Cuellos de botella en sistemas embebidos en la adquisición y transmisión de señales analógicas

Luisina Gómez, Rodrigo González, María José Millán, y Julián Oreggioni,
Universidad de la República, Montevideo, Uruguay
Email: {luisina.gomez, rodrigo.gonzalez.suarez, maria.millan, juliano}@fing.edu.uy

***Resumen*—Los microcontroladores estándar de rango medio especifican la disponibilidad de varios canales analógicos y altas frecuencias de muestreo de sus conversores analógico-digitales. Este trabajo explora de forma preliminar qué tan factible es aprovechar esas altas tasas de muestreo, identificando cuellos de botella, y eventualmente proponiendo soluciones. Para ello, se desarrolló el prototipo de un sistema embebido, basado en el microcontrolador MSP430G2553, que adquiere señales analógicas, las procesa y las transmite a una computadora personal mediante comunicación serial, buscando operar a la máxima capacidad posible.**

I. Introducción

Los microcontroladores estándar de rango medio, como el ATmega328PB de Microchip o el MSP430 de Texas Instruments, especifican la disponibilidad de varios canales analógicos (hasta ocho en los mencionados microcontroladores) y altas frecuencias de muestreo de sus conversores analógico-digitales (ADC), por ejemplo 80 ksps (kilo muestras por segundo) en el caso de ATmega328PB, o 200 ksps en el MSP430. Este trabajo explora qué tan factible es aprovechar esas altas tasas de muestreo, identificando cuellos de botella, y eventualmente proponiendo soluciones. Para ello, se desarrolló el prototipo de un sistema embebido, basado en el microcontrolador MSP430G2553, que adquiere señales analógicas, las procesa y las transmite a una computadora personal (PC) mediante comunicación serial, buscando operar a la máxima capacidad posible. Asimismo, se implementó en Python en el PC una interfaz gráfica de usuario (GUI) que permite recibir los datos en el PC, visualizarlos y configurar ciertas características de la adquisición, tales como el número de canales a adquirir y la frecuencia de muestreo de la señal.

II. Descripción del sistema

En la Fig. 1 se muestra el diagrama funcional y de conexionado del sistema, donde las señales adquiridas provienen del generador de señales Analog Discovery 2 de Digilent.

Se utiliza la placa de desarrollo MSP-EXP430G2ET (Launchpad) del microcontrolador MSP430G2553. Se trata de un microcontrolador de 16 bits, que cuenta con un ADC de 10 bits y tasa de muestreo de hasta 200 ksps, multiplexado en su entrada para admitir hasta 8 canales. Posee además, memoria flash de 16 kB y RAM de 512 bytes, varios modos de operación de bajo consumo, e interfaz de comunicación universal (USCI) de 2 Mbps de tasa máxima de comunicación por UART (Universal Asynchronous Receiver-Transmitter). Si

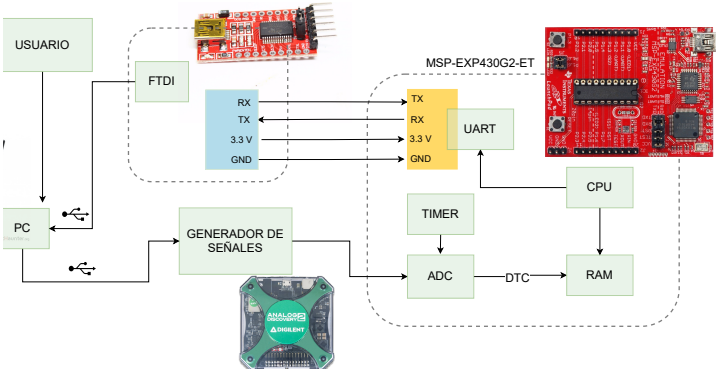

Figura 1. Diagrama funcional y de conexionado.

bien el microcontrolador tiene ocho canales analógicos, en el sistema implementado la cantidad de canales disponibles se reduce a seis, puesto que la UART comparte dos pines con el ADC. El ADC incluye además un "Data Transfer Controller" (DTC), el cual permite la transferencia de los datos digitalizados desde el ADC a un lugar de RAM predefinido, sin la intervención del microprocesador. Dichas transferencias son realizadas en bloques de largo configurable. Esto permite que el microprocesador realice otras tareas en paralelo mientras se realiza la transferencia aumentando la capacidad de adquisición de datos.

Además de proveer un mecanismo para depurar y programar, el Launchpad ofrece un sistema para medir el consumo de energía (Energy Trace) y un adaptador UART/USB cuya velocidad máxima es 115 kbps (ver Tabla 4 en pág. 19 de [1]). Este adaptador constituye por tanto el primer cuello de botella que se solucionó fácilmente incorporando un adaptador UART/USB externo que permite transmitir a una frecuencia máxima de 3 Mbps.

El software se implementó en lenguaje C utilizando el Code Composer Studio (CCS) de Texas Instruments como IDE. Se utiliza *Round-Robin con interrupciones* como arquitectura de software embebido [2]. En la Fig. 2 se presenta un diagrama de tres capas con los distintos módulos de software implementados.

main.c implementa el bucle principal del Round-Robin con interrupciones (ver Fig. 3). *flag_adc* se enciende en la ISR del ADC cuando se finaliza la adquisición de un nuevo bloque de datos. Se apaga en el main, luego de procesar y transmitir el último dato del bloque. *flag_cmd* se enciende en la ISR de recepción de la UART cuando se recibió del PC un

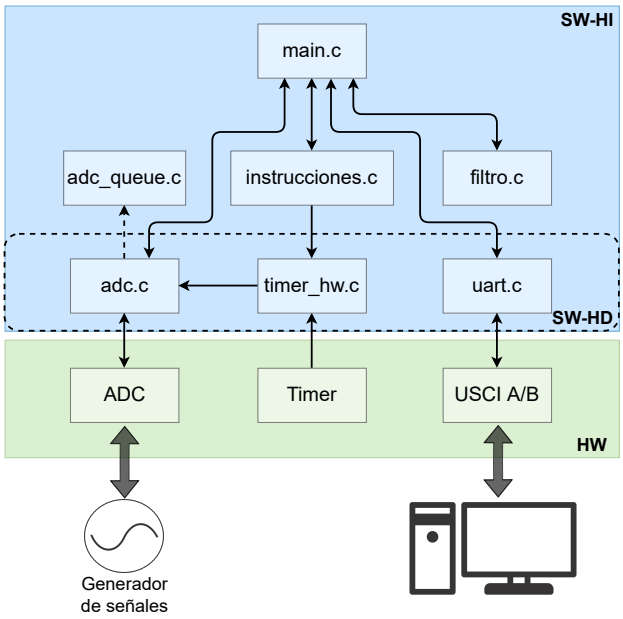

Figura 2. Diagrama de módulos. Las flechas continuas indican la comunicación entre módulos. La flecha punteada señala la transferencia de datos adquiridos por el ADC hacia una sección de memoria específica (DTC).

comando completo. Se apaga en el *main* luego de decodificar el comando recibido por UART.

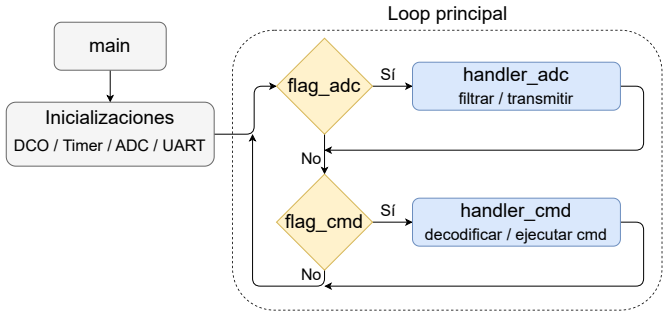

Figura 3. Diagrama de flujo del módulo principal (main.c).

En **adc_queue.c** se definen las funciones de acceso a la cola datos adquiridos mediante el ADC.

El módulo **instrucciones.c** cuenta con la función *get_cmd* que permite decodificar el mensaje obtenido por UART, interpretando el comando y tomando la acción correspondiente según el caso. Los comandos disponibles son: inicio y fin de adquisición, configuración de frecuencia de muestreo (f_s) y baudrate (ver Tabla I) y selección de cantidad de canales de adquisición (en la actual versión se permite adquirir uno o dos).

En el módulo **adc.c**, entre otras cosas, se configura para que el ADC use su oscilador interno (ADC10OSC), que tiene una frecuencia de 5 MHz, para que trabaje en modo *un canal - una conversión*, y para que los resultados de la conversión se guarden directamente en memoria RAM, en bloques de N datos, usando Data Transfer Controller (DTC).

Se implementaron cuatro rutinas de atención a interrupciones ISR: una para el Timer, una para el ADC y dos para la UART (transmisión y recepción).

Tabla I
VALORES QUE PUEDE TOMAR LA FRECUENCIA DE MUESTREO f_s.

Parámetro	baudrate (bps)	f_s $(ksps)$
0	9600	0.48
1	115200	5.7
2	256000	12.8
3	460800	22.8
4	921600	44

Las interrupciones del Timer (*isr_timer*) son periódicas y fijan la frecuencia de muestreo, ya que en su ISR se inicia la conversión del ADC. Cuando se trabaja con más de un canal de adquisición, previo a iniciar la conversión, se realiza la conmutación del canal. La ISR del ADC (*isr_adc*) se dispara cuando se finaliza la transferencia a memoria de un bloque completo de N datos. Dentro de esta ISR simplemente se enciende la bandera *flag_adc* cuyo estado es verificado en el main. Luego de adquirirse un bloque entero de datos, desde el main se inicia la transmisión. Se guardan los datos en la cola de transmisión de la UART y se habilitan las interrupciones de transmisión. En la ISR (*isr_tx*) se copia el dato en el buffer de transmisión y se actualizan los punteros de la cola. Cuando la cola queda vacía se deshabilitan las interrupciones. Las interrupciones de recepción se disparan cuando llega un comando por UART. En esta ISR (*isr_rx*) se obtiene el dato del buffer de recepción y se guarda en la cola de recepción de la UART. Finalmente se enciende la bandera *flag_cmd* cuyo estado es verificado en el main.

III. ANÁLISIS DE LOS CUELLOS DE BOTELLA

III-A. Reloj del sistema

El objetivo de muestrear a la máxima frecuencia posible implica utilizar un reloj que sea capaz de hacer funcionar el sistema en estas condiciones. El microcontrolador tiene integrado un Digitally-Controlled Oscillator (DCO) cuya frecuencia de trabajo puede variar en un rango amplio dentro de ciertos valores discretos. Para determinar con precisión las frecuencias a las que trabaja el DCO se observó en un osciloscopio la señal generada por éste en dos placas diferentes usando dos configuraciones diferentes. En la Tabla II se presentan los mayores valores obtenidos.

Tabla II
MEDIDA DE LA FRECUENCIA REAL DEL DCO EN 2 PLACAS DIFERENTES USANDO RSELX = 15 Y MODX = 0.

Placa	Frecuencia (DCOx = 3)	Frecuencia (DCOx = 7)
1	13.7 MHz	19 MHz
2	13.4 MHz	18 MHz
Valor esperado [3]	12-18.5 MHz	16-26 MHz

Ya que el reloj principal del microprocesador no puede trabajar a más de 16 MHz, se configura el DCO para que trabaje a f_{DCO} = 13.7 MHz, quedando determinada en ese valor la frecuencia del MCLK y el SMCLK.

III-B. Velocidad del ADC

La frecuencia máxima de adquisición del ADC se midió conmutando una señal digital de salida en cada interrupción/adquisición del ADC. Se obtuvo una señal cuadrada de frecuencia 104 kHz por lo que se concluye que la frecuencia de muestreo es 208 kHz (ver Fig. 4), valor que supera los 200 ksps especificados en la hoja de datos.

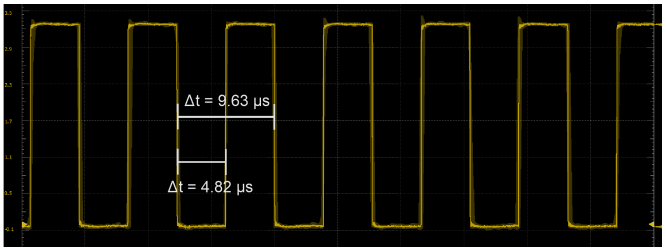

Figura 4. Máxima velocidad de adquisición del ADC

III-C. Procesamiento de los datos

El *tiempo de procesamiento* t_{proc} se define como el tiempo entre que se dispara la adquisición de una muestra y ésta se guarda en el buffer de transmisión, e incluye cualquier procesamiento que se haga con los datos. Asimismo, se considera el mínimo procesamiento posible, que consiste solamente en guardar el dato adquirido en la cola de transmisión. El ADC con DTC interrumpe cuando finaliza la adquisición de un bloque. La medida del tiempo de procesamiento arrojó $t_{proc} = 8{,}1\ \mu s$ en la primer muestra del bloque y $t_{proc} = 7{,}15\ \mu s$ para cada una de las restantes, pues en estas últimas no se ingresa a la ISR del ADC. El bloque utilizado fue de tamaño 50 muestras. En la Ec. (1) se presenta el cálculo del tiempo requerido para el procesamiento de N = 50 muestras con DTC.

$$t_{proc}^{conDTC} = 49 \times 7{,}15 + 8{,}1 = 358{,}45\ \mu s \tag{1}$$

De esta manera, se obtiene que la condición que debe cumplir la frecuencia de muestreo está dada por la Ec. 2.

$$f_s < \frac{N}{t_{proc}^{conDTC}} \Rightarrow f_s < 123{,}4\ kHz \tag{2}$$

III-D. Transmisión por UART

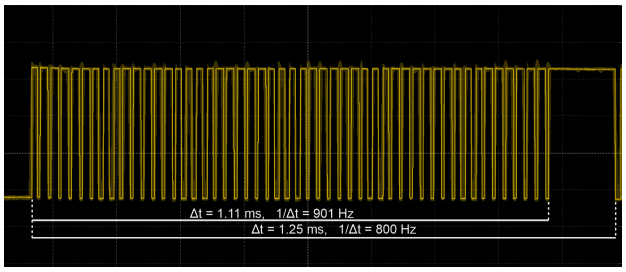

Figura 5. Tiempos de transmisión a una tasa de 921.6 kbps

Se configura la UART para enviar datos a una tasa de 921.6 kbps (se intentó usar valores más altos pero no se logró que funcionara). En la Fig. 5 se muestran las medidas realizadas. Se registra un tiempo de transmisión de 22.7 μs y 1.11 ms para las 50 muestras, se concluye por tanto que la máxima frecuencia de muestreo para una tasa de 921.6 kbps en trasmisión es:

$$f_s^{max} = 44\ kHz$$

IV. Discusión y conclusiones

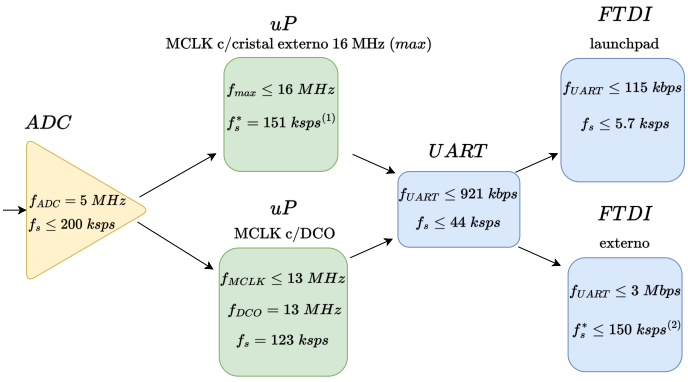

Figura 6. Máxima f_s alcanzada según cada etapa. (1) y (2) son estimaciones

El resumen de las restricciones se condensa en la Fig. 6. Se verificó que el ADC es capaz de adquirir 200 ksps. El procesamiento mínimo (encolar el dato adquirido para transmitir) reduce a 123 ksps la frecuencia de muestreo usando el DCO. Una posible solución a la variablilidad del DCO entre placas sería usar un cristal de 16 MHz, esto además permitiría llevar la velocidad de reloj interno del microprocesador a su límite, aumentando el tiempo disponible para procesamiento y por tanto la máxima frecuencia de muestreo a 151 ksps. El primer cuello de botella detectado fue el adaptador USB/UART del Launchpad que solo permite muestrear a 5,7 ksps. Eso se solucionó agregando un adaptador externo, que permite llevar la frecuencia de muestreo a 150 ksps. Finalmente, el cuello de botella es el periférico UART del microcontrolador, que limita la frecuencia de muestreo a 44 ksps. A futuro nos proponemos sustituir el AD2 por el Front-End analógico desarrollado en nuestro laboratorio basado en [4], y realizar pruebas en personas usando el sistema desarrollado.

Agradecimientos

Este trabajo fue financiado por CSIC (Comisión Sectorial de Investigación Científica, Uruguay) y Erasmus+ Project NEON, 618942-EPP-1-2020-1-AT-EPPKA2-CBHE-JP.

Referencias

[1] *MSP Debuggers User's Guide*, Texas Instrument, 2015. [Online]. Available: https://www.ti.com/lit/ug/slau647o/slau647o.pdf
[2] D. E. Simon, *An embedded software primer*. Addison-Wesley Professional, 1999, vol. 1.
[3] *Mixed signal microcontroller*, Texas Instrument, 2013. [Online]. Available: https://www.ti.com/lit/ds/symlink/msp430g2553.pdf
[4] J. Oreggioni, A. Caputi, and F. Silveira, "Current-efficient preamplifier architecture for cmrr sensitive neural recording applications," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 12, no. 3, pp. 689–699, jun 2018.