

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

“Diseño e implementación de un sistema multiparamétrico de adquisición y acondicionamiento para sensores químicos resistivos”

TRABAJO FINAL DE GRADO

Autor/a:
Daniel Roig Muñoz

Tutor/a:
José Pelegrí Sebastià

GANDIA, 2013

RESUMEN

En este trabajo final de grado se presenta un sistema *multiparamétrico* de *adquisición y acondicionamiento* para sensores químicos resistivos. Dicho sistema multiparamétrico está formado básicamente por un dispositivo de interfaz de sensores (sistema de adquisición de datos) y un *microcontrolador*. Se ha desarrollado un software para controlar el sistema bajo la plataforma *LabVIEW*. La comunicación entre el microcontrolador y el dispositivo de interfaz de sensores se realiza mediante bus *SPI* y la puerta de enlace con el ordenador mediante comunicación serie a través de USB.

El sistema multiparamétrico desarrollado es muy *económico* en comparación con los sistemas existentes en el mercado.

ABSTRACT

In this final degree work we present a *multiparametric* of *acquisition and conditioning* system for chemical-resistive sensors. Said multiparametric system basically consists of a sensor interface device (data acquisition system) and a *microcontroller*. We have developed software to run the system under *LabVIEW* platform. The communication between the microcontroller and the sensor interface device is done through *SPI* bus and the gateway with the computer is done through serial communication over USB.

The multiparametric system developed is very *economical* compared with existing systems in the market.

La realización de este proyecto va dedicada a:

Mi esposa e hijo, los motores de mi vida, los que llenan mis momentos más felices y por los que lucho día a día.

A mis padres por su apoyo, consejos y sabiduría.

A José Pelegrí Sebastià por su labor docente, personal y profesional como tutor del proyecto.

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

ÍNDICE GENERAL

1. INTRODUCCIÓN..... 11

1.1. Presentación. 11

1.2. Objetivos..... 11

1.3. Metodología..... 12

1.4. Problemas surgidos..... 12

2. DESCRIPCIÓN DEL SISTEMA HARDWARE. 17

2.1. Elección del bus de comunicaciones..... 17

2.1.1. Descripción del Bus I²C..... 17

2.1.2. Descripción del Bus SPI..... 19

2.1.3. I²C vs SPI. Justificación de la elección del bus SPI..... 21

2.2. Elección de los componentes electrónicos..... 22

3. DESCRIPCIÓN DEL SISTEMA SOFTWARE. 25

3.1. Organización del programa. Jerarquía..... 25

3.2. Descripción de los Nodos Secundarios (SubVIs). 27

3.3. Descripción del VI Principal (Programa Principal)..... 34

4. RESULTADOS EXPERIMENTALES..... 37

4.1. Aplicabilidad del sistema multiparamétrico de adquisición de datos..... 40

5. CONCLUSIONES..... 41

5.1. Líneas futuras de trabajo. 41

6. BIBLIOGRAFÍA..... 43

6.1. Libros..... 43

6.2. Páginas Web..... 43

6.3. Referencias..... 43

ANEXO I. MANUAL DE USUARIO DEL SOFTWARE

ÍNDICE DE ILUSTRACIONES

Figura 1: Registros para configurar la cadena de adquisición y almacenamiento del resultado de la conversión 13

Figura 2: Bus I²C con dos dispositivos conectados. Las líneas SDA y SCL están conectadas con Vcc a través de resistencias de Pull-up 17

Figura 3: Estructura de una trama de 2 bytes de datos en el protocolo I²C 18

Figura 4: a) Bus SPI con conexión entre un único maestro y un único esclavo y b) un maestro y múltiples esclavos 19

Figura 5: Comunicación SPI simple por flanco de bajada 20

Figura 6: Modos de comunicación SPI, a) Modo A, b) Modo B, c) Modo C d) Modo D 20

Figura 7: Diagrama de bloques del SX8724S 23

Figura 8: Etapas del conversor SX8724S..... 23

Figura 9: Placa de evaluación del SX8724S 24

Figura 10: Jerarquía del programa 26

Figura 11: Diagrama de flujo del funcionamiento del programa principal..... 35

Figura 12: Tarjeta de conexión y acondicionamiento de sensores 37

Figura 13: Ejemplo fichero de resultados sensor Canal 1 38

Figura 14: Ejemplo fichero de resultados sensor Canal 2 38

Figura 15: Ejemplo fichero de configuración..... 39

Figura 16: Ejemplo gráfica Canal 1 39

Figura 17: Ejemplo gráfica Canal 2 40

Figura 18: Creación de carpetas en la ruta C:\DATOS..... 45

Figura 19: Panel Frontal al iniciar el programa 45

Figura 20: Selección del Puerto COM 46

Figura 21: Identificación de los sensores conectados 46

Figura 22: Configuración de los sensores. Conexión de cada sensor 47

Figura 23: Configuración cargada con éxito 47

Figura 24: Panel Frontal con 2 sensores..... 48

Figura 25: Finalización del programa 48

Figura 26: Archivos obtenidos 49

Figura 27: Gráficas obtenidas 49

Figura 28: Selección del fichero de configuración 50

1. INTRODUCCIÓN.

1.1. Presentación.

El presente trabajo final de grado nace a partir de la idea de crear un sistema multiparamétrico de adquisición de datos de distintos tipos de sensores y acondicionamiento de las señales adquiridas. El sistema creado es controlado mediante un software desarrollado bajo la plataforma LabVIEW. La base del sistema se centra en un dispositivo de interfaz de sensores de la marca SEMTECH denominado SX8724S (de ahora en adelante le llamaremos conversor SX8724S) que puede interconectar de forma directa con hasta 6 sensores referenciados a masa o 3 sensores diferenciales.

A lo largo de este capítulo se presentan, de una forma global, todos los contenidos que van a ir tratándose en las secciones posteriores. Por ello se presentan los objetivos, la metodología seguida, las fases en las que se divide el trabajo y los problemas surgidos en cada etapa.

En los capítulos posteriores se va a tratar la elección de los componentes electrónicos para el diseño del sistema, una descripción detallada del software creado y la aplicabilidad del sistema creado. Por último se añade un Anexo con un manual de usuario del software.

1.2. Objetivos.

El objetivo principal del trabajo final de grado es diseñar un sistema de adquisición con interfaz USB para cuatro sensores químicos resistivos.

Los objetivos secundarios son:

- Elegir el Bus de comunicaciones (SPI o I²C) entre el conversor y el microprocesador.
- Diseñar una etapa de adquisición de datos con 16 bits de conversión con Amplificador de Ganancia Programable (PGA) para amplificar la señal antes de llegar al conversor.
- Diseñar el software de control del sistema multiparamétrico.
- Estudiar la aplicabilidad del diseño dentro de los múltiples campos de utilidad (sanitario, seguridad, industria, medio ambiente).

1.3. Metodología.

Los pasos seguidos para la obtención de los objetivos marcados se han dividido en 4 fases:

- FASE I: BUS DE COMUNICACIONES.
 - Elección del bus de comunicaciones (SPI o I²C) idóneo que mejor se adapte a las especificaciones requeridas para el sistema, tales como velocidad de transmisión, consumos de energía, etc.
- FASE II: COMPONENTES ELECTRÓNICOS DEL SISTEMA.
 - Búsqueda y adquisición de los componentes electrónicos necesarios para diseñar el sistema.
- FASE III: SOFTWARE.
 - Diseño del software de control mediante el entorno de programación LabVIEW.
- FASE IV: APLICABILIDAD DEL DISEÑO.
 - Se han estudiado los posibles campos de utilidad para el diseño desarrollado (sanitario, seguridad, industria, medio ambiente).

1.4. Problemas surgidos.

Durante la ejecución de cada una de las fases descritas en el apartado anterior han ido surgiendo una serie de problemas que se han ido solventando de manera satisfactoria hasta conseguir los objetivos marcados. Los principales problemas han surgido durante la FASE III, en el diseño del software de control.

Para llevar a cabo la ejecución del trabajo se ha adquirido una placa de control fabricada por la empresa SEMTECH (SX8724SEVK) para evaluar el convertidor SX8724S. Dicha placa, básicamente está formada por un microcontrolador y el convertidor SX8724S. La comunicación entre el ordenador y la placa de control se realiza mediante comunicación serie a través de USB. El microcontrolador se encarga de la comunicación con el convertidor SX8724S mediante bus SPI. La problemática ha surgido durante la comunicación entre el entorno de programación LabVIEW y el convertidor SX8724S a través del microcontrolador, a la hora de obtener el valor de voltaje de varios sensores conectados simultáneamente y la estabilidad (ruido) de la lectura obtenida. Dicha problemática ha consumido la mayor parte del tiempo dedicado a la FASE III del diseño del software. A continuación se explican con detalle los problemas surgidos en la FASE III.

El convertor SX8724S funciona mediante registros de 8 bits cada uno. Básicamente trabaja con 8 registros cada uno con una dirección distinta (en formato hexadecimal). El valor de los registros es variable (también en formato hexadecimal) en función de la variación del valor de las distintas variables que contiene cada registro.

Register Name	Bit position							
	7	6	5	4	3	2	1	0
RegACOutLsb	Out[7:0]							
RegACOutMsb	Out[15:8]							
RegACCfg0 Default values:	Start 0-1	SetNelconv 00-11		SetOsr 000-111			Continuous 0-1	SampleShiftEn 0-1
RegACCfg1 Default value:	IbAmdAdc 00-11		IbAmpPga 00-11		Enable 0000-1111			
RegACCfg2 Default value:	SetFs 00-11		Pga2Gain 00-11		Pga2Offset 0000-1111			
RegACCfg3 Default value:	Pga1Gain 0-1	Pga3Gain 0000000-1111111						
RegACCfg4 Default value:	DataReadyEn 0-1	Pga3Offset 0000000-1111111						
RegACCfg5 Default value:	Busy 0-1	Def 0-1	Amux 00000-1111				Vmux 0-1	

Figura 1: Registros para configurar la cadena de adquisición y almacenamiento del resultado de la conversión [1]

A continuación se realiza una descripción de cada registro:

- RegACOutLsb (dirección 0x50): Almacena el valor “bajo” (primeros 8 bits) de voltaje obtenido por las entradas.
- RegACOutMsb (dirección 0x51): Almacena el valor “alto” (siguientes 8 bits) de voltaje obtenido por las entradas.
- RegACfg0 (dirección 0x52): Almacena el valor del número de conversiones elementales y el *oversampling rate*.
- RegACCfg1 (dirección 0x53): Almacena la habilitación del convertor A/D y las ganancias.
- RegACCfg2 (dirección 0x54): Almacena el valor de la frecuencia de muestreo, la segunda etapa de ganancia y el offset de la segunda etapa.
- RegACCfg3 (dirección 0x55): Almacena el valor de la ganancia de la primera etapa y el valor de ganancia de la tercera etapa
- RegACCfg4 (dirección 0x56): Almacena el valor de offset de la tercera etapa.

- RegACCFg5 (dirección 0x57): Almacena el valor del tipo y número de canal de entrada.

Estos registros, una vez configurados según el valor de sus variables internas se envían mediante comunicación serie hasta el conversor SX8724S mediante una trama en formato hexadecimal con los siguientes campos:

NAME	TYPE	SIZE	COMAND ID	ADDRESS	BYTE
BYTE	1	2	3	4	5
VALUE	0x01	0x03	0x01	0xaa	0xdd

Por ejemplo para enviar el RegACCFg5 con el valor del primer canal de entrada referenciado a masa (valor 0x24) se enviaría la siguiente trama:

NAME	TYPE	SIZE	COMAND ID	ADDRESS	BYTE
VALUE	0x01	0x03	0x01	0x57	0x24

El conversor devuelve la confirmación del envío de cada registro mediante una trama con el siguiente formato:

NAME	TYPE	SIZE	RETURN CODE
BYTE	1	2	3
VALUE	0x02	0x01	0x00 (*)

(*) Hay 7 códigos el 0 indica que no ha habido ningún error

Seguida de los datos que devuelve el conversor:

NAME	TYPE	SIZE	RETURN CODE	DATA MSB	DATA LSB
VALUE	0x03	0x03	0x00	0xdd	0xdd

Esta comunicación sería trivial y fácil de interpretar pero, en realidad, el conversor empieza a enviar bytes de datos de forma continua (hasta que se produce una interrupción, por ejemplo el cambio de canal de entrada), intercalando bytes de confirmación entre ellos, lo que hace más difícil la extracción de los 2 bytes (DATA MSB y DATA LSB) donde se almacenan los datos, además el número de bytes que se van almacenando en el buffer del puerto USB no es fijo, varía muy rápidamente, aunque se fije el valor de dicho buffer, ya que como el conversor envía los datos de forma continua, siempre hay datos disponibles en el bus. Esto causa una inestabilidad en el dato adquirido y una lentitud a la hora de procesar los datos del puerto USB.

Cuando se configuran 6 canales referenciados a masa para adquirir su valor de forma simultánea, cabría esperar que el formato de la trama de datos sería una cadena de bytes del tipo:

```
030300 DATA MSBsensor 1DATA LSBsensor1030300 DATA MSBsensor 2DATA LSBsensor2030300...
```

Pero no ocurre así. El conversor está preparado para leer un canal cada vez por eso cuando se recibe la trama de datos siempre es la misma:

```
030300 DATA MSBsensor 1DATA LSBsensor1030300 DATA MSBsensor1DATA LSBsensor1030300...
```

Por ello, se deben ir efectuando interrupciones para ir adquiriendo la lectura de cada canal.

El primer byte DATA MSB y el primer byte DATA LSB que se recibe después de cada interrupción son valores absurdos por lo que se descartan a la hora de adquirir el dato de cada canal.

En un principio cada comunicación serie se abría y cerraba para adquirir el dato de cada canal con lo cual, cuando se adquiría el valor de 6 sensores (número máximo de sensores conectado) la comunicación se ralentizaba en gran medida, cayendo el número de muestras por segundo a 1. Se abría y cerraba la comunicación debido a que si no se realizaba de dicha forma, los valores adquiridos no se correspondían con el valor de entrada, estaban intercalados entre ellos, es decir en cuando se intentaba adquirir el valor del canal 1 se obtenía el valor del canal 2 y así respectivamente. Al final se ha conseguido establecer una sola comunicación serie para 6 sensores simultáneos, habilitando y deshabilitando la activación del conversor y vaciando el buffer recibido (buffer de la comunicación serie).

Otra problemática surgida ha sido a la hora de adquirir el valor de la señal de entrada que ofrece el sensor, ya que el conversor devuelve el valor a su salida en función de la entrada y después de pasar por tres etapas de ganancia con 2 de ellas con un valor de offset.

Es por ello por lo que se ha creado una ecuación para obtener dicho valor en función de su salida:

$$V = \frac{OUT_{ADC} \times V_{REF} \times OSR}{2^{16} \times (OSR + 1)} (GD_{TOT} - (GD_{off3} + GD_3 \times GD_{off2}) \times S \times V_{REF})$$

Donde,

- OUT_{ADC} es el valor a la salida del conversor.
- V_{REF} es el valor de referencia al que se alimenta la placa de evaluación SX8724SEVK.
- OSR es el valor de Over Sampling Rate.
- GD_{TOT} es el valor de la multiplicación de las ganancias de cada etapa ($GD_1 \times GD_2 \times GD_3$), previas a la entrada del conversor.
- GD_{off3} es el valor de offset de la etapa 3.
- GD_3 es el valor de ganancia de la etapa 3
- GD_{off2} es el valor de offset de la etapa 2.
- S vale "-1" o "1" en función de si el tipo de entrada de cada sensor está invertida o no cuando los configuramos.

Finalmente, después de un arduo trabajo, se han obtenido unos resultados satisfactorios en cuanto a estabilidad (ruido) del valor de voltaje obtenido de distintos sensores conectados de forma simultánea y la rapidez de obtención (velocidad de transmisión) de dicho valor.

2. DESCRIPCIÓN DEL SISTEMA HARDWARE.

2.1. Elección del bus de comunicaciones.

Para llevar a cabo la elección del bus de comunicaciones que mejor se adapte a las especificaciones requeridas del diseño se ha realizado una comparativa entre ambos buses:

2.1.1. Descripción del Bus I²C.

I²C (Inter-Integrated Circuit) es un protocolo de comunicación serie diseñado por Philips que se utiliza esencialmente entre dispositivos que pertenecen al mismo circuito, por ejemplo, sensores con un microcontrolador. Es una comunicación de tipo half-dúplex, es decir que es una comunicación bidireccional por la misma línea pero no de forma simultánea.

Es un protocolo de comunicación multi-maestro de 2 hilos, usado para comunicarse con periféricos de baja velocidad (normalmente, la velocidad es de 100 kbps, para el modo estándar, 400 Kbps para el modo rápido o 3,4 Mbps para el modo de alta velocidad).

I²C usa dos líneas bidireccionales: Serial DATA (SDA) y Serial Clock (SCL) las cuales están conectadas a una línea de referencia mediante resistencias de pull-up, y una línea GND común de la interconexión entre todos los dispositivos conectados al bus.

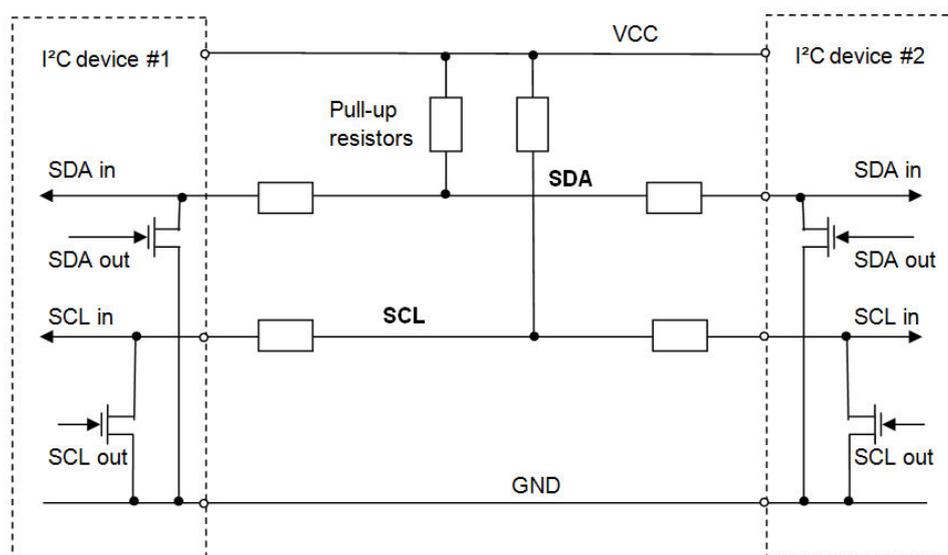


Figura 2: Bus I²C con dos dispositivos conectados. Las líneas SDA y SCL están conectadas con Vcc a través de resistencias de Pull-up.

2.-DESCRIPCIÓN DEL SISTEMA HARDWARE

Cualquier número de maestros y cualquier número de esclavos se pueden conectar a las líneas SCL y SDA y comunicarse entre sí utilizando el protocolo, que define:

- 7 bits para direcciones de los esclavos. Cada dispositivo conectado al bus tiene una única dirección.
- Datos divididos en bytes de 8 bits.
- Un bit de START para controlar el inicio de la comunicación, un bit de STOP para controlar el final de la comunicación un bit de Lectura/Escritura y un bit de acuse de recibo (ACK).
- Un bit para lectura o escritura

START	Slave address	Rd/nWr	ACK	Data	ACK	Data	ACK	STOP
1 bit	7 bits	1 bit	1 bit	8 bits	1 bit	8 bits	1 bit	1 bit

Figura 3: Estructura de una trama de 2 bytes de datos en el protocolo I²C

La condición inicial de la comunicación (bus libre) es cuando las señales SDA y SCL están en estado lógico alto, en este caso, cualquier dispositivo maestro (sólo maestro, un esclavo no puede iniciar una comunicación) puede ocupar el bus, estableciendo la condición de inicio (start). Esta condición de inicio es cuando un maestro pone en estado bajo la línea SDA, dejando la línea SCL en estado alto. El primer byte que se transmite, después de la condición de inicio, contiene siete bits que componen la dirección del dispositivo que se desea seleccionar, y un octavo bit que corresponde a la operación que se quiere realizar con él (0 para escritura y 1 para lectura). Si el dispositivo cuya dirección corresponde a la que se indica en los 7 bits, está presente en el bus, éste contesta con un bit en estado bajo, (bit de reconocimiento, ACK) ubicado inmediatamente del octavo bit que ha enviado el dispositivo maestro. Esto le indica al dispositivo maestro que el esclavo reconoce la solicitud y está en condiciones de comunicarse. Si el bit de lectura/escritura es 0 (escritura), el maestro envía datos al esclavo. Esto se mantiene mientras se continúen recibiendo señales de ACK, y la comunicación concluye cuando se hayan transmitido todos los datos.

En el caso contrario, cuando el bit de lectura/escritura es 1, el maestro genera pulsos de reloj para que el esclavo pueda enviar datos. Después de cada byte recibido, el maestro genera un pulso de reconocimiento. El maestro puede dejar libre el bus generando una condición de parada [2].

2.1.2. Descripción del Bus SPI.

SPI (Serial Peripheral Interface) es un protocolo de comunicación serie síncrono creado por Motorola. SPI es un protocolo de comunicación de un solo maestro (un dispositivo central inicia todas las comunicaciones con los esclavos), que opera con un modo full dúplex, es decir que es una comunicación bidireccional de forma simultánea. SPI no está limitado a palabras de 8 bits, por lo tanto se puede enviar un tamaño de mensaje arbitrario.

SPI está formado por 4 líneas: una señal de reloj (SCLK, Serial Clock), enviada por el maestro a todos los esclavos, todas las señales van sincronizadas con esta línea de reloj, una señal de selección para cada esclavo (SSn, Slave Select), utilizada para seleccionar el esclavo con el cual el maestro se comunica, una línea de datos desde el maestro a los esclavos (MOSI, Master Output Slave Input) y una línea de datos de los esclavos al maestro (MISO, Master Input Slave Output).

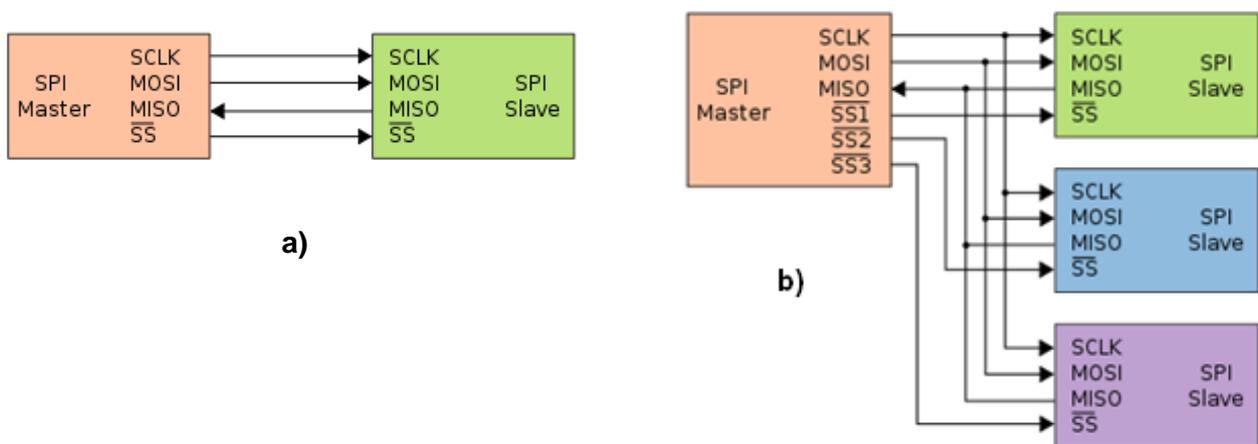


Figura 4: a) Bus SPI con conexión entre un único maestro y un único esclavo y b) un maestro y múltiples esclavos

Los datos se transmiten de manera síncrona con los pulsos del reloj (en cada pulso, el maestro envía 1 bit). Para empezar la transmisión de la cadena de bits, el maestro pone la señal de SS del esclavo que quiere activar a nivel bajo y empieza la transmisión.

2.-DESCRIPCIÓN DEL SISTEMA HARDWARE

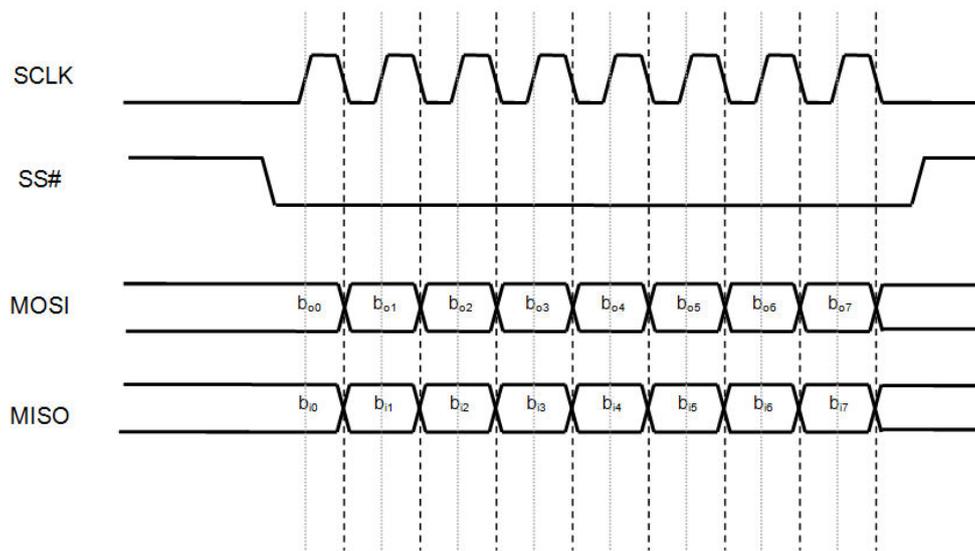


Figura 5: Comunicación SPI simple por flanco de bajada

Los pulsos de reloj pueden estar programados de manera que la transmisión de cada bit se realice de cuatro modos diferentes:

- Modo 0.- Con el flanco de subida sin retraso.
- Modo 1.- Con el flanco de subida con retraso.
- Modo 2.- Con el flanco de bajada sin retraso.
- Modo 3.- Con el flanco de bajada con retraso [3].

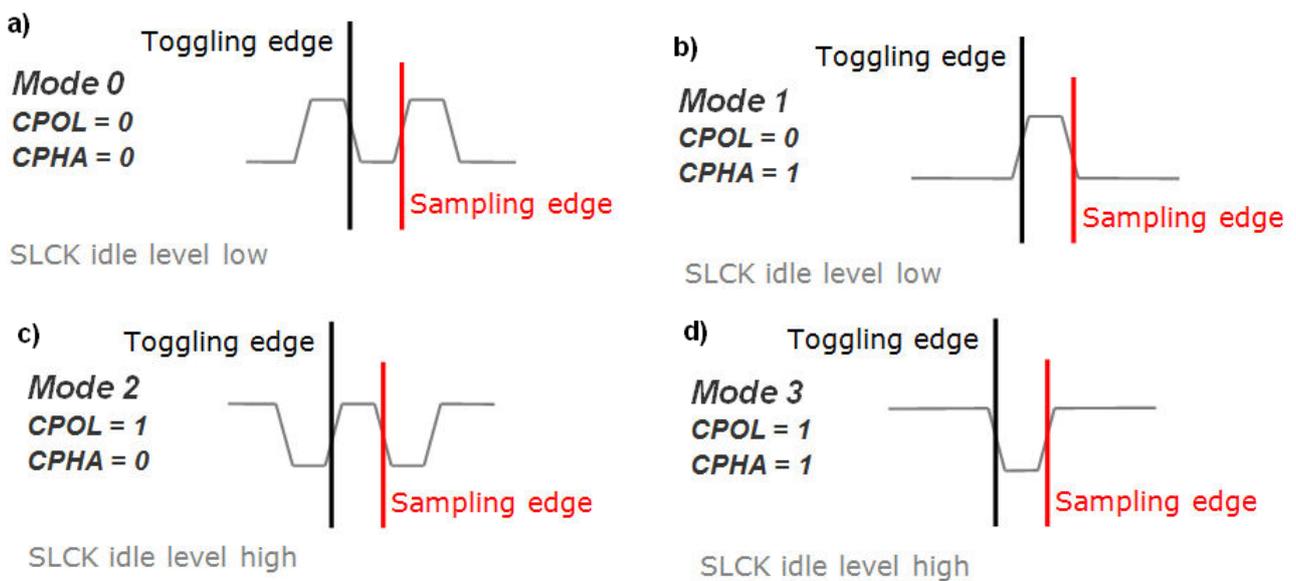


Figura 6: Modos de comunicación SPI, a) Modo A, b) Modo B, c) Modo C d) Modo D

2.-DESCRIPCIÓN DEL SISTEMA HARDWARE

2.1.3. I²C vs SPI. Justificación de la elección del bus SPI.

A continuación se muestra una tabla comparativa con las principales características de ambos protocolos:

SPI	I ² C
Se necesitan 4 líneas de bus: <ul style="list-style-type: none">• SCLK• SS_n• MOSI• MISO	Se necesitan 2 líneas de bus: <ul style="list-style-type: none">• SDA• SCL
La interfaz no requiere resistencias pull-up, con lo que consume menos energía	La interfaz requiere resistencias pull-up, con lo que consume mayor energía
No tiene especificaciones oficiales, depende del componente en cuestión.	Tiene especificaciones oficiales (el protocolo I ² C fue creado por Philips).
Mayor tasa de datos (superior a 10 MHz)	Soporta velocidades de transmisión entre 100 KHz y 400 KHz
Más eficiente en comunicaciones punto a punto (un solo maestro y un solo esclavo)	Más eficiente en aplicaciones multimaestro – multiesclavo.
No tiene direccionamiento de dispositivos	Tiene direccionamiento esquematizado y directo
No posee un mecanismo de reconocimiento de confirmación de recepción de datos	Posee un mecanismo de reconocimiento de confirmación de recepción de datos
Menor sobrecarga con aplicaciones punto a punto	Mayor sobrecarga con aplicaciones punto a punto
Mejor para aplicaciones de flujo de datos	Mejor para comunicaciones entre dispositivos de una misma placa y que se accede a ellos de forma ocasional

El bus I²C es más sencillo al tener menos líneas. Esto significa que necesita menos pines con la interfaz de un circuito integrado. Cuando se comunica con más de un esclavo, I²C tiene la ventaja de direccionamiento en banda, en lugar de tener una línea de selección de chip para cada esclavo. Con I²C te aseguras de comunicar con un esclavo, ya que este devuelve un reconocimiento (ACK). Con SPI, un maestro puede enviar datos a nada en absoluto y no se entera.

Por otro lado, el bus SPI permite la comunicación bidireccional con un mayor rendimiento que I²C (velocidad de transmisión más elevada). El bus SPI no se limita a las palabras de 8 bits, puede enviar cualquier tamaño de los mensajes con contenido y propósito arbitrario. La interfaz SPI no requiere de resistencias Pull-up, lo que se traduce en menor consumo de energía, muy importante en las tendencias actuales de eficiencia energética. SPI es más adecuado para aplicaciones que tienen que ver con los flujos de datos más largos como por ejemplo cuando estamos trabajando con un convertidor de señal analógico-digital de forma continuada.

2.-DESCRIPCIÓN DEL SISTEMA HARDWARE

Debido al mayor rendimiento de SPI frente a I²C (velocidad de transmisión más elevada), a un menor consumo gracias a que no utiliza resistencias de Pull-up y a un flujo de datos más largo se decide trabajar con el protocolo SPI.

2.2. Elección de los componentes electrónicos.

Se realiza una búsqueda exhaustiva de los componentes electrónicos existentes en el mercado siguiendo una serie de especificaciones técnicas requeridas por el sistema multiparamétrico:

- Sistema de adquisición de datos con 16 bits de conversión.
- El amplificador tiene que tener Ganancia Programable.
- Se necesitan varias entradas para conectar al menos 4 sensores.
- Interface SPI.
- Salida digital.
- Conexión USB con el ordenador.

Con estas especificaciones encontramos distintos conversores en el mercado:

Marca	Modelo	Ganancia Programable	Señales de Entrada	Bits de resolución	Interfaz
SEMTECH	SX8725S	1/12 hasta 1000	1 diferencial o 2 de un solo terminal	Hasta 16	SPI
SEMTECH	SX8724S	1/12 hasta 1000	3 diferencial o 6 de un solo terminal	Hasta 16	SPI
SEMTECH	SX8723S	1/12 hasta 1000	2 diferencial o 4 de un solo terminal	Hasta 16	SPI
TEXAS INSTRUMENTS	MSC1210	1 a 128	8 diferencial / un solo terminal	24	SPI
MICROCHIP	MCP3423	1, 2, 4 o 8	2 diferencial	18	I ² C
MICROCHIP	MCP3903	Indica que tiene	4 diferencial	16	SPI
ANALOG DEVICES	AD 7194	1 a 128	8 diferencial / 16 pseudo-diferencial	24	SPI

Consultando las fichas técnicas de los distintos conversores nos decidimos por el modelo SX8724S de la marca SEMTECH. A continuación se muestra una figura del diagrama de bloques:

2.-DESCRIPCIÓN DEL SISTEMA HARDWARE

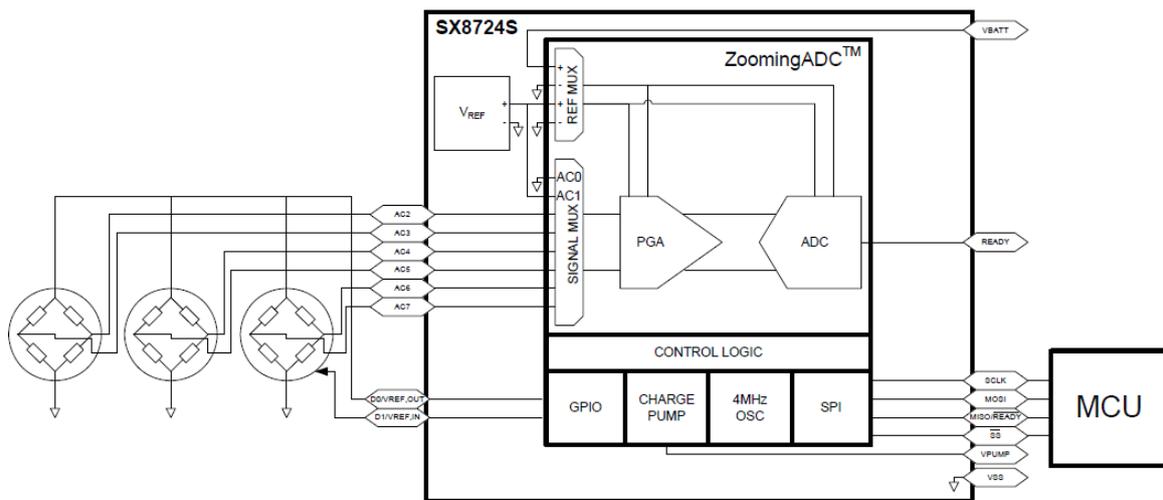


Figura 7: Diagrama de bloques del SX8724S.

Las especificaciones técnicas más relevantes del convertor son las siguientes [4]:

- Sistema de adquisición de datos de 16 bits de conversión.
- Ganancia Programable desde 1/12 hasta 1000.
- Compensación de offset.
- 3 entradas diferenciales o 6 entradas referenciadas a masa.
- Resolución programable.
- Referencia de voltaje interna o externa.
- Alimentación de 2.4V a 5.5V.
- Potencia de funcionamiento baja (250µA para 16 bits a velocidad de 500 muestras por segundo).
- Rango de ruido medio en la salida de 205 a 365 µV.
- Error de ganancia de $\pm 0.15\%$
- Error de offset de $\pm 1\%$

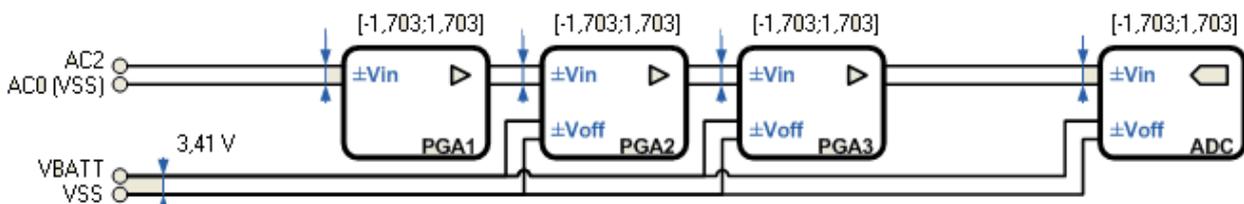


Figura 8: Etapas del convertor SX8724S

3. DESCRIPCIÓN DEL SISTEMA SOFTWARE.

3.1. Organización del programa. Jerarquía.

El software de control del sistema multiparamétrico ha sido desarrollado bajo la plataforma LabVIEW, versión 8.11 (2011).

El programa se ha estructurado en un VI principal y múltiples SubVIs secundarios, además de varias variables globales, todo ello dentro de un mismo proyecto denominado PROJECT SENSORS. El nombre de cada SubVI está guardado en mayúscula y el nombre de cada variable global en minúscula para poder diferenciarlos entre ellos.

En la figura 10 se puede ver una imagen de la jerarquía seguida en el programa principal desarrollado.

Como se puede observar existen 6 niveles diferenciados:

- **Nivel 1:** VI Principal denominado Gas Sensor del que dependen todos los demás SubVIs.
- **Nivel 2:** SubVIs Secundarios en los cuales, internamente dependen de otros SubVIs.
- **Nivel 3:** SubVIs Secundarios dependientes del nivel superior. Estos SubVIs también pueden tener internamente otros SubVIs, como es el caso.
- **Nivel 4:** Es el nivel más bajo de esta jerarquía. Los SubVIs de este nivel, internamente no dependen de otros SubVIs.

3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

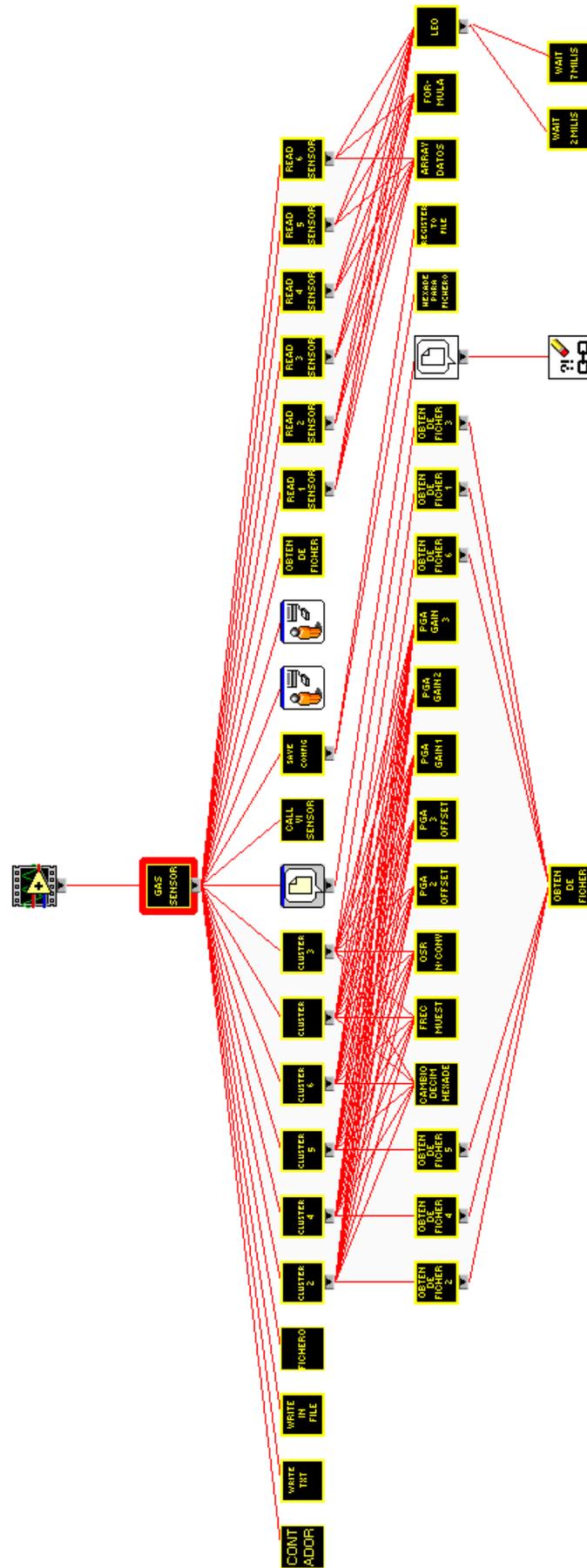
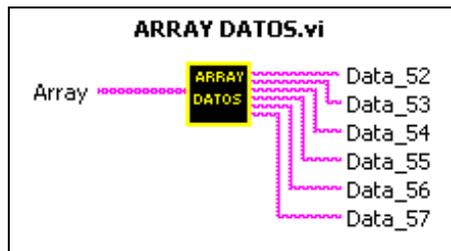


Figura 10: Jerarquía del programa

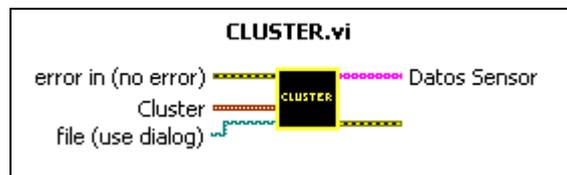
3.2. Descripción de los Nodos Secundarios (SubVIs).

Vamos a describir los distintos nodos secundarios:

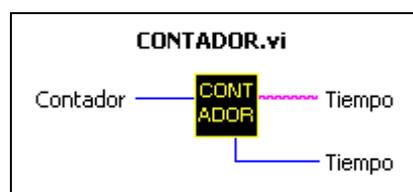
- **ARRAY DATOS.vi:** Tiene como salidas cada elemento del array formado por los valores de cada registro.



- **CLUSTER.vi; CLUSTER2.vi; CLUSTER3.vi; CLUSTER4.vi; CLUSTER5.vi; CLUSTER6.vi:** Crea el array de datos del sensor formado por los valores en hexadecimal de cada registro (provenientes de un cluster formado por las ganancias, los offsets, frecuencia de muestreo, número de conversiones elementales y el *oversampling rate*). Hay 6 CLUSTER porque podemos tener hasta 6 canales de entrada.

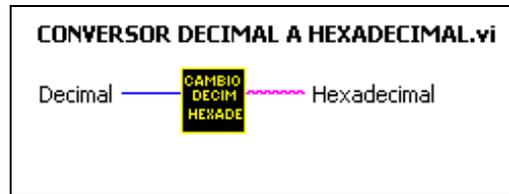


- **CONTADOR.vi:** Es un cronómetro que se pone en marcha cuando se empieza a adquirir el valor de los sensores para tener en todo momento visualizado el tiempo de captura de datos. La salida string muestra el tiempo en el panel frontal, la salida numérica sirve para calcular las muestras por segundo (dividiendo el número de repeticiones del bucle while por este valor) que también se mostraran en el panel frontal.

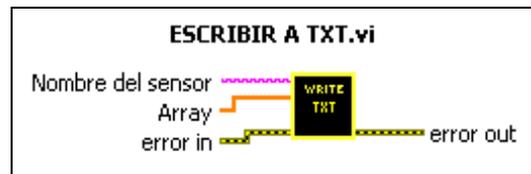


3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

- **CONVERSION DECIMAL A HEXADECIMAL.vi:** Cambia el valor de la entrada en valor decimal a hexadecimal en la salida.



- **ESCRIBIR A TXT.vi:** Guarda los datos de cada sensor (insertados en un array) en un fichero .txt en la ruta especificada C:\DATOS\RESULTADOS denominado Nombre del sensor.txt. El valor del array se cambia a string para guardarlo en el fichero.



- **ESCRIBIR EN FICHERO.vi:** Escribe el valor de los registros en la inicialización del fichero de configuración. Como entrada tiene la ruta del fichero de configuración. En este fichero sólo se guarda el valor del registro RegACCfg5, donde se indica el tipo y número de canal donde tenemos conectado cada sensor.



- **FICHERO.vi:** Crea un fichero en la ruta especificada C:\DATOS\CONFIGURACION denominado CONFIGURACIÓN_fecha_hora.txt y escribe en la primera fila del fichero la fecha y la hora. Se trata de la inicialización del fichero de configuración de los sensores. Como salida ofrece la ruta del fichero de configuración para incluir posteriormente los valores de configuración de cada sensor.



3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

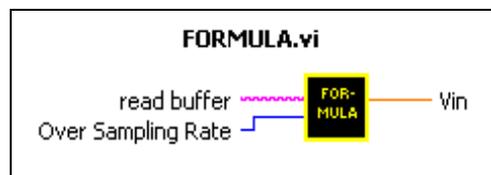
- **FORMATO HEXADECIMAL PARA FICHERO.vi:** El tratamiento de los datos en hexadecimal en el entorno del SX8724S son visualizados como símbolos en un string de LabVIEW. Para ello, a la hora de guardar los valores en hexadecimal en el fichero de configuración es necesario realizar la modificación del valor del registro. Este SubVI realiza dicha modificación.



- **FORMULA.vi:** Es el SubVI donde se realizan las operaciones matemáticas para obtener el valor de voltaje que proviene del sensor. Para ello se captura la trama del Puerto Serie (registros RegACOutLsb y RegACOutMsb) y se realiza la siguiente operación matemática:

$$V = \frac{OUT_{ADC} \times V_{REF} \times OSR}{2^{16} \times (OSR + 1)} (GD_{TOT} - (GD_{off_3} + GD_3 \times GD_{off_2}) \times S \times V_{REF})$$

Donde OUT_{ADC} es el valor de los registros RegACOutLsb y RegACOutMsb.

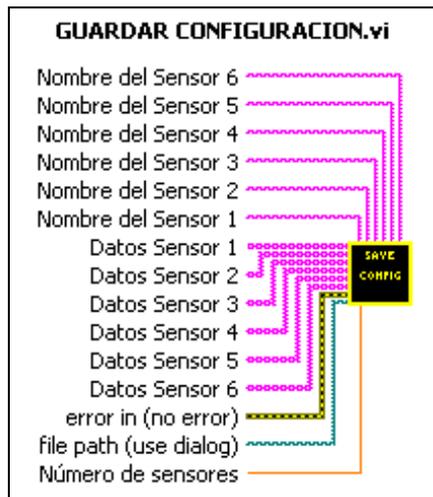


- **FRECUENCIA DE MUESTREO.vi:** La frecuencia de muestreo puede ser 62.5 Hz, 125 Hz, 250Hz y 500 Hz. Cada uno de los valores corresponde a un número en decimal. Este SubVI tiene como salida el valor en decimal de la frecuencia de muestreo a la entrada. Este valor corresponde al registro RegAcCfg2.



3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

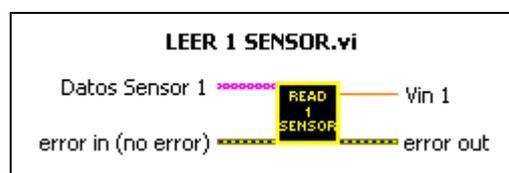
- **GUARDAR CONFIGURACION.vi:** Guarda el fichero definitivo de configuración donde se indica el nombre de cada sensor y el valor de cada registro para cada sensor. Tiene también como entradas la ruta del fichero de configuración creado donde se sustituyen los valores antes guardados menos la fecha y la hora y el número de sensores conectados ya que se guardan solo los valores de los sensores que hay conectados.



- **GUARDAR REGISTROS EN FICHERO.vi:** Guarda el valor de los registros en el fichero de configuración. El formato es el nombre del sensor seguido de los valores de cada registro desde el RegAcCfg0 hasta el RegAcCfg5. La salida registros es lo que se guarda en el fichero de configuración.

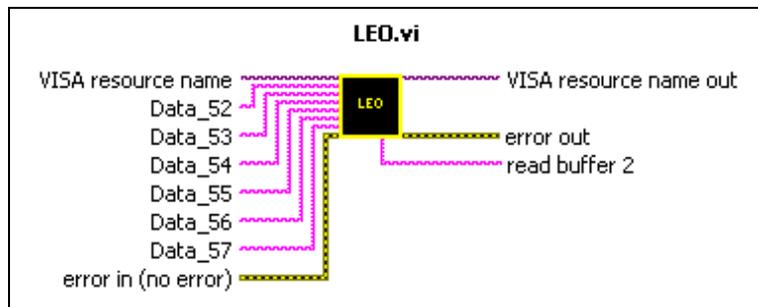


- **LEER 1 SENSOR.vi; LEER 2 SENSORES.vi; LEER 3 SENSORES.vi; LEER 4 SENSORES.vi; LEER 5 SENSORES.vi; LEER 6 SENSORES.vi:** Se obtiene el valor de voltaje ofrecido por el sensor que hay conectado a partir de los valores de los registros almacenados en el array de entrada. Hay 6 porque podemos tener hasta 6 canales de entrada.

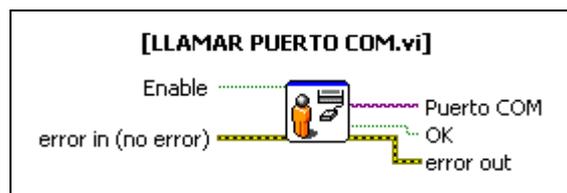


3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

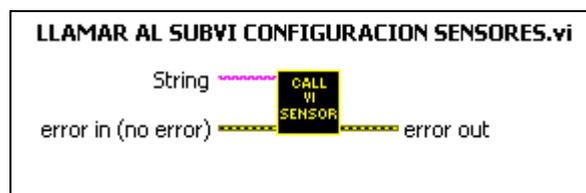
- **LEO.vi:** En este SubVI se envían los valores de los registros mediante el Puerto Serie al SX8742S y se recibe la contestación (trama de datos) donde contiene el valor de voltaje de cada sensor.



- **LLAMAR AL PUERTO COM.vi (PROMPT):** Es un prompt donde se indica el Puerto COM donde tenemos conectado el dispositivo de interfaz de sensores.

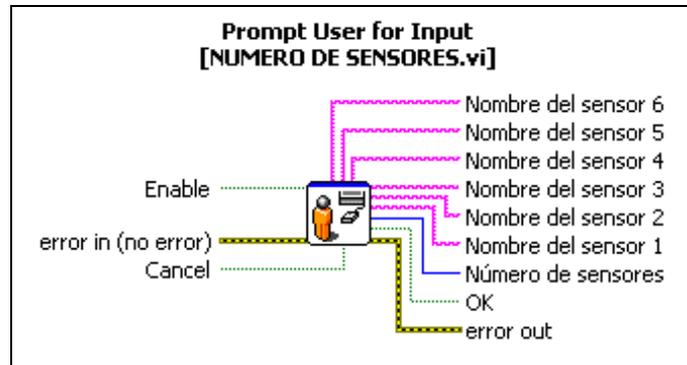


- **LLAMAR AL SUBVI CONFIGURACION SENSORES.vi:** Se ha creado un VI de configuración de sensores. Este SubVI muestra el panel frontal de dicho VI para configurar las entradas de cada sensor. Como entrada se le indica el nombre del sensor.

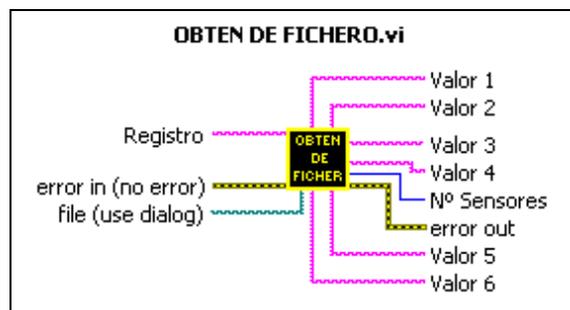


3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

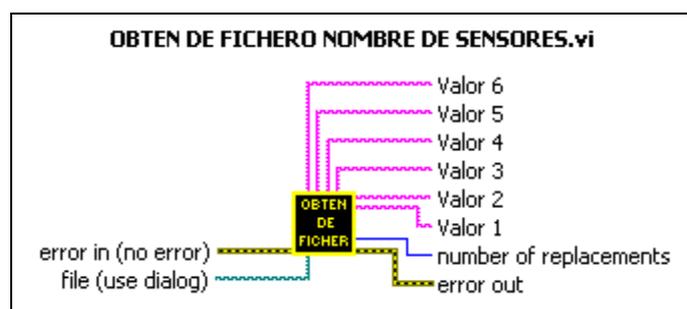
- **NUMERO DE SENSORES.vi (PROMPT):** Es un prompt donde se indica el número de sensores que tenemos conectados y el nombre de cada uno.



- **OBTEN DE FICHERO.vi:** Tiene como salida el valor del registro guardado en el fichero de configuración. La entrada es el nombre del registro y la ruta del fichero de configuración. Como salida tendremos hasta 6 valores del registro indicado (ya que el fichero puede tener hasta 6 valores almacenados) y el número de sensores configurados en dicho fichero.

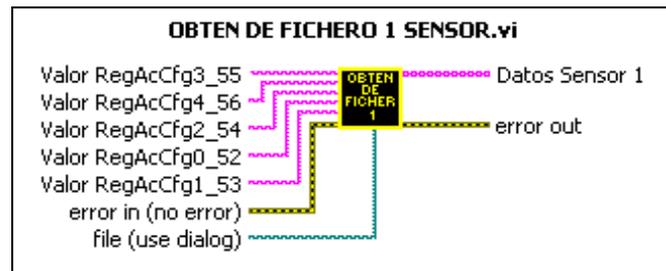


- **OBTEN DE FICHERO NOMBRE DE SENSORES.vi:** Obtiene el nombre de los sensores guardados en el fichero de configuración. También tiene como salida el número de sensores guardados en dicho fichero.



3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

- **OBTEN DE FICHERO 1 SENSOR.vi; OBTEN DE FICHERO 2 SENSORES.vi; OBTEN DE FICHERO 3 SENSORES.vi; OBTEN DE FICHERO 4 SENSORES.vi; OBTEN DE FICHERO 5 SENSORES.vi; OBTEN DE FICHERO 6 SENSORES.vi:** Obtiene el valor del registro RegAcCf5 del fichero de configuración (por eso una de las entradas es la ruta de dicho fichero). El valor de los demás registros son entradas al SubVI y lo inserta todo en el array de Datos de Sensor.



- **OSR Y NUMERO DE CONVERSIONES.vi:** Al igual que la frecuencia de muestreo, el valor del número de conversiones elementales y el *oversampling rate* también corresponde a un número en decimal. Este SubVI tiene como salida el valor en decimal correspondiente a ambos valores. Este valor corresponde al registro RegAcCf0.



- **PGA_GAIN 1.vi, PGA_GAIN 2.vi, PGA_GAIN 3.vi:** Tienen como salida el valor en decimal de la ganancia de la primera, segunda y tercera etapa respectivamente.

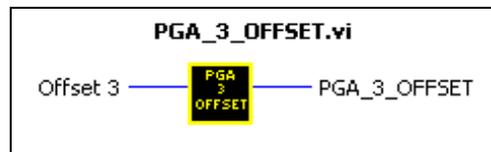


3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

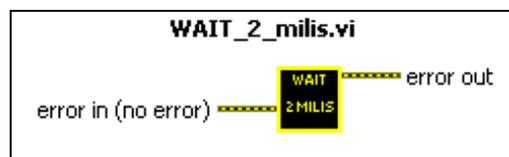
- **PGA_2_OFFSET.vi:** Cada valor de offset de la segunda etapa también corresponde a un número en decimal. Este SubVI tiene como salida el valor de Offset de la segunda etapa en decimal.



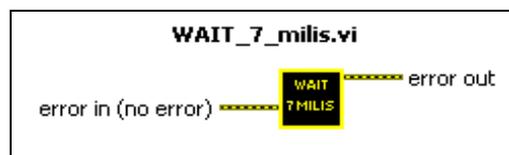
- **PGA_3_OFFSET.vi:** Cada valor de offset de la tercera etapa corresponde a un número en decimal. Este SubVI tiene como salida el valor de Offset de la tercera etapa en decimal.



- **WAIT_2_MILIS.vi:** Se trata de un tiempo de espera al enviar 1 registro por el Puerto COM, para que se pueda procesar el dato de forma correcta.



- **WAIT_7_MILIS.vi:** Se trata de un tiempo de espera al enviar los 6 registros de forma consecutiva por el Puerto COM, para que se puedan procesar los datos de forma correcta.



3.3. Descripción del VI Principal (Programa Principal).

En la figura 11 se muestra el diagrama de flujo del funcionamiento del programa principal.

3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

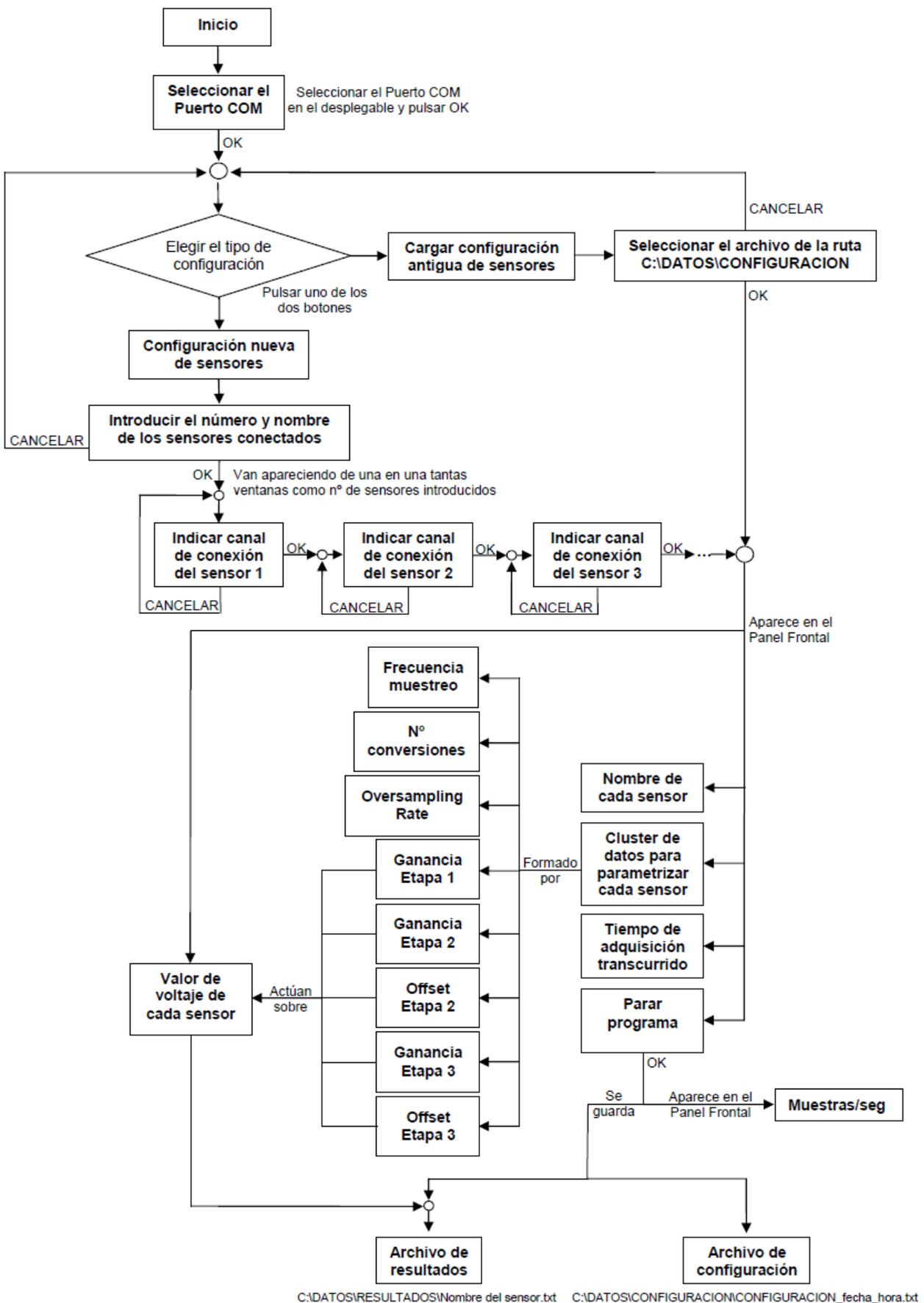


Figura 11: Diagrama de flujo del funcionamiento del programa principal

3.-DESCRIPCIÓN DEL SISTEMA SOFTWARE

El Programa Principal se divide en dos grandes bloques o casos. Cuando arranca el programa aparecen solo 2 botones para pulsar:

- **CONFIGURACIÓN NUEVA DE SENSORES:** Al pulsar este botón se solicita en primer caso el número de sensores que hay conectados y se solicita que se identifique cada uno (se le debe indicar un nombre a cada sensor). A continuación se abre un nuevo panel frontal donde se indica el canal donde está conectado cada sensor. Dicho panel frontal se abrirá tantas veces como sensores hayamos indicado con el fin de configurar cada sensor. Cuando se haya finalizado la configuración de cada sensor aparecen una serie de controles (cluster de parámetros) e indicadores. Como controles aparecen las tres etapas de ganancia y las dos etapas de offset, el número de conversiones elementales, la frecuencia de muestreo y el *oversampling rate*. Como indicador se muestra el valor de voltaje del sensor. Estos controles e indicadores son individuales para cada sensor conectado. También aparecen un indicador general con el Tiempo transcurrido desde el inicio de la captura de datos. También aparece un botón de PARAR PROGRAMA para detener la adquisición de datos y el programa. Cuando finaliza la adquisición de datos (pulsando el botón PARAR PROGRAMA) aparece el número de muestras por segundo obtenido y se crean dos archivos: uno de resultados y otro de configuración.

El archivo de resultados contiene todos los valores de voltaje registrados durante la captura de datos (muestras/segundo) además de los valores finales de las 3 etapas de ganancia, los valores finales de las 2 etapas de offset, el valor final de la frecuencia de muestreo, el valor final del oversampling rate y el valor final del número de conversiones. Dicho archivo de resultados se guarda en la ruta de destino: C:\DATOS\RESULTADOS\Nombre del sensor.txt.

El archivo de la configuración final, se guarda en la ruta de destino C:\DATOS\CONFIGURACIÓN\CONFIGURACION_fecha_hora.txt.

- **CARGAR CONFIGURACIÓN ANTERIOR DE SENSORES:** Al Pulsar este botón se debe indicar la ruta del archivo de configuración que se desea cargar, esto es: C:\DATOS\CONFIGURACION\Nombre del archivo de configuración.txt. A continuación directamente aparecen los controles (cluster de parámetros) e indicadores antes mencionados al igual que el botón PARAR PROGRAMA.

En el Anexo I se explica con detalle y de forma ilustrativa el modo de funcionamiento del programa.

4. RESULTADOS EXPERIMENTALES.

Desde el Departamento de Ingeniería Electrónica se ha creado una tarjeta de conexión y acondicionamiento de sensores químicos resistivos (sensores de gas). Dicha tarjeta adquiere un valor de voltaje de hasta 8 sensores que se conecta a la entrada del conversor, mostrando y tratando los valores de los sensores de gas.

Gracias a esta tarjeta de conexión se ha podido comprobar el correcto funcionamiento tanto del software creado como de todo el sistema hardware multiparamétrico de adquisición de datos. Para ello se han seguido los siguientes pasos:

1. Se conectan 2 sensores químicos resistivos (sensores de gas) a la tarjeta de conexión y acondicionamiento de sensores.

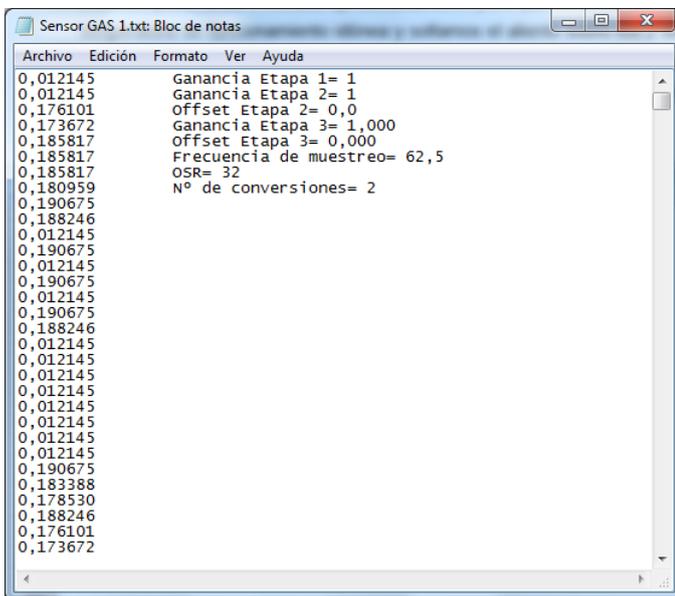


Figura 12: Tarjeta de conexión y acondicionamiento de sensores

2. Se alimenta la tarjeta de conexión a un voltaje de 9 voltios de tensión continua y limitando la corriente a 250 mA.
3. Se conecta la salida del sensor 1 y la salida del sensor 2 de la tarjeta de conexión a la entrada 1 (AC2) y la entrada 2 (AC3) de la placa de evaluación SX8724SEVK respectivamente.
4. Se conecta la placa de evaluación al Puerto USB del ordenador (Se establece la comunicación entre ambos además de alimentar a la placa a una tensión de 5 voltios).
5. Se ejecuta el programa.

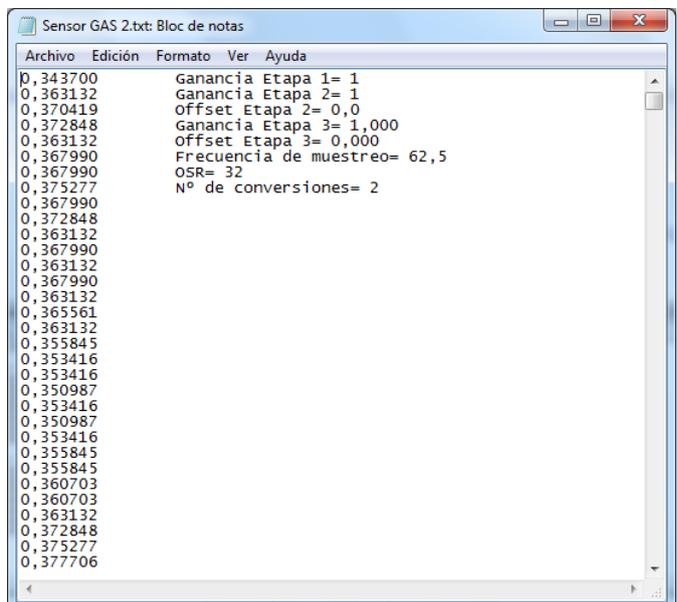
4.-RESULTADOS EXPERIMENTALES

- a. Se establece el puerto COM 4 como el puerto serie donde está conectada la placa de evaluación.
 - b. Se indica que hay 2 sensores conectados, el primero se identifica como Sensor GAS 1 y el segundo como Sensor GAS 2.
 - c. Se configura el Sensor GAS 1 conectado a la entrada AC2, referenciado a masa (Single-ended) y polaridad no invertida.
 - d. Se configura el Sensor GAS2 conectado a la entrada AC3, referenciado a masa (Single-ended) y polaridad no invertida.
6. Acto seguido aparecen en el panel frontal del programa los valores de voltaje de ambos sensores junto al cluster de parametrización (Las 3 etapas de ganancia, las 2 etapas de offset, la frecuencia de muestreo, el oversampling rate y el número de conversiones), el nombre de cada sensor y el tiempo transcurrido.
7. Dejamos pasar unos 3 minutos aproximadamente para que los sensores cojan una temperatura de funcionamiento idónea y soltamos el aliento sobre los 2 sensores durante aproximadamente 5 segundos. Pasados unos 3 minutos más, detenemos el programa obteniendo 3 archivos .txt, uno de resultados por cada sensor y otro de configuración de ambos sensores:



```
Archivo Edición Formato Ver Ayuda
0,012145 Ganancia Etapa 1= 1
0,012145 Ganancia Etapa 2= 1
0,176101 Offset Etapa 2= 0,0
0,173672 Ganancia Etapa 3= 1,000
0,185817 Offset Etapa 3= 0,000
0,185817 Frecuencia de muestreo= 62,5
0,185817 OSR= 32
0,180959 N° de conversiones= 2
0,190675
0,188246
0,012145
0,190675
0,012145
0,190675
0,012145
0,190675
0,012145
0,188246
0,012145
0,012145
0,012145
0,012145
0,012145
0,012145
0,012145
0,012145
0,190675
0,183388
0,178530
0,188246
0,176101
0,173672
```

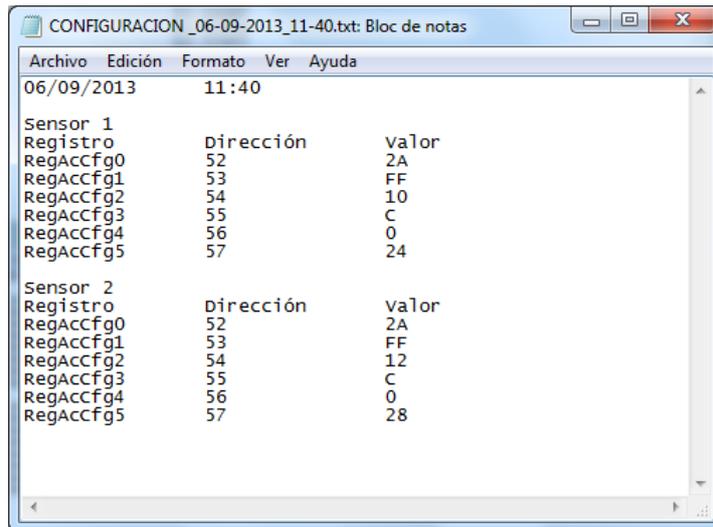
Figura 13: Ejemplo fichero de resultados sensor canal 1



```
Archivo Edición Formato Ver Ayuda
0,343700 Ganancia Etapa 1= 1
0,363132 Ganancia Etapa 2= 1
0,370419 Offset Etapa 2= 0,0
0,372848 Ganancia Etapa 3= 1,000
0,363132 Offset Etapa 3= 0,000
0,367990 Frecuencia de muestreo= 62,5
0,367990 OSR= 32
0,375277 N° de conversiones= 2
0,367990
0,372848
0,363132
0,367990
0,363132
0,367990
0,363132
0,365561
0,363132
0,355845
0,353416
0,353416
0,350987
0,353416
0,350987
0,353416
0,355845
0,355845
0,360703
0,360703
0,363132
0,372848
0,375277
0,377706
```

Figura 14: Ejemplo fichero de resultados sensor canal 2

4.-RESULTADOS EXPERIMENTALES



Sensor 1		
Registro	Dirección	Valor
RegAccfg0	52	2A
RegAccfg1	53	FF
RegAccfg2	54	10
RegAccfg3	55	C
RegAccfg4	56	0
RegAccfg5	57	24

Sensor 2		
Registro	Dirección	valor
RegAccfg0	52	2A
RegAccfg1	53	FF
RegAccfg2	54	12
RegAccfg3	55	C
RegAccfg4	56	0
RegAccfg5	57	28

Figura 15: Ejemplo fichero de configuración

Los archivos de resultados se pueden abrir mediante el programa Excel para obtener las gráficas de los resultados. Para ello abriremos el archivo txt de resultados desde Excel. Nos pedirá una serie de pasos que debemos ignorar pulsando el botón Finalizar. Nos aparecen dos columnas: una con los datos y otra con la configuración de los parámetros del cluster. Seleccionamos la primera columna e indicamos insertar gráfico. Nos aparecen las siguientes gráficas (una para cada archivo):

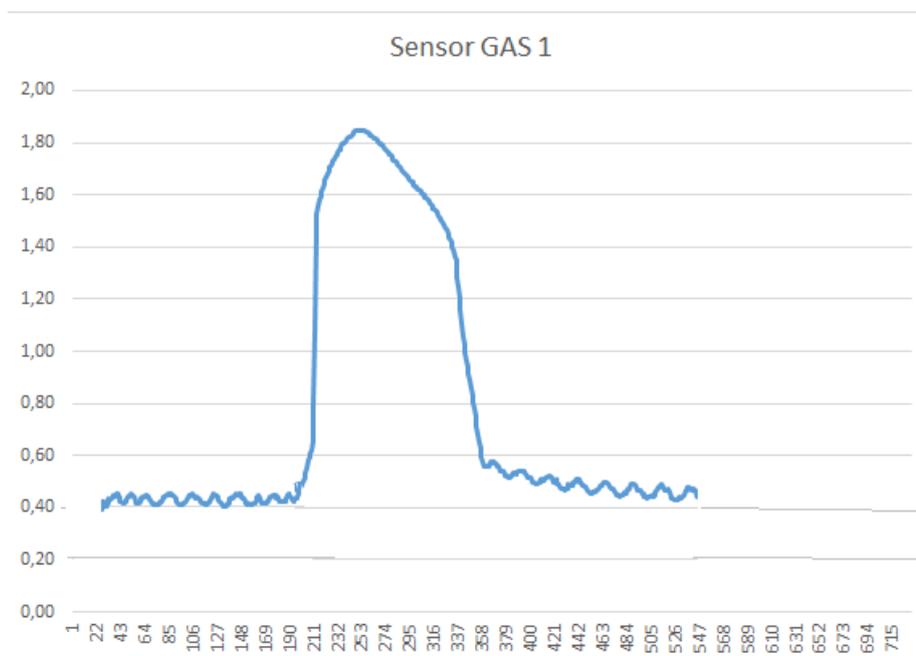


Figura 16: Ejemplo gráfica Canal 1

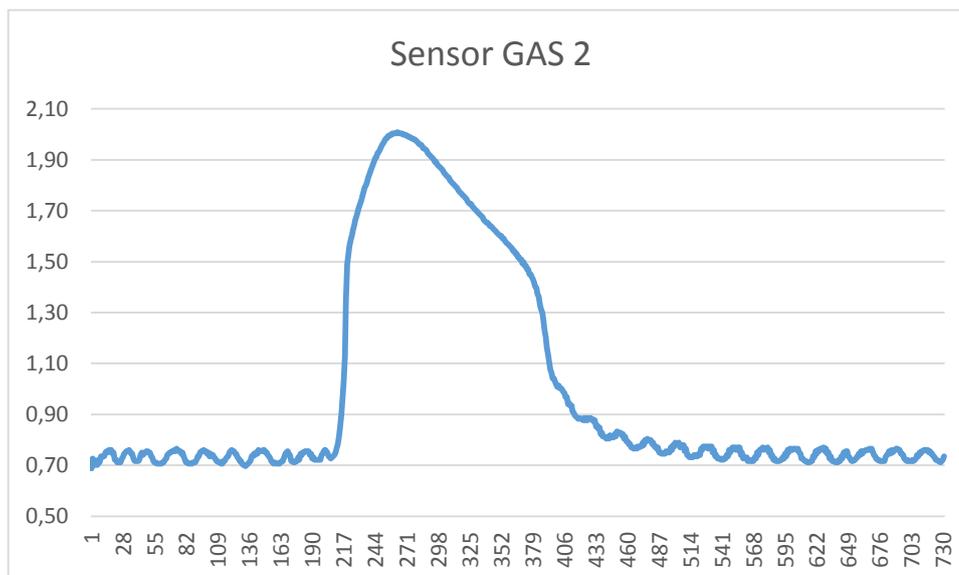


Figura 17: Ejemplo gráfica Canal 2

4.1. Aplicabilidad del sistema multiparamétrico de adquisición de datos.

La idea final del trabajo realizado es obtener un sistema multiparamétrico de adquisición de datos económico con gran capacidad de aplicación en cualquier entorno que se requiera, tal como sanitario, medioambiental, industrial, seguridad-detección, etc.

- En el entorno sanitario se podría crear un sistema de adquisición de niveles de oxígeno o cualquier otro elemento en estado gaseoso.
- En el entorno medioambiental se podría crear un sistema de adquisición de niveles de CO₂ en el ambiente o para indicar zonas de atmósfera explosiva por acumulación de gas.
- En el entorno industrial se podría crear un sistema de adquisición de niveles tóxicos en industrias químicas (atmósferas contaminantes) donde se requiera el uso de equipos de protección individual. También se podría crear un sistema para capturar, identificar y valorar los gases vertidos a la atmósfera en los hornos o calderas de combustión.
- En el entorno de seguridad-detección se podría crear un sistema de detección de incendios con un sistema de adquisición de humo o en los aparcamientos un sistema de detección de CO.

5. CONCLUSIONES.

Al finalizar el trabajo final de grado hemos conseguido los objetivos marcados al inicio del trabajo:

- Diseñar un sistema de adquisición con interfaz USB para cuatro sensores químicos resistivos. Al final hemos conseguido un sistema de hasta 6 sensores.
- El Bus de comunicaciones entre el conversor y el microprocesador es SPI.
- Diseñar una etapa de adquisición de datos con 16 bits de conversión con Amplificador de Ganancia Programable (PGA) para amplificar la señal antes de llegar al conversor. El conversor es de 16 bits con una ganancia programable de 1/12 has 1000.
- Diseñar el software de control del sistema multiparamétrico. Se ha creado un programa mediante la plataforma LabVIEW.11 que cumple todos los requisitos de diseño.
- Estudiar la aplicabilidad del diseño dentro de los múltiples campos de utilidad (sanitario, seguridad, industria, medio ambiente). Como se ha comentado en el punto 4.1, la aplicabilidad del sistema es muy variada dentro de los múltiples campos de utilidad.

Gracias al desarrollo del presente trabajo final de grado he podido adquirir grandes conocimientos en el entorno de programación LabVIEW, además de profundizar en los conocimientos de electrónica adquiridos a lo largo del Grado en Ingeniería de Telecomunicación, Sonido e Imagen.

5.1. Líneas futuras de trabajo.

Como línea futura de trabajo se pretende crear un sistema de calibración de sensores dentro del software desarrollado en esta fase, para así ver la posibilidad de crear un filtrado previo en los distintos tipos de sensores que se pueden conectar al sistema multiparamétrico de adquisición desarrollado.

6. BIBLIOGRAFÍA

6.1. Libros.

PALLAS ARENY, R. (1994). *Sensores y Acondicionadores de señal*. Barcelona: Marcombo.

JACOB FRADEN (1993). *AIP Handbook of Modern Sensors*. AIP Press.

JAMES BRYANT WALT KESTER (1991). *Linear design seminar*. IEEE Computer Society Press.

WEBSTER (1999). *The Measurement, Instrumentation and Sensors Handbook*. CRC Press.

LAJARA VIZCAINO, J.R., PELEGRI SEBASTIA, J. (2011). *Labview. Entrono gráfico de programación (2ª edición)*. Barcelona: Marcombo.

6.2. Páginas Web.

JOHN ARTIUCH. *I²C or SPI Serial Communication, which one to go with?*. 2010.
<http://dev.emcelettronica.com/i2c-or-spi-serial-communication-which-one-to-go>

SWARTHMORE COLLEGE. *Serial Communication*. 2012
[http://www.swarthmore.edu/NatSci/echeeve1/Class/e91/Lectures/E91\(10\)Serial.pdf](http://www.swarthmore.edu/NatSci/echeeve1/Class/e91/Lectures/E91(10)Serial.pdf)

WIKIPEDIA. *I²C*. 2013.
<http://en.wikipedia.org/wiki/I%C2%B2C>

NATIONAL INSTRUMENTS. *NI Discussion Forums: Instrument Control (GPIB, Serial, Visa, IVI)*. (2013).
<http://forums.ni.com/t5/Instrument-Control-GPIB-Serial/bd-p/140>

NATIONAL INSTRUMENTS. *NI Developer Zone* (2013).
<http://zone.ni.com/dzhp/app/main>

6.3. Referencias

[1] [4] Data Sheet SX8724S. ZoomingADC for sensing data acquisition.
<http://www.semtech.com/images/datasheet/sx8724s.pdf>

[2] BYTE PARADIGM. *Introduction to I²C and SPI protocols*. 2013.
<http://www.byteparadigm.com/applications/introduction-to-i2c-and-spi-protocols/>

[3] WIKIPEDIA. *Serial Peripheral Interface Bus*. 2013.
http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

[5] SX8724SEVK USER GUIDE. Evaluation kit for SX9724S Data acquisition System.
http://www.semtech.com/images/datasheet/sx872xsevk-users-guide_ag.pdf

ANEXO I. MANUAL DE USUARIO DEL SOFTWARE

El presente anexo describe cómo instalar y utilizar el programa de una forma sencilla y muy ilustrativa.

En el CD proporcionado está grabado el programa. Simplemente tendremos que crear una carpeta en el directorio C:\ denominada DATOS, y dentro de esta carpeta crearemos dos más denominadas CONFIGURACION y RESULTADOS.

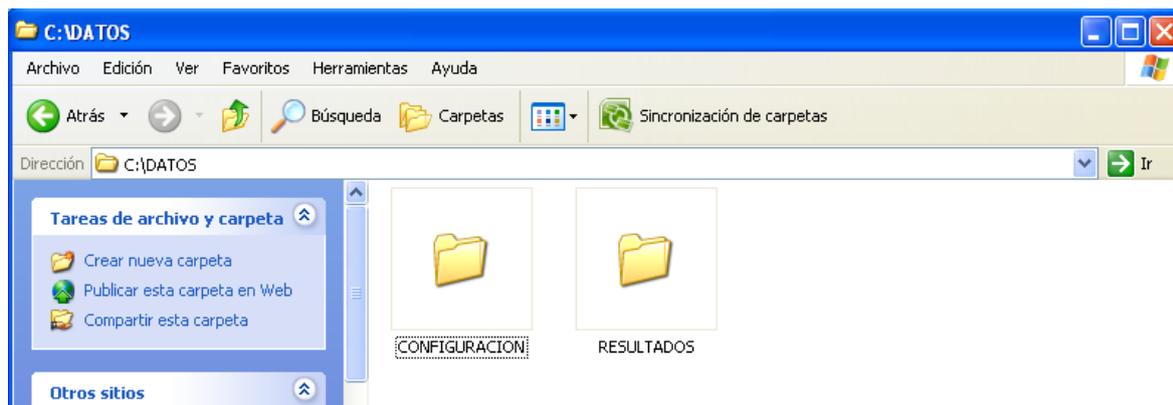


Figura 18: Creación de carpetas en la ruta C.\DATOS

Para poner en funcionamiento el programa haremos doble clic sobre el icono GAS SENSORS.exe y nos aparecerá el siguiente Panel Frontal:

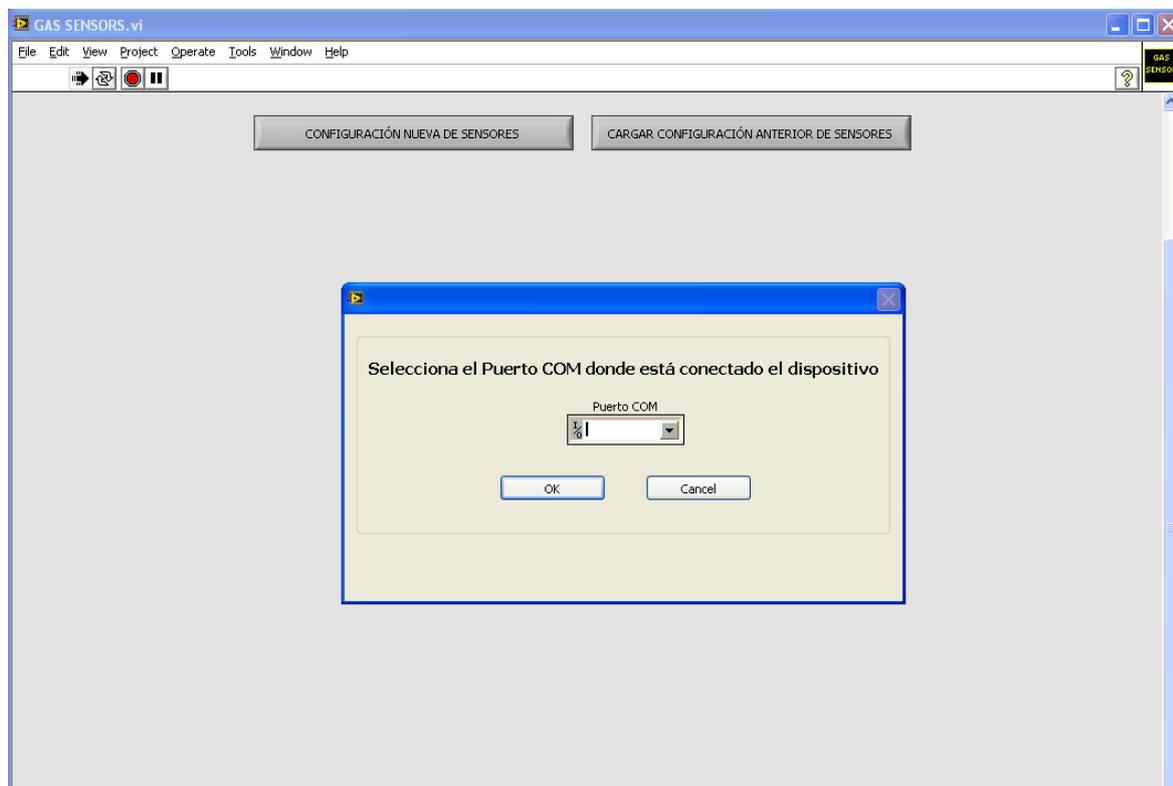


Figura 19: Panel Frontal al iniciar el programa

En el desplegable que aparece en pantalla deberemos seleccionar el Puerto COM donde tenemos conectada la placa de evaluación (SX8724SEVK).



Figura 20: Selección del Puerto COM

A continuación debemos indicar (pulsar el botón) si realizamos una CONFIGURACIÓN NUEVA DE SENSORES o CARGAR CONFIGURACIÓN ANTERIOR DE SENSORES.

- CONFIGURACIÓN NUEVA DE SENSORES:

Se abre una nueva ventana donde indicamos el número de sensores (hasta 6 sensores) y el nombre de cada sensor.

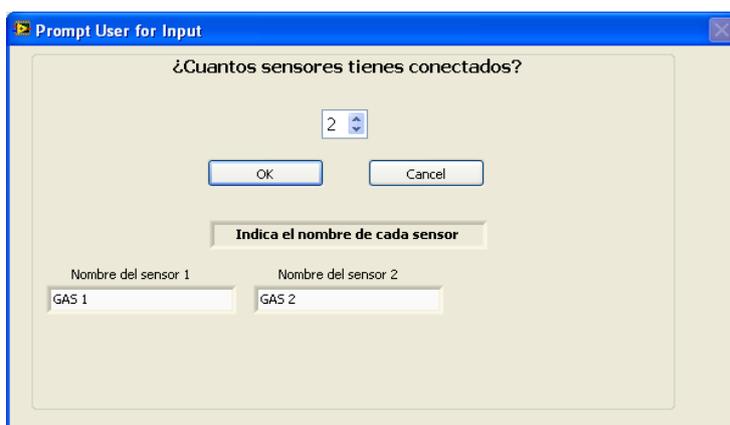


Figura 21: Identificación de los sensores conectados

Al pulsar el botón OK, la ventana se cierra y aparece una nueva ventana para indicar donde tenemos conectado cada sensor. En el desplegable VinP seleccionamos el canal de conexión del sensor indicado (AC2, AC3, AC4, AC5, AC6 o AC7) y pulsamos CARGAR CONFIGURACIÓN.

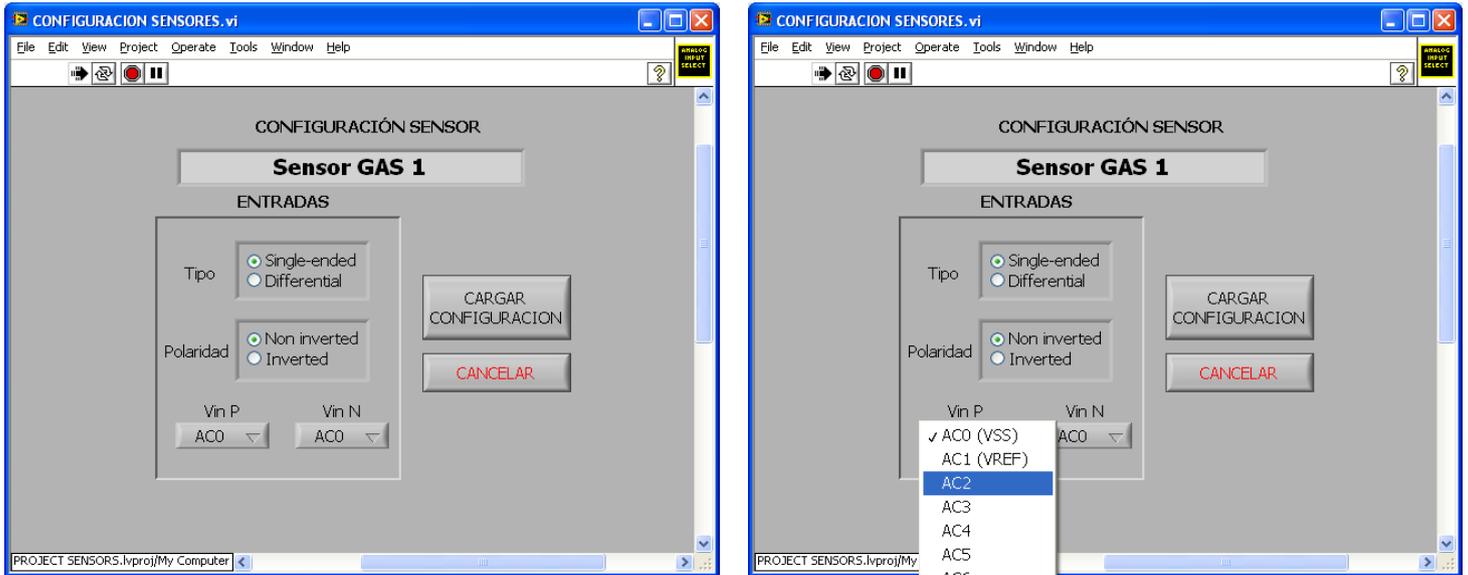


Figura 22: Configuración de los sensores. Conexión de cada sensor

Nos indicará que la configuración se ha cargado con éxito. Pulsamos OK.

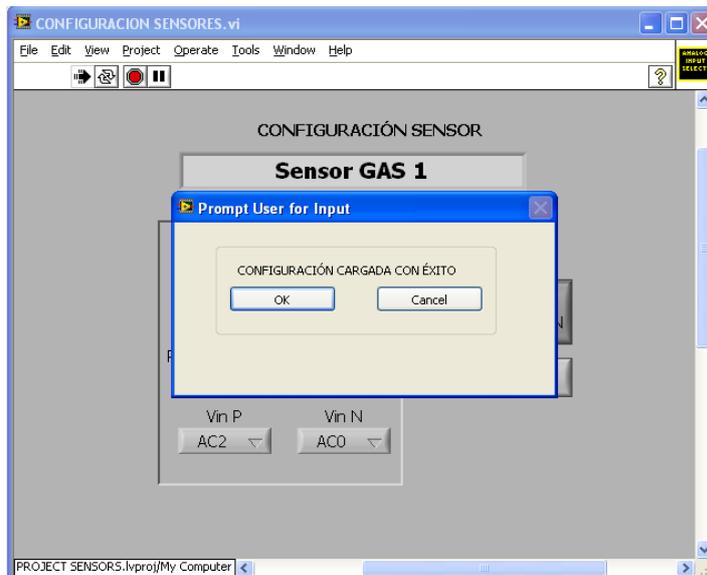


Figura 23: Configuración cargada con éxito

Esta acción se repite también para el sensor 2 y los demás sensores que hayamos indicado en la ventana de la Figura 21. Cada uno conectado en un canal diferente.

Una vez configurados todos los sensores, en el Panel Frontal Principal nos aparecerán los controles (Ganancias de las 3 etapas, offsets de las 2 etapas, frecuencia de muestreo, número de conversiones elementales y el *oversampling rate*) e indicadores (valor de

voltaje del sensor) de cada sensor junto al tiempo transcurrido desde el inicio de la adquisición de datos y el botón de PARAR PROGRAMA (arriba a la derecha).

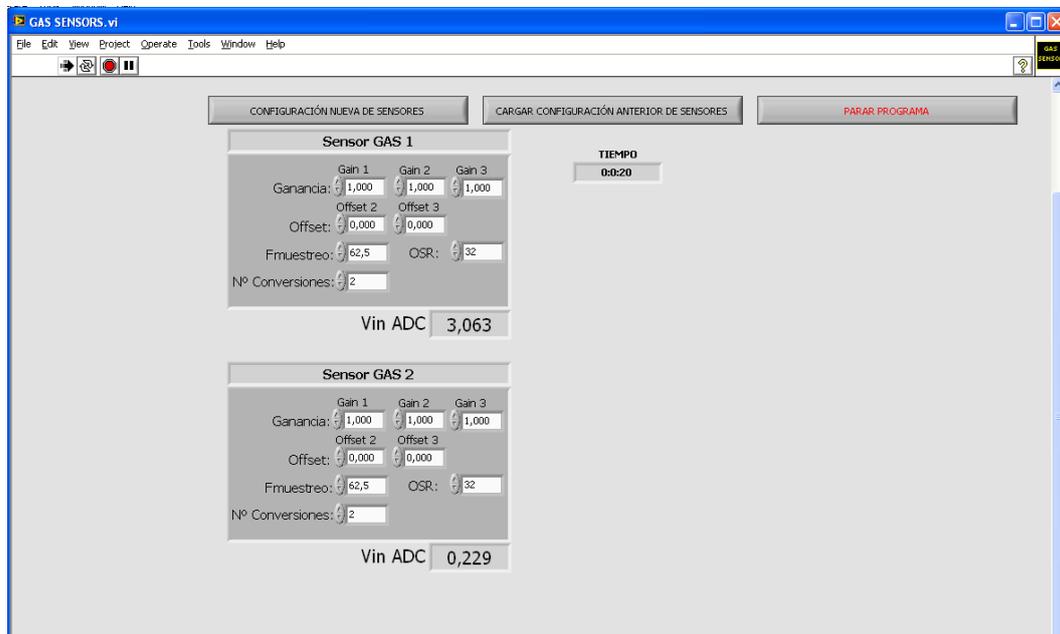


Figura 24: Panel Frontal con 2 sensores

Para terminar la adquisición de datos pulsamos el botón PARAR PROGRAMA. Se indica el número de muestras por segundo adquiridas para cada sensor y finaliza el programa.

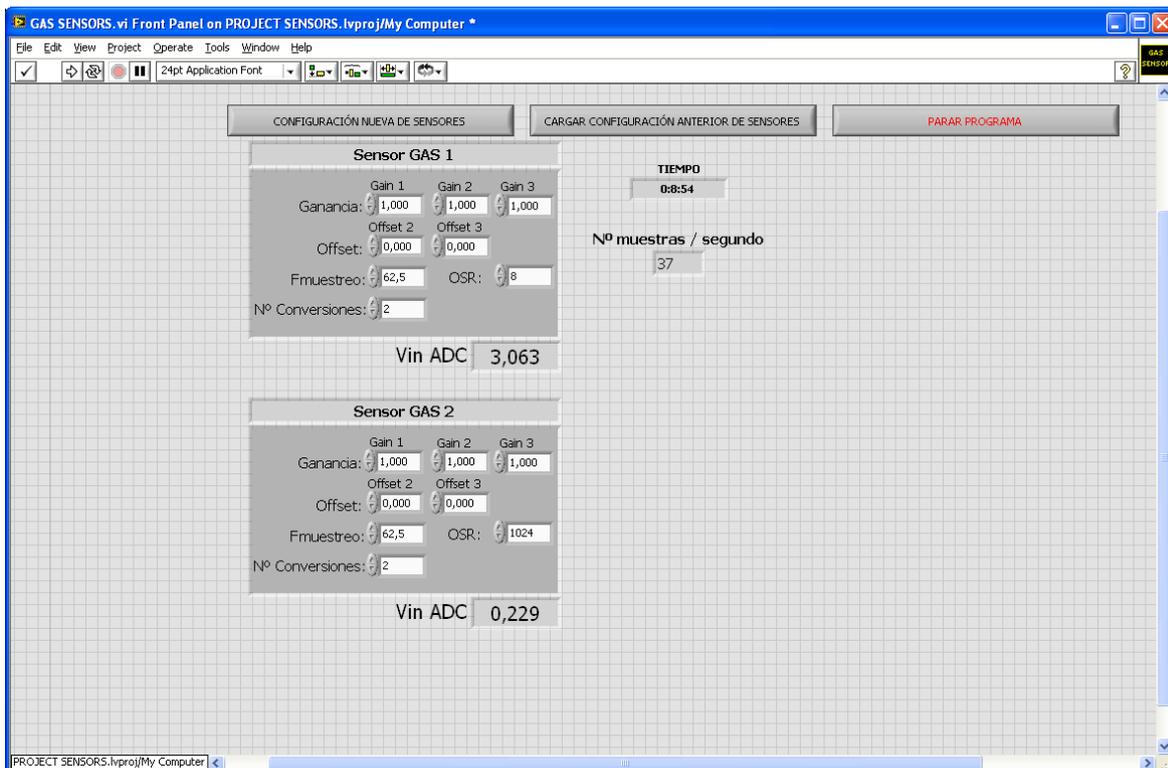
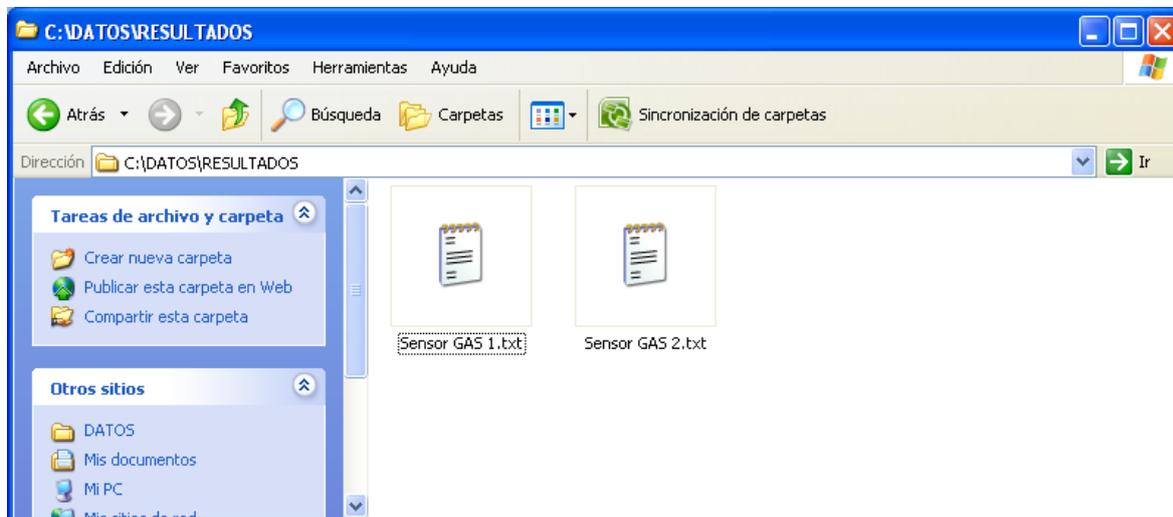
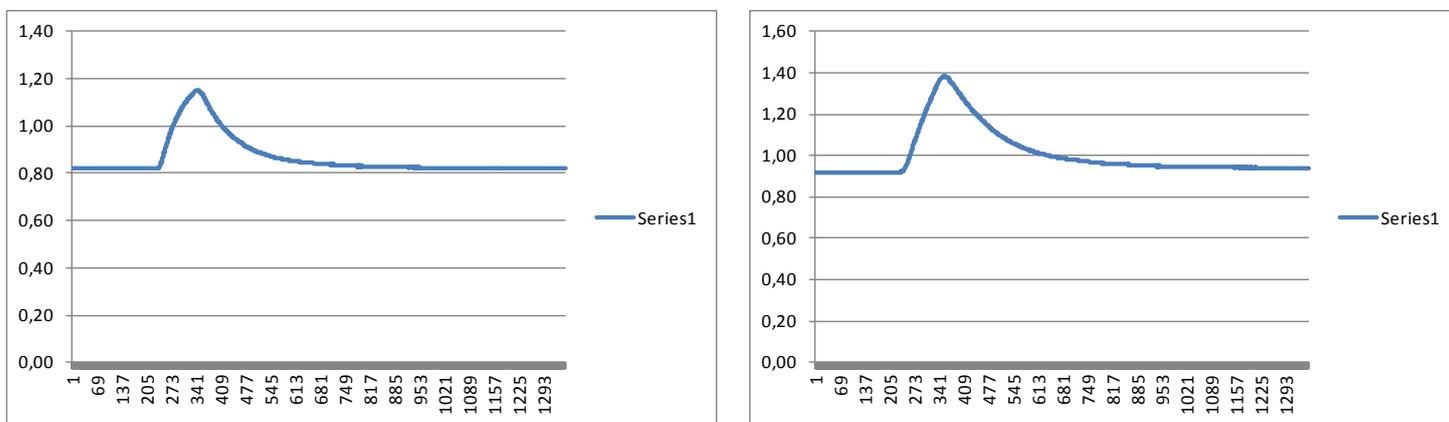


Figura 25: Finalización del programa

En la ruta C:\DATOS\RESULTADOS tendremos los archivos en formato .txt obtenidos de la adquisición realizada. Se crea un archivo por sensor.

**Figura 26:** Archivos obtenidos

Si abrimos estos archivos con Microsoft Office Excel y realizamos los ajustes necesarios, podremos visualizar las gráficas de la lectura de cada sensor.

**Figura 27:** Gráficas obtenidas

- CARGAR CONFIGURACIÓN ANTERIOR DE SENSORES:

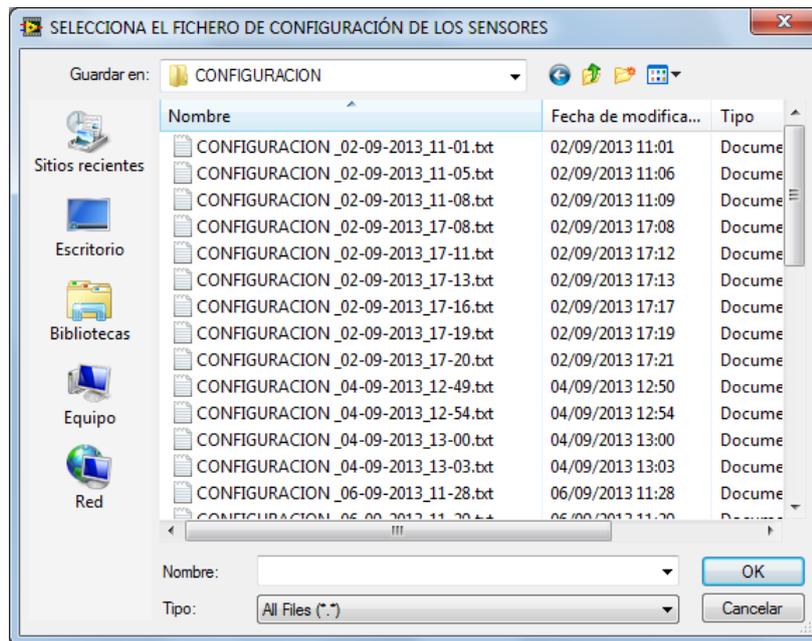


Figura 28: Selección del fichero de configuración

Al pulsar este botón se abre una ventana nueva donde debemos indicar el fichero de configuración que deseamos ejecutar. Al pulsar Ok se siguen los mismos pasos a partir de la figura 24