

# Proyecto Final: Guante MIDI

David Burbano, Miguel Perez, Carlos Pinto

(daaburbanola@unal.edu.co, migaperezber@unal.edu.co, cfpintog@unal.edu.co)

## I. INTRODUCCIÓN

La especificación MIDI (Musical Interface Digital Interface) surge en 1983 como un estándar técnico universal que permitiera la comunicación y conexión entre equipos profesionales de audio, independientemente del fabricante. Mediante un protocolo de comunicación serial, compuesto por tres bytes, fue posible el envío y recepción de información asociada a un instrumento musical específico, su volumen (velocity), tono, paneo, etc.

En principio este protocolo fue concebido para instrumentos musicales cuyas características de diseño digital consistiera en Hardware, sin embargo, debido a un desarrollo paralelo de los computadores personales como instrumentos de producción de audio, el protocolo MIDI tomó un enfoque de co-diseño HW/SW que resulta interesante para el objetivo del curso.

## II. DESCRIPCIÓN DEL DISPOSITIVO.

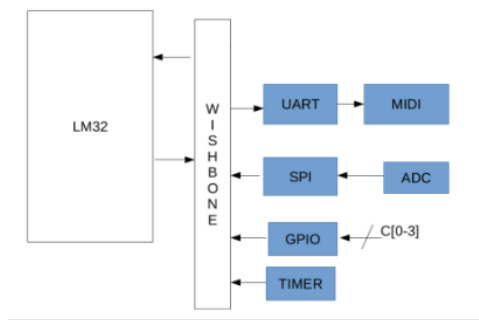


Figura 1. Bloque estructural

El dispositivo se plantea como un instrumento musical en el cual la activación de las notas se hace por medio de movimiento de los dedos, usando un guante que tendrá incorporado sensores fotoeléctricos, dispuestos de tal forma que capten el movimiento de los dedos del usuario.

Una vez captada la posición de los dedos, esta es procesada para luego ser enviada a modo de mensaje de protocolo MIDI, el cual es transmitido por medio de UART a un circuito acondicionador, el cual eleva los niveles lógicos característicos de la FPGA a los 5V lógicos necesarios para el funcionamiento del sistema MIDI, mensajes que serán interpretados por una interfaz para luego ser reenviados a el dispositivo de reproducción.

Debido a las características del protocolo MIDI ya que es el protocolo usado de manera profesional para aplicaciones musicales de alta fidelidad, se cuentan con diversas opciones, tales como cambiar el paneo (Distribución espacial del

sonido), la velocidad de la nota, el instrumento a emular en el dispositivo reproductor de sonido y se puede variar entre 127 notas diferentes.

En nuestro caso en específico se cuenta con la selección de 8 distintas notas, estas debido a las diferentes combinaciones posibles de los dedos índice, corazón, anular y meñique; Además se cuenta con la opción de cambiar el nivel de volumen, esta señal depende del valor medido en un potenciómetro, el cual es cambiado a valores digitales por medio de un conversor Analógico-Digital, el cual se comunica con la FPGA por medio de protocolo SPI, y de la misma manera que la nota, es enviado por medio de MIDI a el dispositivo de reproducción.

## III. ESPECIFICACIONES SOFTWARE/HARDWARE

A continuación se presenta información detallada sobre las especificaciones HW/SW del proyecto.

### Especificaciones de Software

#### Análisis de señales provenientes de sensores:

Uno de los aspectos fundamentales del proyecto consiste en que al recibir una señal específica de un actuador, el sistema esté en capacidad de analizar dicha señal, saber de dónde proviene, y realizar una tarea específica a partir del análisis de la misma. En este caso la señal eléctrica proveniente del ADC es digitalizada, y posteriormente enviada a la FPGA. El análisis de la señal y la ejecución de una acción específica son procesos que resultan más viables desde una perspectiva de diseño de software. En este caso, el procesador montado en la FPGA debe estar en la capacidad de determinar la intensidad (o nivel) de la señal proveniente del sensor, y a partir de esto asignar un nivel del volumen.

Adicionalmente, considerando que el proyecto hace uso de cuatro fototransistores añadidos al guante, el software debe estar en capacidad de determinar el origen de la señal digital. A partir de este análisis se decide que nota se asigna identificadas con un valor en el protocolo MIDI.

### Especificación de Hardware

Periféricos para ADC: La conversión de señales analógicas y digitales requiere de la utilización de circuitos integrados que pasen del dominio discreto digital, al dominio continuo análogo y viceversa. Para el caso de los conversores ADC, basta con la utilización de circuitos ADC de 8 bits. Esto debido a que el protocolo MIDI tiene un rango de máximo 7 bits para el nivel (velocity).

Los demás elementos de hardware van asociados a los periféricos de comunicación usados:

- SPI
- UART
- GPIO
- TIMER

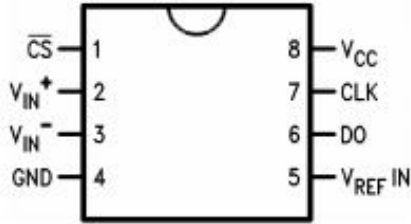


Figura 2. IC del conversor Análogo-Digital ADC08031.

#### IV. PERIFÉRICOS UTILIZADOS, DESCRIPCIÓN Y FUNCIONAMIENTO.

##### UART

La UART (Universal Asynchronous Receiver/Transmitter) corresponde a un Hardware que traduce información de serial a paralelo y viceversa. Lo anterior quiere decir que la UART toma información paralela en forma de Bytes, y envía bit por bit de esta información de manera secuencial.

Esto implica la definición de ciertas condiciones de manera que tanto emisor como receptor puedan comunicarse de manera adecuada. Primero, la comunicación es asíncrona, esto significa que no existe una señal de reloj común entre las dos partes que se comunican. Segundo, se requieren bits de inicio y de parada de manera que las dos partes sepan en que momento se da inicio y finalización a la transmisión de datos. Tercero, se utiliza un único lazo para la transmisión y recepción de información, lo que significa que no existe ninguna de las partes actúa como maestro o esclavo.

Para el proyecto se parte de una UART diseñada previamente teniendo en cuenta la compatibilidad con el Wishbone del LM32, este periférico se implemento para lograr la transmisión del mensaje MIDI; mensaje que puede ser transmitido por medio de una interfaz MIDI o por medio de bluetooth usando el modulo HC-06, pero dicha transmisión no puede ser reconocida por la dificultad para cambiar la velocidad de transmisión de datos por parte del modulo de bluetooth ya que MIDI utiliza una velocidad de 31250 baudios.

En la figura IV se puede observar el diagrama funcional de UART en donde vemos las señales involucradas tanto en el proceso de transmisión como en recepción, en el caso específico de esta aplicación solo es usada la función de UART-tx, ya que este periférico solo es usado para transmisión del mensaje MIDI; Además que la frecuencia de envío debió ser puesta a 31250 baudios.

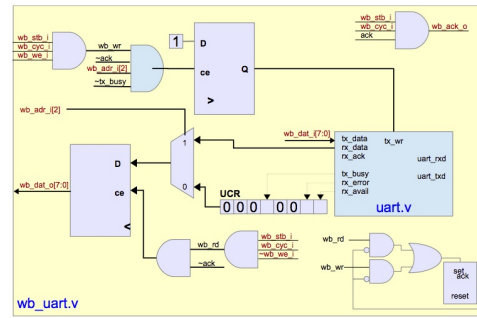


Figura 3. Diagrama lógico de periférico UART.

##### SPI

Para el correcto funcionamiento del proyecto es necesario recibir de manera adecuada una señal analógica proveniente del control de volumen, para esto se hace necesaria la implementación de un conversor AD, usando el ADC0831 el cual tiene una salida serial y se comunica con el procesador por medio de protocolo SPI.

El SPI tiene una comunicación síncrona y utiliza líneas separadas para los datos y el reloj que mantiene a ambos lados en sincronía. Cuenta con 4 señales propias, CLK (reloj), la señal MOSI (Master Output - Slave Input), la señal MISO (Master Input - Slave Output) y un SS (select Slave) o CS. Al producirse una señal de reloj es seleccionado el esclavo con la señal SS, la cual al estar inactiva se encuentra en un estado lógico alto y en el momento en el que es seleccionado un esclavo la señal SS se pone en estado bajo. Las líneas de datos son enviadas en orden. Primero el dato de transmisión, aquel que va por la señal MOSI, y luego el dato de recepción que va por la señal MISO. Este proceso descrito anteriormente se ilustra en la figura IV.

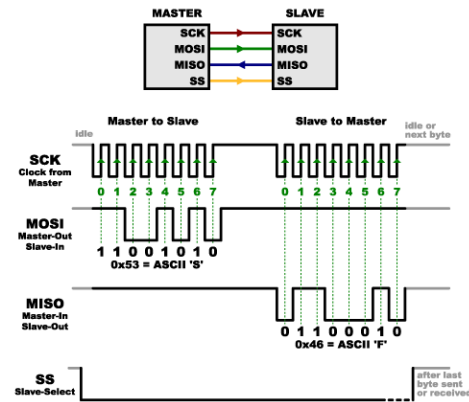


Figura 4. Diagrama de transmisión SPI.

##### Timer

El periférico TIMER es un hardware que permite al usuario implementar un control para la generación de bases de tiempos. Tiene la capacidad de realizar funciones como medición de frecuencia, conteo de eventos, conteo de intervalos, generación de reloj, etc. En la figura se observa la estructura del ciclo

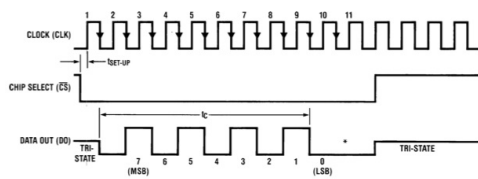


Figura 5. Funcionamiento 0831.

de lectura y escritura del periférico TIMER. Este periférico posee 8 registros y la posibilidad de leer al tiempo estas 8 variables. El timer cuenta con un contador, un comparador y un “tcr” de 32 bits, este registro contiene los bits EN, AR, IRQEN Y TRIG, los cuales corresponden a bits de control. El registro compare (comparador) de 32 bits contiene el tiempo o dato en el que se ha definido el retraso, de esta forma, al obtener el valor de compare, se activa el bit EN el cual inicia el conteo. Una vez el contador alcanza el valor compare se activa el bit AR y se reinicia el contador en 1. las variables son posteriormente enviadas al wishbone por la señal wb\_dat\_o. En la fig. 12 se muestra el funcionamiento del periférico TIMER. Para el desarrollo del proyecto, se implementó un timer o retraso en los mensajes de desactivación de nota, con el objetivo de permitir que la nota o sonido sea reproducida durante un tiempo más largo, de lo contrario la nota solo duraría el tiempo que el procesador toma en leer el mensaje de desactivación de nota lo cual equivaldría a un tiempo muy pequeño.

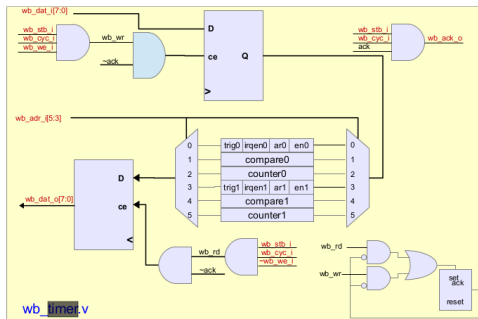


Figura 6. Comunicación Periférico TIMER- Lm32

*GPIO*

El periférico GPIO es un periférico de pines de entrada y salida de propósito general. Permite controlar la dirección de un pin, controlar el valor de un pin de salida y leer el valor de un pin de entrada. En la siguiente figura se muestra el funcionamiento interno del periférico GPIO.

Para el desarrollo del proyecto se implementó un GPIO que a través de unos interruptores en la tarjeta de desarrollo permita cambiar de instrumentos de acuerdo con la especificación midi. Esto con el objetivo de que el proyecto no tenga solo funcionalidad de batería sino que también pueda sonar como diferentes instrumentos seleccionados a través del GPIO. Una de las importantes capacidades del GPIO es la capacidad de leer valores de entrada (por lo general alto = 1, bajo = 0).

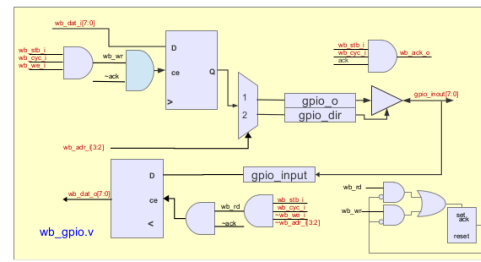


Figura 7. Comunicación Periférico GPIO- Lm32

Teniendo en cuenta esta funcionalidad, para este GPIO se utilizaron 8 interruptores de la Nexys 4, estos 8 interruptores corresponden cada uno a un bit ( sea 0 o 1) seleccionado a partir del usuario, lo cuales permiten un rango de 0-255 en número binario que corresponde en total a 255 diferentes sonidos de instrumentos musicales.

## V. PROTOCOLLO MIDI

MIDI (abreviatura de Musical Instrument Digital Interface) es un estándar tecnológico que describe un protocolo, una interfaz digital y conectores que permiten que varios instrumentos musicales electrónicos, computadoras y otros dispositivos relacionados se conecten y comuniquen entre sí.<sup>1</sup> Una simple conexión MIDI puede transmitir hasta dieciséis canales de información que pueden ser conectados a diferentes dispositivos cada uno.

El sistema MIDI lleva mensajes de eventos que especifican notación musical, tono y velocidad; señales de control para parámetros musicales como lo son la dinámica, el vibrato, paneo, cues y señales de reloj que establecen y sincronizan el tempo entre varios dispositivos. Estos mensajes son enviados mediante un cable MIDI a otros dispositivos que controlan la generación de sonidos u otras características. Estos datos también pueden ser grabados en un hardware o software llamado secuenciador, el cual permite editar la información y reproducirla posteriormente.

Un mensaje MIDI es una instrucción que controla algún aspecto del dispositivo receptor. Un mensaje MIDI consiste en un byte de estatus, el cual indica el tipo del mensaje, seguido de dos bytes de datos que contienen los parámetros.<sup>20</sup> Los mensajes MIDI pueden ser "mensajes de canal", los cuales son enviados a unos de los dieciséis canales y puede ser escuchado solo por los dispositivos en ese canal, o "mensajes de sistema", los cuales pueden ser escuchados por todos los dispositivos. Cualquier dato no relevante para un dispositivo receptor es ignorado.<sup>37</sup> Existen cinco tipos de mensajes: "Channel Voice", "Channel Mode", "System Common", "System Real-Time" y "System Exclusive". En la figura V podemos observar la conexión con la interfaz MIDI, la cual como se observa parte de la señal proveniente de UART enviada por el procesador que se pasa por un amplificador dc-dc en el cual la señal es elevada a los valores lógicos admitidos por MIDI, y se realiza la conexión a el

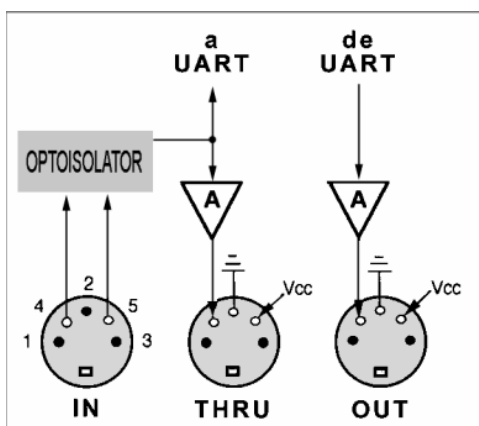


Figura 8. Esquema de conector MIDI

dispositivo, el cual tiene 5 pines, de los cuales solo se usan 3.

Con respecto al mensaje MIDI, este está compuesto por

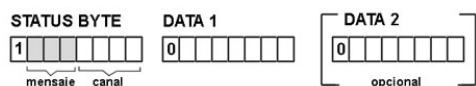


Figura 9. Mensaje MIDI

3 bytes, como se indica en la figura V en los cuales el primer byte indica el tipo de mensaje a ser enviado, y los siguientes dos son para transmitir los datos, por ejemplo, para la activación de una nota el primer byte indicaría que se quiere activar una nota, el segundo el número de la nota (mirar figura V), y el tercero su velocidad.

Para la transmisión del mensaje MIDI, se emplea de la forma

Note	-2	-1	0	1	2	3	4	5	6	7	8
C	0	12	24	36	48	60	72	84	96	108	120
C#	1	13	25	37	49	61	73	85	97	109	121
D	2	14	26	38	50	62	74	86	98	110	122
D#	3	15	27	39	51	63	75	87	99	111	123
E	4	16	28	40	52	64	76	88	100	112	124
F	5	17	29	41	53	65	77	89	101	113	125
F#	6	18	30	42	54	66	78	90	102	114	126
G	7	19	31	43	55	67	79	91	103	115	127
G#	8	20	32	44	56	68	80	92	104	116	---
A	9	21	33	45	57	69	81	93	105	117	---
A#	10	22	34	46	58	70	82	94	106	118	---
B	11	23	35	47	59	71	83	95	107	119	---

Figura 10. Tabla de notas protocolo MIDI

especificada en V:

Note On	Nota musical (m)	Velocidad
0x90	DO,RE,MI,FA,SOL,LA,SI	VEL_ON:0x40,VEL_OFF:0x00

Figura 11. Mensaje midi para activación de nota

Donde el valor 0x90 representa el mensaje de la activación de una nota en el canal 0. El siguiente Byte corresponde a la nota seleccionada por el guante, finalmente el último byte especifica la velocidad de ataque que indica que tan fuerte se está presionando una tecla. Por defecto este valor se coloca en 0x40, para desactivar una nota se envía un 0x00 en este byte.

Para el cambio de volumen se emplea el mensaje especificado en V.

Control Change	Tipo de control	Valor volumen
0xB0	0x07	vol=spl_getchar()

Figura 12. Mensaje midi para variación de volumen

El primer byte representa un mensaje de cambio de control, que permite interactuar de distintas maneras con la calidad del sonido. El segundo byte corresponde al tipo de control que se va a emplear, en este caso es un cambio de volumen. Finalmente el último byte corresponde al valor del volumen de la nota.