

# Adquisición de audio con un micrófono con fpga

Sergio Encina  
Charles Trujillo

## 1. Introducción

En el presente trabajo es parte de un proyecto que busca medir la diferencia de llegada de un impulso sonoro a dos micrófonos ubicados a una distancia fija. Con la diferencia de tiempo obtenida entre los micrófonos se puede determinar una hipérbola desde la cual fue emitido el impulso. Si uno quisiera determinar la posición desde donde fue emitido el sonido, se deberían agregar micrófonos ubicados adecuadamente.

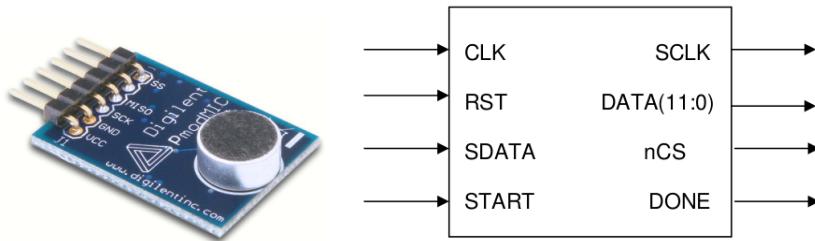


Figura 1: eferencia del componente PmodMic

La distancia de separación de los micrófonos nos da la máxima distancia temporal que se va a medir entre los dos micrófonos.  $L = v_s \Delta t_{max}$  , entonces si tenemos una distancia de un metro entre micrófonos ( $L=1$  m) el  $\Delta t_{max} = 3,3$  ms. El  $\Delta t_{min}$  depende de la frecuencia con la cual se midan los micrófonos, tendría que ser por lo menos un par de ordenes de magnitud menos que el máximo.

Actualmente se trabaja con una frecuencia de muestreo de 44.1kHz, es decir, con 44.100 muestras por segundo. Esta cifra se ha convertido en un

estándar para la mayoría de los proyectos de creación musical. No sólo es la misma frecuencia utilizada en el sistema de CD-Audio, sino también la más baja capaz de reproducir todo el espectro de frecuencia al que el oído humano es sensible: desde unos 20Hz hasta los 20kHz. Con esta frecuencia de muestreo se obtiene un  $\Delta t_{min} = 2,27\mu s$ , por lo que nos parece adecuado adoptar este standar para nuestra aplicación.

Para el proyecto de la detección de sonido se contó con una placa de desarrollo de fpga nexys 2, un módulo de microfono de la marca digilent (pmodMic) que ofrecia una resolución de 12 bits con una frecuencia máxima de sampleo de 1 MSPS. Se decidió utilizar solo 8 bits para el sampleo debido a que se necesitaba campturar la existencia de impulsos sonoros relativamente altos y nos basto esta resolución.

## 2. Aquisición de la señal sonora con un micrófono y envio a una PC

Antes de afrontar el problema con los dos micrófonos, se propuso resolver el problema con uno solo, es decir, poder grabar sonido con la frecuencia de muestro adoptada de 44.1 kHz con 8 bits de resolución y enviarlo a la computadora desde la fpga. A continuación se describen los módulos utilizados para esta primera aproximación al problema.

Para la adquisición de los impulsos sonoros se simplificó el problema en bloques mas sencillos, como se muestra en la figura 3.

El primer módulo es brindado por el fabricante del micrófono, el pmodMic, este realiza la digitalización del sonido a 12 bits y envia los datos digitalizados cada vez que le llega una señal de start.

El siguiente módulo, *lanzador* , se utiliza para seleccionar apropiadamente la frecuencia de muestreo, esto se consigue acondicionando una señal de *start* a 44.1 kHz, manteniendola en alto durante una frecuencia del sclk 12.5 MHz. Además *lanzador* envia una señal que indica cuando estan listos los datos para ser guardados.

El módulo *guardar* se ocupa de enviar los datos a una memoria fifo cuando recibe la señal de la disponibilidad de los datos. El pmodMic nos entrega los datos con una resolucion de 12 bits, en este módulo se desechan los 4 bits menos significativos y se envian los 8bits restantes.

El módulo de la *fifo* fue creada utilizando el 'core generator' del programa ise, con 8 bits, en esta primera etapa de diseño, con la máxima capacidad

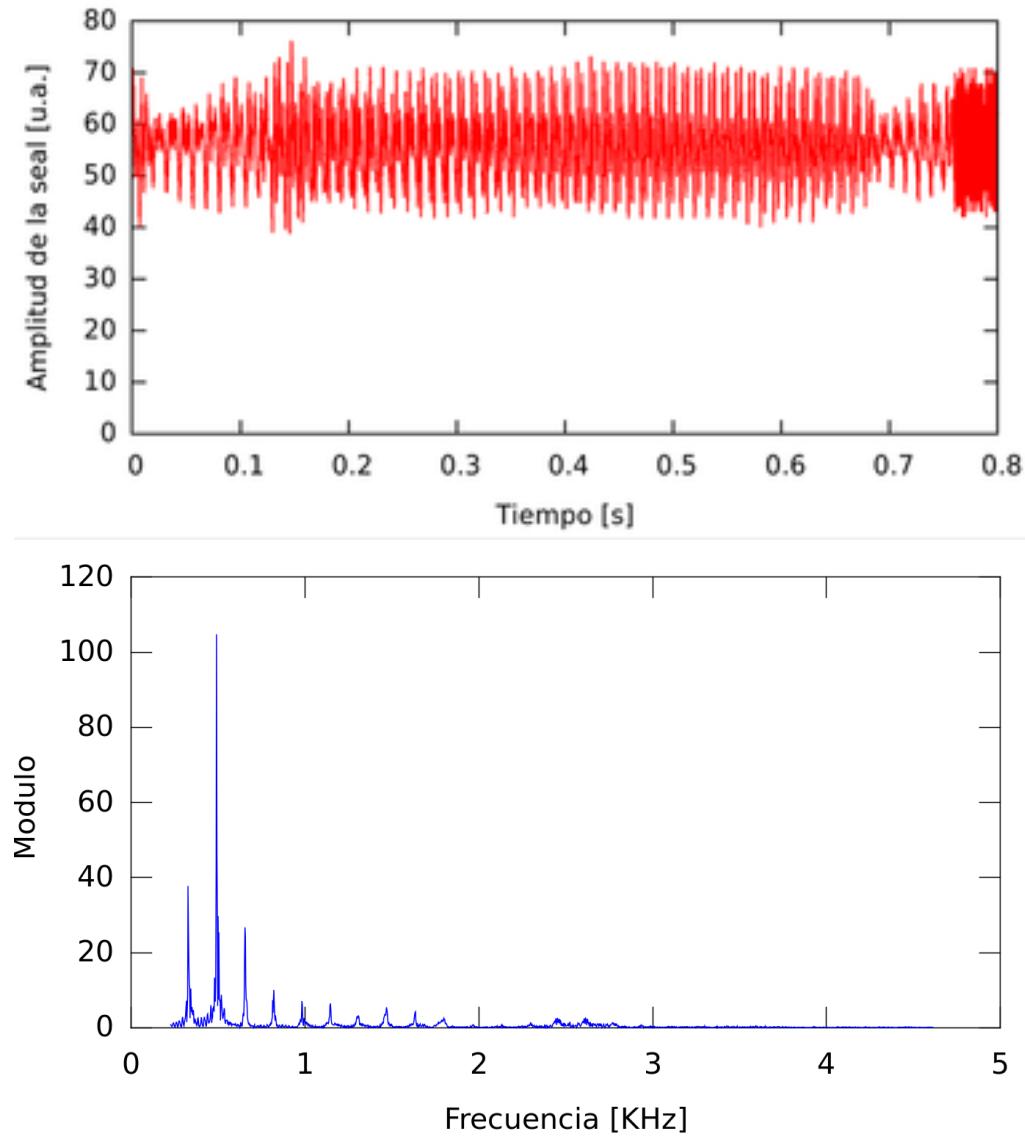


Figura 2: Ejemplo de la adquisicion de audio con un micrófono y su espectro de frecuencias.

disponible, para obtener un tiempo mayor de grabación del sonido.

En el módulo de *envío de datos* se implementa la comunicación entre la pc con la placa de la fpga. Siempre que haya datos disponibles en la fifo, los

envia a través del puerto RS-232 a una velocidad de 19200 baudios. Desde la pc se leía los datos con un programa implementado en python, y si estos estaban disponibles, se guardaban en un archivo \*.dat con la finalidad de ver la señal adquirida, como en la figura 2.

Se realizaron diversos ensayos para comprobar el código implementado. Se procedió a grabar una serie de tonos a frecuencias conocidas, una de ellas se muestra en la figura 2 . Los datos fueron enviados a la pc para graficarla, ademas se realizó el espectro de frecuencias de para comprobar que correspondiera a las frecuencias de los tonos que se emitieron para ser grabados.

### Descripción de los módulos

Se implementó un programa donde se instanciaron los módulos implementados y brindado por los fabricantes (el pmodMic) y generados con el core generator de ise (las memorias FIFO).El diagrama en bloques del programa se muestra en la figura 3.

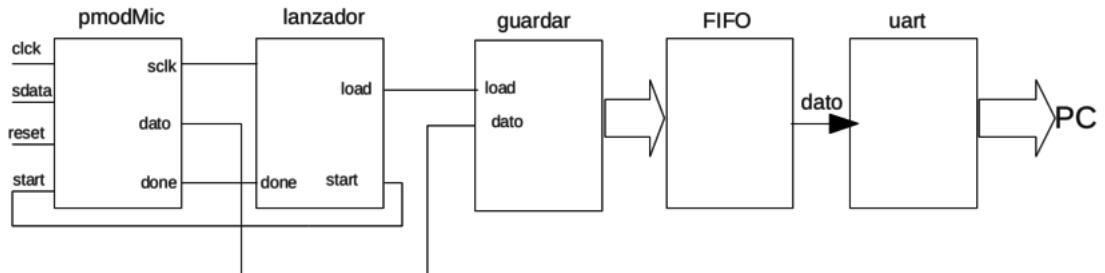


Figura 3: Descripción del programa principal

A continuación se detallan los módulos implementado para la adquisición y envío de la señal, capturadas con los módulos pmodMic para FPGA, hacia la PC mediante el puerto RS-232.

#### 2.1. Lanzador

##### Entradas

- Clock: 12.5 MHz
- reset asincrono.

- medir: variable que ordena al microfono a medir.
- done: micrófono indica que los datos terminaron de procesar.

## Salida

- start: señal para empezar una medición del micrófono.
- load: Indica a la fifo que guarde el dato.

## Señales

- maquina de estados:
  - idle
  - lanzar
  - lanzando
  - leeryguardar
- variables:
  - est\_reg, est\_next
  - velocidad sampleo, 1 o 0 de forma tal que setea la frecuencia de medición
  - auxvelocidad (vector de 9 bits) : para contar cuando poner en 1 la velocidad de sampleo.

## Instanciaciones

- mcounter (contador) con  $M=284$  de 9 bits, utilizando para tener una frecuencia de sampleo de 44.1KHz.

### 2.2. Guardar

#### entradas

- data: vector de 12 bits
- load, reset, clock, full\_fifo.

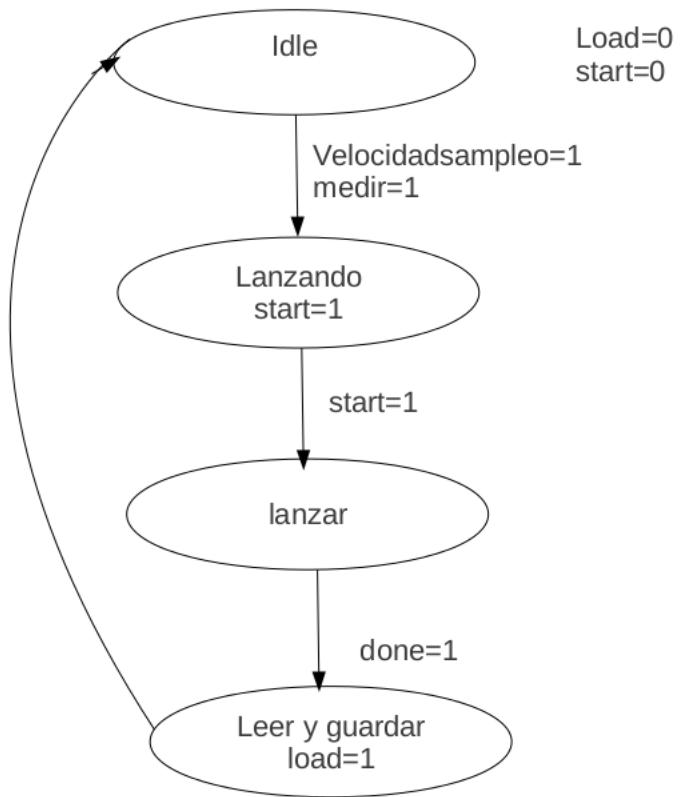


Figura 4: Maquina de estados de lanzador

### Salidas

- data\_fifo (vector de 12 bits)
- write\_read enable\_fifo, clk\_fifo, reset

### Maquinas Estados

- señales: idle qrite

### 2.3. Módulos utilizados

- micrófono (bajado de digilent.

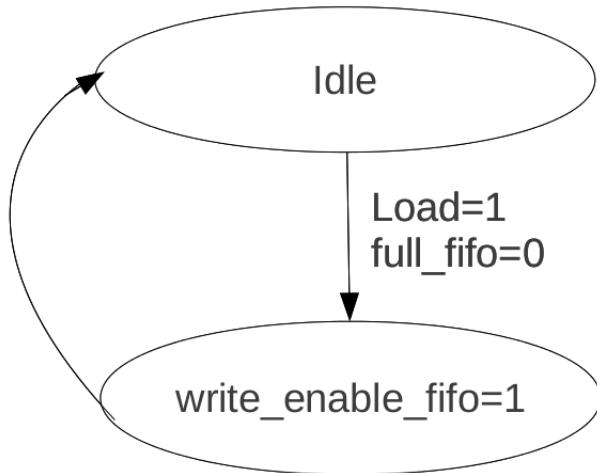


Figura 5: Máquina de estados del módulo guardar

- fifo (generada con el program generator)
- uart: utilizado el ejemplo del libro de FPGA de Chu.

Con estos módulos implementados se logró adquirir el audio, que fue comprobado al analizar el espectro de un tono con frecuencia conocida, como se mostró en la figura 2, para un solo micrófono. Hasta esta parte utilizamos diversos "programitas" implementados a lo largo de la materia, como el contador, para seleccionar la frecuencia adecuada, el uso de máquinas de estado para casi todos los modulos. Además se dejó en marcha la comunicación entre las señales físicas con la pc (manejo del puerto RS-232) a través de un micrófono y la fpga (manejo de tiempos de muestro y almacenamiento en memoria tipo fifo), siguiendo un standar en sonidos.

### 3. Integración del Trabajo

Posteriormente a resolver el problema de la adquisición de audio con un micrófono se procedió a debatir la implementación para adquirir y procesar los datos de dos micrófonos para determinar la diferencia de tiempo en que les llega la señal a cada uno. La idea planteada fue dividir el proyecto en cuatro bloques (ver figura 6 ), que resolvían diferentes etapas del problema. Estos bloques se describen a continuación:

- Adquisición de audio: Comunicación con los micrófonos y guardar los datos en dos fifos. Este bloque actúa como interfaz entre el mundo físico y provee las entradas necesarias a los bloques de análisis y procesamiento.
- Análisis de datos: en este bloque se mide la diferencia de tiempos de la llegada del impulsos sonoro a los micrófonos.
- Mostrar: Recibir y mostrar la diferencia de tiempo medida en un display matricial que recibe como resultado del análisis de los datos.
- Envío de datos: Se envian los datos almacenados en las fifos. Este envío de datos hacia la PC se realiza para una posterior verificación del resultado obtenido y posterior análisis de ser necesario.

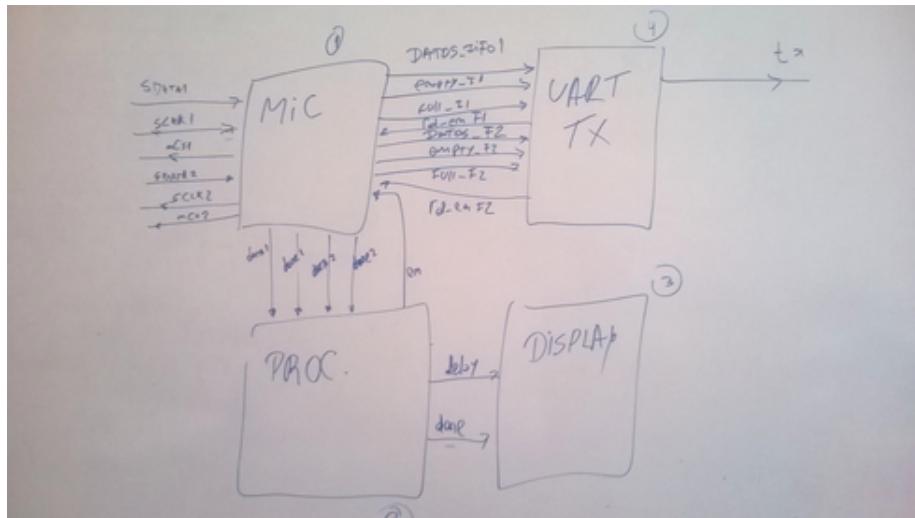


Figura 6: Diagrama del proyecto considerando los dos micrófonos

### 3.1. Adquisición de audio

La idea principal para resolver este bloque era duplicar la implementación realizada para un micrófono, con la excepción de no usar el módulo de comunicación uart con la fifo, y estas son de menor capacidad de almacenamiento. Este bloque es además el que más entradas y salidas posee, debido a que interactúa con los bloques de procesamiento, la comunicación uar y el mundo real.

### 3.2. Análisis de Datos

Las señales adquiridas con los dos módulos de micrófonos, poseén el mismo espectro, lo que es coherente ya que miden lo mismo, pero al estar separados una distancia, estas señales presentan un corrimiento temporal. El corrimiento temporal será el que nos brinde la información acerca del origen del impulso sonoro.

Para determinar la diferencia temporal entre estas dos señales se imple-

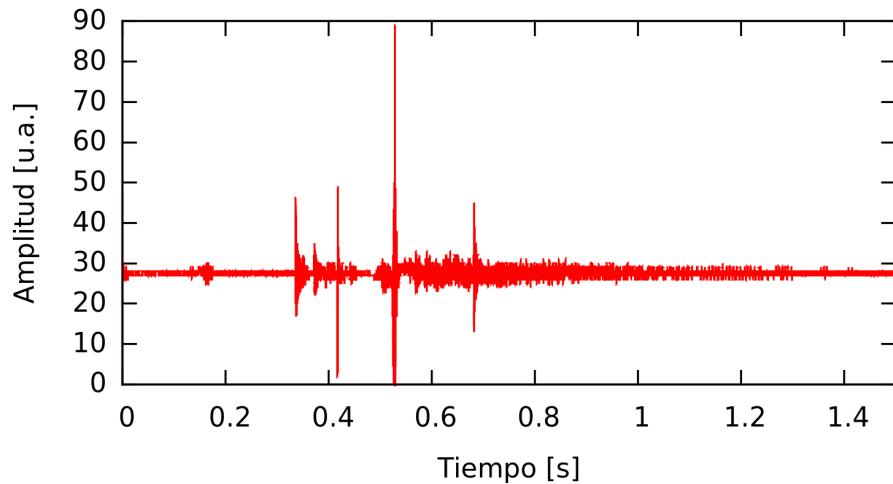


Figura 7: Señal que muestra un serie de aplausos. Se puede observar que el nivel de salto de trigger debería ser de 15 unidades superior al promedio del ruido de fondo.

mentó una maquina de estado que determina la diferencia entre los trigger de ambos micrófonos. Para lanzar los triggers se calcula el promedio del sonido de fondo y cuando llega una señal que supera este promedio en una cantidad fijada adecuadamente se pone en uno el trigger (para esto se analizó una figura de un par de aplausos como se puede ver en 7), iniciando un contador hasta la llegada del segundo trigger, que se activa de la misma forma pero asociado al otro micrófono. La máquina de estados mencionada se muestra en la descripción de los modulos implementados.

Cabe mencionar que este bloque habilita el guardado de datos en la fifo con la orden dada por alguno de los triggers.

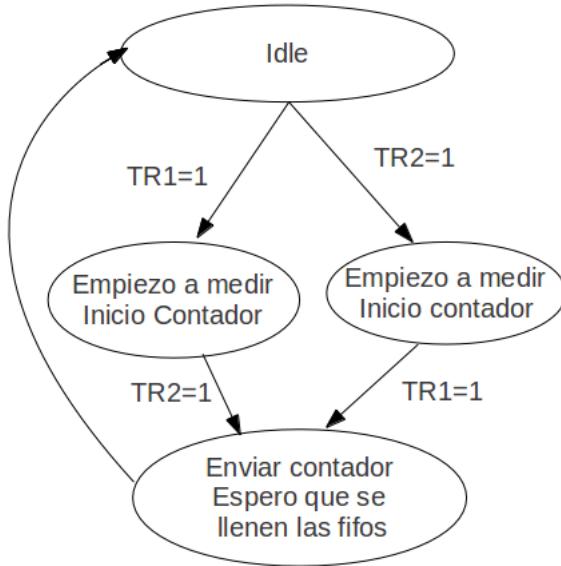


Figura 8: Máquina de estado para lanzar el trigger.

### 3.3. Envio de datos

La estrategia que se utilizó para enviar los datos es esperar a que se llenen ambas fifos. Luego de que esto suceda se empiezan a enviar los datos de la primera fifo y cuando se obtiene una señal de que esta vacía, se procede a vaciar la segunda fifo. Las fifos al ser del mismo tamaño, los datos enviados presentan un vector donde la mitad de los datos corresponden a cada micrófono.

## 4. Conclusiones

El módulo PmodMic para el FPGA al recibir una frecuencia de 50MHz realiza muestras cada 12.5Mhz, para obtener una frecuencia de muestreo de 44.1KHz se implementó con un contador hasta 284 cada 12.5MHz.

La adquisición del sonido fue realizada con el pmodMic y pudo verificarse, a través de diversos ensayos la correcta adquisición de los datos, grabando tonos y observando su espectro para frecuencias ya conocidas. Se armó y se discutió una estrategia en conjunto para grabar y procesar los datos de dos micrófonos, para la integración con los grupos de la materia.

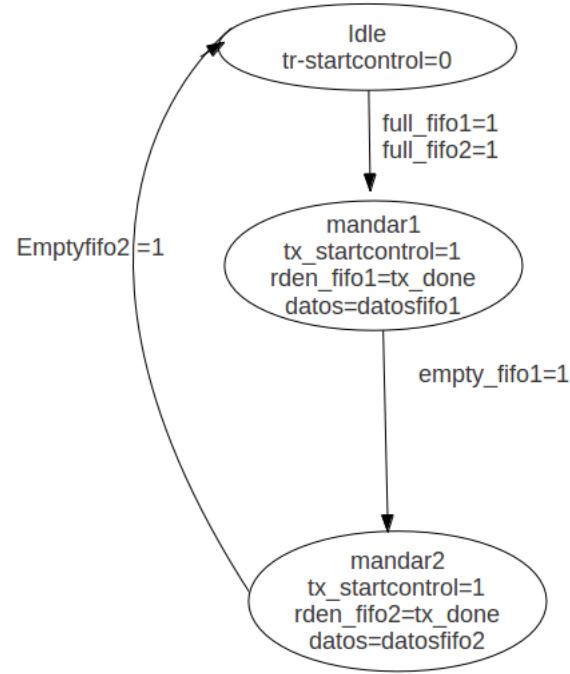


Figura 9: Máquina de estados del módulo de envío de datos de los micrófonos

Se implementaron diversos modulos en cada grupo. Algo interesante para mejorar, sería el manejo de la memoria para grabar audio por más tiempo. También realizar algún cambio entre la comunicación pc-fpga, a altas velocidades sin necesidad de almacenar grandes volúmenes de datos en las fifos. Por ejemplo, si queremos enviar la señal de dos microfonos en tiempo real necesitariamos un protocolo que nos permita enviar a una velocidad de 4Mb/s, lo cual se supera con creces con una interfaz usb 2.0, sería interesante poder cambiar la transmisión de datos a este tipo de tecnología.

La velocidad de procesamiento de datos realizado por la fpga supera con creces a la capacidad de almacenamiento. Por lo cual para calcular la diferencia de tiempo entre microfonos no es necesario almacenar todos los datos adquiridos, solo se necesita un ventana que nos permita calcular el promedio, sin la necesidad de utilizar una FIFO.