrolador pueda leer 2^8 valores (256)
- Canalizar por un puerto del microcontrolador la señal digitalizada
- Convertir la señal discreta en continua
- Remover ruidos de la señal a lo largo del sistema
- Filtrar frecuencias no deseadas
- Amplificar la corriente de salida del sistema

1. Marco Teórico

1.1 Teoría de muestreo

Nyquist afirmaba que una señal Analógica puede ser reconstruida, sin error, de muestras tomadas en intervalos iguales de tiempo, la razón de muestreo debe ser igual o mayor al doble de su ancho de banda de la señal Analógica

La teoría de muestro define que para que una señal de ancho de banda limitado la frecuencia de muestreo FM debe ser mayor que 2 veces su ancho de banda BW medido en Hertz

$$F_m = 2 * BW \quad (1)$$

Si nuestra señal a ser digitalizada es la voz entonces el ancho de banda de la voz es de $4000Hz$ su frecuencia de muestreo será $8000Hz$, equivalente a 8000 muestras por segundo $1/8000$ entonces la razón de muestreo de la voz debe ser de al menos $8000Hz$ para que pueda regenerarse sin error.

$2 * BW$ es la frecuencia de Nyquist o frecuencia de muestreo

El oído humano solo puede detectar sonidos en el rango de $20Hz$ a $20kHz$, de acuerdo a la teoría de muestreo debemos muestrear la señal al menos a $40KHz$ para reconstruir una señal de sonido aceptable al oído humano, el sonido digitalizado se compone de muestras de sonido captadas a diferentes frecuencias. Cuantas más muestras de sonido se capten por segundo, mayor será su calidad, los DVD utilizan $48 kHz$, los CD de audio usan una frecuencia de muestreo de $44,1 kHz$, la Radio FM $22kHz$, y las voces de la redes telefónicas usan $8 kHz$.

La frecuencia de muestreo que seleccione dependerá de la naturaleza del sonido.

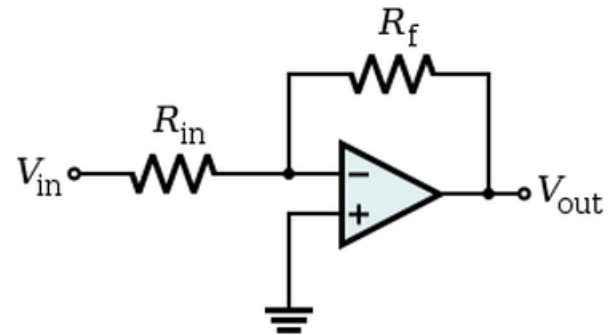


Figure 2. Amplificador no Inversor

1.2 Tamaño de muestra

El tamaño de la muestra de sonido o Taza de bits, (bit rate) determina su margen dinámico. viene definido por la siguiente formula

$$2^N \quad (2)$$

Donde N es el número de bits, el sonido de 8 bits entonces, suministra 255 valores posibles, mientras que el de 16 bits permite más de 65.000 valores posibles.

Las computadoras por lo general reproducen un sonido de 16 bits para la música que permite reproducir música suave como de música fuerte, por ejemplo, la orquestal. Para la voz o la música de volumen más constante, telefonía fija, el sonido de 8 bits suele dar buenos resultados.

Si necesita reducir tamaño de un archivo multimedia, puede reducir el tamaño de la muestra de 16 a 8 bits. Con esto el tamaño del archivo se reduce a la mitad, pero también "empeora" la calidad del audio.

El oído humano no es capaz de percibir la diferencia entre tamaños de muestra mayores a 16 bits, sin embargo existen dispositivos y convertidores de estos valores.

1.3 Amplificador no Inversor

El amplificador es la primera parte del circuito, los transductores manejan amplitudes alrededor de $200mV$, por lo que requerimos incrementar la amplitud a un valor cercano a $2.5v$ sin excederlo la otra función del amplificador es proteger la fuente del audio, (lo que sea que genere el audio, micrófono, teléfono, pc, radio, etc...) del resto del circuito, la señal amplificada requerirá toda la corriente del amplificador, así que cualquier carga puesta después en el acto no será "sentida" por la fuente, utilizaremos nuestros OP AMPS en configuración no inversora, adicionalmente la alta impedancia de nuestro AmpOp protegerá al dispositivo emisor de cualquier falla en nuestro sistema que pueda llegar a presentarse

Alimentaremos nuestros OP AMPS con $\pm 10v$ debido a que la señal no será amplificada nunca más allá de $2.5v$

La siguiente ecuación describe la relación entre amplitudes antes y después del amplificador no inversor

$$V_{out} = V_{in} * \left(1 + \frac{R_2}{R_1}\right) \quad (3)$$

Donde $R2$ es la resistencia de retroalimentación, entre la salida y la terminal no inversora, $R1$ es la resistencia a tierra, V_{out} es la amplitud de la señal de salida, (de la salida del amplificador) y V_{in} es la amplitud de la señal de entrada (La entrada del amplificador).

En nuestro caso $R2$ será 100k y $R1$ un potenciómetro de 10k, girando el potenciómetro, podremos pasar de 0 a 10k.

Cuando el Potenciómetro está proporcionando su resistencia máxima de 10k la relación de entrada salida será:

$$V_{out} = V_{in} * \left(1 + \frac{100k\Omega}{10k\Omega}\right) = 11 \quad (4)$$

la señal que viene del micrófono con una amplitud de 200mv (que es bastante fuerte para un micrófono) será amplificada a:

$$200mv * 11 = 2200mV = 2.2v \quad (5)$$

Este es el rango adecuado que queremos (Amplitud cerca de 2.5v sin pasarnos)

Si giramos el potenciómetro hasta la mitad obtendremos una resistencia de 5kOhms en donde podemos calcular nuevamente la relación de entrada y salida

$$V_{out} = V_{in} * \left(1 + \frac{100k\Omega}{5k\Omega}\right) = 21 \quad (6)$$

La amplitud se multiplicara por 21, esto es demasiado para la señal de 200mV

$$200mv * 21 = 4200mV = 4.2v \gg 2.2v \quad (7)$$

Sin embargo esta amplificación sería perfecta para una señal de 100mv

$$100mv * 21 = 2100mV = 2.1v \quad (8)$$

Girar el potenciómetro más allá mantendrá decreciendo la resistencia de $R1$ e incrementará la amplificación también llamada ganancia, de este amplificador teóricamente hasta el infinito, ajustando el potenciómetro podemos ajustar la ganancia del amplificador para acomodarla a la sensibilidad del micrófono mientras mantenemos un rango adecuado para que ADC del microcontrolador pueda manejar

1.3.1 Capacitor Bypass

El capacitor bypass es un capacitor electrolítico conectado directamente a la fuente de voltaje y a tierra, esto permite que nuestra componente de corriente alterna sea filtrada, el capacitor actúa como una reserva de corriente y ayuda a rellenar cualquier corte que llegue a haber en nuestro voltaje V_{cc} liberándolo cuando el voltaje llegue a caer, el tamaño del capacitor determina que tan grande puede ser ese "hueco" que

se debe rellenar, entra más grande el valor, mayor es el hueco que puede rellenar, un tamaño común utilizado es .1uF, sin embargo el valor preciso del capacitor no es importante, sin embargo altas frecuencias requieren capacitores de menores valores, en nuestro diseño utilizaremos un capacitor de 47nF

1.4 Filtro de CD

A la salida de nuestro primer OPAMP utilizaremos un capacitor electrolítico de 10uF a 12 o 15v en serie, esto para filtrar la componente de CD que pueda generar ruido en nuestro sistema y dejar una señal senoidal lo más fina posible, la parte positiva del capacitor va a la salida de nuestro OPAMP y la negativa al resto del circuito

1.5 Offset de CD

Los micro controladores solo pueden medir voltajes entre 0 y 5v, si medimos los voltajes negativos el micro controlador este solo leerá 0v y terminaríamos cortando el semiciclo negativo de la señal, sin embargo, es posible amplificar y aplicar un offset a esta señal de audio para que esté colocadas entre este rango, idealmente la colocaremos en 2.5v para que su voltaje mínimo sea 0v y su máximo 5v

Con la intención de que el micro controlador sea capaz de leer los voltajes de 0 a 5v es necesario desplazar nuestra señal del origen a un voltaje promedio de 2.5v, esto le permite al microcontrolador procesar los valores de 0 a 256 y de esta manera el voltaje de CD sobre el cual estará montada la señal senoidal, será 2.5V lo significa que la señal estará desplazada hacia arriba 2.5v

$$V_{max} = V_{central} + Amplitud = 2.5v + 2.5v = 5v \quad (9)$$

$$V_{min} = V_{central} - Amplitud = -2.5v + 2.5v = 0v \quad (10)$$

La siguiente señal está amplificada sin embargo aún se encuentra acoplada en 0v oscilando entre -2.5v y 2.5v por lo que para corregir esto tendremos que aplicar un Offset de CD, que significa cambiar el voltaje promedio de la onda con su voltaje central en 2.5v en lugar de 0v, es decir la señal senoidal oscilará alrededor de 2.5v y no de 0v, esta señal está preparada para ser recibida por el puerto analógico de nuestro micro controlador

Para lograr esto el circuito tendrá 2 componentes principales, un divisor de voltaje y un capacitor bypass, para el divisor de voltaje la fórmula es:

$$V_{out} = \frac{R1}{R1 + R2} * V_{fuente} \quad (11)$$

se compone de 2 resistencias de 100kOhms serie conectadas a una fuente de alimentación externa de 5v, que originalmente alimenta al microcontrolador, Con la intención de que el micro controlador sea capaz de leer los voltajes de 0 a 5v es necesario desplazar nuestra señal del origen a un voltaje promedio de

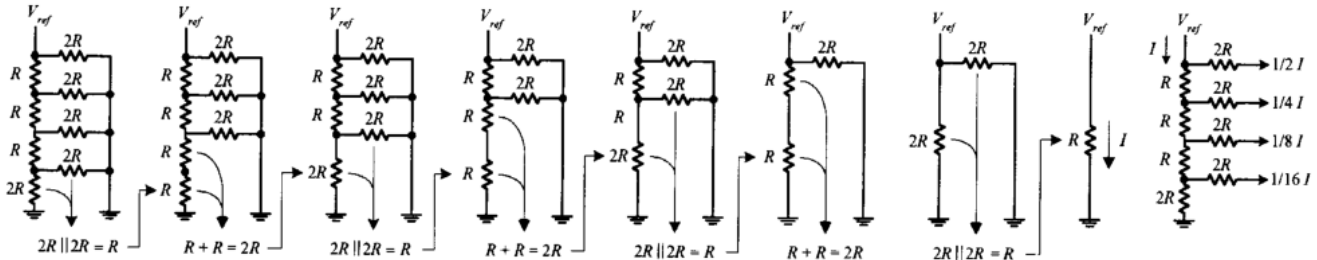


Figure 3. Arreglo Digital to Analog Converter R/2R

2.5v, esto le permite al microcontrolador procesar los valores de 0 a 256 y de esta manera el voltaje de CD sobre el cual estará montada la señal senoidal,

El capacitor se colocara en paralelo hacia tierra m también se le conoce como capacitor bypass y permite filtrar a tierra el ruido de Corriente alterna inherente de cualquier circuito será de 47nF

$$V_{out} = \frac{100k\Omega}{100k\Omega + 100k\Omega} * 5v = 2.5v \quad (12)$$

1.6 Convertidor Analógico-Digital

Un conversor, (o convertidor) de señal analógica a digital, o ADC del inglés ("Analog-to-Digital Converter") es un dispositivo electrónico capaz de convertir una señal analógica de voltaje en una señal digital con un valor binario. Se utiliza en equipos electrónicos como computadora, grabadores de sonido y de vídeo, y equipos de telecomunicaciones. La señal analógica, que varía de forma continua en el tiempo, se conecta a la entrada del dispositivo y se somete a un muestreo a una velocidad fija, obteniéndose así una señal digital a la salida

Nuestro microcontrolador ATMEGA328P tiene 3 puertos dedicados con ADCs de una resolución de 10 bits, es decir hasta 1024 valores, la frecuencia de muestreo por default de este microcontrolador es de 8kHz, pero es posible modificar este valor utilizando los comandos proporcionados en la hoja de datos del fabricante

1.7 Arreglo R/2R DAC

Debido a que nuestro microcontrolador no tiene las capacidades de salida analógicas tenemos que usar un DAC (Digital to Analog Converter) para convertir nuestros datos digitales (Bytes/Números/Enteros) en una forma de onda Analógica (tensión oscilante) una forma económica y sencilla de hacer esto es utilizar una escalera R/2R, Esencialmente toma la salida digital de 0v a 5v los pesa y los suma para producir un voltaje entre 0v y 5v se puede pensar de este arreglo como un divisor de voltaje de múltiples niveles

En este caso particular utilizaremos un ADC R2R de 8 bit, esto significa que puede producir 2^8 (256) diferentes voltajes entre 0v y 5v, y los conectaremos a los pines 0-7 de un puerto en nuestro microcontrolador

Un arreglo de este tipo utiliza una escalera de resistencias utilizando solo 2 valores R y $2R$ la figura muestra un arreglo simple de 4 bits , por ahora asumiremos que switches lógicos controlaran las entradas, la corriente siempre es la misma no importa si se activa un switch o varios , la corriente fluye a tierra si el switch se apaga , sin embargo si este prende fluirá hacia una tierra virtual de la terminal no inversora de un OPAMP convertirá la señal no inversora en 0v vía retroalimentación negativa

1.8 Filtro Pasa Bajas

El propósito del filtro pasa bajas es suavizar la salida del DAC con la intención de reducir el ruido, usando un filtro se pueden reducir los pasos, de nuestra señal digital, , para lograr esto se requiere una resistencia y capacitor en serie hacia tierra la señal que pase entre estos dos será filtrada , para calcular los valores de estos se puede utilizar la siguiente formula

$$F_c = \frac{1}{2 * \pi * R * C} \quad (13)$$

El teorema de Nyquist establece que para una señal con una sampling de x Hz la frecuencia más alta producida debe ser 2 veces ese valor, debemos establecer nuestra frecuencia de corte en $2*B$

Si tuviésemos una frecuencia de corte de 40khz (standard para audio musical) entonces la frecuencia máxima que podemos reproducir es de 20khz (el límite superior del espectro audible), y la frecuencia de corte del filtro pasa bajas debe estar alrededor de los 20khz

Principalmente estaremos digitalizando voz, por lo que para evitar el ruido en el sistema, debemos limitar nuestro sistema para poder reducir los tonos agudos por lo que calcularemos nuestro filtro a poco más de 8khz

Para una frecuencia de corte de 8000hz y una resistencia de 1.5kohms

$$8000Hz = \frac{1}{2 * \pi * R * 0.01\mu F} \quad (14)$$

$$R = 2k\Omega \quad (15)$$

$$F_c = \frac{1}{2 * \pi * 2kOhm * 0.01uF} = 7957.75Hz \quad (16)$$

Estos valores nos ayudaran a elegir los componentes adecuados para filtrar nuestra señal y remover ruido no deseado

1.9 Seguidor de Tensión

Usualmente utilizamos los AMPOPS para amplificar voltaje sin embargo es posible utilizarlos para incrementar la corriente de la señal para que pueda manejar una carga (Altavoz) en esta etapa del circuito acoplamos nuestros LM741s en paralelo, en la configuración de seguidores de voltaje

es decir la salida del potenciómetro a la terminal no inversora de ambos OPAMPs, entonces se conectan como seguidores de voltaje y sus salidas se unen, cada OPAMP puede manejar 20mA, combinados pueden manejar hasta 40mA

1.10 Un-Offset de CD

Antes de enviar nuestra señal a nuestro altavoz, tenemos que asegurarnos de que este oscilando alrededor de 0v, hasta este momento la señal que hemos estado manipulando se encuentra oscilando alrededor de 2.5v, para arreglar esto tenemos que usar un capacitor grande, como indicamos en el esquemático, utilizamos uno de 220uF para bajar esa señal de su posición actual y que pueda ser así reproducida en nuestro altavoz

2. Diseño

2.1 Esquemático

El diagrama esquemático que se utilizó se encuentra en los anexos del documento para no afectar al formato de este reporte

El diseño de nuestro circuito se realizó en Altium designer 14, utilizando los estándares de la IPC 2221A y reglas de diseño apropiadas de OSHPark.com esto para para en caso poder maquilar el circuito impreso por medio de los archivos GERBER; los componentes se adquirieron por medio de Newark element14 debido a la facilidad de seleccionar los componentes y sus datasheets en línea

Los archivos Extended GERBER, o RS-274X generados fueron los siguientes

1. .DRL Archivo para taladrar
2. .GBL Bottom Layer
3. .GBS Bottom Soldermask
4. .GKO Board Outline
5. .GTL Top Layer
6. .GTO Top Silkscreen
7. .GTS Top Soldermask

para la generación de la placa de Circuito Impreso (PCB) se enviaron todos los archivos juntos en un archivo comprimido ZIP por medio de internet a OSH Park en estados unidos en donde se procesaron con las siguientes características

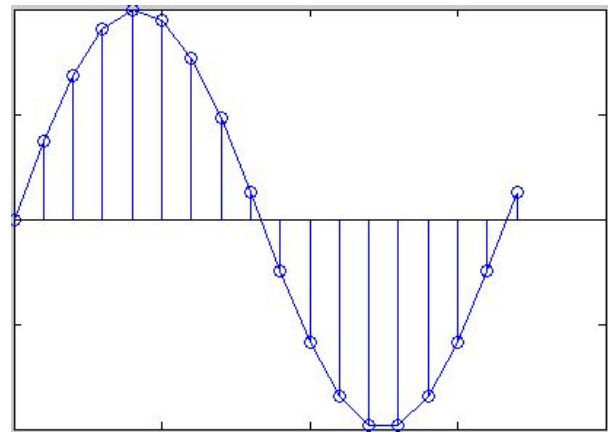


Figure 4. Señal Analógica o continua

2.2 Simulación en MATLAB

Para verificar el proceso de muestreo y Cuantización es necesario simular el proceso a través de MATLAB, este software nos permite simular, una señal continua (analógica) y su conversión a señal discreta (digital) por medio del proceso de la teoría de Cuantización, también nos permite utilizar el ordenador para generar una muestra de sonido de ambas, la señal continua y la señal discreta (digital)

Podemos ver en qué consiste el proceso de muestreo o cuantificación de la onda, donde a partir del modelo original intentamos reconstruirla mediante la generación de puntos sobre su trazo. En la imagen inferior se aprecia cómo la fidelidad de la reconstrucción depende también del número de bits utilizado (bitrate).

2.2.1 Simulación de señal continua

A continuación se muestra el código requerido para poder graficar una señal analógica utilizando una frecuencia de muestreo de 8000kHz, es posible cambiar ambos, la frecuencia, y la frecuencia de muestreo, la función `plot` graficará nuestra gráfica en donde podremos ver y escuchar vía el altavoz de la computadora el sonido analógico esta es una simulación de la señal de entrada (sin offset) que entrara en nuestro microcontrolador

```
clf
t=0:1:16.66
y=sin(2*pi*0.06*t)
plot(t,y)
hold on
plot(t,y,'ko')
hold off
n=[0:1:10000]
ftono=440
fm=8000
Tm=1/fm
A=1
y=A*sin(2*pi*ftono*Tm*n)
sound(y,fm)
xlabel('Tiempo [ms]')
ylabel('Tension RMS [v]')
```

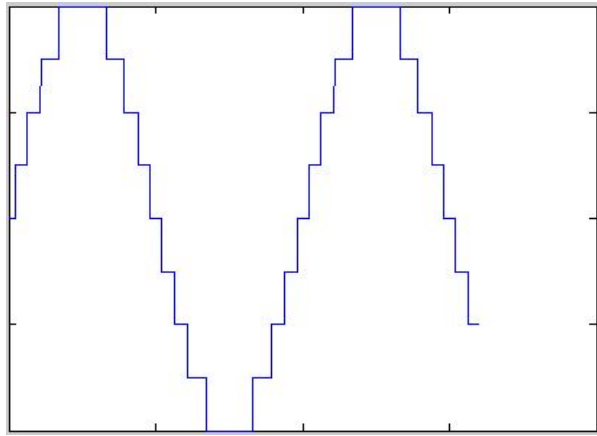



Figure 5. Señal digital o discreta

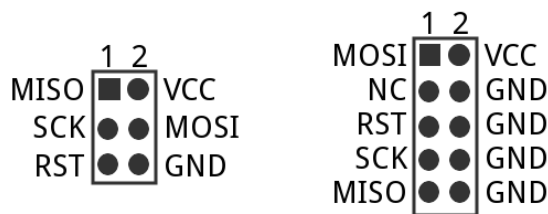


Figure 6. Diagrama de conexión para conector ISP

title('Simulación de conversión Analógica Digital')

2.2.2 Simulación de señal discreta

El siguiente código nos permite modificar el número de muestras, al mismo tiempo de graficar nuestra señal en el tiempo en forma discreta, este código también nos permite reproducir el sonido que genera esta señal digital en el altavoz de nuestro computador, es posible variar N de 1 a 16 bits, siendo 16 bits la mejor calidad de audio que el oído humano puede percibir

```
clf
t = 0:1:1600
fm = 1000
Tm=1/fm
ftono = 440
N = 2^2
senal = sin(2*pi*t*ftono*Tm)
y = round(senal*N)/N
plot(round(sin(2*pi*t/1000)*N)/N)
sound(round(sin(2*pi*t*ftono*Tm)*N)/N, 1000)
```

2.3 Lista de Materiales

Los componentes se adquirieron por medio de newark element14 esto debido a la escasez de componentes especializados como las resistencias de precisión de 10k 20k y 3k, así como por conveniencia para cumplir los plazos del proyecto, los costos de los componentes fueron los siguientes

2.4 Programación ISP

Programación en el sistema (In-system programming) o ISP, por sus siglas en inglés, es la habilidad de algunos dispositivos

Table 1. Tabla de componentes

Cant.	Componente	Costo(USD)
1	Resistencia 10	0.11
1	Resistencia 8.2k	0.11
7	Resistencia 10k	0.77
9	Resistencia 20k	0.99
3	Resistencia 100k	0.33
1	Resistencia 3k	0.11
2	Clema para PCB	2.00
1	Cristal Oscilador 12Mhz	1.20
5	OPAMP LM741	4.00
1	ATMega328P	2.82
5	Capacitor 10uF	0.11
2	Capacitor 22nF	0.33
1	Capacitor 0.01uF	0.50
1	Header Male 5x2	1.50
2	Trimpot 50M	1.50
1	Bocina 8Ohms	3.00
1	Push Button NC	0.21
1	Maquilacion PCB	15.00

lógicos programables, microcontroladores y otros circuitos electrónicos de ser programados mientras están instalados en un sistema completo, en lugar de requerir que el chip sea programado antes de ser instalado dentro del sistema. Para programar nuestro microcontrolador es necesario un programador externo, en nuestro caso nos apoyaremos del arduino dado a su facilidad en costo y accesibilidad, sin embargo debemos diseñar nuestra tarjeta para poder utilizar pines de programación directamente en el circuito impreso, estos nos permitirán reprogramar el microcontrolador directamente conectado en el PCB, el diseño de nuestro PCB tiene contemplado el ruteo de los pines VCC, MISO, MOSI, SCK, CC y GND, para cumplir este propósito, así como los componentes requeridos para que el microcontrolador pueda funcionar de manera autónoma en nuestro PCB

2.5 Códigos de programación en C

En los 4 siguientes códigos de programación, se demuestran diversos métodos para leer la señal de voz de entrada en el puerto A0 como un numero entre 0 y 1023 lo almacenan como la variable *IncomingAudio*, desde aquí, es posible almacenar este valor para un uso posterior, así como realizar operaciones matemáticas con el mismo

Se utilizó un DAC de 8 bit para poder visualizarlos puntos de información que el microcontrolador almacena como la variable *IncomingAudio* y poder compararla que tan cerca está de la señal original, podemos calcular la tasa de muestreo con la siguiente formula

Es posible utilizar esta información para almacenar el valor para un uso posterior o incluso para realizar operaciones matemáticas con él, o hacer cualquier otro tipo de manipu-

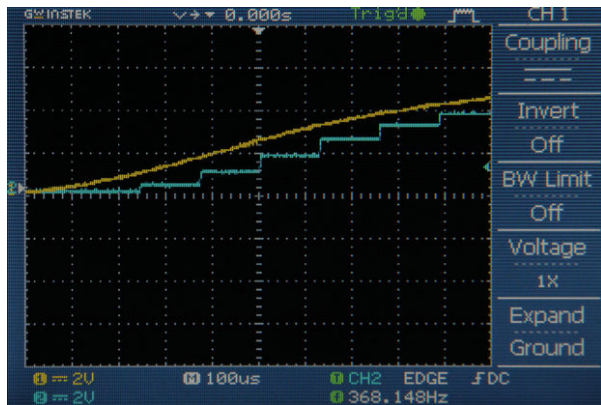


Figure 7. Salida digital muestreada cada 125µs

lación con el

$$T_{\text{muestreo}} = \frac{1}{125\mu\text{s}} = \frac{1}{0.000125\text{s}} = 8000\text{Hz} \quad (17)$$

Como se puede ver en la figura, el microcontrolador toma una muestra cada 125µs de A0

2.5.1 Muestreo a 8kHz en puerto A0

Por default el comando *AnalogRead* solo nos proporciona una tasa de muestreo de 8kHz, más que suficiente para muestrear señales de voz Digital para audio telefónico, para poder lograr tasas de muestreo más elevadas es necesario saltarnos la función *AnalogRead*, y utilizar la hoja de datos para modificar la tasa de muestreo del ADC int incomingAudio;

```
void setup()
for (byte i=0;i<8;i++) pinMode(i,OUTPUT);
void loop()
incomingAudio = analogRead(A0);
incomingAudio = (incomingAudio+1)/4 - 1;
if (incomingAudio<0)
incomingAudio = 0;
```

2.5.2 Muestreo a 8kHz en A0 con salida a DAC de 8 bits

El código anterior permite al microcontrolador leer los valores a 8kHz sin embargo solo hemos leído la señal no hemos redirigido a ningún puerto la señal digital, para poder visualizarla en el osciloscopio es necesario un DAC R/2R de 8 bit para poder convertir la señal en algo que se parezca a una señal Analógica, este código permite redirigir la señal del microcontrolador hacia el puerto B en donde se utilizaran sus 7 canales para conectar nuestro DAC

```
int incomingAudio;
void setup()
for (byte i=0;i<8;i++)
pinMode(i,OUTPUT);
void loop()
incomingAudio = analogRead(A0);
incomingAudio = (incomingAudio+1)/4 - 1;
if (incomingAudio<0)
incomingAudio = 0;
PORTD = incomingAudio;
```

2.5.3 Muestreo a 38.5kHz por el puerto A0

Es posible ordenar al microcontrolador omitir las otras entradas analógicas con el propósito de obligar una medición constante y en tiempo real del pin A0, de esta manera mientras el microcontrolador entra en un loop infinito, el microcontrolador se mantendrá actualizando la variable llamada *ADCH* con nuevos valores de A0 a un ritmo de 38.5kHz (Una muestra cada 26µs), para esto es necesario reducir la resolución de las medidas analógicas un poco, para obtener una mayor tasa de muestreo

Utilizando la hoja de datos, del microcontrolador podemos modificar el convertidor analógico digital de la entrada Analógica A0, en el *ADCH* directamente (En el primer código solo leímos los 8 bits más significativos, del ADC de 10 bits, para trabajar de la manera más rápida posible y ahorrar tiempo a la hora de programar) podemos colocar el contador del ADC a 500kHz debido a que el ADC toma 13 ciclos en leer el nuevo valor analógico

$$T_{\text{muestreo}} = \frac{500\text{kHz}}{13} = 38.5\text{kHz} \quad (18)$$

Valor que nos acerca a los 40kHz (Valor estándar de muestreo musical) sin inducir ninguna clase de ruido adicional, esto permite al microcontrolador tomar muestras cada

$$T_{\text{muestreo}} = \frac{13}{500000\text{Hz}} = 26\mu\text{s} \quad (19)$$

El código para lograr esta velocidad de lectura en el puerto A0 es el siguiente

```
int incomingAudio;
void setup()
ADCSRA = 0;
ADCSRB = 0;
ADMUX = (1 << REFS0);
ADMUX = (1 << ADLAR);
ADCSRA = (1 << ADPS2) | (1 << ADPS0);
ADCSRA = (1 << ADSC);
ADCSRA = (1 << ADIF);
ADCSRA = (1 << ADIF);
ADCSRA = (1 << ADIF);
void loop()
incomingAudio = ADCH;
```

2.5.4 Muestreo a 38.5kHz en puerto A0 con salida a DAC de 8 bits

```
void setup()
for (byte i=0;i<8;i++)
pinMode(i,OUTPUT);
ADCSRA = 0;
ADCSRB = 0;
ADMUX = (1 << REFS0);
ADMUX = (1 << ADLAR);
ADCSRA = (1 << ADPS2) | (1 << ADPS0);
ADCSRA = (1 << ADSC);
ADCSRA = (1 << ADIF);
ADCSRA = (1 << ADIF);
ADCSRA = (1 << ADIF);
```

```
ADCSRA —= (1 << ADSC);
void loop()
PORTD = ADCH;
```

2.6 Circuito Impreso

Inicialmente el circuito se armó en protoboard para verificar su funcionamiento, al verificar su correcto funcionamiento se realizó el esquemático en Altium Designer para poder generar una versión imprimible del mismo la cual se pasó a PCB y nuevamente se probó, al verificar su funcionamiento y corregir la posición de los componentes, se generaron los archivos GERBER requeridos para su maquilación, esto nos permitió enviar a maquilar nuestro circuito

3. Conclusiones y Observaciones

Este arreglo no da la mejor calidad de audio para música, sin embargo es más que suficiente para voz, es posible utilizar Circuitos integrados ADC de 16bit y 24bit con estos es posible obtener una mejor calidad de audio, algunos modelos específicos incluso pueden proporcionar salidas estéreo sin embargo habría que recurrir a Circuitos integrados más avanzados y especializados

Se utilizaron LM741 debido a que se tienen de sobra de todas las prácticas a lo largo de la carrera, se pudo utilizar empaquetados más pequeños que contienen 4, LM741 integrados sin embargo se optó por utilizar los componentes existentes

La calidad de sonido de salida es baja debido al proceso de conversión del ADC de 8 bits de resistencias, es probable incrementar la calidad del sonido utilizando un ADC de 16 bits sin embargo se tendría que encontrar un microcontrolador capaz de manejar 16 puertos de salida simultáneamente es de un audio analógico de 8 bits, por lo que canciones con una mejor calidad de audio

El led azul indicador SMD de 10v se quemó durante las pruebas iniciales, esto se debió al cálculo de la resistencia limitadora de corriente fue "demasiado exacto" por lo que la tolerancia paso a jugar en nuestra contra, el led funcionó aproximadamente 30 minutos antes de quemarse, se estima que la corriente que paso por el fue de 21mA cuando la corriente máxima soportada por el mismo era de 20mA, este led ya fue reemplazado en la tarjeta final

Se tuvieron distintos problemas de distintos tipos de esta magnitud debido a que se tuvo que llevar un proceso de comprobación de teoría, verificación en laboratorio, repetidas veces, incluso se tuvo que descartar un primer circuito de una sola cara, que se llegó a imprimir soldar, estañar y probar de manera casera, esto nos llevó a tener la problemática de tener que reparar distintas pistas que o no se pasaron bien a la placa o se arrancaron en el proceso de estañado, finalmente se comprobó que lo ideal es pasar de la protoboard al PCB siempre y cuando se tenga muy en claro cómo se deberá pasar de protoboard a PCB, así como procurar siempre un diseño modular con puntos de prueba incluidos en el PCB para poder tener una maniobrabilidad incluso después de que el PCB esté terminado

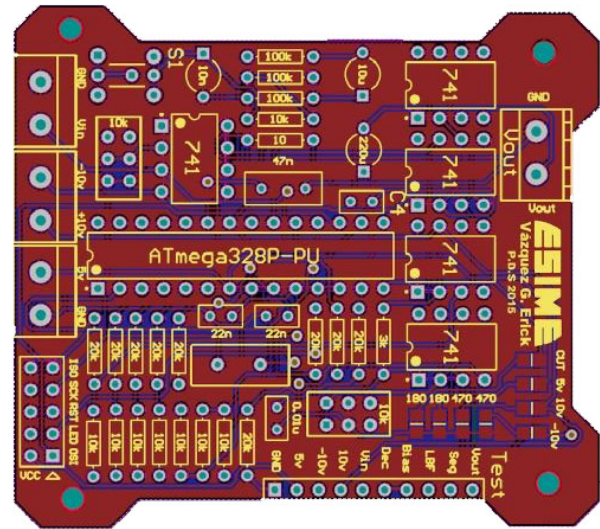


Figure 8. PCB Layout doble capa enviado a Maquilacion

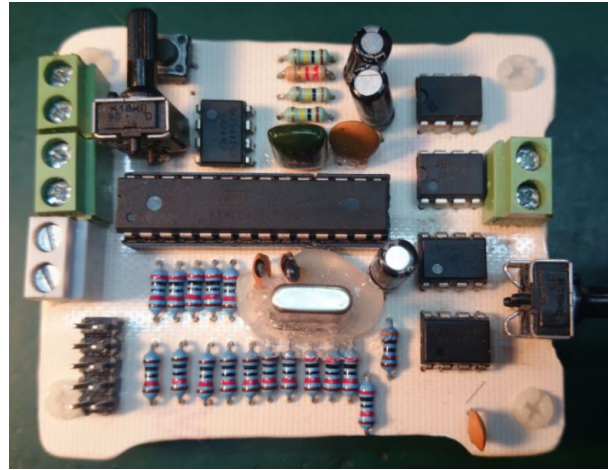


Figure 9. Circuito impreso hecho en casa

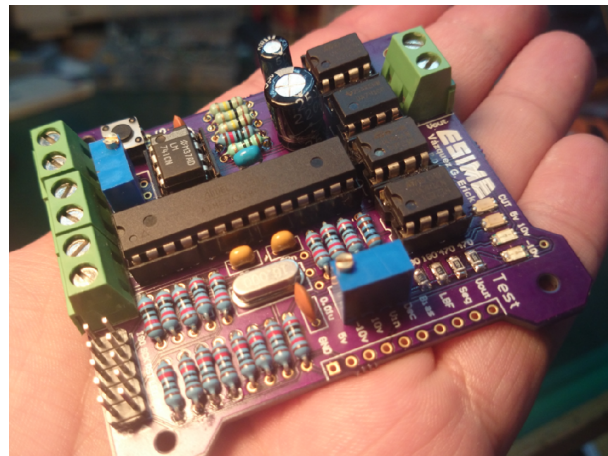


Figure 10. Tarjeta maquilada en OSHPark.com

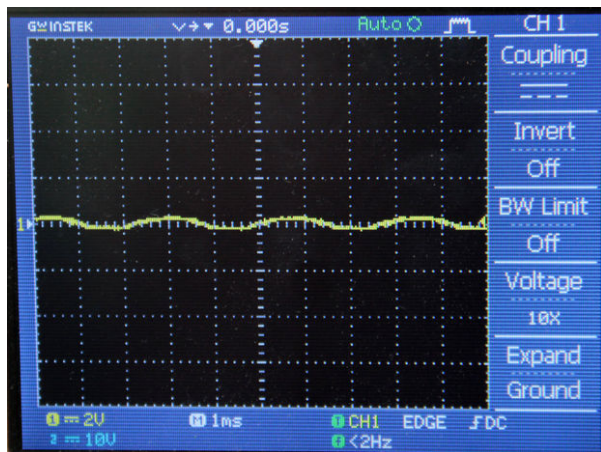


Figure 11. Señal de Salida de un transductor

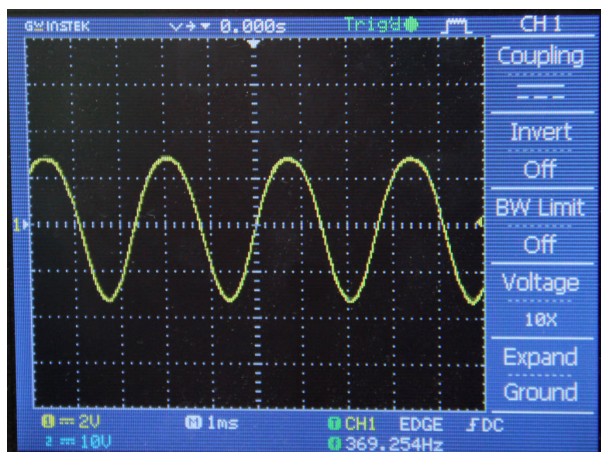


Figure 12. Señal analógica amplificada

Aunque nosotros oímos la música como onda analógica (la misma que hace vibrar nuestro tímpano), cuando ésta llega al cerebro la información es transformada en pulsos eléctricos. Es decir, que nuestro propio cerebro realiza un proceso de “digitalización” del sonido transformándolo en quantos de información que puedan ser transmitidos por la corteza cerebral y otras partes del cerebro interno. Por tanto, en el proceso de escucha de cualquier canción existen tanto partes analógicas como digitales que se van alternando continuamente incluso dentro de nosotros mismos.

4. Bibliografía y referencias

- Practical Electronics for inventors, Feb 2013, Paul Scherz
<http://electronics.stackexchange.com/>
- Analog Sampling Basics
<http://www.ni.com/white-paper/3016/en/>
- Analog Input Subsystem
<http://www.mathworks.com/help/daq/analog-input-subsystem.html>
- PCB Design Tutorial
alternatezone.com/electronics/files/PCBDesignTutorialRevA.pdf
- Basic Circuits - Bypass Capacitors

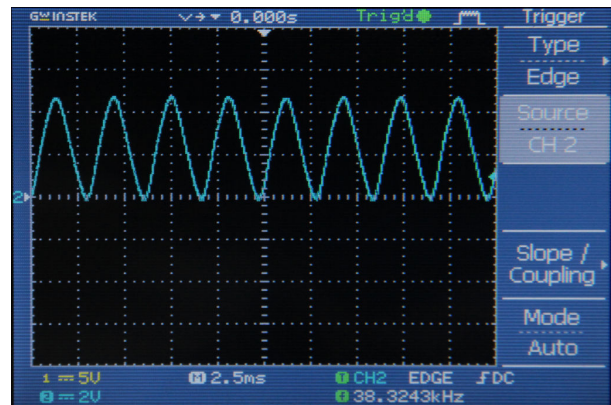


Figure 13. Señal Amplificada con offset de DC

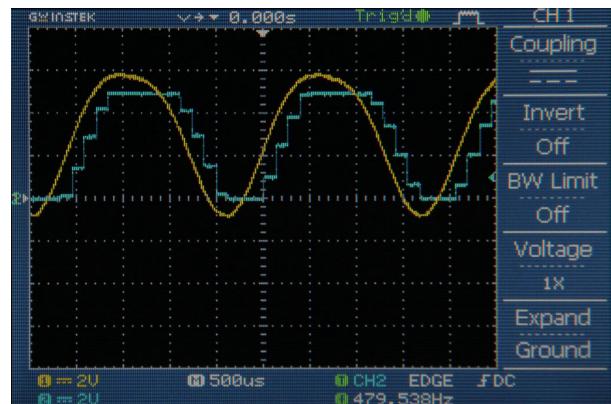


Figure 14. Señal Acondicionada vs señal de salida

<http://www.seattlerobotics.org/encoder/jun97/basics.html>
 -¿Se puede tocar el sonido? Sonido analógico versus digital blogs.heraldo.es/ciencia/?p=2845