



Tecnológico de Monterrey

Proyecto Final: Generador de sonido binaural

Equipo 4

Nicol Gómez Tisnado A00227180

Andrés Antonio Bravo Orozco A01630783

Laboratorio de Procesamiento Digital de Señales

Profesor Fernando Peña Campos

Zapopan, Jalisco a 13 de Junio de 2021

Motivación

El término binaural significa escuchar con dos oídos, en la psicoacústica se refiere a una de las capacidades del oído humano de percibir la direccionalidad y espacialidad del sonido, es decir, de qué dirección vienen los sonidos y que tan grande o pequeño es el espacio en el que nos encontramos. En la actualidad la utilización de la tecnología que utiliza HRTF para simular sonido binaural está en expansión ya que este tipo de sonido brinda una mejor experiencia de audio a los usuarios y les permite ubicar la procedencia de los sonidos. Esto tiene una amplia aplicación en videojuegos o incluso en la mejora de la calidad del audio de plataformas musicales. El proyecto está basado en la recreación del proyecto KEMAR del MIT a partir de la implementación de un código en C cuyo funcionamiento se puso en práctica a través de una tarjeta ESP32 y mediante la simulación del mismo a través de Matlab. En el presente reporte se describirá detalladamente la manera en la que se elaboró el generador de sonido binaural.

Marco Teórico

Función de transferencia relacionada con la cabeza

La función de transferencia relacionada con la cabeza (HRTF) por sus siglas en inglés, es una función que sirve para caracterizar cómo un oído recibe el sonido desde un punto en el espacio. Conforme el sonido impacta en el oyente, las características del oyente transforman el sonido afectando la forma en la que este es percibido al aumentar unas frecuencias y atenuar otras. Actualmente, se utiliza HRTF para los dos oídos sintetizando un sonido binaural que parece provenir de un punto particular en el espacio (XYZ). Es, por lo tanto, una función de transferencia que describe cómo un sonido de un punto específico llegará al oído.

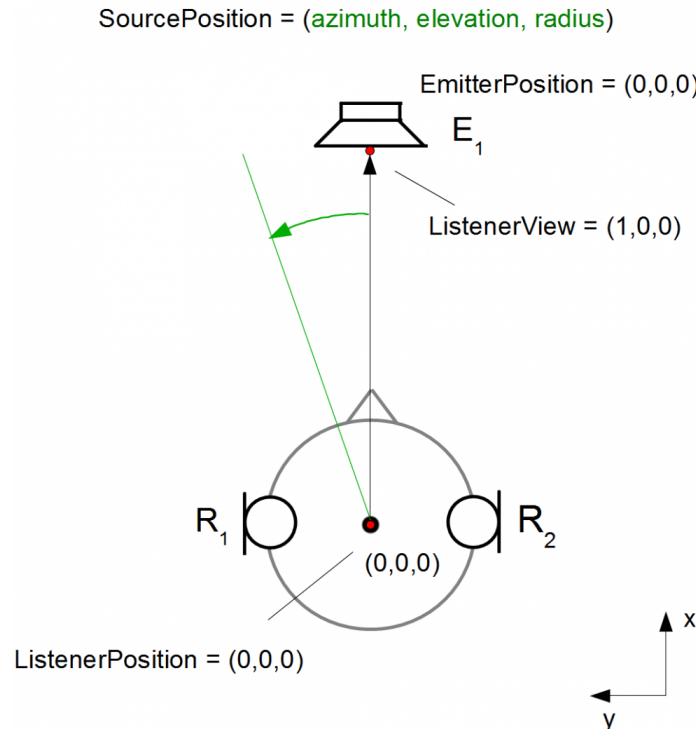


Figura 1. Representación de HRTF

Proyecto KEMAR MIT

El proyecto KEMAR es un proyecto perteneciente al MIT que consiste de un extenso grupo de funciones de transferencia relacionadas con la cabeza obtenido a partir de mediciones a una cabeza simulada de KEMAR la cual contiene micrófonos en la parte interna de los oídos. Las mediciones consisten en la respuesta al impulso del oído izquierdo y el oído derecho desde un altavoz montado a 1.4 metros de distancia de la cabeza. Se utilizaron secuencias binarias pseudoaleatorias para obtener las respuestas al impulso en una frecuencia de muestreo de 44.1 Hz. En total se tomaron 710 muestras distintas a elevaciones que van de los -40° a +90°. Además se midió la respuesta al impulso de la bocina en campo libre y múltiples audífonos ubicados en el KEMAR.

DAC

Un DAC también llamado Digital to Analog Converter, es un conversor capaz de transformar la señal digital de cualquier fuente en una señal analógica. Los DACs convierten las palabras digitales en un valor de tensión analógica capaz de ser procesada por un sistema electrónico. A continuación, el sistema electrónico proporcionará a los correspondientes altavoces, una

señal que determine su actuación dando lugar finalmente al sonido que escuchamos. La calidad de un DAC radica en la resolución, es decir, la cantidad de bits por muestra que se pueden convertir en señales analógicas, su tasa de muestreo y la linealidad.

I2S

I2S es un estándar de interfaz de bus serie que se usa para conectar dispositivos de audio digitales. Un bus de I2S tiene señales de reloj y de datos en serie independientes, así como una parte destinada a la selección de la palabra, siendo 0 el canal izquierdo y 1 el canal derecho. Este protocolo puede utilizarse para comunicar la señal digital con un DAC externo.

Filtro FIR

Un filtro FIR es aquel que tiene una respuesta finita al impulso y se caracteriza por ser un sistema no recursivo. Un filtro FIR de orden N se describe mediante la siguiente ecuación:

$$y(n) = a_0 x(n) + a_1 x(n - 1) + a_2 x(n - 2) + \dots + a_N x(n - N)$$

donde la secuencia a_0 , son los coeficientes del filtro.

En este tipo de filtrado no existe retroalimentación. Además, la respuesta al impulso es de duración finita ya que si la entrada se mantiene en cero durante N periodos consecutivos la salida entonces será cero. La estructura de dicho filtro se puede observar en la figura 1.

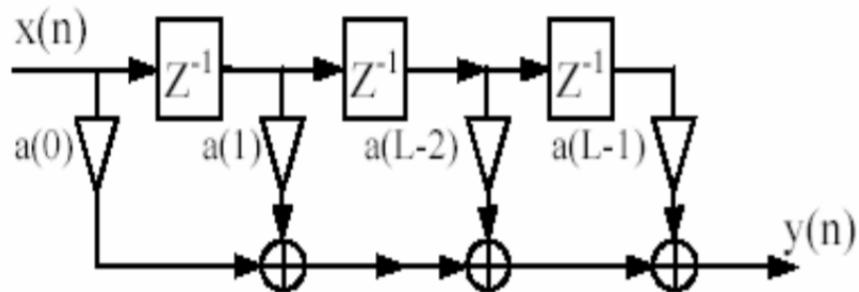


Figura 2. Estructura de un filtro FIR

Este tipo de filtros tienen un alto interés en aplicaciones de audio ya que son se pueden diseñar para ser lineales y son estables debido a la falta de elementos en el denominador de su función de transferencia.

Descripción de la solución e implementación

Diagrama de bloques de funcionamiento

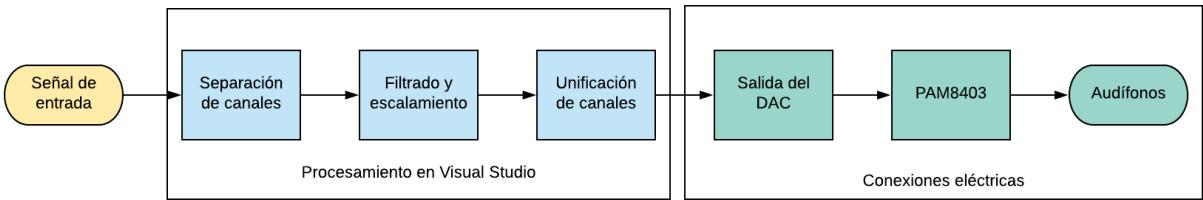


Figura 3. Diagrama de bloques de funcionamiento

El proyecto está conformado por distintas etapas que van desde que la señal de entrada es recibida hasta que es escuchada por el usuario. Para describir el flujo es posible comenzar con el momento en que la señal de entrada es recibida y enviada a través de Bluetooth (utilizando un dispositivo móvil) a la ESP32. La señal es procesada a través de Visual Studio, donde se separa en dos canales para tratarla por separado y poder obtener una salida binaural. Una vez que la señal es procesada, se filtra a través de un filtro FIR el cual amplifica o atenúa la señal según sea el caso para crear el efecto de que el sonido proviene de un punto en particular. Luego de esto la señal es escalada para evitar saturación a la salida del DAC y se realiza una unificación de ambos canales que permite obtener un único arreglo de salida. Para terminar la señal se envía a los 2 canales de salida del DAC, que se encuentra conectada a una tarjeta PAM8403, la cual funciona como un amplificador digital y finalmente se escucha el resultado mediante audífonos que permiten apreciar el sonido binaural.

Diagrama de conexiones eléctricas

Para lograr extraer el sonido binaural del ESP-32 se utilizaron los 2 canales de DAC del sistema los cuales son los pines G25 y G26. Estos fueron conectados a una tarjeta PAM8403 que funciona como amplificador digital. Se conectaron unos audífonos a la salida de esta tarjeta para así poder escuchar el resultado del proyecto y modificar el volumen haciendo uso del PAM.

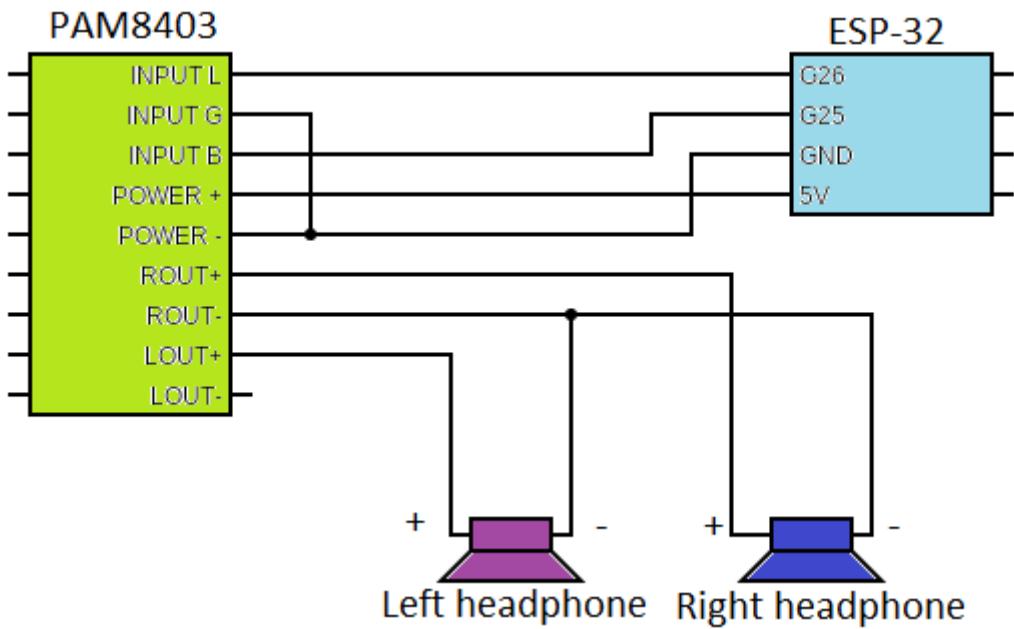


Figura 4. Diagrama de conexiones eléctricas

Diagrama de flujo del software implementado

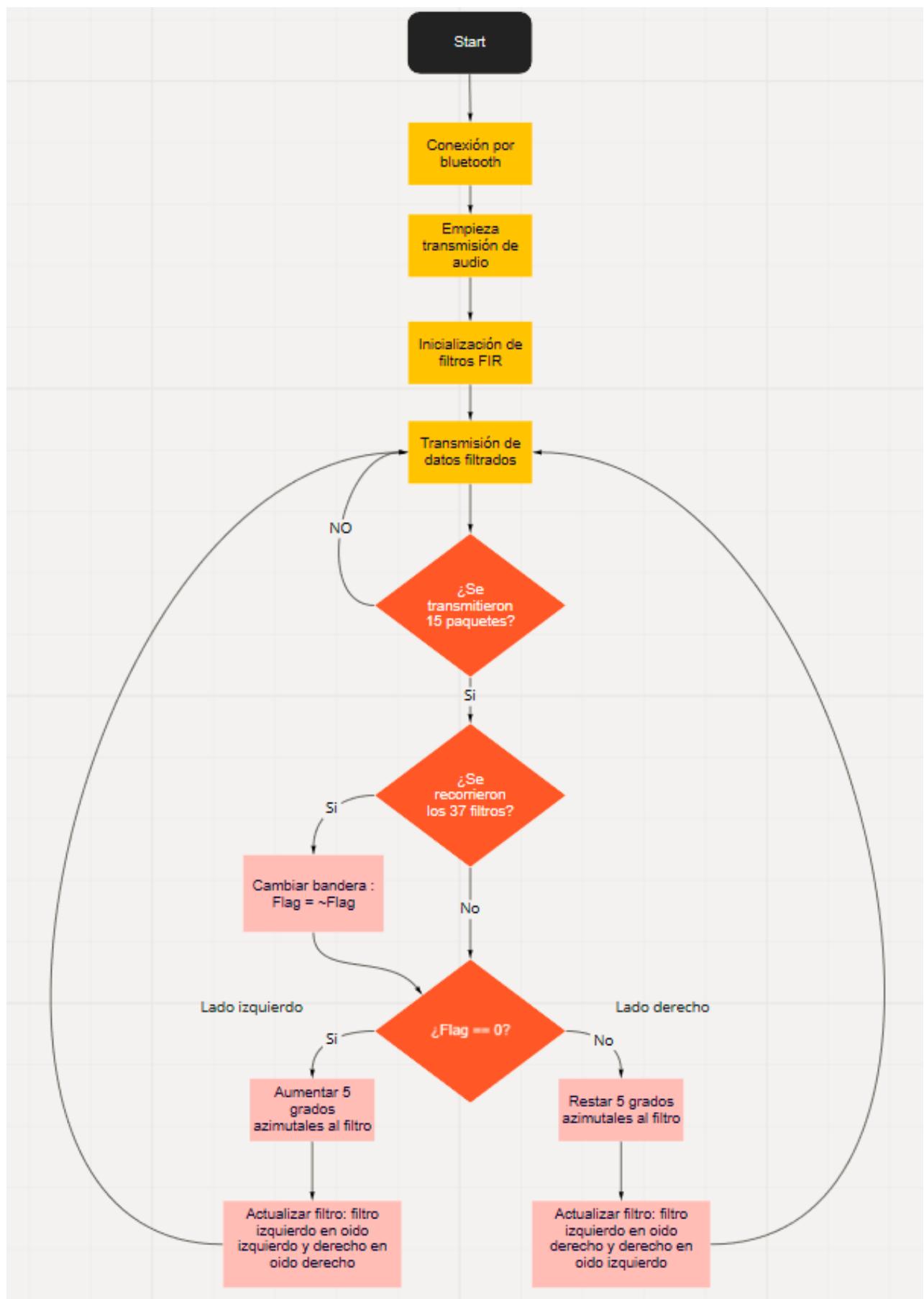


Figura 5. Diagrama de flujo del software

Código fuente de secciones relevantes

```
static void IRAM_ATTR bt_i2s_task_handler(void *arg)
{
    uint8_t *data = NULL;
    int16_t *temp = NULL;
    size_t item_size = 0, i = 0;
    size_t bytes_written = 0;
    uint16_t received_pack=0;
    uint8_t z = 0;
    uint8_t flag = 0;

    //FIR filter initialization

    dspS_fir_init_f32(&myfirL, (float *)&left[z*COEFF_L], statesL,
COEFF_L);

    dspS_fir_init_f32(&myfirR, (float *)&right[z*COEFF_L], statesR,
COEFF_L);

    for (;;) {

        data = (uint8_t *)xRingbufferReceive(s_ringbuf_i2s, &item_size,
(portTickType)portMAX_DELAY);

        if (item_size != 0) {

            received_pack++;

            if ( received_pack % 15 == 0 ){

                //Do something each 15 received items

                z++;

                if(z == 37) {
```

```

z = 1;

if(flag == 0) {

    flag = 1;

} else{

    flag = 0;

}

}

if(!flag) {

    printf("azimuth: %d°\n", z*5);

    myfirL.coeffs = (float *)&left[z*COEFF_L];

    myfirR.coeffs = (float *)&right[z*COEFF_L];

} else{

    printf("azimuth: %d°\n", 180+z*5);

    myfirL.coeffs = (float *)&left[(36-z)*COEFF_L];

    myfirR.coeffs = (float *)&right[(36-z)*COEFF_L];

}

}

//input converted to separated float channels L and R

for(i=0;i<item_size;i+=4){

temp=(int16_t*)(&data[i]);

tempinl[i>>2]=(float)(*(temp))/32768.0;

tempinr[i>>2]=(float)(*(temp+1))/32768.0;

}

}

//FIR Filter to decide in which ear it is going

```

```

    if (!flag) {

dsps_fir_f32_ae32(&myfirL,tempinl,tempoutl,item_size>>2);

dsps_fir_f32_ae32(&myfirR,tempinr,tempoutr,item_size>>2);

} else{

dsps_fir_f32_ae32(&myfirR,tempinl,tempoutl,item_size>>2);

dsps_fir_f32_ae32(&myfirL,tempinr,tempoutr,item_size>>2);

}

//convert [-1,1] float to integer, i=f*32768 + 32768

dsps_mulc_f32_ae32(tempoutl,tempinl,item_size>>2,32768.0,1,1);

dsps_mulc_f32_ae32(tempoutr,tempinr,item_size>>2,32768.0,1,1);

dsps_addc_f32_ae32(tempinl,tempoutl,item_size>>2,32768.0,1,1);

dsps_addc_f32_ae32(tempinr,tempoutr,item_size>>2,32768.0,1,1);

//Output converted to merged interleaved LR output

for(i=0;i<item_size;i+=4){

    temp=(int16_t*) (&data[i]);

    *temp=(int16_t) (tempoutl[i>>2]);

    *(temp+1)=(int16_t) (tempoutr[i>>2]);

}

i2s_write(0, data, item_size, &bytes_written,
portMAX_DELAY);

```

```
    vRingbufferReturnItem(s_ringbuf_i2s, (void *)data);  
}  
  
}  
  
}
```

Fotografías del sistema integrado

El sistema integrado físicamente se muestra en las siguientes figuras.

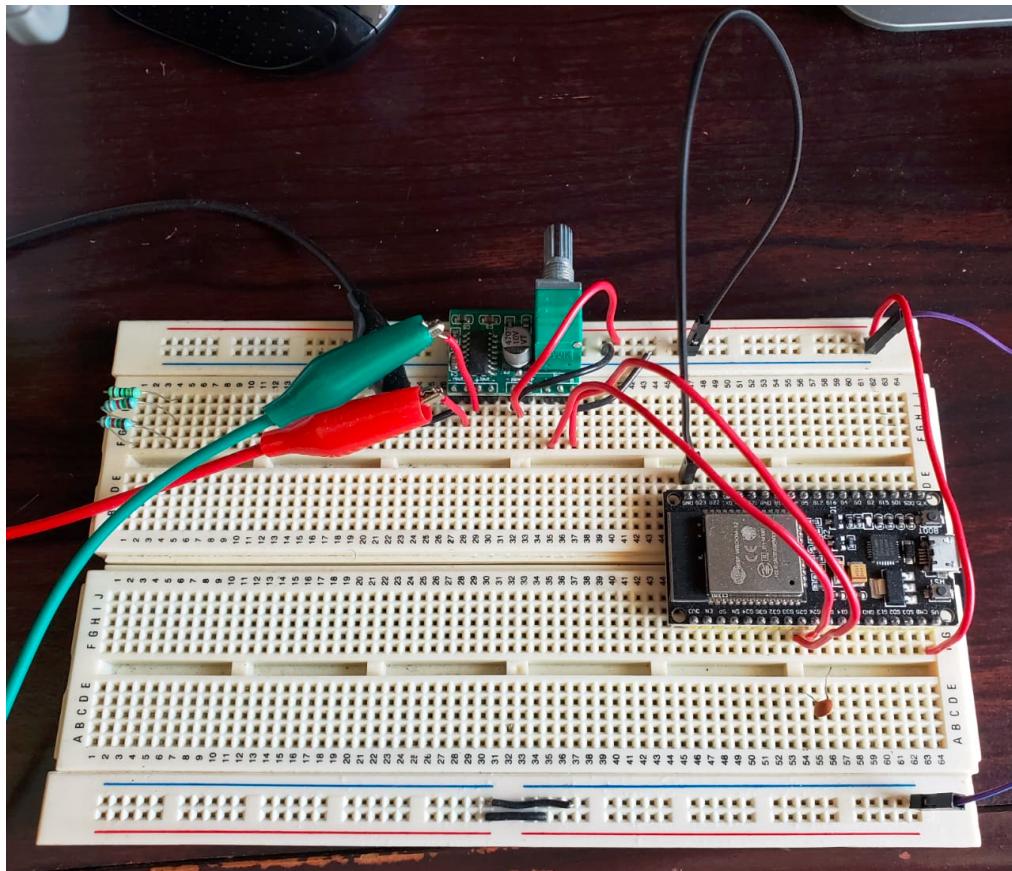


Figura 6. Conexión real del sistema

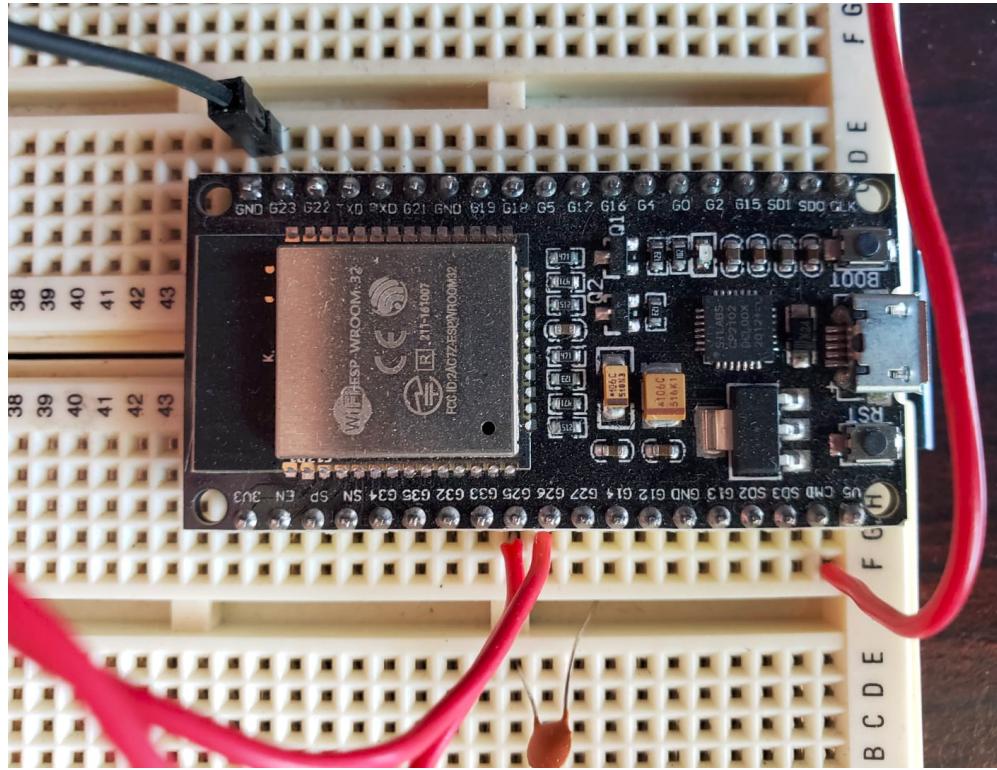


Figura 7. Conexiones del módulo ESP-32

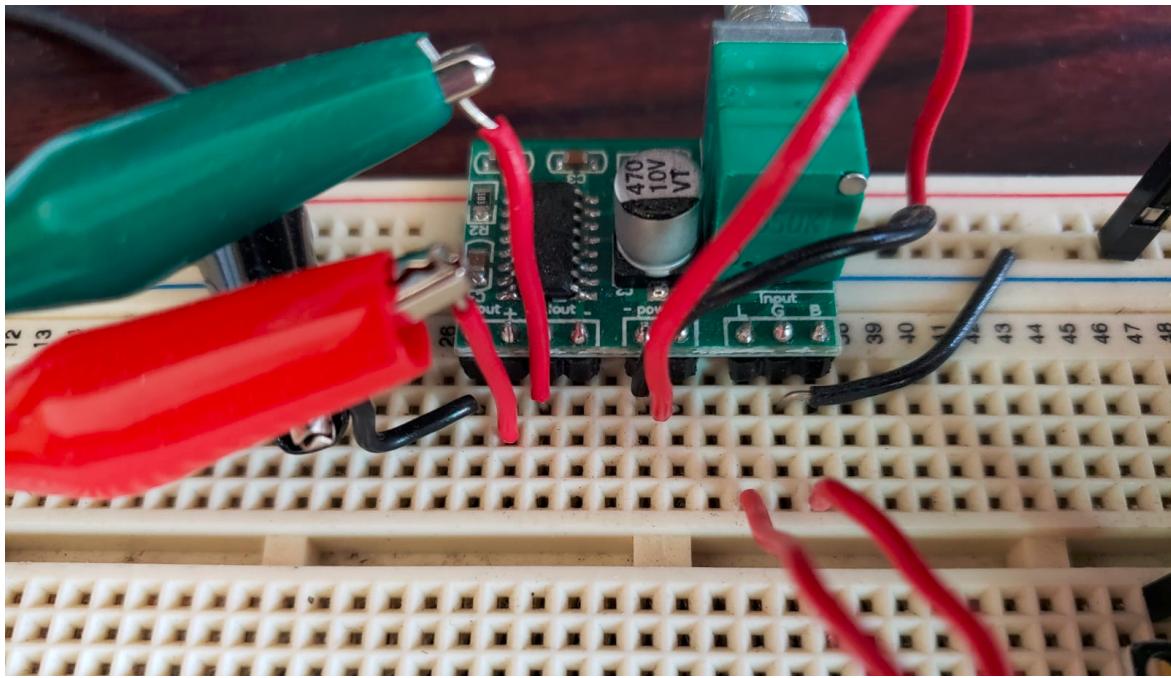


Figura 8. Conexiones a la tarjeta PAM8403

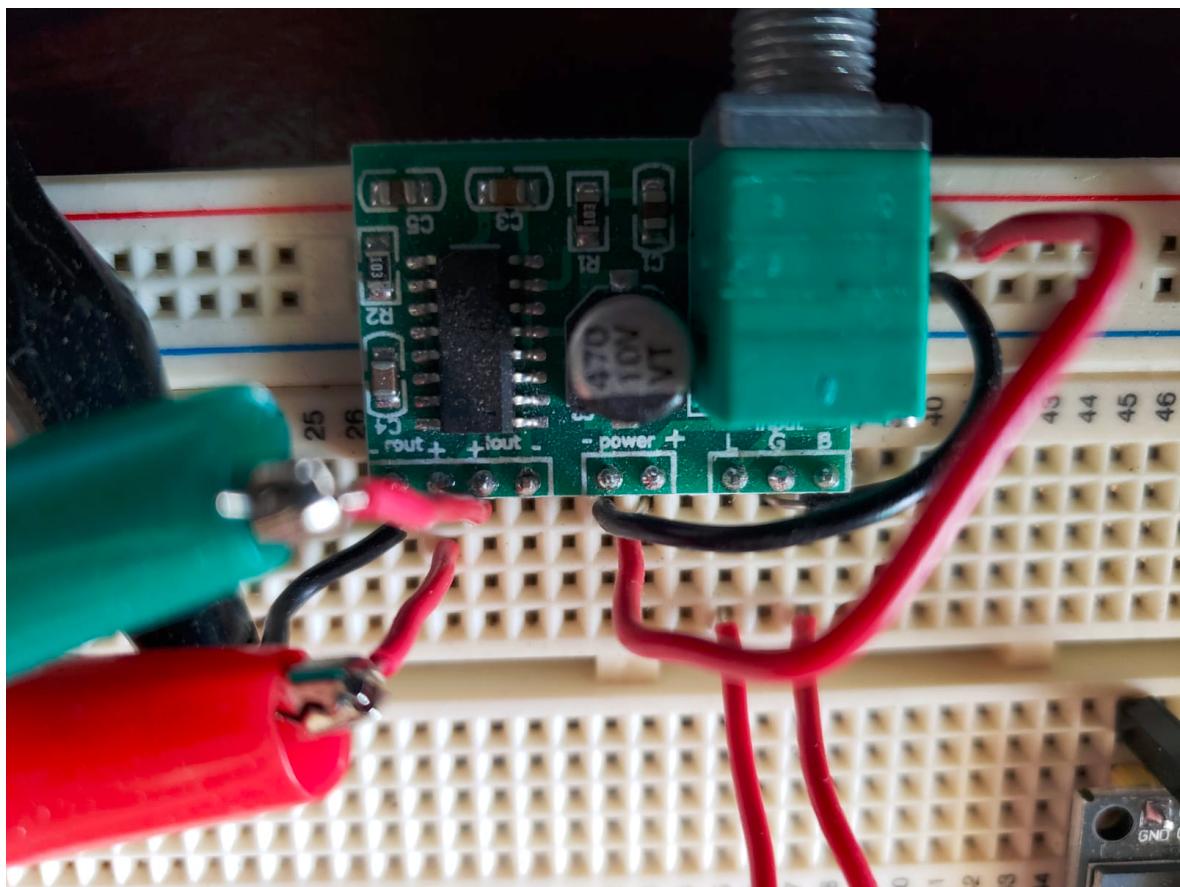


Figura 9. Conexiones de la tarjeta PAM8403

Para conectar el auxiliar a la tarjeta pam se utilizó el siguiente diagrama





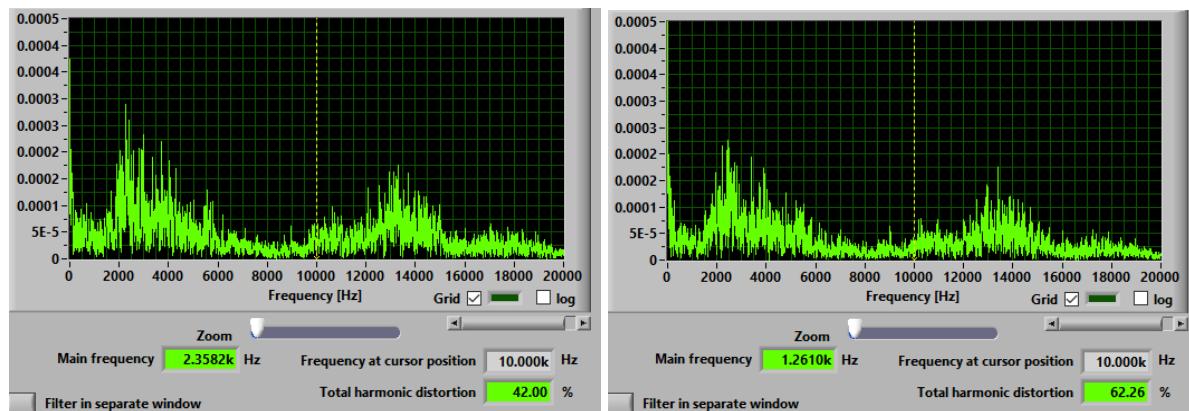
Figura 10. Conexión a auxiliar.

Resultados

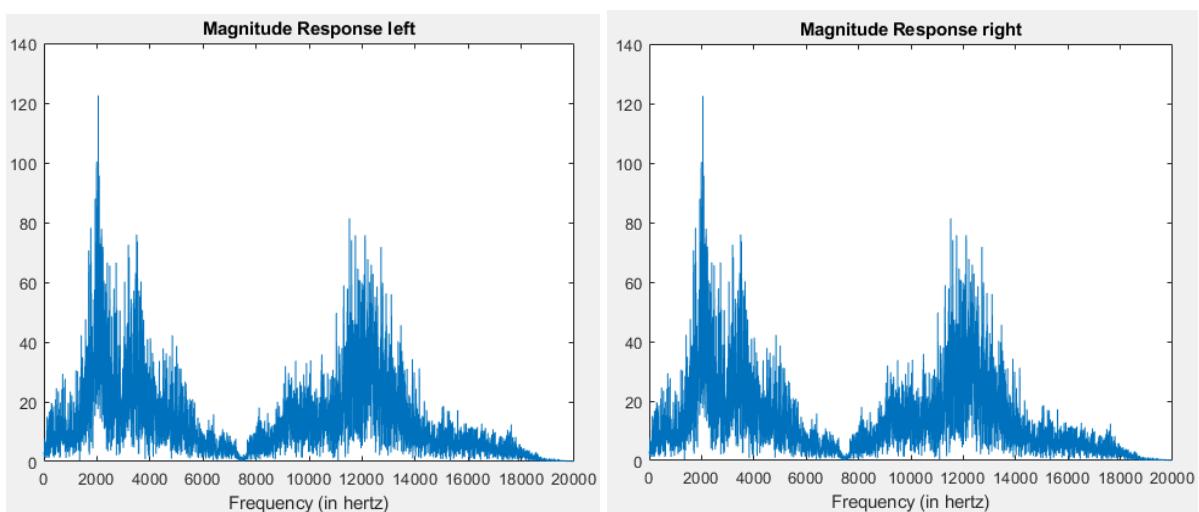
Resultados cuantitativos:

Se hicieron pruebas de espectro utilizando ruido blanco para así comparar el comportamiento del sistema tanto para el canal izquierdo como el derecho. Para esto se utilizó la herramienta “Soundcard Oscilloscope” que, a través de un cable auxiliar cortado, se conectó a la tarjeta un canal a la vez para medir su respuesta. El programa se modificó para permanecer en un ángulo azimutal específico y poder medir apropiadamente el resultado. Estos resultados obtenidos del microcontrolador fueron corroborados con los valores teóricos del espectro obtenidos a través de Matlab. Al hacer la comparación de la forma del espectro, ambos sistemas son bastante similares y muestran resultados casi idénticos.

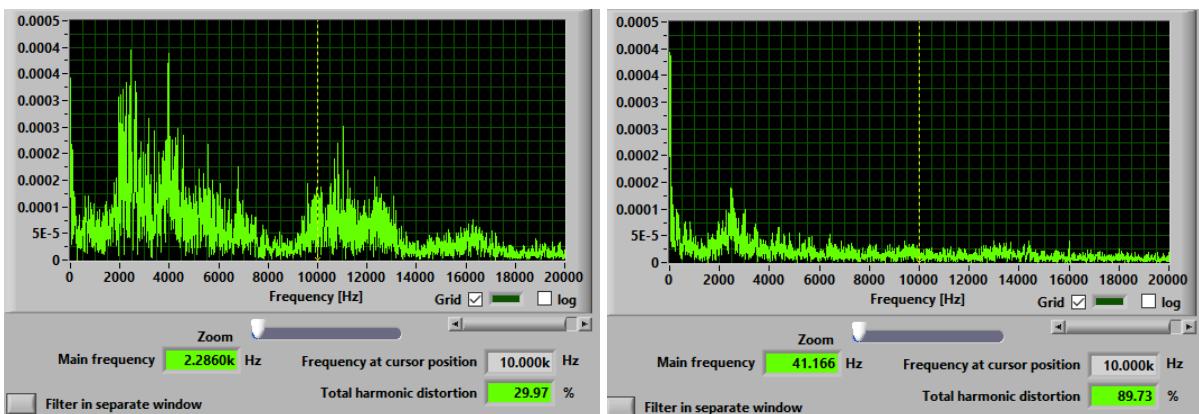
NOTA: el eje de frecuencias en las gráficas de matlab se encuentra desfasado por alrededor de 1kHz



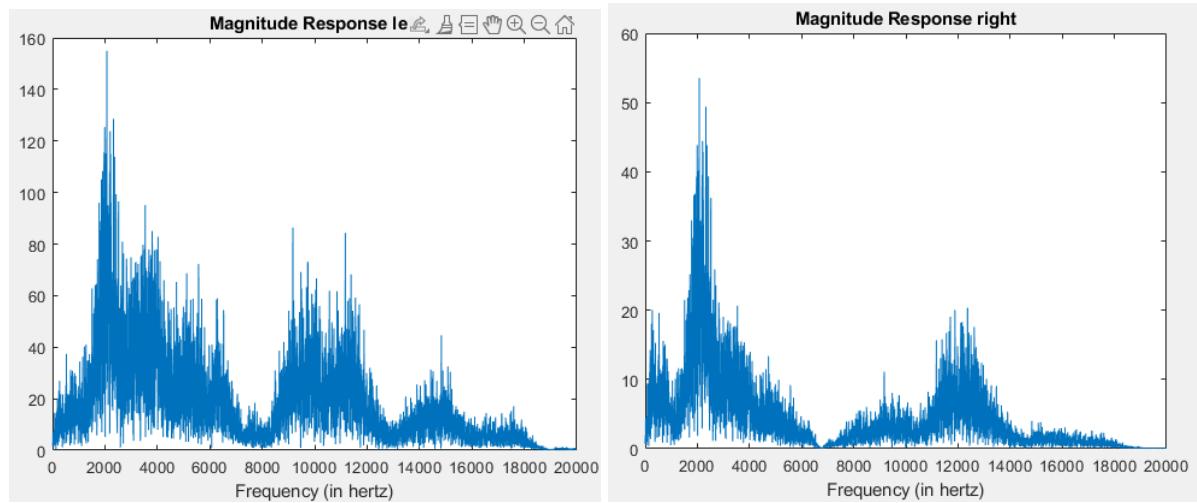
Figuras 11 y 12. 0°



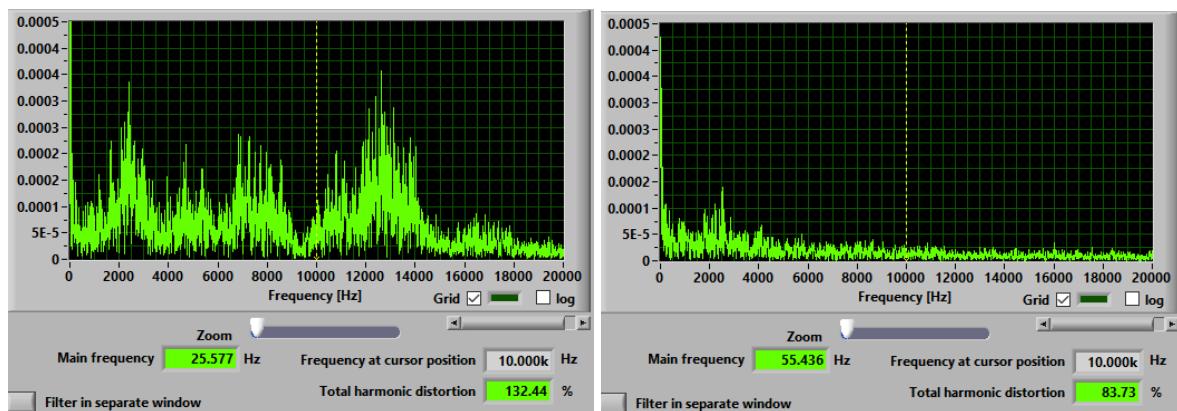
Figuras 13 y 14. 0° Matlab



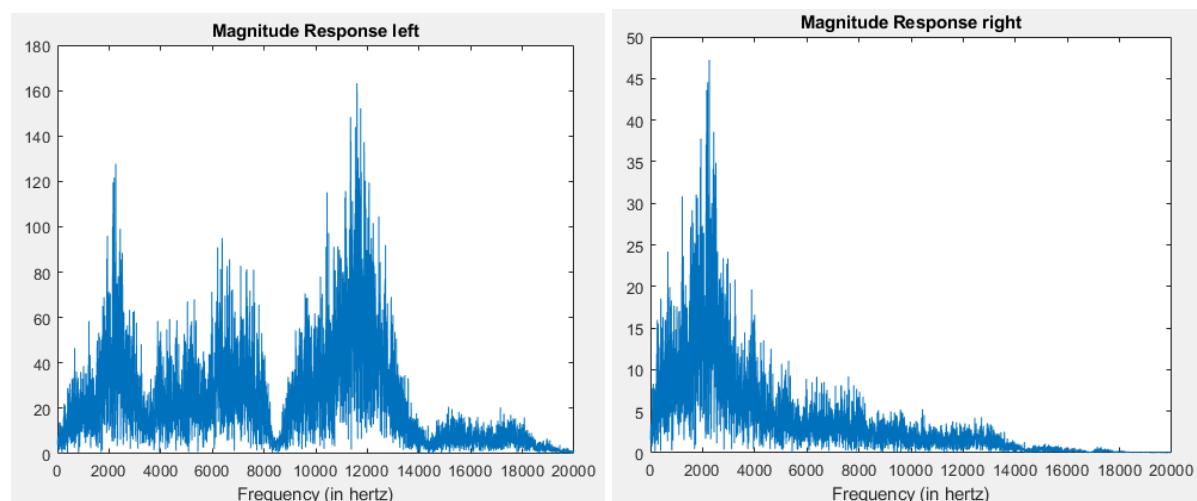
Figuras 15 y 16. 45°



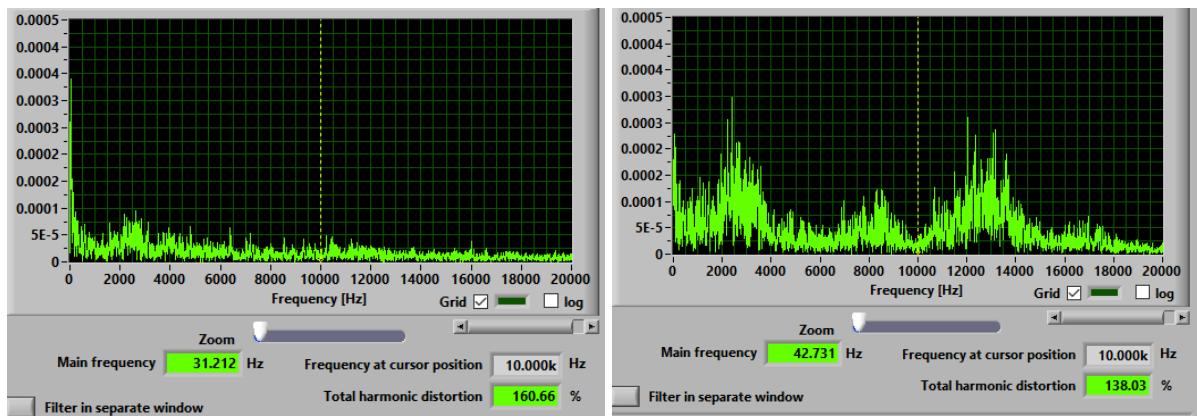
Figuras 17 y 18. 45° Matlab



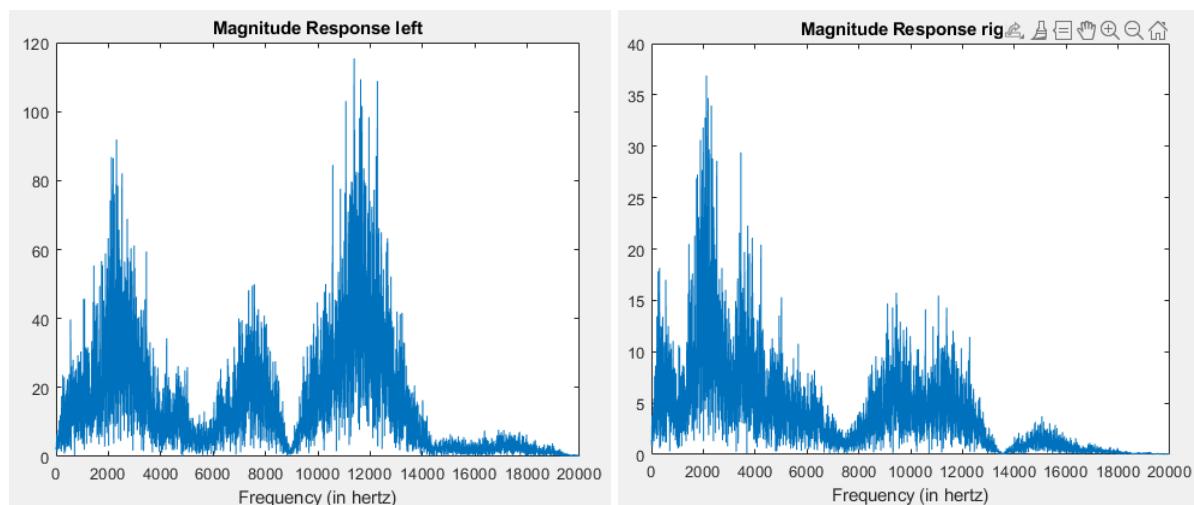
Figuras 19 y 20. 90°



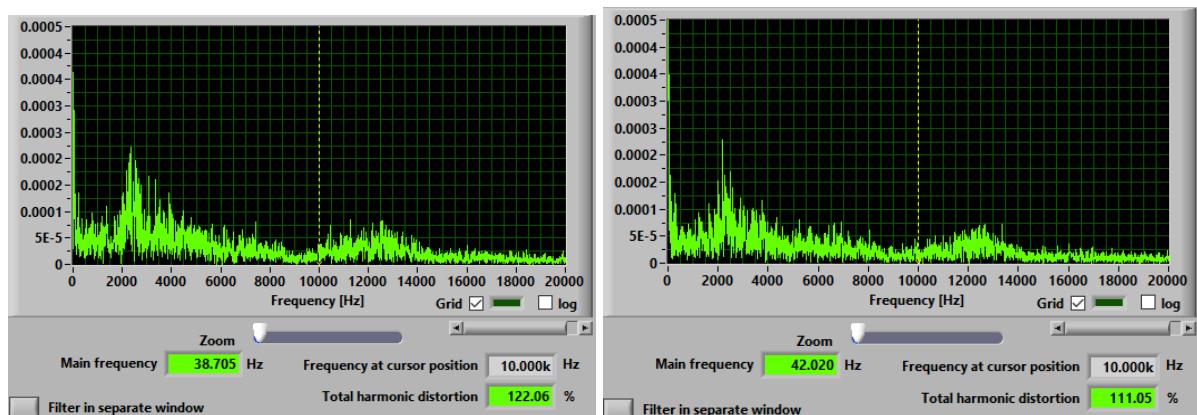
Figuras 21 y 22. 90° Matlab



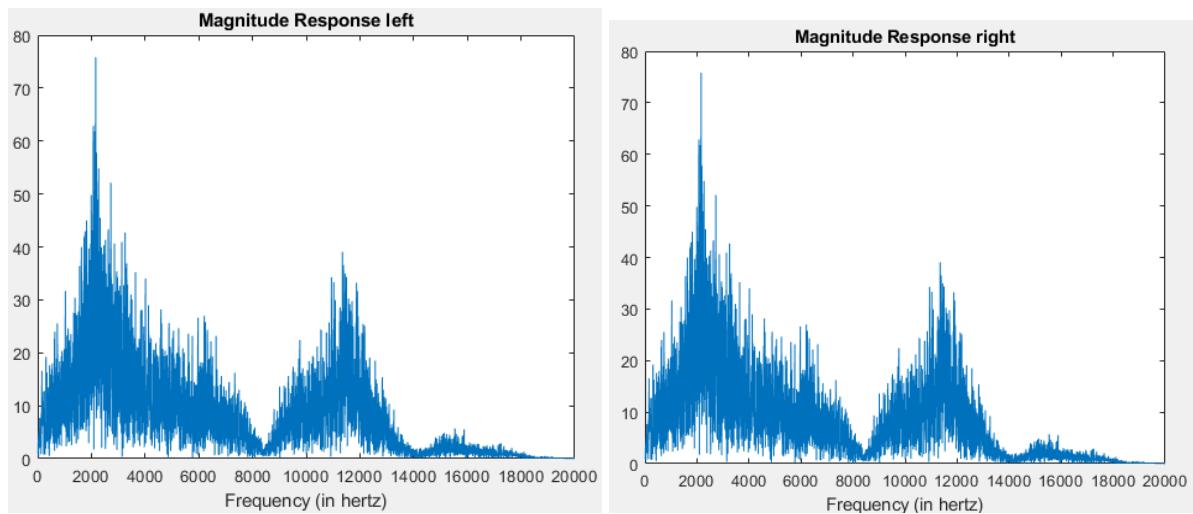
Figuras 23 y 24. 135°



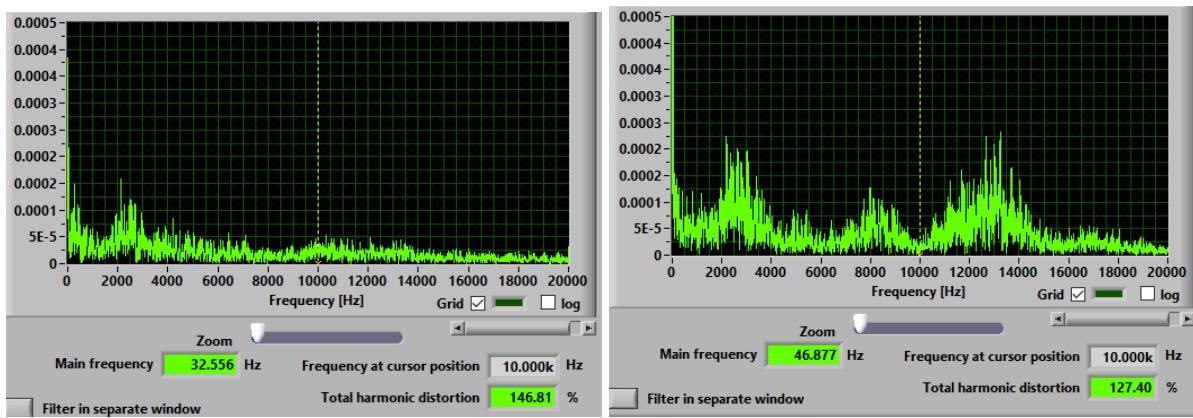
Figuras 25 y 26. 135° Matlab



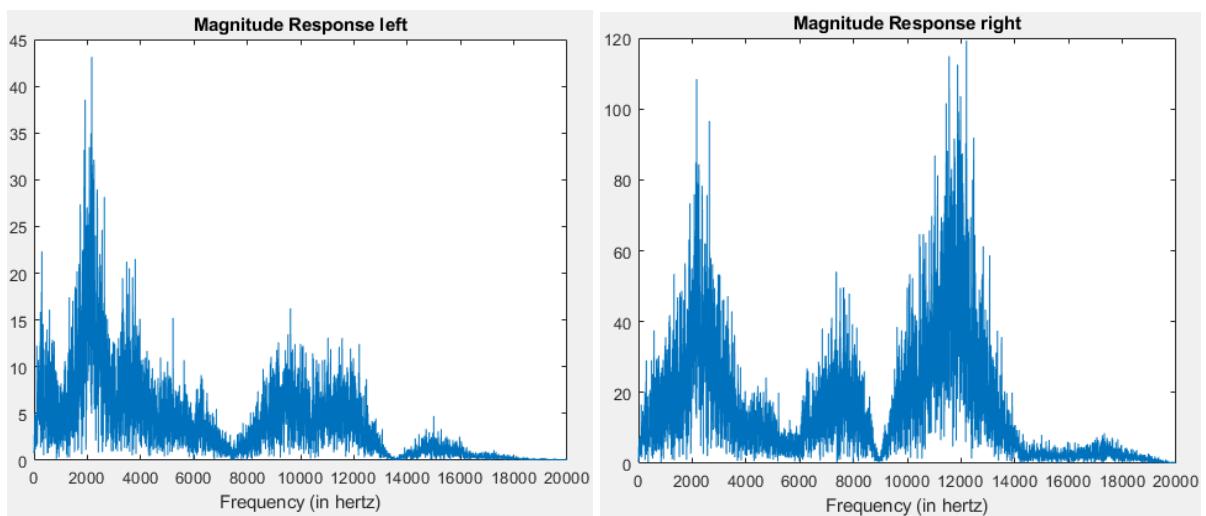
Figuras 27 y 28. 180°



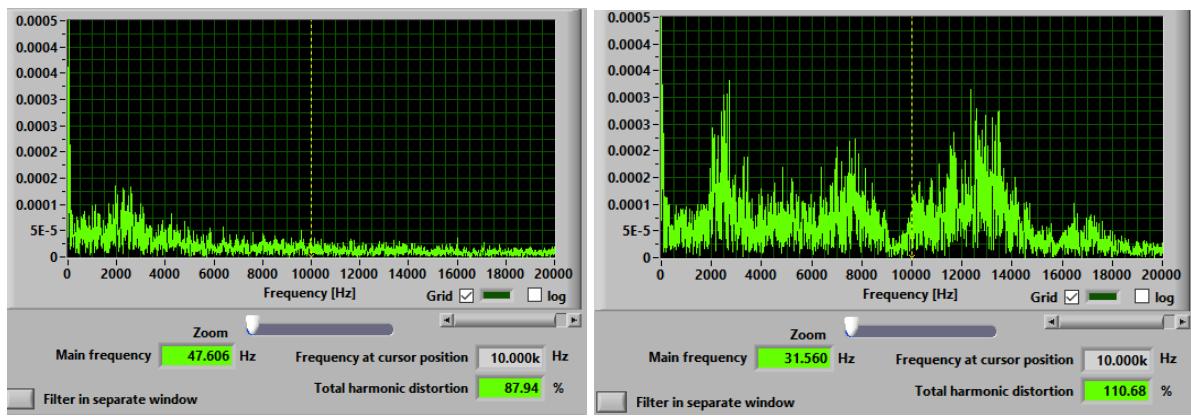
Figuras 29 y 30. 180° Matlab



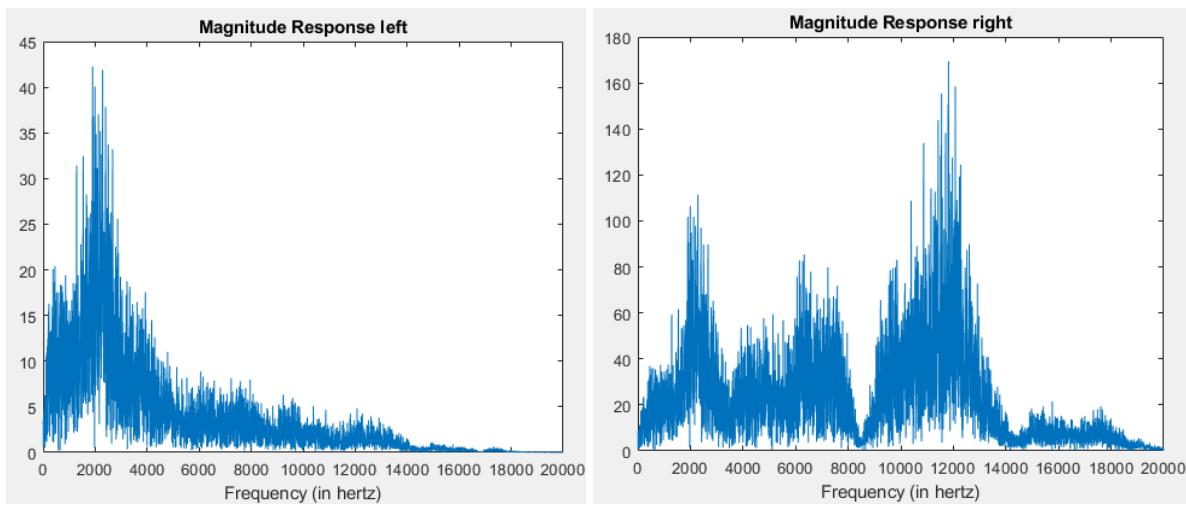
Figuras 31 y 32. 225°



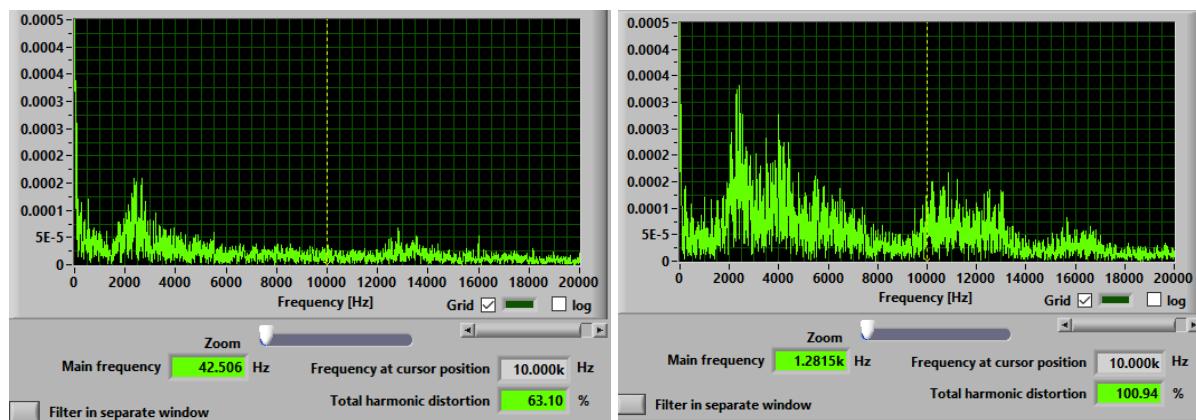
Figuras 33 y 34. 225° Matlab



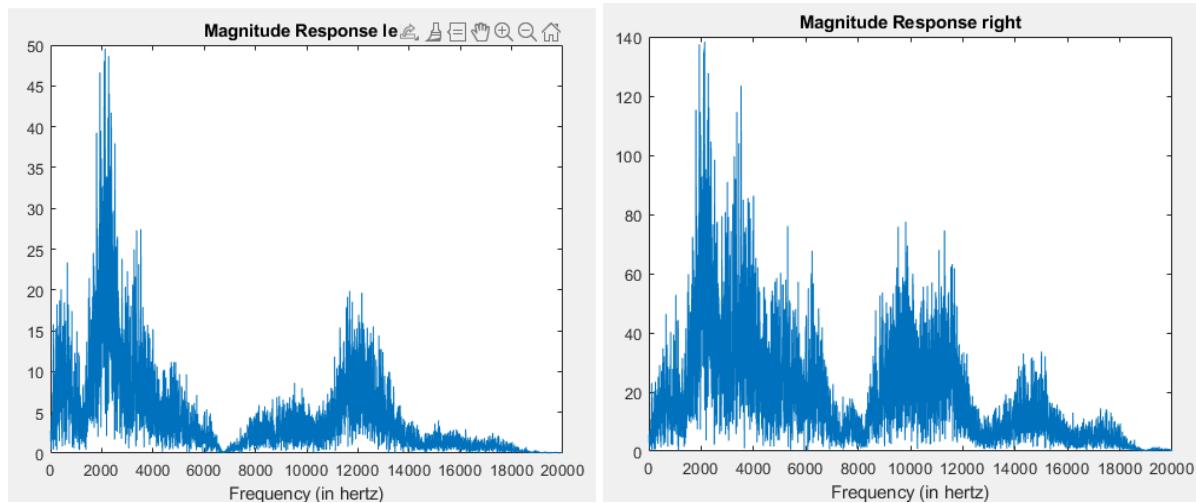
Figuras 35 y 36. 270°



Figuras 37 y 38. 270° Matlab



Figuras 39 y 40. 315°



Figuras 41 y 42. 315° Matlab

Resultados cualitativos:

Para conocer si en verdad es perceptible el cambio de ángulo azimutal en el sistema para una persona, se realizaron distintas pruebas con diferentes sujetos de prueba, donde se les hizo escuchar al sistema con ruido blanco, música y voz. Estos experimentos se pueden observar en la siguiente tabla

Sujeto	Prueba	Resultado
Sujeto 1	Ruido blanco: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	Música: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	10
	Voz: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	¿Qué sonido fue más fácil de distinguir?	Música
Sujeto 2	Ruido blanco: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	10
	Música: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	8

	Voz: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	¿Qué sonido fue más fácil de distinguir?	Ruido blanco
Sujeto 3	Ruido blanco: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	Música: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	Voz: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	10
	¿Qué sonido fue más fácil de distinguir?	Música
Sujeto 4	Ruido blanco: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	10
	Música: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	8
	Voz: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	¿Qué sonido fue más fácil de distinguir?	Ruido Blanco
Sujeto 5	Ruido blanco: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	Música: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	10
	Voz: ¿Qué tan fácil es distinguir que el sonido te rodea? del 1 al 10	9
	¿Qué sonido fue más fácil de distinguir?	Música

Tabla 1. Resultados cualitativos

Conclusiones

Es importante reconocer que este tipo de proyectos requieren de tiempo y dedicación para realizarse y si bien, los resultados finales fueron satisfactorios para el equipo hubo algunos problemas con los que nos topamos en el camino, principalmente con la implementación del código en C que debía procesar la señal. Se experimentaron distintas situaciones al realizar el proyecto desde que en ocasiones al correr el código funcionaba de manera óptima y otras veces dejaba de funcionar sin razón aparente, o que en ocasiones el código presentaba problemas para uno de los integrantes del equipo y para el otro no, hasta el hecho de que se escuchara una señal con mucho ruido e interferencia o que los auriculares no reprodujeran el sonido. Sin embargo, a pesar de los inconvenientes se tuvieron bastantes aprendizajes sobre el procesamiento de señales y sus aplicaciones e inclusive incrementamos nuestro conocimiento y habilidades en cuanto a la programación en C, lo cual para nuestra carrera es bastante importante.

Referencias bibliográficas

- (1) Marín de la Rosa, J.. (s.f).Fundamentos teóricos: Filtros. Junio, 2021, de Sindicato Español Universitario Sitio web:
<http://bibing.us.es/proyectos/abreproj/11375/fichero/MEMORIA%252FFundamentos+teoricos.pdf>
- (2) Keith, M. Gardener, B. (2020). *HRTF Measurements of a KEMAR Dummy-Head Microphone*. MIT. Recuperado de
<https://sound.media.mit.edu/resources/KEMAR.html>
- (3) Alonso, R. (2020). *HRTF, la tecnología que permite escuchar audio posicional en juegos*. Hard Zone. Recuperado de
<https://hardzone.es/reportajes/que-es/hrtf-tecnologia-audio-posicional/>
- (4) Anónimo. (2012). ¿QUÉ ES UN DAC?. Junio 2021, de Sound&Pixel Planet Sitio web: <http://www.sound-pixel.com/blog/%C2%BFqu%C3%A9-es-un-dac>
- (5) Ánonimo. (2020). Uso de I2S en aplicaciones con respuesta en tiempo real. Junio,

2021, de Microsoft Sitio web:

<https://docs.microsoft.com/es-es/azure-sphere/app-development/i2s-rt>