RIELAC, Vol. XXXIX 3/2018 p. 68-76 Septiembre – Diciembre ISSN: 1815-59

RIELAC
Revista de Ingeniería Electrónica, Automática y Comunicaciones

Adquisición de datos analógicos con alta precisión usando una Computadora de Placa Única

Fidel Alejandro Rodríguez Corbo, Arturo Hernández González, Jorge Ramírez Beltrán

RESUMEN / *ABSTRACT*

La utilización de las computadoras de placa única en el campo de la adquisición continua de datos analógicos conduce a la reducción de tiempos de fabricación, fácil implementación y a sistemas más compactos y portables. Como desventajas tienen las pérdidas de datos y errores temporales entre muestras. Sistemas Operativos de Tiempo Real y componentes externos al sistema pueden reducir estos errores, pero tienden a complejizar el proyecto y aumentar su costo. Este trabajo se propone la implementación de un sistema de alta precisión para la adquisición continua de valores analógicos utilizando el BeagleBone Black. La gestión de la frecuencia de muestreo del convertidor analógico digital interno se realiza a través del Subsistema Programable de Tiempo Real, también al interior del BeagleBone Black, garantizando la reducción de componentes externos y una alta precisión temporal. En las pruebas prácticas realizadas se obtuvo un error de 38 ppm de muestras al usar el Subsistema Programable de Tiempo Real mejorando las pérdidas de datos que se producen al gestionar la frecuencia de muestreo desde un temporizador o desde software.

Palabras claves: BeagleBone Black, sistema de adquisición de datos, computadora de placa única

The use of single-board computers in the field of continuous analog data acquisition leads to the reduction of manufacturing times, easy implementation and more compact and portable systems. Disadvantages include data loss and temporary errors between samples. Real-time Operating Systems and components external to the system can reduce these errors, but tend to make the project more complex and increase its cost. This work proposes the implementation of a high precision system for continuous analog data acquisition using the BeagleBone Black. The Programmable Real-Time Subsystem, also inside the BeagleBone Black, guarantees the reduction of external components and a high temporal precision, by imposing the sampling frequency of the internal digital to analog converter. In the practical tests carried out, an error of 38 ppm of samples was obtained when using the Programmable Real-Time Subsystem, improving the data losses that occur when managing the sampling frequency from a timer or from software.

Key words: BeagleBone Black, data acquisition system, single board computer

High precision system for analog data acquisition using a Single Board Computer

1.- INTRODUCCIÓN

Una computadora de placa única (SBC por sus siglas en inglés) es una computadora completamente funcional construida en una placa reducida, con microprocesador, memoria, puertos de entrada-salida y otras funcionalidades adicionales para aplicaciones de propósito general. En la mayoría de los casos estas funcionalidades están incluidas en un chip monolítico llamado Sistema en un Chip (SoC por sus siglas en inglés). Aunque la aparición de las SBC puede remontarse hasta hace más de 30 años, solo hace unos pocos años que se amplió su uso en el campo de la ingeniería eléctrica y automática con el surgimiento de modelos de baja potencia, costo reducido y bien soportados por la comunidad de desarrolladores como la Raspberry Pi y la BeagleBone Black. Este avance tecnológico abrió una puerta para las aplicaciones científicas y en especial al campo de la adquisición continua de datos gracias a su pequeño tamaño, bajo consumo de potencia y facilidades de desarrollo de aplicaciones en corto tiempo [1].

Recibido: 3/5/2018 Aceptado: 20/7/2018

Proyectos académicos que van desde aplicaciones para estudiantes a complejos prototipos en múltiples campos vieron en estas plataformas un lugar donde desarrollarse. La utilización de las SBC en el diseño e implementación de sistemas de adquisición continua facilita la replicación a la vez que disminuye el tiempo de fabricación. Su reducido tamaño se debe principalmente a la utilización de los SoC como núcleo central de la SBC, siendo también su principal característica descriptiva y fuente de sus potencialidades.

Otra de las ventajas de la utilización de las SBC es la gran variedad de lenguajes de programación que son capaces de utilizar. Sus potencialidades para ejecutar un sistema operativo permiten implementaciones en los lenguajes más populares [2]. Uno de los lenguajes más utilizados entre los desarrolladores es el Python, un lenguaje interpretado multiplataforma con gran soporte en las comunidades virtuales. Este lenguaje agiliza la implementación del código gracias a su sintaxis limpia, facilitando el trabajo de desarrolladores no avezados en la programación. La combinación de las SBC como plataforma de desarrollo hardware para la adquisición de datos y el Python como lenguaje de programación suponen un paso importante hacia la creación de sistemas de rápida y fácil implementación.

En la actualidad coexisten un gran número de SBC, en algunos casos con marcadas diferencian entre ellas pero conservando en la mayoría su arquitectura alrededor del SoC. Un proyecto interesante basado en estas plataformas para la adquisición continua de datos utiliza la SBC TS-7250-V2. En este trabajo se enfatizan las ventajas de utilizar una SBC para la creación de registradores de datos, las facilidades de programación de estas plataformas y sus características de bajo consumo de potencia. Aunque se desarrolla un sistema capaz de adquirir y registrar datos desde 20 instrumentos, es de señalar que el costo del prototipo asciende a 1000 dólares, relacionado principalmente al alto precio de esta SBC en comparación con otras disponibles en el mercado [3].

A pesar de la presencia de varias SBC en la actualidad, en el campo de la adquisición continua de datos, la Raspberry Pi y la BeagleBone Black (BBB) suponen las principales exponentes, esencialmente debido a la relación costo-potencialidades así como el soporte por parte de la comunidad de desarrolladores. Proyectos interesantes basados en la adquisición continua de datos analógicos tienen como núcleo central la Raspberry Pi [4-7]. Trabajos realizados para la adquisición de variables acústicas en el fondo marino emplean la Raspberry Pi como núcleo de su plataforma [4]. En esta investigación se utilizan los datos capturados desde un hidrófono hacia la Raspberry Pi y son guardados en la µSD. Una de las principales desventajas del uso del Raspberry Pi para estas funciones es que no trae integrado convertidor analógico-digital, y tampoco un reloj de tiempo real, por lo que para su utilización como registrador de datos analógicos estos periféricos deben ser agregados, aumentando la complejidad del sistema y constituyendo otra fuente de consumo de potencia.

En otro proyecto basado en Raspberry Pi se implementa un registrador de datos para la adquisición de variables en vehículos potenciados por humanos. Nuevamente la utilización de componentes externos a la plataforma es necesario en orden de adquirir las variables analógicas transmitidas por los sensores. Un convertidor analógico-digital fabricado por Microchip Technology MCP3208 es utilizado en conjunto con un reloj de tiempo real externo. Es señalado en esta investigación que las operaciones de escritura y lectura en la memoria µSD pueden detener el lazo de adquisición por un tiempo inaceptable de hasta 600 ms. En este trabajo se evalúa el sistema para un proceso de adquisición durante 10 minutos a un periodo de muestreo de 2 ms utilizando la distribución original del kernel Linux, reportando 990 intervalos mayores a los 2 ms del periodo entre muestras, para una afectación de 3493 ppm de intervalos erróneos. Aunque este problema es evitado utilizando un kernel de tiempo real y escribiendo los valores registrados en la memoria RAM, es reportado por esta investigación que el Raspberry Pi en la configuración utilizada no soporta frecuencias de muestreo por encima de los 500 Hz utilizando los 8 canales simultáneamente [5].

En la automatización de una estación meteorológica Salcedo y Cendrós usan una Raspberry Pi asociada a sensores digitales. En este trabajo se enfatiza la limitación de esta SBC para ser acoplada a sensores analógicos pues impondría el uso de un convertidor analógico-digital externo. Respecto a la frecuencia de muestreo, no se aborda este tema por lo que se supone que los requisitos de la aplicación no son exigentes en este sentido [6]. Una alternativa a la carencia de convertidor analógico-digital se usa en el trabajo de Nikhade donde la adquisición de los datos analógicos la realiza un microcontrolador y este le transmite los datos a la Raspberry Pi por vía inalámbrica. En este caso se usa la SBC en un sistema de adquisición de datos como concentrador de varios nodos de sensores [7].

La BBB, más enfocada en las aplicaciones hardware, contiene 7 entradas analógicas de hasta 200000 muestras/s y un RTC interno [8], convirtiéndola en un sistema más compacto y sencillo para los sistemas de adquisición continua [9-10]. En una investigación conducida hacia el registro de vibraciones mecánicas por principio óptico se implementa la adquisición analógica de la BBB y se usa el lenguaje de programación Python. El sistema desarrollado se destaca por su sencillo diseño y reducción de componentes externos. Aunque no se especifica la frecuencia máxima de adquisición, sin dudas es muy baja pues se gobierna la misma desde el software Python y la salida de los datos es por la consola de Linux [9].

Un proyecto más elaborado plantea la utilización de la BBB en conjunto con una tarjeta de expansión desarrollada por los autores para la adquisición de variables analógicas. En este trabajo se especifica la necesidad de alcanzar altas frecuencias

de muestreo. En la tarjeta de expansión desarrollada se implementan otros 3 convertidores analógico-digitales que trabajarán en conjunto con el convertidor interno de la BBB. En la elección de la BBB ante un microcontrolador seleccionan la primera por las ventajas de tener un sistema operativo Linux completamente funcional. Aunque se especifica como máxima frecuencia de muestreo de la aplicación 1000 muestras/s, los autores reconocen que esta limitación está dada por la sobrecarga de los procesos de usuario en Linux. Esta frecuencia de 1000 muestras/s no es evaluada en el trabajo y los resultados prácticos que se reportan son tomados a una muestra por segundo [2].

En la bibliografía consultada se observa que las SBC presentan deficiencias frente a la estabilidad del disparo de conversión debido a que los sistemas operativos nativos no tienen límites de tiempo para atender los procesos. Esto genera espaciamiento temporal erróneo entre muestras y muchas veces pérdidas de las mismas. La utilización de un sistema de tiempo real (RTOS por sus siglas en inglés) podría disminuir estos errores, pero su instalación en estas plataformas no es sencilla y para usuarios desconocedores implicaría una dificultad extra. Otra deficiencia común a las SBC cuando son programadas en Python es la generación de disparos de conversión utilizando las funciones de la librería Time como lazo de espera. Al ser el Python un lenguaje interpretado y existir varias capas de abstracción hacia el hardware, la función time.sleep() utilizada para este fin no tiene una duración precisa ni estable. Una alternativa a esta función seria la utilización de los temporizadores contenidos en el SoC pero los tiempos de adquisición todavía estarían a cargo del Python, disminuyendo la precisión de estos.

Es recurrente en la bibliografía consultada, la no conciliación de frecuencias altas de muestreo con sistemas de gestión a alto nivel. Esto es debido a que una atención continua al sistema de adquisición puede verse afectado por las demoras de gestión internas o implementadas de un sistema operativo no determinístico. Aunque en algunos casos esta deficiencia es disminuida con la utilización de sistemas operativos de tiempo real, este problema aun constituye una limitante para alcanzar altas frecuencias de muestreo. Este problema resulta en una reducción de la confiabilidad de las SBC en las tareas de adquisición analógica de datos con alta precisión.

Una característica distintiva de la BBB es la inclusión en el SoC de un subsistema llamado Programable Real-Time Unit Subsystem (PRUSS) basado en dos procesadores RISC de 32 bits independientes al procesador central y con acceso a todos los periféricos del SoC [11]. La utilización de estos procesadores para establecer la frecuencia de muestreo y atención al convertidor analógico-digital permitirían a la BBB adquirir señales analógicas a una frecuencia constante y evitar pérdida de datos. En este artículo se presenta una solución para disciplinar el convertidor analógico-digital en la BBB utilizando el PRUSS y obtener un sistema de adquisición de señales analógicas de alta precisión gestionado desde una aplicación Python. Una caracterización rigurosa del sistema de muestreo permite validar a la BBB como un sistema de adquisición de datos analógicos de alta precisión y confiablidad, a la vez de permitir la gestión desde un lenguaje de alto nivel como el Python y liberar al procesador ARM de las principales cargas asociadas a la adquisición de los datos.

2.- BEAGLEBONE BLACK

La BBB pertenece a la gama de las BeagleBoards, hasta la actualidad la más vendida y popular de la serie. Catalogada como una SBC fuertemente enfocada al desarrollo de hardware, sus principales características son: procesador ARM Sitara AM3358 1GHz, 512MB de RAM, Ethernet, SPI, I2C, 69 GPIO, 4 temporizadores, 7 entradas analógicas y otras de uso más específico [12].

El convertidor analógico-digital (ADC) o el Touchscreen Controller como es identificado en el Manual de Referencia Técnica del AM335x, es un ADC de 12 bits y 8 canales de propósito general con soporte opcional para paneles táctiles resistivos [8]. De los 8 canales analógicos en el procesador, solo 7 se encuentran accesibles a través de los puertos de expansión de la BBB ubicados en el puerto P9 mostrado en la figura 1. En esta figura se muestran encerrados en rojo los pines relacionados con el ADC. Como el rango de estas entradas analógicas es de 0 a 1,8 V, junto a la tierra analógica está disponible una tensión de alimentación de 1,8V.

Es tarea del sistema de adquisición de datos ejecutar el inicio de conversión y establecer la frecuencia de muestreo. Para garantizar la estabilidad del muestreo y que no existan pérdidas de datos, el disparo de conversión así como la atención a los datos guardados en la FIFO del ADC serán atendidos por el PRUSS, un subsistema al interior del SoC AM3358 mostrado en la figura 2.

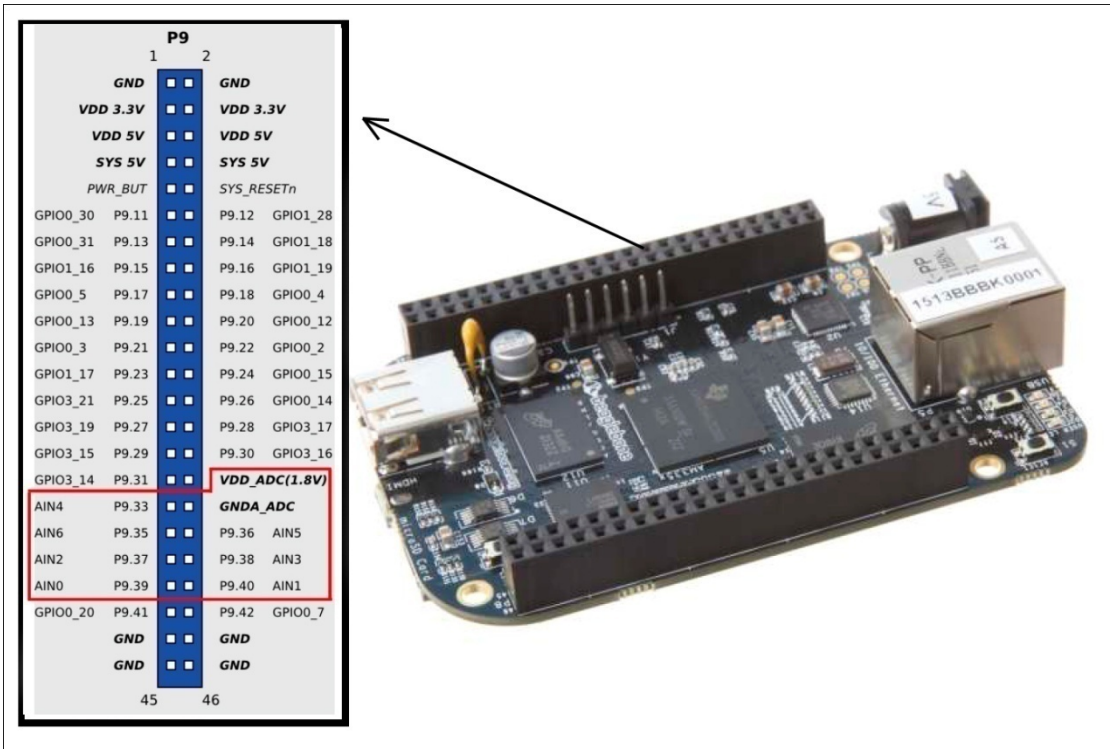

Figura 1

Entradas analógicas en la BBB situadas en el puerto P9 junto a la referencia y alimentación para este fin.

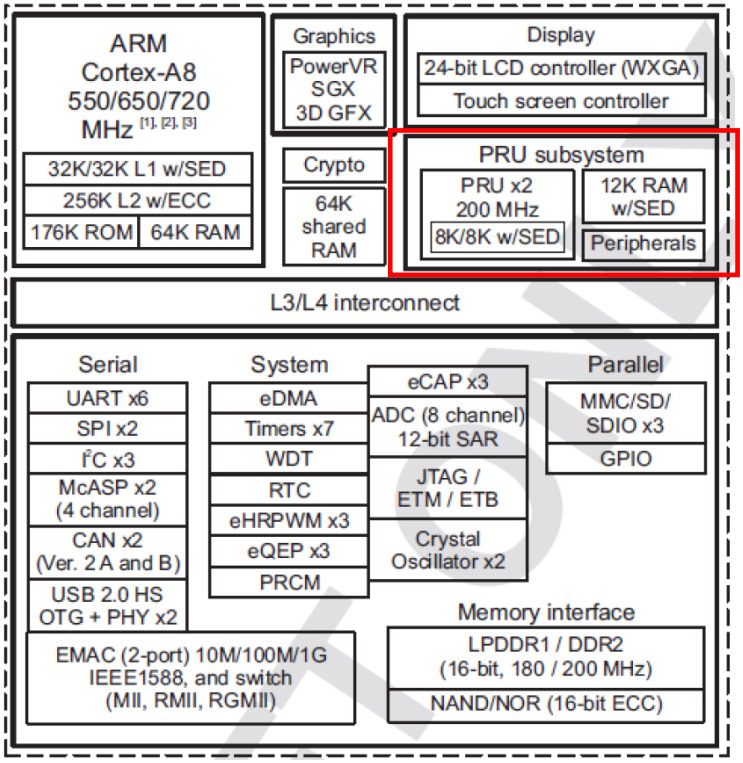

Figura 2

Arquitectura interna del AM335X [8].

El subsistema consta de dos procesadores RISC de 32 bits a una frecuencia de 200 MHz llamados Programable Real-Time Unit (PRU0 y PRU1). Los PRUs funcionan de forma independiente uno del otro y tienen acceso a todos los periféricos del SoC. Por diseño tienen un rápido acceso al resto de los sistemas así como atención a eventos externos con una estabilidad de tiempo base en el orden de las 50 ppm y una resolución temporal de 5ns. Estas cualidades convierten al PRUSS en una buena opción para sistemas críticos en tiempo. El sistema diseñado utiliza esta cualidad para disciplinar el ADC interno de la BBB.

El código de los PRUs está escrito en lenguaje ensamblador garantizando un mayor control sobre los tiempos de disparo y lazos de espera entre muestras. Esta ventaja supondría un problema para usuarios no adiestrados en ensamblador y las funciones de los PRUs dificultando el uso del sistema por parte de usuarios generales y desarrolladores no expertos. En respuesta a esta desventaja se creó un sistema de alto nivel que gestionará las aplicaciones implementadas en los PRUs que se distribuirían como un firmware para la adquisición de señales analógicas.

La BBB en su revisión C tiene como sistema operativo Linux Debian y trae de forma nativa un intérprete de Python. Las facilidades de uso de este lenguaje lo hacen una buena opción para gestionar la aplicación principal del sistema de adquisición. Para establecer comunicación con los PRUs se utilizó la librería PyPRUSS disponible en internet, que se encarga de escribir los programas en espacio de ejecución en cada uno de los PRUs y establecer comunicación con ellos a través de interrupciones. Un esquema del funcionamiento es mostrado en la figura 3.

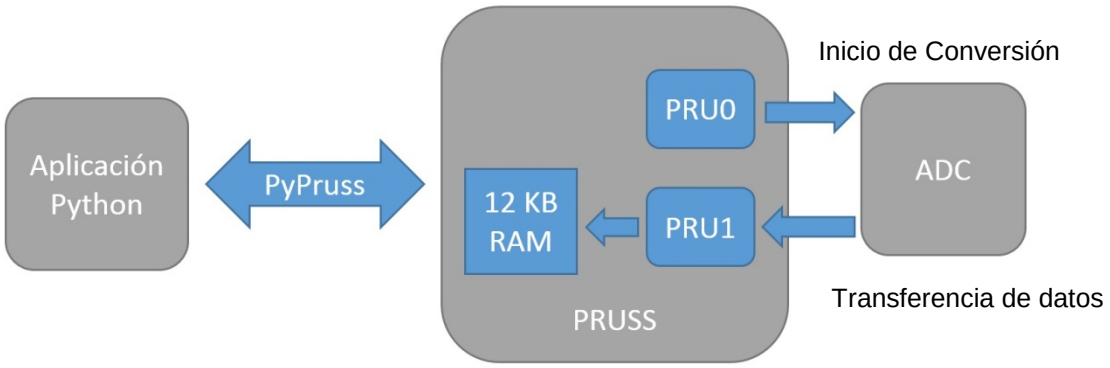

Figura 3
Esquema de funcionamiento.

Una vez que los programas compilados son descargados en los PRUs utilizando la librería PyPruss, el PRU0 se encarga de inicializar el ADC y comenzar la conversión. A la vez, el PRU1 está a la espera de los datos en la FIFO del ADC para ser salvados. Ambos PRUs comparten una RAM interna de 12Kb para el intercambio y almacenamiento de datos la que es usada por el PRU1 para almacenar de manera temporal las muestras extraídas del ADC. Cada valor leído del convertidor es un número binario de 32 bits, por lo que el espacio para muestras se ve reducido a 3000 muestras [11]. En dependencia de la frecuencia de muestreo y los canales utilizados este espacio se puede llenar en un tiempo relativamente rápido. Es tarea del programa principal salvar estos datos hacia archivos en memoria no volátil. El PyPruss permite establecer comunicación con los PRUs a través de interrupciones por lo que la aplicación principal recibirá una interrupción cíclica de parte del PRU1 cada vez que la memoria interna se encuentre lista para ser salvada. Esto garantiza la continuidad de los datos adquiridos y que no ocurran pérdidas de los mismos.

3.- RESULTADOS Y DISCUSIÓN

Para la validación del sistema se capturó la cantidad de muestras adquiridas durante un tiempo impuesto por un sistema GPS. Se tomó como referencia de tiempo la salida de pulso por segundo (PPS, por sus siglas en inglés) del Adafruit Ultimate GPSV3, con resolución de 10 ns y muy alta sensibilidad [13]. Esta señal es emitida como un pulso en alto por 50ms en cada inicio de segundo permitiendo contar de manera precisa la cantidad de segundos transcurridos al contar la

cantidad de pulsos. El número de muestras adquiridas durante un tiempo base es comparada con la cantidad de muestras esperadas en el mismo tiempo transcurrido a una frecuencia constante, mostrando los errores durante la adquisición.

Como método comparativo también fue evaluada la función time.sleep() del Pyhton y un temporizador interno del SoC para realizar el disparo de conversión del ADC. El temporizador utilizado está embebido en el periférico de Ethernet industrial que también forma parte del PRUSS [11].

La señal PPS es pasada al PRU1 a través de su puerto de expansión ubicado en el P9.26 mostrado en la figura 4, el cual es reflejado en su registro de entrada R31 en el bit 16 [11]. Este bit es encuestado por el PRU1 contando la cantidad de segundos transcurridos en cada transición del mismo.

P9	17	A16	I2C1_SCL	pr1_uart0_txd		
	18	B16	I2C1_SDA	pr1_uart0_rxd		
	19	D17	I2C2_SCL	pr1_uart0_rts_n		
	20	D18	I2C2_SDA	pr1_uart0_cts_n		
	21	B17	UART2_TXD	pr1_uart0_rts_n		
	22	A17	UART2_RXD	pr1_uart0_cts_n		
	24	D15	UART1_TXD	pr1_uart0_txd	pr1_pru0_pru_r31_16 (Input)	
	25	A14	GPIO3_21*	pr1_pru0_pru_r30_5 (Output)	pr1_pru0_pru_r31_5((Input)	
	26	D16	UART1_RXD	pr1_uart0_rxd	pr1_pru1_pru_r31_16	
	27	C13	GPIO3_19	pr1_pru0_pru_r30_7 (Output)	pr1_pru0_pru_r31_7 (Input)	
	28	C12	SPI1_CS0	eCAP2_in_PWM2_out	pr1_pru0_pru_r30_3 (Output)	pr1_pru0_pru_r31_3 (Input)
	29	B13	SPI1_D0	pr1_pru0_pru_r30_1 (Output)	pr1_pru0_pru_r31_1 (Input)	
	30	D12	SPI1_D1	pr1_pru0_pru_r30_2 (Output)	pr1_pru0_pru_r31_2 (Input)	
	31	A13	SPI1_SCLK	pr1_pru0_pru_r30_0 (Output)	pr1_pru0_pru_r31_0 (Input)	

Figura 4

Entrada de la señal PPS al PRU1 a través del puerto de expansión [12].

Tres frecuencias de muestreo fueron evaluadas: 400 muestras/s, 1000 muestras/s y 5000 muestras/s. Fueron elegidas en el marco de futuros proyectos para la utilización de las señales adquiridas en la detección de eventos en tuberías de agua. En el caso de 400 m/s y 1000 m/s para la detección y localización de rupturas súbitas y fugas de fondo en tuberías plásticas, y en el caso de 5000 m/s para la detección de fugas de fondo en tuberías metálicas usando métodos vibro-acústicos. Cada prueba fue repetida 10 veces, permitiendo evaluar también la desviación típica de cada captura y por lo tanto la estabilidad del muestreo.

La figura 5, presenta la cantidad de muestras adquiridas a 400 muestras/s en un rango de tiempo de un minuto y cinco minutos.

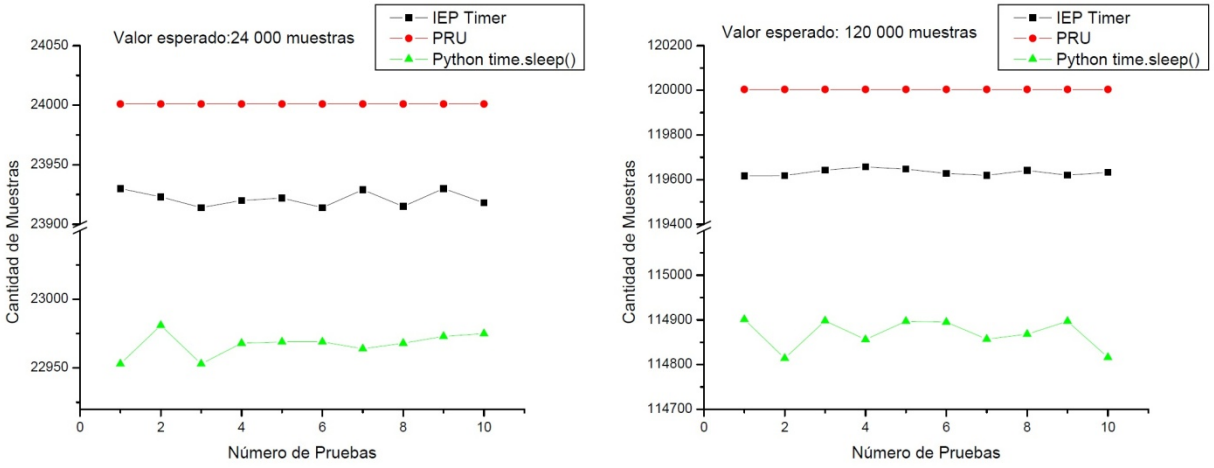

Figura 5

Muestras obtenidas a 400 muestras/s durante 1 minuto (izquierda) y 5 minutos (derecha).

El valor esperado durante un minuto a 400 muestras/s es de 24000 muestras adquiridas. El valor medio de las muestras adquiridas utilizando la función de Python time.sleep() es de 22969, asumiendo una pérdida promedio de 1031 muestras en un minuto de adquisición y una desviación típica de 8,86 muestras. A esta misma frecuencia de adquisición, pero durante 5 minutos, la cantidad de muestras esperadas es de 120000. El valor promedio de la adquisición para esta función de Python fue de 114880, suponiendo una pérdida de 5120 muestras y una desviación típica de 33,75 muestras.

La utilización de un temporizador aumenta notablemente la precisión de la adquisición. Para un valor esperado de 24000 muestras en un minuto, el disparo por el temporizador alcanza un valor medio de 23921, para una pérdida promedio de 79 muestras y una desviación típica de 6,43 muestras. Para 5 minutos de adquisición se alcanzan valores medios de 119630, para una pérdida promedio de 370 muestras y una desviación típica de 14,14 muestras.

Realizando el disparo de conversión con los PRUs se obtiene para un minuto de adquisición 24001 muestras para un error promedio de una muestra y desviación típica de 0 muestras. En 5 minutos un valor de 120004 muestras, para un error promedio de 4 muestras y desviación típica nuevamente de 0 muestras.

Las figuras 6 y 7 muestran los valores adquiridos a 1000 muestras/s y 5000 muestras/s respectivamente durante uno y cinco minutos de adquisición. La tabla 1 muestra un resumen de los valores registrados en todas las pruebas.

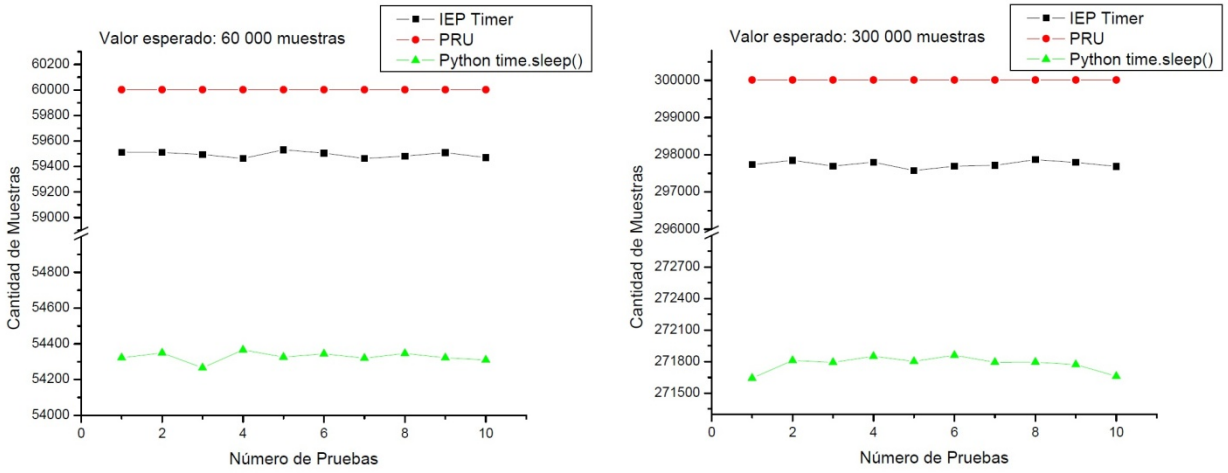

Figura 6

Muestras obtenidas a 1000 muestras/s durante1 minuto (izquierda) y 5 minutos (derecha).

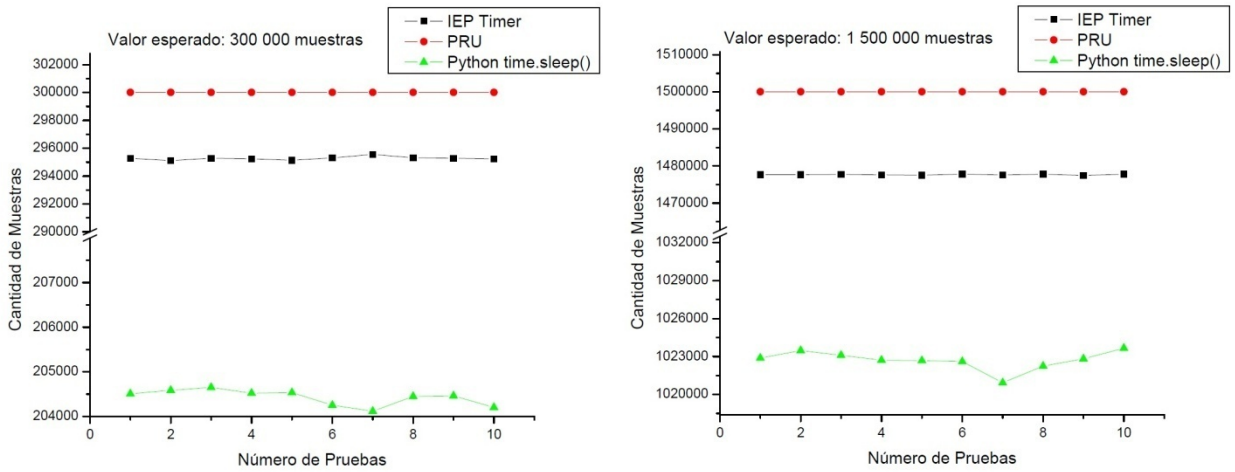

Figura 7

Muestras obtenidas a 5000 muestras/s durante1 minuto (izquierda) y 5 minutos (derecha).

En los resultados presentados en la Tabla 1 se observa como el error absoluto promedio se incrementa notablemente al elevarse la frecuencia de muestreo. En el caso de la función time.sleep() a 400 muestras/s se tiene una pérdida promedio del 4,3 % de las muestras adquiridas, esta cifra asciende al 31,84 % de las muestras cuando se encuentra en régimen de 5000 muestras/s. En el caso del temporizador, al no depender de las demoras del sistema operativo, los resultados son superiores a los alcanzados con la función de Python. Para 400 muestras/s se tiene una pérdida promedio del 0,33 % ascendiendo a 1,57 % de las muestras a 5000 muestras/s. Ambos casos presentan una desviación típica con tendencia a ascender con la frecuencia de adquisición. En el caso del sistema implementado para el disparo de conversión utilizando los PRUs, el error absoluto promedio se mantiene en el rango de 0,3 a 0,4 %. La dispersión es en todos los casos de 0 muestras, demostrándose la estabilidad de la frecuencia de muestreo impuesta por el PRU0. El error absoluto generado por el PRU es probablemente debido al reloj interno del subsistema, el cual es un Lazo de Seguimiento de Fase (PLL por sus siglas en inglés), que usa como base de tiempo el oscilador a cristal de la BBB.

Tabla 1

Valores registrados en las pruebas realizadas.

	Frecuencia de Muestreo [muestras/s]	Python time.sleep()		Temporizador		PRUSS	
		1 min	5min	1 min	5min	1 min	5min
Error absoluto promedio [muestras] y error relativo entre paréntesis [% de las muestras esperadas]	400	1031 (4,3%)	5120 (4,3%)	79 (0,33%)	370 (0,31%)	1 (0,004%)	4 (0,003%)
	1000	5676 (9,46%)	28210 (9,4%)	537 (0,89%)	2275 (0,76%)	2 (0,003%)	11 (0,004%)
	5000	95520 (31,84%)	477200 (31,81%)	4720 (1,57%)	22345 (1,49%)	11 (0,004%)	58 (0,004%)
Desviación Típica [muestras]	400	8,86	33,75	6,43	14,14	0	0
	1000	27,42	72	23,48	88,2	0	0
	5000	176,85	752,65	120,57	116,57	0	0
Error relativo promedio para peor caso de 5000 muestras/s y 5min [partes por millón (ppm) de las muestras esperadas]		318133 ppm		14897 ppm		38 ppm	

Al final de la tabla se muestra el error relativo promedio, esta vez expresado como partes por millón de muestras referidas a las muestras esperadas. En el caso del time.sleep() y el temporizador este valor es calculado para la máxima frecuencia de muestreo durante 5 minutos por ser el peor caso, debido a la inestabilidad del error absoluto frente a la frecuencia de adquisición. En el caso de la utilización del PRU y su alta estabilidad en el error absoluto en todas las frecuencias de muestreo empleadas, este valor aproximado de 38 ppm puede ser utilizado para estimar su precisión en todos los rangos de frecuencia de muestreo.

4.- CONCLUSIONES

Con la utilización del Subsistema Programable de Tiempo Real de la BeagleBone Black para gestionar el disparo de conversión del ADC y la atención a los datos generados por la adquisición se logra reducir las pérdidas de datos a 38 ppm. Si se usara un temporizador, la pérdida de datos promedio sería de 14897 ppm y con un disparo por software sería de 318133 ppm.

La desviación típica de los valores adquiridos con el uso del PRUSS es de cero muestras, lo que garantiza una alta estabilidad en la frecuencia de muestreo.

El sistema diseñado es capaz de garantizar una adquisición de señales analógicas con una alta precisión en la BeagleBone Black, evitando aumentar los costos por la utilización de componentes externos. La solución aquí implementada permite

obtener una gran estabilidad temporal de muestreo usando lenguajes de alto nivel como el Python en una computadoras de placa única.

La gestión de la adquisición desde el PRUSS permite que el procesador principal de la BBB solamente maneje el movimiento de los datos adquiridos desde la RAM hacia la micro SD. Este proceso no afecta la atención a la adquisición de los datos.

REFERENCIAS

1. Piri D, Nagy TG, Barta V, Bán D, Bányai L, Bór J, et al. Universal Raspberry Pi based data logger developed for the NCK Geophysical Observatory–IAGA division 5. Observatory, instruments, surveys and analyses. Geomatikai Közlemények. 2015;18(1):93-94.
2. Laros JH, Pokorny P, DeBonis D,. PowerInsight. A commodity power measurement capability. IEEE International Green Computing Conference Proceedings. Arlington; USA; 2013.p. 27-29.
3. Dickhudt PJ, Hathaway KK. Custom data logger for real-time remote field data collections. Army Engineer Research and Development Center Vicksburg United States, 2017.
4. Caldas-Morgan M, Alvarez-Rosario A, Padovese LR. An autonomous underwater recorder based on a single board computer. PloS One. 2015;10(6):1-18.
5. Ambro M. Raspberry Pi as a low-cost data acquisition system for human powered vehicles. Measurement. 2017;100:7-18.
6. Salcedo M, Cendrós J. Uso del minicomputador de bajo costo “Raspberry Pi” en estaciones meteorológicas. Télématique. 2016;15(1):62-84.
7. Nikhade SG. Wireless sensor network system using Raspberry Pi and ZigBee for environmental monitoring applications. IEEE International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM). Chennai; India; 2015.p. 6-8.
8. Texas Instruments. AM335x ARM Cortex-A8 Microprocessors (MPUs) Technical Reference Manual; 2011.
9. Jolly AR, Chakravarthi MK. A standalone data logger for fiber optic vibration measurement system using Beaglebone. IEEE 10th International Conference on Intelligent Systems and Control (ISCO). Coimbatore; India; 2016.p. 519-522.
10. Lütjohann DS, Jung N, Bräse S. Open source life science automation: design of experiments and data acquisition via “dial-a-device”. Chemometrics and Intelligent Laboratory Systems. 2015;144:100-7.
11. Texas Instruments. AM335x PRU-ICSS Reference Guide; 2013.
12. Coley G. Beaglebone Black system reference manual. Texas Instruments, Dallas. 2013.
13. Pokhsraryan D. Fast Data Acquisition system based on NI-myRIO board with GPS time stamping capabilities for atmospheric electricity research. 5th International TEPA Symposium Thunderstorm and Elementary Particle Acceleration. Aragatsotn; Armenia; 2015.p. 23-27.

AUTORES

Fidel Alejandro Rodríguez Corbo. Ingeniero en Telecomunicaciones y Electrónica. Profesor Instructor de la Universidad de La Habana, La Habana, Cuba. fidel.rodriguez@flacso.uh.cu. Como aspirante al título de máster en Diseño de Sistemas Electrónicos investiga en sistemas embebidos, instrumentación y acondicionamiento de señales.

Arturo Hernández González. Ingeniero en Telecomunicaciones y Electrónica. Profesor Instructor de la Universidad Tecnológica de La Habana José Antonio Echeverría, La Habana, Cuba. Investiga en instrumentación, sensores inteligentes y acondicionamiento de señales.

Jorge Ramírez Beltrán. Ingeniero Electrónico. Investigador Titular de la Universidad Tecnológica de La Habana José Antonio Echeverría, La Habana, Cuba. Investiga en instrumentación y detección y localización de eventos en tuberías.