UNSAM
UNIVERSIDAD
NACIONAL DE
SAN MARTÍN

Universidad Nacional de San Martín

Escuela de Ciencia y Tecnología

Electrónica Digital II

Proyecto final:

Loopera para aplicaciones de audio

Autores: Fonseca Giraudo, Maximiliano Gabriel -

Müller, Nahuel Agustín

maxigfonseca@yahoo.com.ar

nahuelam.93@gmail.com.ar

- 2017

A continuación se detallará el desarrollo de una Loopera, implementada en EDU-CIAA con procesador LPC4337 Cortex M4. La misma consta de un rango de medición de 0 a 3.3 V, con una resolución de 10 bits, muestreados con una frecuencia de 80 KHz. Se desarrollará en lenguaje C, a nivel de programación de registros.

La funcionalidad del mismo es la de grabar cierta cantidad deseada de datos en frecuencias de audio (de 20 Hz a 20 Khz aproximadamente) en memoria, y repetir lo grabado en forma de loop, con posibilidad de grabar sobre grabado y borrado de memoria. La aplicación inmediata es la música, en donde se desee grabar una base rítmica en vivo, y tocar y regrabar sobre lo grabado.

1) Loopera.

El esquema consta de un microprocesador colgado en un while(1), que se encarga de responder a los flags; dos canales de DMA, encargados de recibir y enviar datos desde y hacia el ADC y DAC correspondientemente; un ADC que muestrea la señal a la frecuencia establecida, un DAC encargado de entregar la señal analógica a partir de una palabra digital, y una memoria en la que se volcará la información a guardar.

2) Entrada

El sampleo de la señal, en este caso audio, la realiza el conversor analógico-digital, a la frecuencia de 80 KHz, definida por configuración de un divisor de frecuencia del CLK principal. Es manejado por DMA, tomando las muestras y alojándolas en orden en 2 listas concatenadas, que apuntan a un buffer cada uno. El tamaño de los mismos es de 100 muestras de 32 bits. Cada vez que el DMA termina de llenar una lista, se efectúa una interrupción. Esta, interrumpe al microprocesador, el cual ahora se dedica a preguntar qué lista (buffer) está completa y asignarla al buffer de memoria.

3) Memoria

En la memoria se guardarán paquetes de 100 samples en 50 bloques, siendo 75 la máxima cantidad de bloques permitida por falta de memoria. Aquí es donde, dependiendo del flag “RECORDING” activado por la interrupción de TEC_1, se copiara el buffer de entrada.

4) Reproducción:

El segundo canal de DMA toma los datos sumados y los envía al conversor digital-analógico. Se toman los datos desde la memoria y se suman con los que entrega el otro canal de DMA que previamente alojó desde el ADC en buffer. El resultado se deposita en uno de los buffers del segundo canal de DMA que no esté ocupado, indicado desde el flag DAC_LLI0_ATR. Este flag se lee en la interrupción del DMA, que salta cuando una lista fue escrita completamente. Cada vez que una suma es completada, un contador de bloques se incrementa. Este se reinicia en caso de que se llegue al final de la memoria. Es así como se efectúa el loop.

5) Borrado:

El borrado de la memoria se produce cuando una interrupción GPIO es activada por TEC_2. En ella, un flag “ERASE_MEM” se activa, produciendo que el main ponga ceros recorriendo toda la memoria.

6) Conclusiones:

En esta loopera se utilizan dos canales de DMA con dos buffer cada uno. Con estos canales de DMA el uP se independiza de tener que ocuparse de recibir o enviar datos a los conversores. El doble buffer es necesario para que el uP escriba o lea (dependiendo el conversor) un vector el cual el DMA no esté ocupando. Las interrupciones de DMA, TEC_1, o TEC_2 solo ocupan al uP para activar flags, que tendrán efecto solo en el main, optimizando al rendimiento del uP.