

Consideraciones para una correcta implementación de un enlace SPI entre dos dispositivos dsPIC

Alejandro J. Uriz*, Pablo Agüero†, Jorge Castiñeira Moreira*, Juan C. Tulli†, Esteban González† and Francisco Denk‡

*CONICET / Laboratorio de Comunicaciones - Facultad de Ingeniería - Universidad Nacional de Mar del Plata
Mar del Plata, Argentina

Email: ajuriz@conicet.gov.ar, casti@fi.mdp.edu.ar

†Laboratorio de Comunicaciones - Facultad de Ingeniería - Universidad Nacional de Mar del Plata
Mar del Plata, Argentina

Email: (pdaguero,jctulli)@fi.mdp.edu.ar

‡Base Belgrano II

Dirección Nacional del Antártico
Antártida, Argentina

Email: franciscodenk@fi.mdp.edu.ar

Resumen—Hoy en día la mayoría de los microcontroladores poseen módulos integrados que permiten implementar de una forma sencilla protocolos estándar de comunicaciones. Si bien los fabricantes de cada dispositivo ofrecen la documentación necesaria para que un desarrollador pueda implementar enlaces, existen casos donde deben ser tomadas consideraciones adicionales con el fin de llevar a cabo la implementación. En este trabajo se presenta una serie de consideraciones que deben ser tenidas en cuenta con el fin de implementar correctamente un enlace mediante el protocolo SPI entre dos dsPIC de Microchip. Se realizan mediciones experimentales para validar las consideraciones propuestas.

Palabras clave—Comunicación SPI, dsPIC, sistemas embebidos, correlación cruzada.

I. INTRODUCCIÓN

A la hora de implementar un sistema complejo, suele ser necesario establecer una conexión entre dos dispositivos ubicados dentro o fuera de dicho sistema. Existe un gran número de protocolos estándar, los cuales permiten al desarrollador llevar a cabo una implementación de una forma simple y eficiente. Entre los protocolos más difundidos aparecen desde los protocolos tradicionales como el I^2C (*Inter-Integrated Circuit*) [1] y el RS-232 [2], hasta los más modernos como el SPI (*Serial Protocol Interface*) [3] y USB (*Universal Serial Bus*) [4], entre otros. En todos los casos y dependiendo de la aplicación que se desee llevar a cabo existen microcontroladores que tienen integrados módulos que permiten establecer la comunicación entre los dispositivos que componen el sistema. Si bien el fabricante de cada dispositivo suele ofrecer la documentación necesaria para establecer el enlace de una forma apropiada, en algunos casos la documentación puede carecer de la información que se requiere para extrapolar dichas aplicaciones a otras más complejas.

El objetivo de este trabajo es estudiar en profundidad las consideraciones necesarias para llevar a cabo un enlace utilizando el protocolo SPI (*Serial Peripheral Interface*) entre dos dispositivos dsPIC de Microchip. Para este caso

en particular se presentarán los pasos llevados a cabo para establecer una transferencia de datos de forma periódica a alta velocidad. Además, con el fin de sincronizar las señales de reloj de ambos dispositivos, se estudian las condiciones para compartir la fuente de reloj para ambos dispositivos.

El objetivo específico de este trabajo es implementar un enlace mediante el protocolo SPI entre dos dispositivos dsPIC33FJ128GP802 de Microchip, los cuales serán utilizados para implementar un algoritmo de reducción de ruido mediante el método de la correlación cruzada [6], [7].

Este trabajo está organizado de la siguiente manera. La Sección II describe las características relevantes del protocolo SPI. La Sección III describe la implementación llevada a cabo. Por último, la Sección IV presenta las conclusiones del trabajo y las líneas de trabajo a seguir en el futuro.

II. EL PROTOCOLO DE COMUNICACIÓN SPI

El estándar de comunicación serie SPI (*Serial Peripheral Interface*) es un protocolo serie sincrónico para comunicaciones entre dispositivos. Generalmente, este protocolo se implementa en un microcontrolador para establecer un enlace con dispositivos tales como memorias EEPROM, convertidores analógicos-digital, digital-analógicos, e inclusive con otro microcontrolador. Este trabajo hace incapié en las consideraciones que deben tenerse en cuenta a la hora de implementar un enlace SPI con otro microcontrolador.

En un enlace SPI al menos intervienen dos dispositivos, un **maestro**, el cual administra la comunicación, y un **esclavo** el cual obedece a las órdenes del maestro.

El protocolo SPI permite establecer un enlace bidireccional, el cual es implementado mediante cuatro líneas de comunicación:

- SDIx: *Serial data input*. Esta línea funciona como entrada de datos.
- SDOx: *Serial data output*. Esta línea opera como salida de datos.

- **SCKx**: *Shift clock input or output*. Esta línea transporta la señal de reloj entre los dispositivos. Dependiendo del modo de operación SPI en el que esté operando el enlace y de la forma en que esté configurado cada dispositivo. Puede ser configurada como una línea de entrada o salida.
- **SSx/FSYNCx**: *Active-Low slave select or Frame synchronization I/O pulse*. Esta línea puede funcionar como línea de habilitación entre los dispositivos. En dicho caso el dispositivo maestro permite la habilitación de algún dispositivo esclavo cuando sea necesario. Por otro lado, también puede ser configurada como línea de sincronización de datos entre dispositivos.

Un esquema de un enlace SPI se presenta en la Figura 1. En dicha figura se aprecian los principales componentes del mismo. En primer lugar se puede ver que la comunicación ha sido establecida entre dos microcontroladores, un Maestro (*Master*) y otro Esclavo (*Slave*), los cuales interactúan entre sí mediante las cuatro líneas de comunicación previamente descritas, en las que se indica el sentido de flujo de los datos.

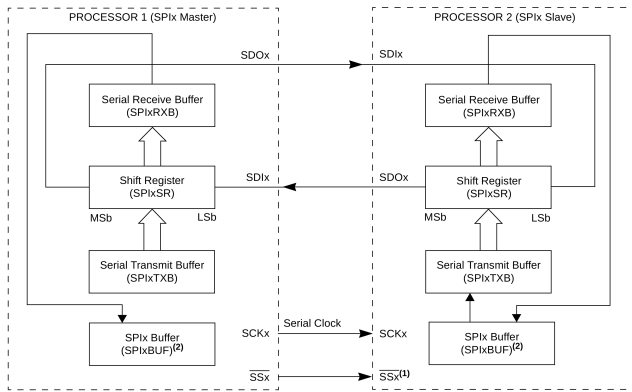


Fig. 1. Esquema del enlace SPI.

Por otro lado, la Figura 1 muestra los registros que intervienen en la comunicación SPI. En primer lugar se aprecia al registro de desplazamiento *SPIxSR*, el cual en el caso de que el sistema funciona en modo transmisor opera en modo entrada paralelo/salida serie. Esto se realiza cargando en forma paralela los datos contenidos en el registro *SPIxTXB*, para luego transmitirlos en formato serie a través del pin *SDOx*. Por otro lado, si el sistema opera como receptor SPI, el registro *SPIxSR* opera en modo entrada serie/salida paralelo. En este caso, la entrada de datos se realiza de forma sincrónica a través del pin *SDIx*, mientras que una vez que una palabra es recibida, la misma es transferida en forma paralela al buffer *SPIxRXB*, desde el cual es copiada al buffer SPI de uso general *SPIxBUF*.

A. Modos de Operación

Como se comentó previamente, un dispositivo que opera en modo SPI puede ser configurado de dos formas: maestro o esclavo. Además, el módulo puede ser configurado para operar con datos de 8 o 16 bits. Por otro lado, existen dos modos de establecer la comunicación: el modo **normal** y el modo **framed**. Este último modo es propuesto por Microchip y se

caracteriza por establecer un enlace SPI en el cual la señal de reloj está constantemente activa. Esto si bien genera un aumento en el consumo de energía, trae aparejado una mejora en el sincronismo de los datos.

III. CASO DE APLICACIÓN: IMPLEMENTACIÓN DE UN ARREGLO DE MICRÓFONOS PARA DETERMINAR LA DIRECCIÓN DE ARRIBO DE UNA SEÑAL

Como se mencionó previamente, el objetivo de este trabajo es llevar a cabo una comunicación SPI entre dos dispositivos dsPIC de Microchip. El fabricante ofrece la documentación necesaria para llevar a cabo la implementación [3], [5], [10], [11]

A. Algoritmo Aplicado

En este trabajo se analiza el caso en que es necesario establecer un vínculo entre dos dispositivos dsPIC con el fin de conformar un arreglo de micrófonos. Dicho arreglo se utiliza para reducir el ruido del sonido adquirido, lo cual se realiza por medio del algoritmo de correlación cruzada [6], [7]. En este método, se toma la señal adquirida por cada elemento que compone el arreglo de micrófonos, y se la correlaciona con el fin de establecer el retardo existente entre la señal recibida por ambos dispositivos. Una vez determinado dicho desplazamiento temporal, se desplaza una de las señales con el fin de compensarlo y luego, se las promedia. Si se asume que ambas señales adquiridas están contaminadas por ruido blanco gaussiano, la señal resultante del promedio reduce el ruido que contiene tantas veces como micrófonos tenga el arreglo. En la Figura 2 se aprecia una representación de la topología presentada.

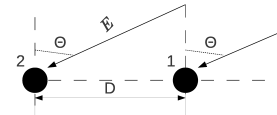


Fig. 2. Esquema del arreglo de micrófonos utilizado.

Este método se basa en el hecho que la señal proveniente de la dirección θ_s arriba primero al micrófono 1, luego recorre una distancia ξ y, es recibida por el micrófono 2. Entonces, si: $\xi = D \cdot \sin \theta_s$, la dirección de arribo de la señal puede ser determinada usando la Ecuación 1.

$$\theta_s = \sin^{-1}\left(\frac{v \cdot \tau}{D}\right) \quad (1)$$

donde v es la velocidad del sonido, y τ es el tiempo que le lleva a la señal atravesar la distancia v . Entonces, una vez que se obtiene un valor de τ del rango $-T < \tau < T$, la dirección de arribo puede ser estimada. Pero, en este trabajo, τ es usado para determinar el desplazamiento temporal entre las señales que son adquiridas por cada micrófono. Entonces, con el fin de ajustar temporalmente ambas señales adquiridas, una de las mismas es desplazada d muestras, donde $d = \frac{\tau}{T_s}$, y T_s la frecuencia de muestreo. En este trabajo, la distancia τ

es obtenida valor máximo de la correlación cruzada $\Phi(\tau)$ (ver Ec. 2) entre los vectores de entrada $X_1(t)$ y $X_2(t)$.

$$\Phi(\tau) = \frac{1}{N} \sum_{t=0}^{N-1} X_1(t)X_2(t + \tau) \quad (2)$$

Donde $\Phi(\tau)$ es un vector de $2.N - 1$ elementos.

Se asume que la señal que llega a ambos micrófonos es exactamente la misma pero desplazada temporalmente τ . Si el elemento de valor máximo del vector $\Phi(\tau)$ coincide con el centro del vector calculado, se concluye que no hay defasaje entre ambas señales, y por lo tanto $\tau = 0$. Por otro lado, si el valor máximo no coincide con el centro del vector obtenido mediante correlación cruzada, se puede concluir que ambas señales han arribado a los micrófonos con un retardo τ . Cabe destacar que el mínimo valor de retardo está asociado al período de muestreo del conversor analógico-digital que adquiere la señal de ambos micrófonos, ya que este intervalo de tiempo está asociado al valor obtenido de τ cuando $d = 1$.

Una vez que se obtiene el retardo entre ambas señales $d = \frac{\tau}{T_s}$, el algoritmo propuesto desplaza una de las señales para compensar dicho retardo y luego promedia ambos vectores. Como consecuencia de esta operación, el ruido blanco contenido en la señal promediada se reduce respecto a cada una de las señales originales tantas veces como micrófonos tenga el arreglo.

La Figura 3 presenta el resultado del algoritmo propuesto para un arreglo de dos micrófonos. El gráfico muestra la relación señal a ruido o SNR (*Signal-to-Noise ratio*) obtenida en función de distintos valores de ruido. En trazo color azul se presenta la SNR para el caso en que no se aplica el algoritmo de mejora, mientras que en trazo rojo se presenta la SNR obtenida al aplicar el algoritmo de correlación cruzada para un arreglo compuesto por dos micrófonos. En la Figura 3 es posible ver que la mejora obtenida es de 3dB para niveles de ruido mayores a 0.01Vpp. Para valores por debajo de ese umbral, la mejora no es significativa.

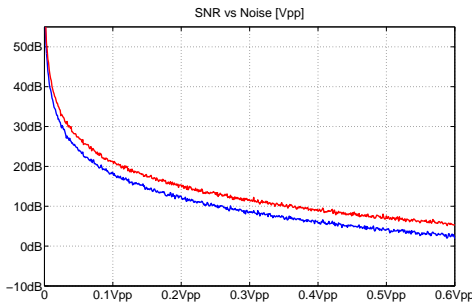


Fig. 3. Relaciones señal a ruido SNR para el sistema base (sin mejora) en trazo azul y del sistema con la mejora implementada en trazo rojo.

La implementación circuital del sistema se desarrolla en la subsección siguiente.

B. Implementación

Para llevar a cabo la implementación, se utilizan dos sistemas para procesamiento de señales de voz desarrollados en

[9]. Este sistema funciona como base para el desarrollo de un sistema de asistencia auditiva de bajo costo, el cual se compone de las siguientes etapas:

- **Etapas de Entrada.** Con el fin de acondicionar la señal para ser procesada, se utiliza un circuito preamplificador de micrófono, un filtro anti-solapamiento y un circuito para control automático de ganancia.
- **Procesamiento.** El procesamiento de la señal adquirida es realizado utilizando un dsPIC33FJ128GP802 [8] de Microchip, el cual está configurado para operar a 40MIPS, tiene una frecuencia de muestreo de 16Khz, y opera con datos de 16 bits.
- **Etapas de Salida.** Con el fin de acondicionar la señal de salida, se diseña un circuito amplificador, el cual controla la ganancia de la señal de forma tal que sea percibida en un nivel apropiado.

En la Figura 4 se presenta un diagrama de bloques del sistema base construido.

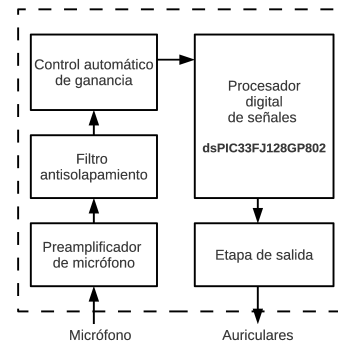


Fig. 4. Diagrama de bloques del prototipo implementado.

Con el fin de aplicar el algoritmo de correlación cruzada se debe implementar un enlace de datos entre cada uno de los dispositivos. Debido a que es necesario que el sistema opere en tiempo real, las muestras que el primer sistema adquiere y que deben ser procesadas por el segundo deben transmitirse a la velocidad mas alta posible. Como se mencionó previamente, el dsPIC a utilizar dispone de un módulo de comunicaciones SPI. Este módulo se caracteriza por poder operar con tasas de transferencia de hasta 10Mbps, lo cual permite transmitir gran cantidad de datos en un breve intervalo de tiempo.

Tal como lo establece el manual del fabricante, se debe definir a uno de los dos dispositivos del enlace como maestro y al segundo como esclavo. En este caso, el sistema que adquiere datos y se los transmite al segundo, definido como maestro. Esto se realiza con el fin de que este sistema defina el momento de transmitir los datos, lo cual debe estar ubicado en una parte específica del ciclo de ejecución de programa. Por otro lado, con fines de sincronizar el flujo de los programas el sistema esclavo, tiene definido dentro de su ciclo de ejecución un punto en que espera los datos del transmisor. El esquema del sistema implementado se aprecia en la Figura 5.

Por otro lado, dado que el dispositivo esclavo va a procesar la señal proveniente de ambos procesadores, es necesario

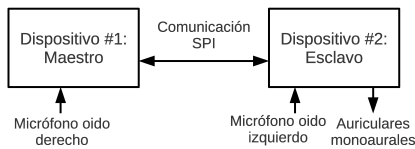


Fig. 5. Esquema del enlace SPI entre ambos dispositivos. En este caso, el dispositivo maestro ubicado en el oído derecho envía información al esclavo, ubicado en el oído izquierdo.

que los convertidores analógico-digital de ambos procesadores operen con la misma frecuencia de muestreo y de forma sincronizada. Cabe destacar que si bien se podría utilizar un oscilador para cada sistema, aparecerían diferencias entre las frecuencias de reloj. Si bien los dispositivos dsPIC disponen de un registro para la sintonía fina del oscilador llamado <OSCTUN>, generalmente la diferencia entre las frecuencias de trabajo de ambos dispositivos suele ser menor a la resolución que brinda este registro. Por dicho motivo, con el fin de asegurar el correcto funcionamiento del sistema, se debe utilizar el mismo oscilador para ambos dispositivos. Por ello, se configura al dsPIC que actúa como maestro de la comunicación para que opere utilizando un oscilador externo a cristal. Con el fin de que el sistema opere de acuerdo a los tiempos establecidos por el fabricante, se utiliza un cristal de cuarzo de 7.3724MHz. Una vez que el dispositivo maestro funciona correctamente, se extrae la señal de reloj del pin *OSC2* del dsPIC, y se utiliza dicha señal para excitar al dsPIC que opera como esclavo en la comunicación SPI, el cual según el fabricante debe configurarse para ser excitado por un oscilador externo. En consecuencia, la topología propuesta se representa en la Figura 6.

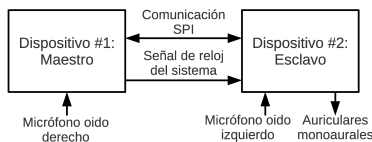


Fig. 6. Sistema Implementado. La señal de reloj del sistema maestro es usada para excitar al sistema esclavo. De esta forma ambos dispositivos quedan sincronizados.

Si bien la solución adoptada en la Figura 6 ha sido implementada siguiendo las recomendaciones del fabricante, el reloj de sistema de ninguno de los dos dsPIC oscila a la frecuencia a la que fueron configurados. Por otro lado, si se desconecta el cable que une ambos osciladores, el dsPIC primario funciona correctamente. Con el fin de determinar la naturaleza del corrimiento en frecuencia del oscilador, se analizó la configuración circuital del mismo. En la Figura 7 se aprecia el circuito asociado al oscilador externo utilizado para los dispositivos.

Tal como se ve en la Figura 7, el circuito corresponde a un **oscilador Pierce**. Por lo tanto, la capacidad equivalente generada tanto por el cable como por la compuerta de entrada del dispositivo maestro, se conecta en paralelo al capacitor *C2*. Se midieron las capacidades equivalentes tanto de la entrada

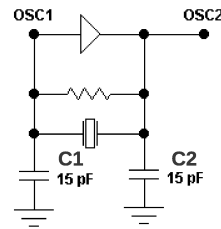


Fig. 7. Oscilador Pierce utilizado para excitar el dispositivo maestro.

del oscilador *OSC1* del dsPIC maestro, como así también del cable que une ambos osciladores. La capacidad equivalente medida en la entrada del oscilador del dsPIC esclavo fue de alrededor de 60pF, mientras que la del cable fue de alrededor de 10pF. Por lo tanto, se forma un capacitor equivalente C_2 de aproximadamente 85pF. Por este motivo, el oscilador del primer dsPIC deja de operar correctamente. En la Figura 8 se aprecia una representación del circuito resultante.

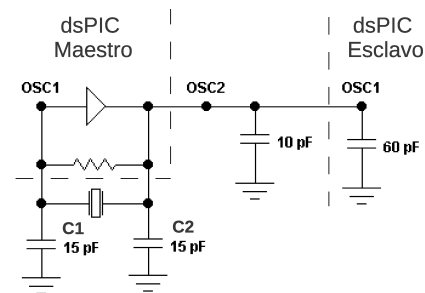


Fig. 8. Circuito eléctrico equivalente que resulta de conectar la señal de reloj del sistema maestro al dispositivo esclavo. Se puede apreciar que en paralelo a un capacitor del oscilador principal aparece una capacidad de 10pF debido al cable y una de 60pF debido a la compuerta de la entrada C1 del dispositivo esclavo.

Con el fin de compensar el efecto producido por estas capacidades equivalentes, se agrega un capacitor en serie entre la salida *OSC2* del dsPIC maestro y el cable que une los dispositivos. El agregado de esta capacidad en serie genera un divisor capacitivo, el cual hace que la *capacidad vista* desde la salida del oscilador del dsPIC maestro sea menor a la capacidad serie introducida. En este caso, debe tenerse en cuenta que la amplitud de la señal de entrada al oscilador del segundo dsPIC sea la suficiente para excitar la entrada. En consecuencia, el circuito resultante se aprecia en la Figura 9.

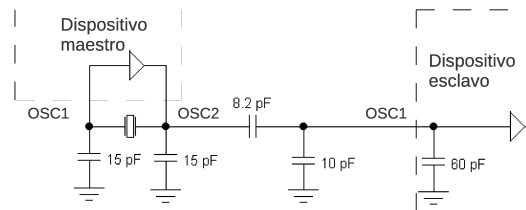


Fig. 9. Circuito eléctrico equivalente luego de la mejora. Con el fin de reducir la capacidad equivalente que aparece en paralelo al capacitor de 15pF se conecta un capacitor de 8.2pF en serie con el cable.

Por último, si bien el fabricante para este caso indica que el oscilador del dispositivo maestro debe estar configurado en modo oscilador externo (*EC*), esta configuración no siempre funciona. Motivo por el cual, se debe configurar al oscilador como si estuviera siendo excitado por un circuito oscilador Pierce entre los pines *OSC1* y *OSC2* (modo *XT*).

Un aspecto que debe tenerse en cuenta a la hora de implementar el prototipo es la radiación electromagnética generada por las líneas de comunicación SPI. Si bien la transmisión se realiza a 10Mbps, si se utiliza el modo estándar en el cual la señal de reloj se activa solo al momento de transferir datos, la señal de reloj aparece modulada por otra señal rectangular de período 15,72ms y tiempo en alto 800µs. Por dicho motivo se genera un batido de frecuencias que da como resultado que componentes de dicha interferencia estén presentes en la banda de audio. En la Figura 10 se presenta una captura utilizando un osciloscopio conectado a la salida del sistema para una señal de prueba de forma diente de sierra.

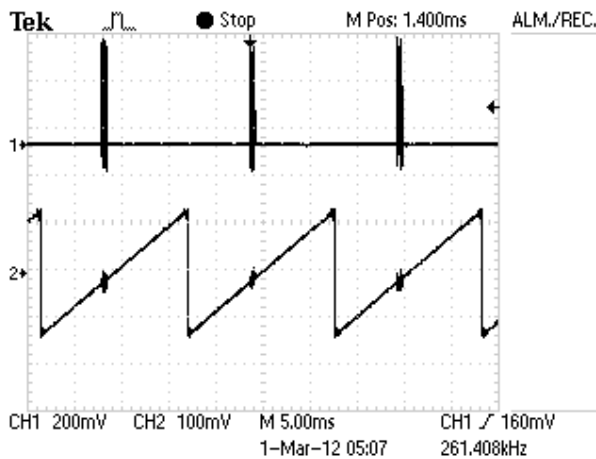


Fig. 10. Captura de las señales de reloj de la comunicación SPI (parte superior) y de los datos de salida del sistema (parte inferior). Se puede ver que en el momento en que el reloj se activa, se produce una interferencia de alrededor de 100mV en los datos de salida, la cual es tiene un período de 15.82ms.

En la parte superior de la Figura 10 se aprecia la señal de reloj. Puede verse que la línea permanece en bajo excepto durante los 800µs durante los cuales se transmiten datos entre los dispositivos. En la parte superior de la figura, se puede ver la señal de salida de forma diente de sierra, la cual es interferida en el instante en el que se activa la señal de reloj.

Con el fin de resolver este problema, en primer lugar se tomaron las precauciones necesarias para blindar las líneas de la comunicación SPI de forma tal de minimizar las radiación emitida por las mismas. Ello se realizó utilizando cables mallados para cada una de las líneas del enlace. Si bien se redujo la amplitud de dicha interferencia, esta seguía apareciendo en la señal sintetizada, se optó por cambiar el modo de comunicación al **modo framed**. En este modo, como se comentó previamente, la señal de reloj permanece activa durante todo el tiempo. De esta forma, dicha interferencia solo tiene como componente espectral la frecuencia de reloj

de 10MHz, y por lo tanto, no se encuentra dentro del espectro de señal útil.

Por último, si bien el dsPIC33FJ128GP802 dispone de 2KB de memoria DMA (acceso directo a memoria), la misma se utiliza para los conversores analógico-digital y digital-analógico. Cabe destacar que se utiliza un doble buffer de 256 datos de 16 bits cada uno. Bajo estas condiciones, la memoria DMA está ocupada en su totalidad. Por dicho motivo, el enlace SPI se implementa utilizando un ciclo el dispositivo maestro y una rutina interrupción en el esclavo, la cual es llamada cada vez que se reciben datos válidos. Con el fin de sincronizar el flujo del programa, cada vez que se ejecuta un ciclo de programa en el algoritmo del dispositivo esclavo, el programa principal queda a la espera de la recepción de nuevos datos a través del enlace SPI. Esto se realiza con una bandera llamada *flagSPI*, la cual solo es activada cuando se finaliza la recepción de los datos.

La rutina de transmisión en el dispositivo maestro se implementa de la siguiente forma:

```
int N;
for (i=0;i<2*N;i++)
{
WriteSPI1(*pSPICountT);
pSPICountT++;
while(SPI1STATbits.SPITBF);
}
```

Dónde, *pSPICountT* es un puntero que apunta a la dirección de memoria a partir de la cual se hayan almacenados los últimos $2 \cdot N$ datos adquiridos. Aquí puede apreciarse que el dato a transmitir es cargado en el buffer de escritura, se incrementa el puntero para preparar al dato siguiente y por último, se espera a que la transmisión del dato finalice.

Por otro lado, la rutina de interrupción para recepción en el esclavo es la siguiente:

```
void __attribute__((interrupt,
no_auto_psv)) _SPI1Interrupt(void)
{
IFS0bits.SPI1IF= 0;
IEC0bits.SPI1IE= 0;
*pSPICountR=getcSPI1();
pSPICountR++;
if (pSPICountR==&spiBufferR[2*N])
{
pSPICountR=&spiBufferR2[0];
pSPICountR2=&spiBufferR[N/4];
flagSPI=0;
}
else if (pSPICountR==&spiBufferR2[2*N])
{
pSPICountR=&spiBufferR[0];
pSPICountR2=&spiBufferR2[N/4];
flagSPI=0;
}
IEC0bits.SPI1IE= 1;
```

En este caso, los primeros dos pasos dentro de la rutina son limpiar la bandera que indica la activación de la interrupción y deshabilitar dicha interrupción para evitar múltiples disparos. La carga de los datos se realiza utilizando dos registros `spiBufferR` y `spiBufferR2`, los cuales se llenan de forma alternada de forma tal que mientras uno de ellos está siendo utilizado por el programa principal, el otro está disponible para almacenar los datos recibidos. El procedimiento es controlado por el puntero `pSPICountR`, el cual cada vez que almacena un dato, se incrementa en una posición para almacenar el dato siguiente. Una vez que el puntero llega al final del registro correspondiente, el puntero es conmutado al otro registro de los que componen el doble buffer y se indica que la recepción de los datos necesarios para procesamiento ha finalizado (`flagSPI=0`).

C. Mediciones realizadas

Con el fin de analizar el funcionamiento del equipo se midió el uso de recursos del sistema implementado. Dichas mediciones se realizaron utilizando $N = 256$, $f_{sampling} = 16KHz$ y el enlace de datos SPI se implementó con una tasa de transferencia de $10Mbps$. En primer lugar se analizó el uso de memoria del algoritmo propuesto. Los resultados se presentan en la Tabla I

RAM	DMA	ROM
95%	100%	8%

TABLA I
USO DE MEMORIA DEL ALGORITMO IMPLEMENTADO EN EL dsPIC.

En la primera columna de la Tabla I se muestra que el algoritmo implementado consume casi la totalidad de memoria de datos del dsPIC utilizado. En contraparte, en la tercera columna se puede ver que la memoria de programa es muy poco utilizada. En la segunda columna, se puede ver que la memoria disponible para acceso directo a memoria (DMA) es totalmente utilizada por los registros destinados a las conversiones analógica-digital y digital-analógica.

Si bien, los resultados presentados previamente demuestran que es posible utilizar el dispositivo propuesto para esta aplicación, aun se debe demostrar que es posible ejecutar el algoritmo de mejora en tiempo real. Para cumplir dicha condición, el tiempo total de procesamiento del sistema debe ser inferior al tiempo de adquisición de datos. En la Tabla II se aprecian los principales tiempos involucrados en la ejecución de programa.

En la Tabla II se puede ver que la suma de los tiempos involucrados en la comunicación de los datos a través de enlace SPI ($0.82ms$) y de la aplicación del algoritmo de mejora ($5.5ms$) es inferior al tiempo de adquisición del sistema ($15.72ms$). Debido a lo planteado previamente, el sistema opera correctamente en tiempo real.

Por último, se midió el rendimiento del sistema reductor de ruido y se pudo apreciar que la mejora obtenida en el rango de señales de entrada es del orden de los 3dB, los cual es acorde a las estimaciones teóricas realizadas.

TABLA II
TIEMPOS MEDIDOS PARA LA COMUNICACIÓN SPI, EJECUCIÓN DEL ALGORITMO DE MEJORA, Y ADQUISICIÓN DE DATOS.

Com. SPI	Alg. mejora	Adquisición
$0.82ms$	$5.5ms$	$15.72ms$

IV. CONCLUSIONES

En este trabajo se presenta un conjunto de pautas para optimizar un enlace de dos dispositivos dsPIC mediante el protocolo SPI.

En este trabajo se han presentado aspectos críticos a la hora de implementar una comunicación SPI entre dos dispositivos dsPIC, algunos de los cuales no están desarrollados con la suficiente profundidad en el manual del fabricante. Por este motivo, se desarrollaron en detalle las consideraciones llevadas a cabo para que ambos dispositivos compartan oscilador.

Por otro lado, se demostró que en el modo SPI estándar, pueden aparecer problemas debidos a la radiación electromagnética generada por las líneas del enlace SPI. Motivo por el cual, en este caso se utilizó el modo SPI Framed.

Finalmente, a modo de trabajo futuro se desea implementar el algoritmo de reducción de ruido en un dispositivo dsPIC que disponga de mas memoria DMA, de esta forma, se podrá implementar la comunicación SPI a través de este medio. Esta mejora permitirá quitar la rutinas de interrupción para transmisión y recepción SPI del flujo del programa principal. Y por ende, el sistema implementado será mas eficiente.

REFERENCIAS

- [1] J.M. Irazabal y S. Blozis, *AN10216-01 - I2C Bus Application Note (Philips Semiconductors)*. www.nxp.com/, 2003.
- [2] Dallas Semiconductor, *Application Note 83: Fundamentals of RS232 Serial Communications*. <http://www.maxim-ic.com/>, 1998.
- [3] Microchip Inc., *Serial Peripheral Interface (SPI) - dsPIC33F/PIC24H FRM*. <http://www.microchip.com/>, 2011.
- [4] USB-IF, *Universal Serial Bus 2.0 Specification*. <http://http://www.usb.org/>, 2000.
- [5] Microchip Inc., *Serial Communications using the dsPIC30F SPI*. http://www.microchip.com, 2005.
- [6] A. K. Tellakula, *Acoustic Source Localization Using Time Delay Estimation*, Degree Thesis. Bangalore, India: Supercomputer Education and Research Centre Indian Institute of Science, 2007.
- [7] S. Takahashi, T. Morimoto, *Development of Small-size and Low-priced Speaker Detection Device Using Micro-controller with DSP functions*. Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol.1, 2011.
- [8] Microchip Inc., *dsPIC33FJ128GPX02/X04 Data Sheet, High Performance 16-bit Digital Signal Controllers*. <http://www.microchip.com/>, 2009.
- [9] F. Denk, P. Agüero, A. Uriz, J.C. Tulli, E. González, J. Garín y S. Bourguigne, *Assistive Listening Device based on a dsPIC*. VI Jornadas Argentinas de Robótica (JAR). pp.24-29, 2010.
- [10] Microchip Inc. *16-Bit Language Tools Libraries*. <http://www.microchip.com/>, 2005.
- [11] Microchip Inc. *dsPIC Language Tools*. <http://www.microchip.com/>, 2004.