

## CAPÍTULO 6. FASE III: PLACA DE CALIBRACIÓN

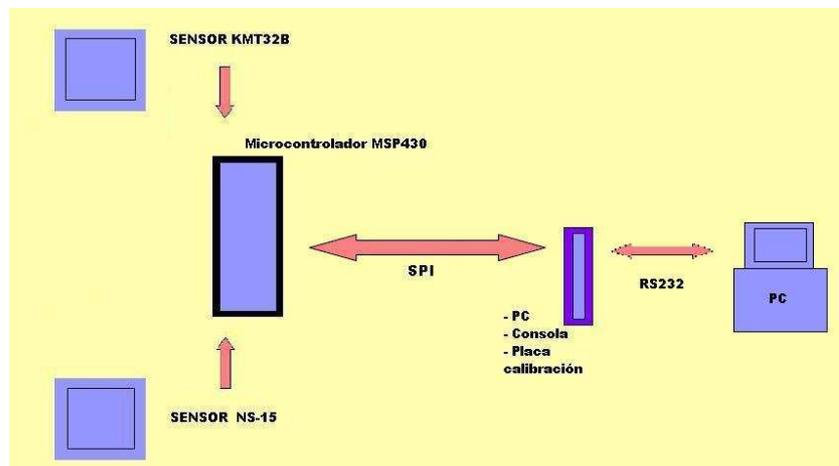
### 6.1 INTRODUCCIÓN

A la hora de calibrar el sistema final, necesitamos una placa de calibración dotada de interfaz SPI y que, además, conozca el “idioma” esto es, los comandos con los que trabaja el sistema.

Con ese fin la última fase de este proyecto es la programación del microcontrolador PIC que lleva una nueva placa de calibración creada a la que se quiere capacitar con ese sistema de comunicación. La placa esta capacitada para calibrar hasta 10 sistemas a la vez y puede comunicarse a través de SPI o I<sup>2</sup>C .

### 6.2 ESQUEMA DE FUNCIONAMIENTO

La placa de calibración es un interfaz entre el PC y el sistema a calibrar. En el PC habrá un programa que se encarga de controlar la temperatura del horno climático, el motor que hace girar el puesto de medida y de tomar las medidas de los sensores en cada posición. Este programa, cuando quiere saber algo del sensor le pregunta mediante el uso de comandos establecidos. Esos comandos en realidad le llegan a la placa de calibración, que actúa como intérprete entre el PC y el sistema sensor. Así, el PC envía un comando que le llega a la placa, y ésta, según qué le hayan pedido, se lo pide al micro del sistema sensor. Éste, a su vez, le responde a la placa que le entrega el resultado al PC. La comunicación entre PC y placa es a través de RS232. A continuación podemos ver el esquema final de todo el conjunto:

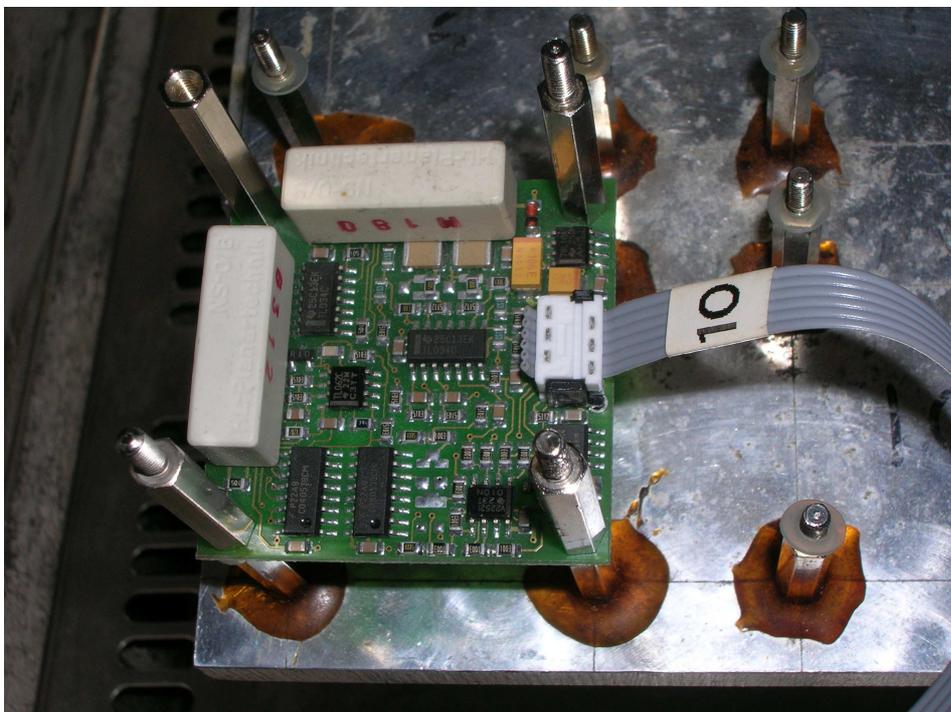


La idea era no alterar, en la medida de lo posible, el programa del PC, si no que, con los mismos comandos con los que se comunica con sistemas no SPI pudiera hacerlo con éstos. De esta manera, aunque se han añadido algunos comandos, la estructura de los mismos no se ha alterado. De esta forma se simplifica mucho la adaptación del programa del PC.

### 6.3 DESCRIPCIÓN DEL FUNCIONAMIENTO

La forma en la que se trabaja con la placa para calibrar el sistema se detalla a continuación.

En la misma placa se pueden conectar hasta 10 sistemas. Una vez conectados, la placa girará hasta su posición inicial. Una vez allí hará una medida del ángulo y almacena el valor obtenido. Esta operación se repite para cada uno de los valores de inclinación que queremos meter en la tabla de la memoria flash (LUT). En nuestro caso tomamos 31 medidas. 15 negativas, el cero y 15 positivas. Una vez que tenemos los distintos valores para cada ángulo de inclinación se rellena la tabla de la memoria flash. Este proceso, además se hace dentro de un horno climático para obtener los valores a distintas temperaturas.



## 6.4 PROGRAMA IMPLEMENTADO

Para la placa se usó el software implementado en las placas anteriores con una serie de actualizaciones para las nuevas características que tiene la placa. Básicamente se trataba de dotar a ésta de comunicación SPI con los sistemas. Además hacerlo de manera que los comandos usados anteriormente no variaran.

El programa debe leer el comando que se le manda vía PC y transmitírselo a el sistema de medida. Además de leer el sistema de medidas y mandárselo al PC.

Se añadió un nuevo comando que permitiera cambiar entre comunicación por SPI o comunicación por I<sup>2</sup>C .

El módulo principal del programa **tsezmain.c** se encarga de configurar las interrupciones, fijar pines de entrada y salida, llamar a las funciones para inicializar los módulos UART, SPI, I<sup>2</sup>C , inicializar estados y variables e inicializar interrupciones. Luego es un bucle infinito donde se manejan los datos llegados vía RS232 al micro.

Para todo ello se usa dentro del módulo **protocolv1\_4.c** la función **ProtocolHandler**. Esta función lleva como parámetro el byte enviado por el PC, y según sea éste así va actuando sobre el sistema. En una sección posterior se podrán ver los distintos posibles comandos. La función lo que persigue es distinguir los distintos tipos de instrucciones que se le van dando y según sean, pedir una cosa u otra al sistema. De esta forma tiene que distinguir si se le pasan datos, direcciones de memoria, etc. Para ello se usan funciones switch y variables de tipo flag que nos van llevando hasta el destino deseado.

La nueva función íntegra es la de la parte de comunicación SPI que se encarga de enviar y recibir los datos al y del sistema de medidas. Dentro del módulo **spi.c** hay tres funciones: **SPI\_Init**, **Read\_Slave\_SPI** y **SPI**. La primera se encarga de inicializar el protocolo de comunicación, la última envía el byte y lee lo que le manda el esclavo. La función **Read\_Slave\_SPI** se encarga de gestionar el número de bytes que envía y recibe. Básicamente hay tres tipos de bytes, los comandos, las direcciones y los datos. Según sea uno de estos así procede la función. La encargada de distinguir el tipo de byte es la función **ProtocolHandler**. Usa un sistema de flags de la siguiente manera:

	Comando	Dirección	Dato
cFlag = 1	X		
cFlag = 0 i = 1		X	
cFlag = 0 i = 2			X

Así, según sea el byte se le asigna el caso comando, dirección o dato y al llamar a la función **Read\_Slave\_SPI** se le pasa, además del byte, el caso del que se trata.

## 6.5 COMANDOS

A la hora de comunicarnos con nuestro sistema lo haremos de una forma concreta. Para ello se usan una serie de comandos. Hay que distinguir entre dos tipos de comandos. Los comandos asignados al sistema sensor y los comandos asignados a la placa. El primer tipo son comandos de tipo byte (char) que el micro del sistema medidor reconoce y actúa en consecuencia. Están definidos dentro del programa del MSP430. El segundo tipo de comandos son más complejos y se le pasan a la placa para que ésta actúe sobre el sensor. Los reconoce el software implementado en el PIC.

Los comandos que entiende el sistema sensor (MSP430) son:

instrucción	Byte	SIMO (hacia el sensor)	SOMI (desde el sensor)	Explicación
NOP	1	00H		
Set Zero (1 +4 Byte)	1 2 3 4 5	01H	00 MSB.. LSB.. MSB.. LSB	Fija el valor actual del ángulo como cero. Offset para nuevas medidas.
Setback Zero (1 Byte)	1	02H	00	Pone el offset a cero.
Software reset (1 Byte)	1	03H	00	El controlador espera hasta que se reinicie vía Watchdog (tarda 4ms).

Lectura Inclinación  (5 Bytes)	1 2 3 4 5	06H 00H 00H 00H 00H	00 MSB eje X LSB eje X MSB eje Y LSB eje Y	Valor en grados, el MSB primero, con: 8000H = overflow negativo 7FFEh = overflow positivo 7FFFH = error interno Ejemplo de valores válidos: C568H = -15000 => -15° 0000H = 0 => 0° 3A99H = +15001 => +15.001°
Lectura ángulo de rotación  (4 Bytes)	1 2 3 4	07H 00H 00H 00H	00 1.Byte (MSB) 2.Byte 3.Byte (LSB)	Valor en grados. Rango: 0° - 179.999° Ejemplo de valores válidos: 000000H = 0 => 0° 02BF1FH = 179999 => 179.999°
Lectura de valores del sensor de inclinación.  (5 Bytes)	1 2 3 4 5	16H 00H 00H 00H 00H	00 MSB eje X LSB eje X MSB eje Y LSB eje Y	Valores dados por el sensor de inclinación, sin pasar a grados.
Lectura de valor del seno y del coseno (ángulo de rotación)  ( 7 Bytes)	1 2 3 4 5 6 7	17H 00H 00H 00H 00H 00H 00H	00 1.Byte (MSB) Sine 2.Byte Seno 3.Byte (LSB) Seno 1.Byte (MSB) Coseno 2.Byte Coseno 3.Byte (LSB) Coseno	Valor del Seno y del Coseno, sin offset. Rango: 0 - 166000 Ejemplos de valores válidos: 000000H = 0 028870H = 166000
Temperatura (3 Bytes)	1 2 3	10H 00H 00H	00 MSB Temperatura LSB Temperatura	Salida de la temperatura medida, en grados centígrados. Ejemplo de valor válido: 0135H = 309 => 30.9°

Leer Flash (3 Bytes)	1 2	20H xxH – Dirección	00 52 xx - Contenido	Leer y devolver un byte de la memoria Flash.
Escribir en Flash (4 Bytes)	1 2 3 4	21H xxH - Dirección xxH – Contenido 00H	00 77  xx – Dirección  xx – Contenido	Escribir el contenido especificado en la dirección especificada. El valor devuelto sirve para verificar la escritura
Escribir continuamente en Flash (3 or more Bytes)	1 2 3...	57H – 'W' xxH – Dirección Inicio xxH - Content	00 57  xx – Dirección Inicio  xx – Contenido	Escribe continuamente uno o más bytes en la memoria Flash, empezando en la dirección de inicio. El valor devuelto se usa para verificación.
Borrar los datos de la memoria Flash (1 Byte)	1	23H	00	Borra todos los datos de todos los segmentos de la memoria Flash

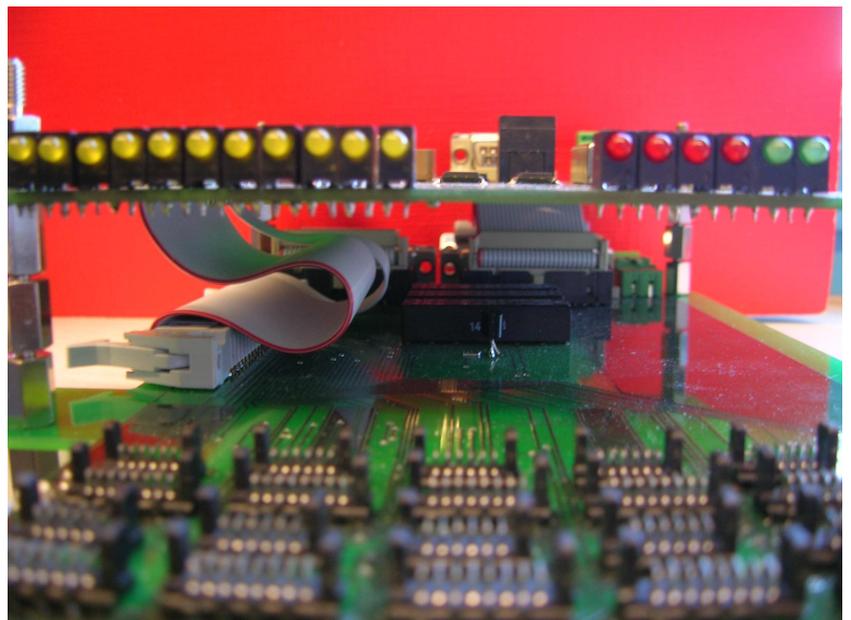
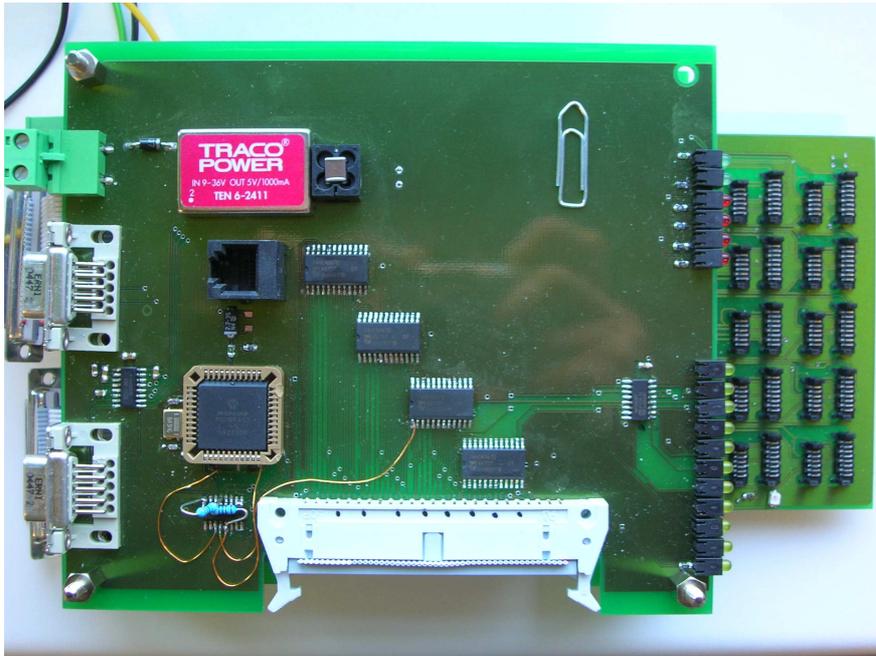
Los comandos que entiende la placa de calibración (PIC) son:

Comando	Enviado a la placa	Devuelto por la placa	Comentario
Sacar versión	"V"	"HL-Planar Microsystems_Demob oard V1.4"	Este comando puede ser usado para comprobar la conexión con la placa.
<b>DPL2</b> + <b>KMS32</b> Set Zero	"N" \$4 "S" \$01 "R"	Devuelve cuatro bytes interpretados como binarios. Primer par de bytes representan la inclinación en el eje X. Segundo par de bytes representan la inclinación en el eje Y.	Fija el valor actual del ángulo como cero. Offset para nuevas medidas.
<b>DPL2</b> + <b>KMS32</b> Setback Zero	"N" \$0 "S" \$02 "R"	No Respuesta.	Pone el offset a cero.
<b>DPL2</b> + <b>KMS32</b> Software Reset	"N" \$0 "S" \$03 "R"	No Respuesta.	El controlador espera hasta que se reinicie vía Watchdog (tarda 4ms).
<b>DPL2</b> + <b>KMS32</b>	"N" \$1 "S" \$4 "R"	\$8	Lee en la memoria Flash en la dirección

Comando	Enviado a la placa	Devuelto por la placa	Comentario
Leer contenido memoria FLash	"S" \$Register "R"	Contenido del registro	"Register"
<b>DPL2 + KMS32</b> Escribir en Flash	"N" \$1 "S" \$5 "R" "S" \$Register "R" "S" \$Data "R"	\$10 \$11 Contenido del registro	Escribe en la memoria Flash en la dirección "Register" el dato "Data"
<b>DPL2 + KMS32</b> Leer ángulo de inclinación	"N" \$4 "S" \$6 "R"	Responde con 4 bytes. El primer par de bytes representan eje X y el Segundo eje Y	
<b>DPL2 + KMS32</b> Leer ángulo de rotación	"N" \$3 "S" \$7 "R"	Responde con tres bytes para la rotación	
<b>DPL2 + KMS32</b> Leer Temperatura	"N" \$2 "S" \$16 "R"	Responde con 2 bytes para la temperatura	
<b>DPL2 + KMS32</b> Tomar valores medidos por el sensor de inclinación, si ángulo. Para rellenar la tabla (LUT)	"N" \$4 "S" \$22 "R"	Responde con 4 bytes. El primer par de bytes representan eje X y el Segundo eje Y	
<b>DPL2 + KMS32</b> Tomar valores de seno y coseno, sin offset	"N" \$6 "S" \$23 "R"	3 primeros bytes del seno 3 siguientes bytes del coseno	
<b>DPL2 + KMS32</b> Borrar la memoria Flash	"N" \$0 "S" \$35 "R"	No responde	Borra todos los segmentos de la memoria Flash.

La estructura del comando es la siguiente. Existen dos posibilidades, que se trate de un comando o que se trate de un dato (dirección o dato propiamente). El segundo caso siempre va precedido del primero. En el

caso del comando primero va la letra “N” que nos introduce a la orden comando, a continuación se especifica el número de bytes devueltos, luego la letra “S” que introduce el comando, luego el número que identifica a cada comando y por último la letra “R” para enviar. Cuando hay que dar además, direcciones y/o datos, la estructura empieza con “S” seguido de la dirección o dato y por último la “R” para enviar.



Proyecto Final de Carrera:  
Diseño y desarrollo de sistema para medida de la inclinación y rotación