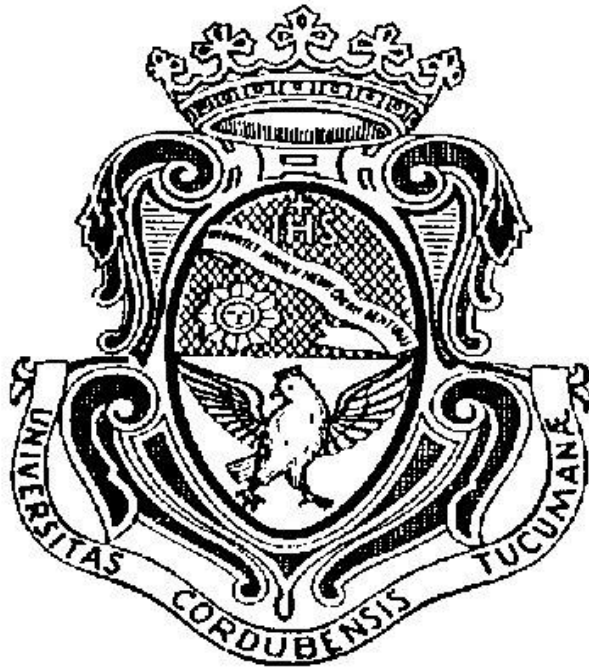


Universidad Nacional de Córdoba

Facultad de Cs. Exactas, Físicas y Naturales



Electrónica Digital III Detector de alcohol en el aire

Trabajo Práctico Integrador

Grupo: Bombinhas

Profesor:

- Ing. Fernando Gallardo

Alumnos:

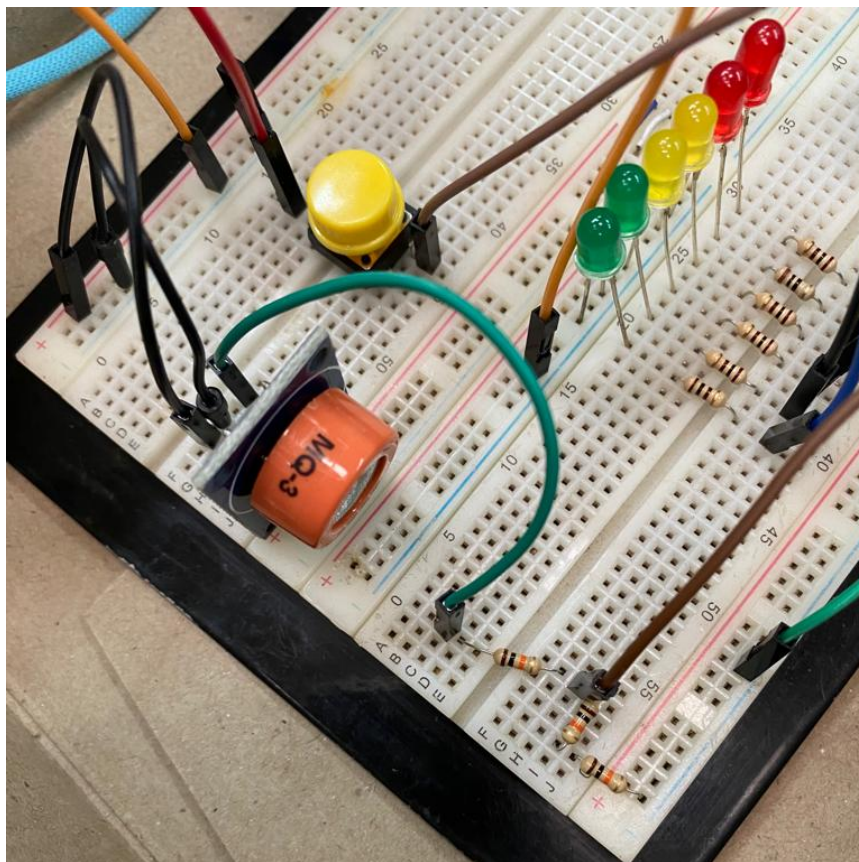
- | | |
|--------------------|----------|
| - Fausto Lavezzari | 42344364 |
| - Franco Viotti | 42051491 |

INTRODUCCIÓN

El siguiente trabajo busca la utilización de módulos periféricos de la placa de desarrollo LPC1769 para realizar un detector de alcohol en el aire y representar las mediciones por medio de mensajes vía comunicación UART, LED's y una alarma.

Principio de funcionamiento:

- Se utilizará el módulo detector de alcohol MQ-3 conectado a una entrada del ADC de la placa. El datasheet se puede encontrar en: [Datasheet - MQ-3](#).
- La placa realizará la conversión pertinente y establecerá el valor obtenido dentro de alguno de los 6 niveles predefinidos.
- Mediante GPIO se refleja estos niveles en una tira de luces led de colores verde, amarillo, y rojo, 2 de cada uno, ubicados en orden creciente segun la medición.
- A través del uso de DMA y el módulo DAC generamos una señal senoidal a distintas frecuencias también dependiendo del nivel, a mayor sea la medición, mayor será la frecuencia. Esta señal la enviamos a un pequeño buffer a modo de alarma.
- Esta “barra de crecimiento” generada también es enviada por transmisión serie de forma continua usando DMA



DESARROLLO

El MQ3:

El sensor MQ-3 mide la concentración de alcohol en el aire y lo traduce a voltios. Inicialmente, tiene un valor “piso” que es cuando no existe alcohol en el aire. Ese valor se utilizará más adelante para calibrar las mediciones y compararlas con el inicial. Para un correcto funcionamiento del sensor, éste debe ser calentado por unas 24 hs (aproximadamente) si no ha sido utilizado recientemente. En caso contrario, con unos minutos de calentamiento sería suficiente. Dada esta naturaleza variable del sensor es que es necesario medir de forma reiterada el valor de piso.

El rango de trabajo del sensor es de 0 a 5V, lo cual puede generar un problema a la hora de trabajar con el LPC1769 ya que la entrada analógica va a medir solamente hasta 3,3V en su configuración por defecto. Para paliar esta problemática aplicamos simplemente un divisor resistivo a la entrada para ajustar el rango

CALIBRACIÓN:

Para comenzar el funcionamiento del sistema, configuramos una entrada de la placa como interrupción externa para que, al interrumpir, pueda tomar diez valores iniciales a través del ADC y promediarlos para luego ser utilizados como referencia. A partir de allí, se esperará que los valores medidos sean mayores al aumentar la concentración de alcohol en el aire.

El pin configurado fue el P2.10, para interrumpir en flancos descendentes. Dentro del manejo de la interrupción, se establece en “1” una flag de calibración que indica al manejo de interrupción del ADC que las mediciones tomadas deben ser para el valor de referencia.

También se inicia el timer utilizado para disparar al ADC y se configuran los parámetros de inicio de conversión de éste último.

ADC:

El módulo adc será utilizado mediante conversión por flanco descendente del match 1 del timer 0. Aún así el tiempo de conversión seteado va a ser el máximo (200k) para obtener una medición precisa. Al no necesitar tantos datos simplemente el módulo quedará esperando la mayor parte del tiempo a que se le ordene convertir nuevamente.

Una vez terminada la conversión se va a ejecutar una interrupción, en su handler se pueden hacer 2 cosas. Como antes mencionamos una es obtener las mediciones para la calibración, la otra es la medición normal a realizar. En el segundo caso se obtendrá el valor de la muestra y se agrega a un buffer circular de 10 elementos representado por un array. Cuando se llegan a las 10 conversiones consecutivas se hace un cálculo del promedio de los valores del buffer y se pasa a una función para asignar el nivel correspondiente al que pertenece.

DISPARO DE CONVERSIÓN:

Las conversiones son manejadas a través del timer 0. Se busca que el ADC convierta diez veces por segundo, es decir, cada 100 milisegundos. Para ello, el timer realizará un toggle en el pin P1.29 cada 50 milisegundos, lo que genera un flanco de bajada cada 100ms que iniciará la conversión del ADC. Al alcanzar el valor de match, el timer se reseteará, mas no interrumpirá ni se detendrá ya que no es necesario.

ASIGNACIÓN DE NIVELES:

Cada vez se realizan diez conversiones consecutivas y se promedian, se llama a la función que asigna el nivel correspondiente a las mismas. La función recibe como parámetro la diferencia entre el promedio de conversiones y el valor de referencia obtenido durante la calibración. Luego, va revisando cada intervalo y accediendo a estos si el valor de la diferencia es mayor que el valor pre establecido como límite. De ser así, irá encendiendo el LED correspondiente en cada intervalo, completa un arreglo donde se almacenará el mensaje luego enviado por UART y activa la alarma con su correspondiente frecuencia en los intervalos impares. Al final, el arreglo de mensaje almacena un salto de línea en su última posición: '\n'.

DAC:

Utilizaremos el módulo DAC para generar la señal a enviar al buffer de alarma. Antes que nada es necesario tener en algún lugar de memoria almacenada la forma de onda a enviar, en nuestro caso un seno, guardado en un array de 60 elementos definidos para utilizar la máxima excursión del conversor. La señal a transmitir será siempre la misma, simplemente variamos la frecuencia.

Para enviar los datos utilizaremos DMA en modo memoria a periférico. Utilizando el sistema de linked list, enlazamos continuamente una única sección de memoria, así se envía la señal constantemente. Para variar la frecuencia nos valemos del TimeOut (tiempo que tarda en solicitar un nuevo dato el DAC al DMA) con una sencilla fórmula podemos ajustar la frecuencia deseada.

$$time_out = 25000000/(frecuencia_deseada*60);$$

Variamos este time_out mediante una función llamada en la en función de asignación de niveles.

UART:

Utilizamos la transmisión UART para poder comunicarnos con el exterior. Como dijimos previamente, se enviará la barra de progreso creada en la asignación de niveles. Utilizando un canal de DMA accedemos continuamente a la posición de memoria en la que está almacenado el mensaje a transmitir y lo envía continuamente.

Este mensaje lo va a recibir una placa ARDUINO la cual cumple simplemente la función de interfaz con la computadora para poder ver los mensajes mediante el monitor serie.

DMA:

El DMA es utilizado en dos aplicaciones:

1. Para el funcionamiento del DAC: Dentro de la configuración, se crea una lista donde se define la dirección de memoria de la señal a transmitir y la dirección de destino, en este caso el registro DACR, que almacena el próximo valor a ser convertido por el DAC. Esta lista se autoreferencia para que la transmisión sea continua sobre el mismo arreglo de datos. Se utiliza el canal cero del DMA, con un tamaño de transferencia de 60 datos de memoria a periférico.
2. Para el funcionamiento de la transmisión vía UART: En este caso, también se crea una lista con fuente en la dirección de memoria del mensaje a enviar y destino el registro THR (transmit holding register, donde se almacena el siguiente carácter a ser transmitido). El tipo de transferencia será también de memoria a periférico con un tamaño de transferencia que dependerá del tamaño del mensaje a enviar. El canal a utilizar será el número uno.

CONCLUSIÓN:

Logramos hacer funcionar el proyecto planteado, si bien presentamos dificultades, muchas supimos resolverlas para poder avanzar, algunas quedaron en el camino pero aún así estamos contentos con el trabajo realizado.

Algunos de los inconvenientes que tuvimos fueron:

- Encontrar una función fiable de conversión de valores de voltaje medidos a unidades de concentración en el aire. El datasheet del sensor no es el más completo y la información disponible en internet puede ser confusa.
- Problemas en la configuración de los canales del módulo DMA al encenderlo y apagarlo para detener transmisiones.
- Enviar mensajes individuales controlados mediante UART por el mismo problema de DMA.
- Drenajes de corriente tanto al sensor como a la placa por mala colocación de resistencias o resistencias con valores imprecisos (distintos al medirlos de los que deberían ser).
- Mal funcionamiento de una de las placas obtenidas del pañol.
- Entre otros.

Pese a esto logramos terminar el trabajo de forma satisfactoria. Fue muy útil para pasar más de la teoría a la práctica, ya que hay un abismo de distancia entre ellos. No es lo mismo hacer que los códigos simplemente compilen o directamente programar en papel, a que de verdad funcione en una placa y se comporte como debe. Además, adquirimos experiencia con el funcionamiento de la placa y las librerías de CMSIS.