



Departament d'Enginyeria Electrònica Elèctrica i Automàtica

Sistema para el control en lazo cerrado de la temperatura de trabajo de una agrupación integrada de microsensores de gases

AUTOR: Jordi Colomer Campos.
DIRECTOR: Eduard Llobet Valero.

FECHA: Junio / 2003.

Índice

1. Introducción	5
1.1 Objeto del proyecto	5
1.2 Antecedentes	5
1.3. Estudio de los sensores	6
1.3.1 Principio de funcionamiento.....	6
1.3.2 Fabricación	9
1.3.3 Aplicaciones	11
1.4 Posibles soluciones	11
1.4.1 Control analógico	11
1.4.2 Control digital	11
 2. Soluciones	 13
2.1 Primera solución	13
2.1.1 Modelo matemático con Matlab	13
2.1.2 Simulación con Pspice	18
2.1.3 Montaje y pruebas con el prototipo	19
2.2 Solución adoptada	22
2.2.1 Diseño del circuito	24
2.2.1.1 Circuito regulador de tensión	24
2.2.1.2 Circuito de consigna	25
2.2.1.3 Filtro <i>Butterworth</i>	26
2.2.1.4 Fuente de corriente Howland	28
2.2.1.5 Circuitos adaptadores de señal	33
2.2.2 El microcontrolador	34
2.2.3 Programa	39

3. Fabricación del circuito	43
3.1 Fabricación del prototipo de la placa de control	43
3.2 Fabricación de la placa base	45
3.3 Fabricación de la placa de control	49
4. Ensayos	54
4.1 Pruebas del control de temperatura	54
4.1.1 Pruebas con prototipo	54
4.1.2 Pruebas con el sistema modular	68
4.2 Pruebas con gases	79
5. Presupuesto	87
5.1 Cuadro de precios	87
5.1.1 Material	87
5.1.2 Diseño y montaje	88
5.2 Aplicación de precios	89
5.2.1 Placa base	89
5.2.2 Placa de control	90
5.2.3 Diseño y montaje	89
5.3 Resumen presupuesto	92
6. Evaluación del sistema	94
6.1 Mantenimiento y ajustes	94
6.2 Aplicaciones	95
7. Conclusión	97

Anexo 1:	Programa en lenguaje C	99
	Programa en lenguaje ensamblador	104
	Archivo pic12f6x.h	113
Anexo 2:	Manual de consulta rápida para el usuario	118

1. INTRODUCCIÓN

1.1 Objeto del proyecto

Realización de un lazo de control que regule la temperatura de los sensores de gases de tecnología *micro-hotplate*. El sistema será inicializado por ordenador, y utilizará los sensores del grupo de investigación de sensores de gases y nariz electrónica del departamento de ingeniería electrónica de la Universidad Rovira i Virgili. Se realiza el proyecto con el fin de conseguir una mejora en las mediciones de gases, y evitar perturbaciones debidas a las variaciones de la temperatura de trabajo en la capa activa del los sensores.

1.2 Antecedentes

Los sensores utilizados para este proyecto requieren una temperatura elevada para lograr su óptimo funcionamiento. Están basados en un principio de reacción electroquímica, altamente relacionado con la temperatura del sensor.

El sistema utilizado actualmente para fijar su temperatura está basado en una fuente de corriente Howland.

Donde una corriente fija, dependiente de una tensión de referencia, es aplicada a cada membrana del sensor.

La membrana es tratada como una resistencia fija, y no se tiene en cuenta que su valor varía con la temperatura. Las variaciones de la temperatura del entorno modificarán la resistencia del *heater*, variando así la temperatura del mismo y produciendo errores en las señales de lectura. Debido a que el sistema actual no dispone de un lazo de control que garantice la estabilidad de la temperatura de trabajo, podemos obtener resultados diferentes a temperaturas de referencia iguales, ya que estos dependen de la temperatura del entorno.

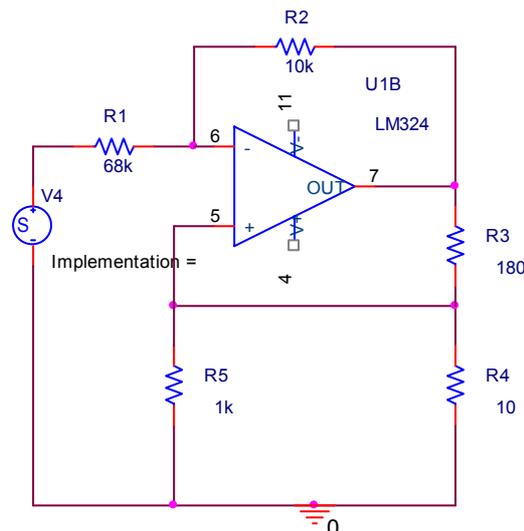


Figura 1. Circuito actual de excitación del *heater*

Como se aprecia en la figura 1, el circuito de excitación del *heater* no dispone de ninguna realimentación sobre la tensión V_1 controladora, con la que pueda realizar el control. Es una fuente Howland controlada por tensión quien determina la temperatura aproximada del sensor. Sin tener en cuenta sus variaciones por las perturbaciones de la temperatura del entorno.

Debido a la necesidad de hacer las medidas de una forma fiable, sin que afecte el factor externo de la temperatura, se necesita realizar un control automático que regule la temperatura en el *heater* y que asegure su estabilidad.

1.3 Estudio de los sensores

1.3.1 Principio de funcionamiento

Los sensores de gases utilizan óxido metálico como material sensor, normalmente SnO_2 . Cuando un cristal de óxido metálico como el SnO_2 es calentado a una temperatura elevada, el oxígeno es absorbido en la superficie del cristal con una carga negativa. Las cargas negativas provienen de la banda de conducción del óxido metálico. Las cargas atrapadas en la superficie generan una zona de vaciamiento alrededor de las partículas de óxido. De este modo, en la frontera entre granos aparece una barrera de potencial que se opone al paso de cargas (eléctricas) ver figura 2.

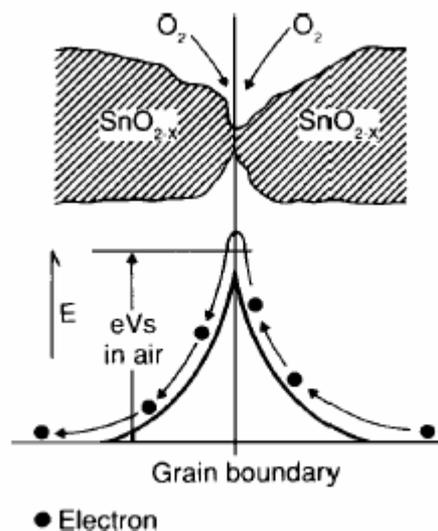


Figura 2. Modelo de barrera de potencial de un entre-grano (en ausencia de gases)

La resistencia eléctrica del sensor es atribuida a esta barrera de potencial. En presencia de un gas reductor, la densidad de las cargas negativas de la superficie disminuye, así pues, la altura de la barrera en la interfaz entre granos se reduce (figuras 2 y 3). La reducción de la barrera disminuye la resistencia del sensor.

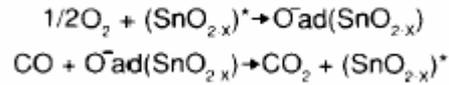


Figura 3. Esquema de la reacción entre CO y el oxígeno absorbido en el SnO_2

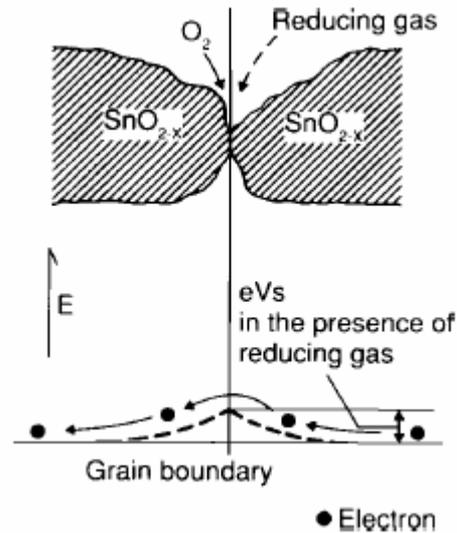


Figura 4. Modelo de barrera de potencial de un entre-grano (en presencia de gases)

La relación entre la resistencia del sensor y la concentración de gas reductor puede ser expresada por encima de un cierto rango de concentración de gas, con la siguiente ecuación.

$$R_s = A[C]^{-\alpha}$$

Donde: R_s = Resistencia eléctrica del sensor

A = Constante

C = Concentración de gas

α = Pendiente de la curva de R_s

1.3.2 Fabricación

1.3.2.1 Fabricación del *micro-hotplate*

El integrado de *micro-hotplates* con series de cuatro microsensores son fabricados en obleas de doble cara pulida, con sustratos de Si tipo P <1 0 0>, de 300 μm de espesor ($4-40 \Omega \cdot \text{cm}$). La estructura del dispositivo se compone de básicamente una capa sensible al gas, los electrodos, capas aislantes y una calentadora de polisilicio. El proceso tecnológico de fabricación de los sensores consta de los siguientes pasos:

- (1) Deposición de la capa de membrana. Las membranas dieléctricas se componen por una capa de Si_3N_4 de 0.3 μm de grosor depositada por LPCVD. Cada pastilla tiene 4 membranas, de $900 \times 900 \mu\text{m}^2$.
- (2) Deposición y aplicación de un *heater* con forma de serpentín, de polisilicio dopado con POCl_3 de 6 Ω /cuadrados de resistencia. El coeficiente de la temperatura de resistividad (TCR) del polisilicio depende del nivel de dopaje, y para el nivel de dopaje aplicado, un valor de 6.79×10^{-4} fue medido en los dispositivos. El *heater* se utiliza también como un sensor de temperatura.
- (3) Una capa de 0.8 μm de grosor de SiO_2 es depositada para aislar el *heater* de los electrodos y la película sensora.
- (4) Abrir de contactos del *heater* para ser accesibles.
- (5) La definición de los electrodos de Pt de 0.2 μm^2 se realiza por *lift-off*. Una capa delgada (20 nm) de Ti fue depositada antes del Pt para promover la adhesión de electrodo. El área del electrodo era $400 \times 400 \mu\text{m}^2$. La figura 1 muestra el *heater* y la configuración de los electrodos.

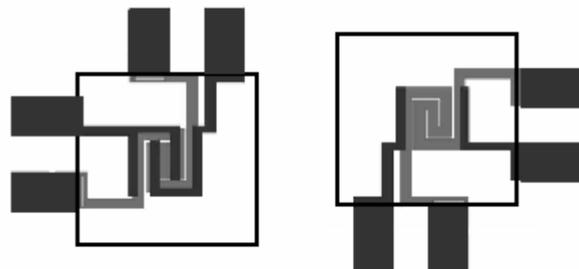


Figura 5. Planta de la membrana

- (6) Aplicación de la máscara del atacador de la cara posterior.
- (7) Deposición de la capa sensora en el área de electrodo.
- (8) Atacado del silicio de la cara posterior con KOH a 70 $^{\circ}\text{C}$ (40 % del peso) para crear las membranas térmicamente aisladas.
- (9) Conexión del hilo y encapsulado. Cada pastilla se monta en un encapsulado TO-8. Los hilos de oro de 52 μm de diámetro se utilizan para realizar una conexión con ultrasonidos. Para prevenir la rotura de las membranas debido a la expansión del aire en la cavidad situada bajo las membranas a la hora de calentar el *heater*, las pastillas no fueron pegadas directamente a la superficie del paquete metálico, sino que se mantienen elevadas

utilizando dos espaciadores laterales de silicio. La figura 2 muestra una vista de una pastilla encapsulada donde las 4 membranas son claramente visibles.

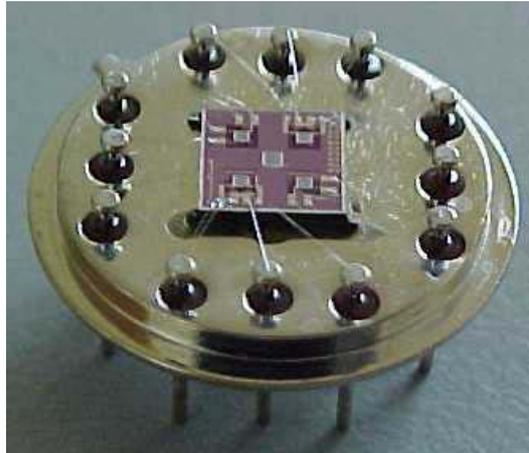


Figura 6. Pastilla encapsulada

1.3.2.2 Deposición de la película activa

Mientras en el proceso descrito previamente, la deposición de la capa sensora es generalmente el último paso - especialmente cuándo la serigrafía o las técnicas de capa de gota son utilizadas -, en el caso informado aquí, el serigrafiado se hace antes del atacado del silicio de la cara posterior y envasado. Este procedimiento evita dañar la membrana de sensor, y así, la lleva a un rendimiento excelente (alrededor de 95 % del trabajo de microsensores después de troquelado y encapsulado).

La capa sensora se compone nanopolvo de SnO de un 5 μm de grosor. El espesor de la capa activa se puede controlar entre 2 y 20 micras después del disparo de átomos, ajustando la viscosidad de pasta orgánica, la máscara y el espesor impreso. El óxido del estaño se compra en Sigma-Aldrich, Inc. (tin (IV) Oxide nanopowder, referencia del producto. 54,965-7).

La pasta que se imprime está preparada utilizando un medio orgánico basado en el therpineol. Después que el nanopolvo de óxido de estaño se mezcla con el medio, se obtiene una pasta con la viscosidad requerida (la mezcla pesa el 58 % del total). La pasta se imprime en las obleas semi-fabricadas utilizando una máquina alta precisión de serigrafía que permite la alineación de la máscara en una cara.

Para obtener una adherencia buena entre el sustrato y la película activa, la oblea es calentada a 60 °C. Después de la impresión, las obleas se dejan reposar para recuperar la temperatura ambiente. A continuación son secadas durante 15 minutos a 125 °C para que el medio orgánico quede completamente seco, y entonces pueda ser transportada a un horno de cinta rotativa durante una hora a una temperatura homogénea de 600 °C.

Las obleas depositadas están entonces listas para el atacado de la capa posterior, el troquelado y envasado. Para proteger la superficie de la capa delantera y el material sensible depositado, las obleas se cubren con un protector mecánico.

1.3.3 Aplicaciones

Algunas de las posibles aplicaciones de los sensores de gas están en la detección de gases como el hidrógeno en el mantenimiento de transformadores, las baterías, uso en la industria siderúrgica, los escapes domésticos de gas, las alarmas, los detectores portátiles de gas, los analizadores de alcohol, los detectores de vapor orgánico o los detectores de disolventes en fábricas o en la industria de los semiconductores.

Los sensores de este tipo pueden utilizarse en infinidad de aplicaciones no solo del tipo industrial. Los sensores de la llamada “nariz electrónica” se aplican en la industria de la alimentación, donde detectan los aromas que desprenden los alimentos y los comparan con patrones para detectar el compuesto volátil.

1.4 Posibles soluciones

Una vez determinado el objetivo del proyecto, se plantean las diferentes opciones con las que se puede realizar el proyecto.

1.4.1 Control analógico

La posibilidad de hacer un control analógico es la que más atrae desde un principio, y todos los esfuerzos se centran en esta dirección.

Un control basado en la regulación de la potencia del *heater* es el que se considera más oportuno. Se seguirá utilizando la fuente Howland que se utilizó en el diseño del proyecto anterior y se realimentará el sistema para lograr un control de temperatura.

1.4.2 Control digital

Se plantea en un principio la posibilidad de utilizar un microcontrolador para hacer el lazo de control. Esta alternativa es descartada debido a la pérdida en la velocidad del sistema un dispositivo programable. Esta pérdida de velocidad se producirá debido a la adquisición de los datos, y a su procesamiento. Un control analógico resolvería este problema.

2. SOLUCIONES

2.1 Primera solución

La realización del circuito analógico se basa en la inyección de corriente para elevar su temperatura deseada. Una tensión de referencia determinará la temperatura de trabajo del sensor. La realimentación se realizará mediante la medida del voltaje en bornes del sensor, para así lograr un control de la temperatura.

2.1.1 Modelo matemático con Matlab

Desde un principio se trabaja para realizar un lazo de control analógico. Se empiezan a recopilar datos para hacer un modelo del sensor y poder estudiar el modelo matemático del lazo de control.

El dato que se desconocía antes de realizar el proyecto, y que era necesario para la realización del mismo era la τ del sensor, es decir, el tiempo de reacción. Fueron hechas una serie de medidas en el laboratorio de sensores por Alex Vergara, y los datos resultantes son estos.

Tabla respuesta RH de sensores						Sensores de capa fina
Sensor	Columna	Medida 1	Medida 2	Medida 3	Promedio	Sensores de ccapa gruesa
	Medida (-5: -6)	Rh(tau)	(ms)	(ms)	(ms)	
1	2	*	*	*		
2	4	20.324	21.846	20.266	20.812	
3	6	20.198	18.501	21.040	19.913	
4	8	20.518	19.872	18.500	19.630	
5	10	14.209	12.320	12.691	13.073	
6	12	*	*	*		
7	14	14.375	13.101	13.271	13.582	
8	16	13.429	12.773	12.911	13.038	
Medida (-4: -6)						
	V					
1	2	*	*	*		
2	4	20.442	20.876	20.594	20.637	
3	6	19.245	20.877	19.512	19.878	
4	8	18.425	19.109	19.682	19.072	
5	10	12.723	13.289	13.007	13.006	
6	12	*	*	*		
7	14	13.307	12.851	11.943	12.700	
8	16	13.167	11.984	12.183	12.445	

Medida (-3: -6) V					
1	2	*	*	*	
2	4	22.598	21.491	21.679	21.923
3	6	19.761	20.037	21.349	20.382
4	8	19.459	19.459	19.966	19.628
5	10	12.634	12.111	13.010	12.585
6	12	*	*	*	
7	14	13.357	13.468	12.539	13.121
8	16	13.420	13.924	12.875	13.406
Medida (-2: -6) V					
1	2	*	*	*	
2	4	21.703	21.373	21.287	21.454
3	6	20.414	20.116	20.414	20.315
4	8	19.586	19.258	19.212	19.352
5	10	13.313	12.430	13.124	12.956
6	12	*	*	*	
7	14	11.753	12.156	11.753	11.887
8	16	13.243	13.065	13.081	13.130

Tabla 1. Tiempos de respuesta (τ) de los sensores

La τ de los sensores que se utilizarán en el proyecto es de aproximadamente 20 ms.

Así que gracias a estas medidas realizadas se comprueba que los sensores se pueden modelar como un sistema de primer orden, con $\tau=20$ ms.

$$G(s) = \frac{1}{1 + 0.02 * s}$$

Una vez hallado el modelo del sensor, se puede trabajar con modelos matemáticos realizados en Matlab.

Se conoce que la resistencia del sensor responde a esta recta:

$$R = R_o \cdot (1 + 6,79 \cdot 10^{-4} \cdot (T - T_o))$$

Con las mediciones realizadas en los laboratorios, que nos permiten saber las temperaturas del sensor respecto de su voltaje e intensidad, la función de transferencia y la recta de la temperatura se realizó este modelo en Matlab.

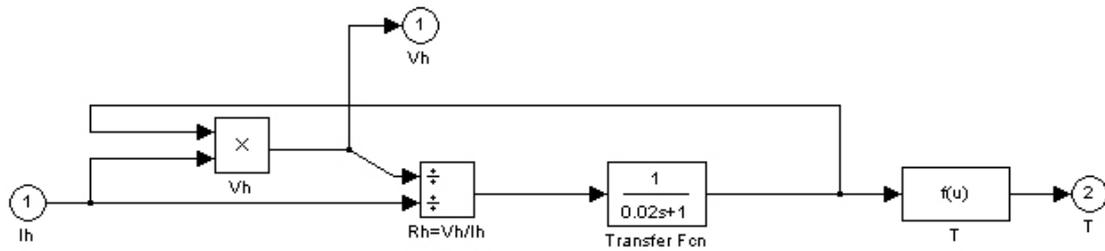


Figura 7. Esquema del modelado del sensor

El sensor es modelado de forma que tiene una entrada por la que se introduce I_h , la intensidad que pasa por el sensor, y las salidas de V_h y T , que son la tensión en bornes del *heater* y la temperatura del mismo, respectivamente.

Una vez hecho el modelo del sensor se hizo tres lazos de control distintos. Un lazo proporcional, un PI y un PID. De esta forma se comprobó cual podía ser el apropiado para nuestro sistema.

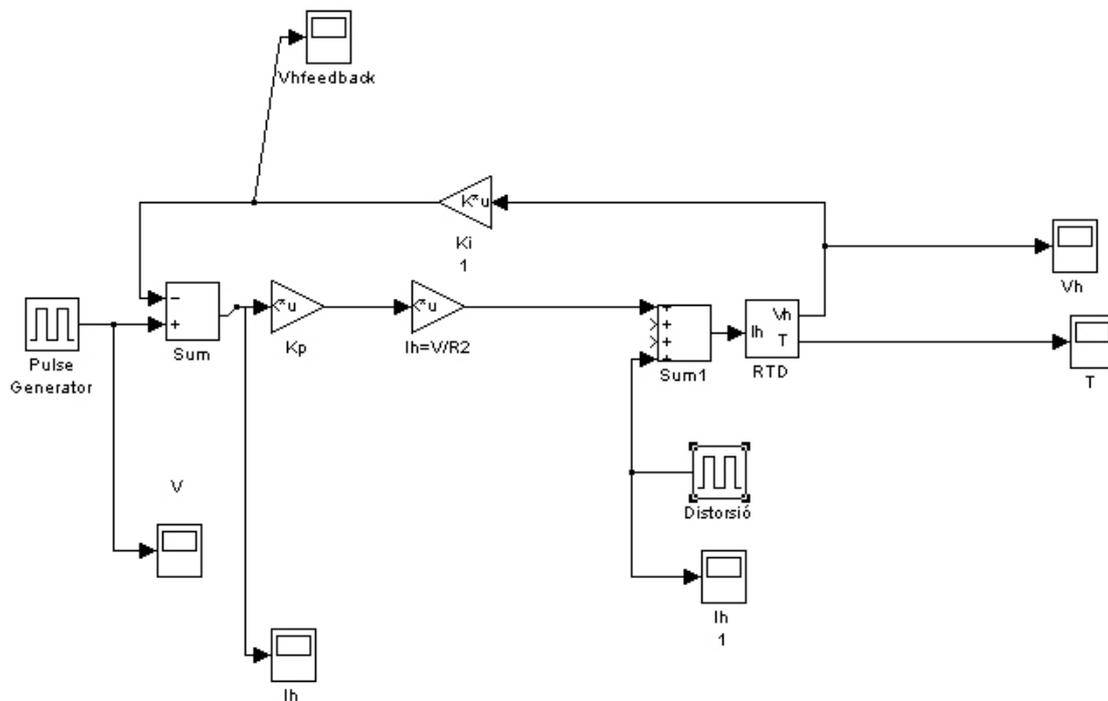


Figura 8. Lazo de control proporcional (P) para el control de la temperatura en un sensor de gas.

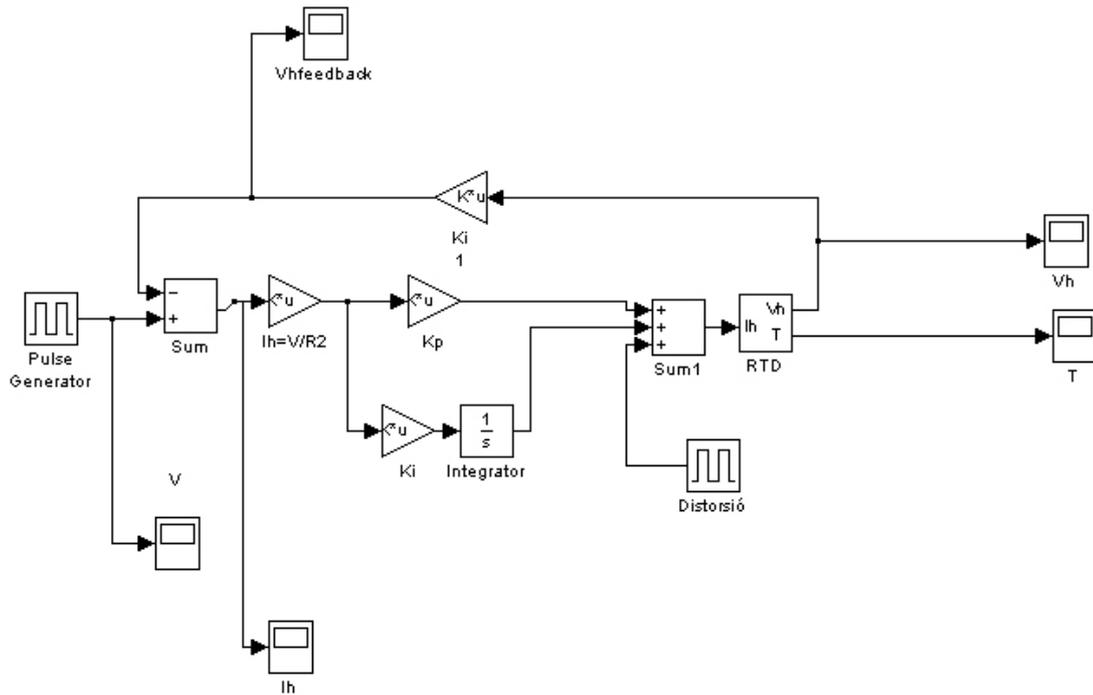


Figura 9. Lazo de control proporcional-integral (PI) para el control de la temperatura en un sensor de gas.

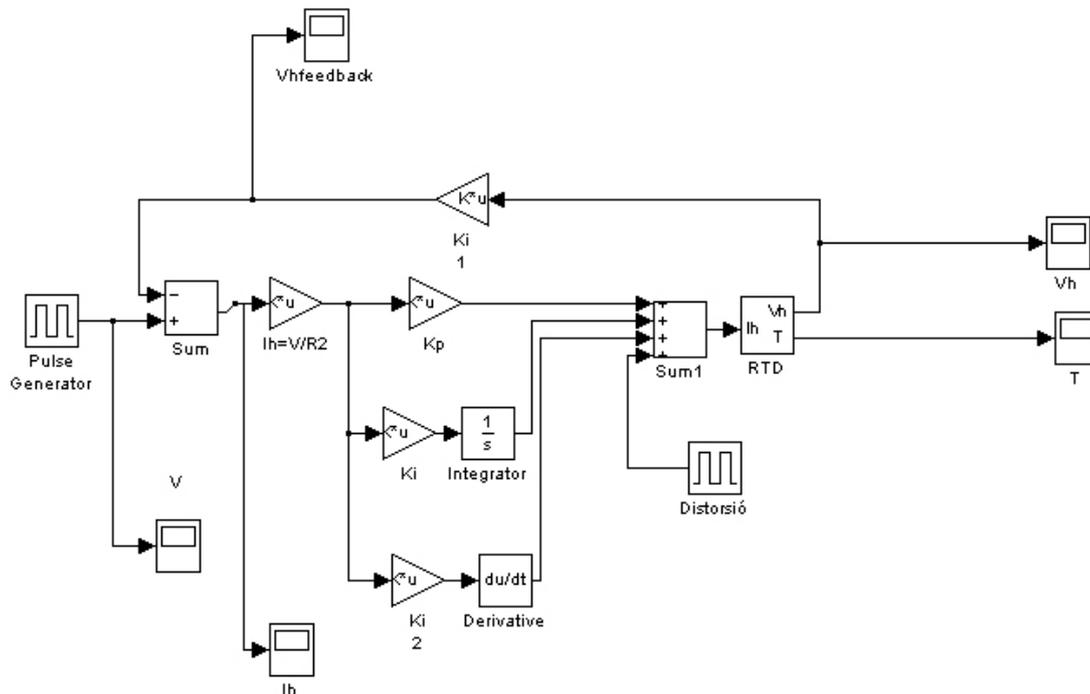


Figura 10. Lazo de control proporcional-integral-derivativo (PID) para el control de la temperatura en un sensor de gas.

Las simulaciones realizadas en Matlab confirmaban lo que se sabía de antemano. Para un sistema de primer orden, los controles PI y PID no son necesarios. Un controlador proporcional será suficiente para nuestro sistema.

Vemos como hace el control de temperatura, fijada a 383 °C, con una perturbación de la temperatura del entorno, equivalente a ± 100 °C.

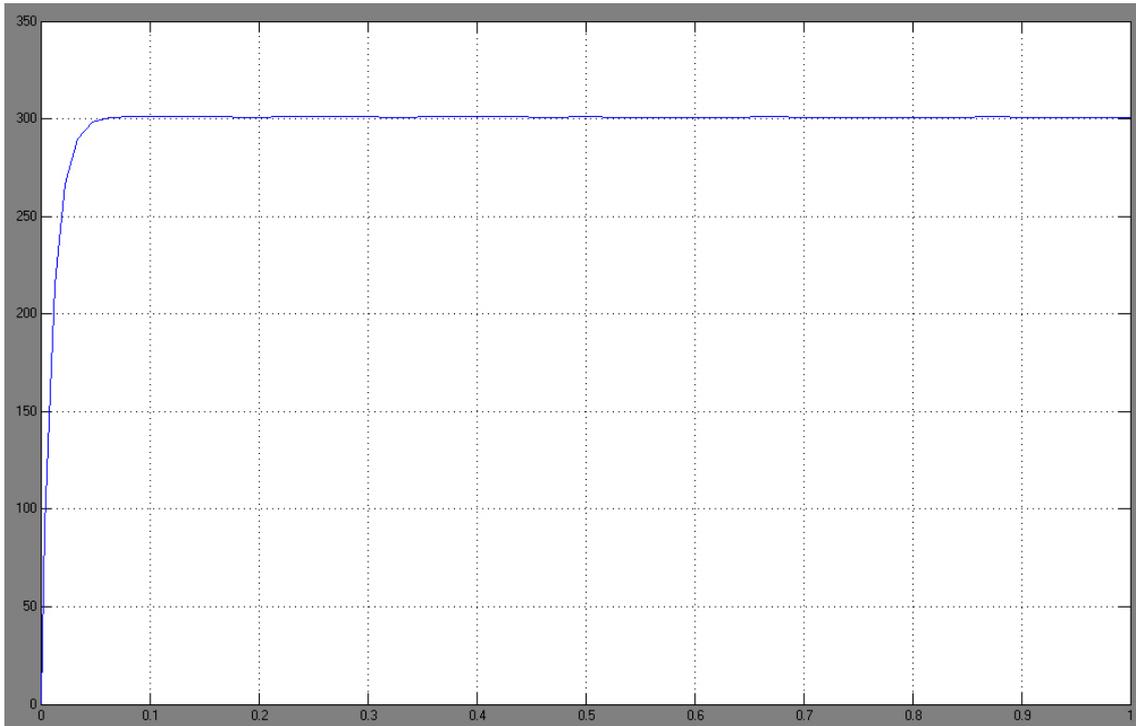


Figura 11. Respuesta del controlador proporcional (P)

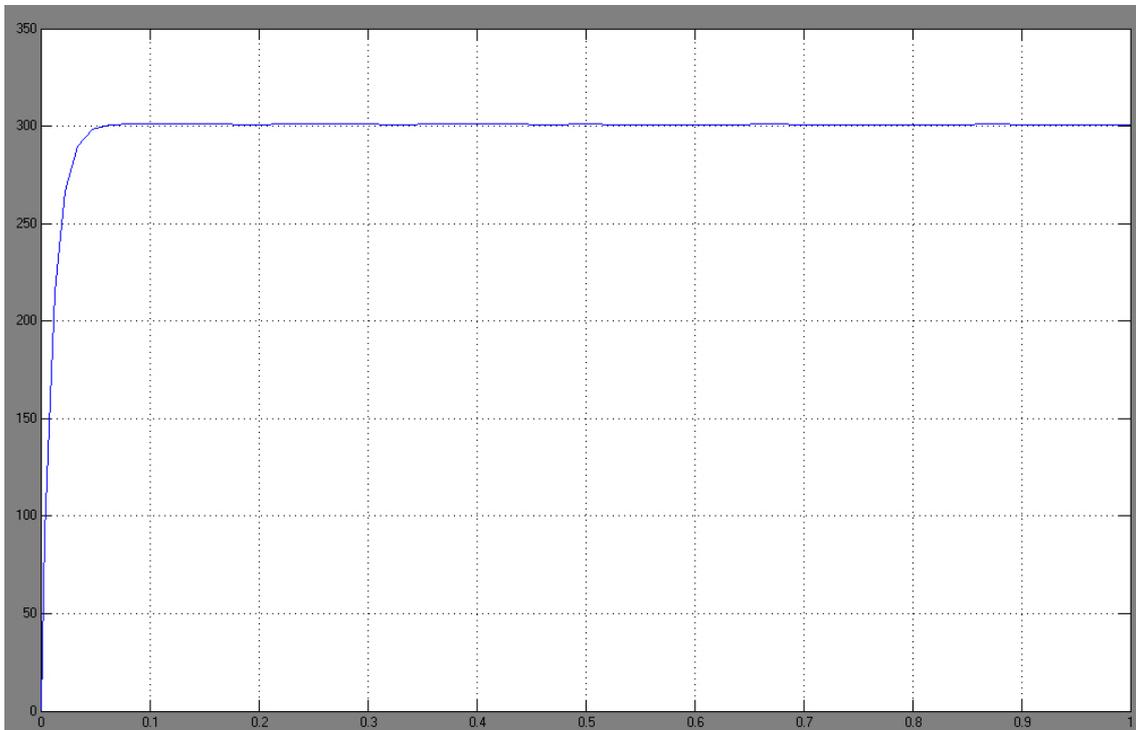


Figura 12. Respuesta del controlador proporcional-integral (PI)

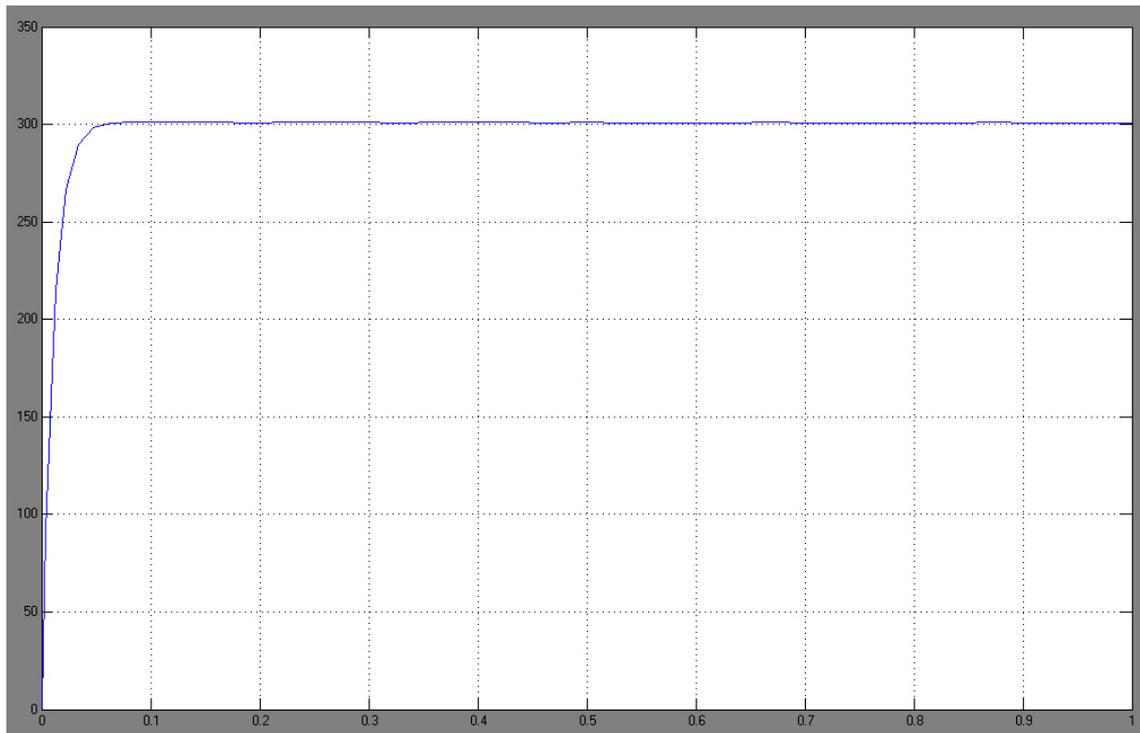


Figura 13. Respuesta del controlador proporcional-integral-derivativo (PID)

Vemos que las tres gráficas son prácticamente iguales. Lo que demuestra que los controles PI y PID no son importantes para el sistema que se quiere desarrollar.

El problema que hubo en este modelo fue que la temperatura se hallaba a través de los datos de ensayos anteriores, y se calculaba mediante la ecuación de una recta. Lo que se conseguía controlar en el modelo matemático era la potencia en el sensor, que era a partir de la cual obtenía la temperatura. Lo cual se conseguía de forma inmejorable.

2.1.2 Simulación con Pspice

Una vez vistos los resultados obtenidos con el Matlab, se pasó a realizar el circuito con el programa Pspice, para después realizar el circuito práctico.

La base del circuito es la fuente de corriente Howland controlada por tensión, igual a la utilizada en el proyecto anterior. Se emplea un amplificador sumador para hallar el error, un amplificador operacional para hacer la lectura de la tensión en el sensor (V_h) en alta impedancia y un amplificador que amplifica el error para hacer el control proporcional.

Los valores elegidos para la fuente son los mismos que los de la fuente actual. De esta forma la relación voltaje de entrada-temperatura del sensor será la misma que en la fuente realizada en el proyecto anterior, pero esta vez controlada.

Vemos el esquema del circuito

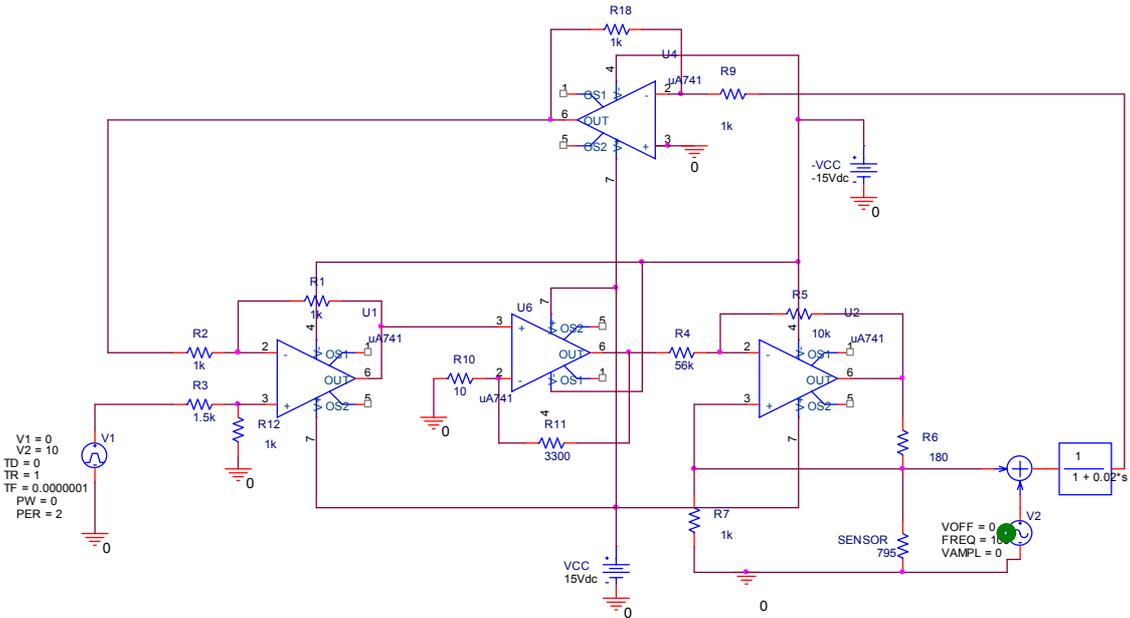


Figura 14. Circuito del lazo de control proporcional (P)

Como se puede observar el sensor se ha modelado para hacer la simulación con Pspice, y se ha implementado mediante una resistencia fija y un elemento de retardo. También se incluye una fuente de tensión senoidal que permite simular las perturbaciones a las que será expuesto el sensor, y que nos permiten ver si el circuito reacciona correctamente.

La lectura de la tensión en el sensor, V_h , se hace en alta impedancia para evitar valores erróneos.

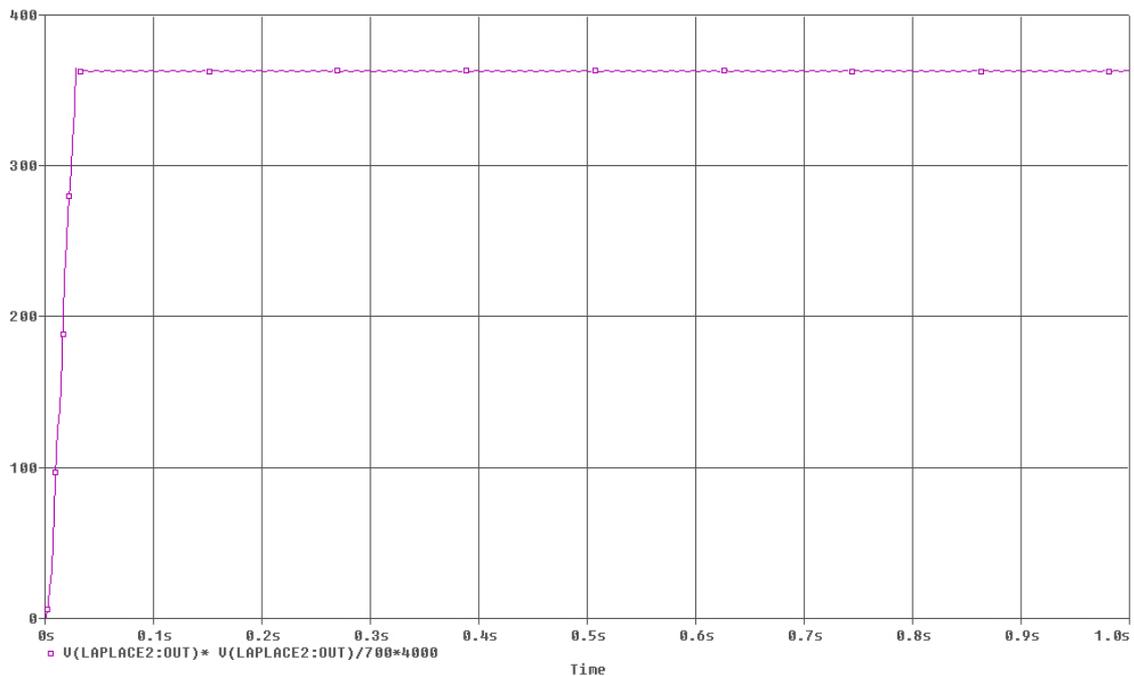


Figura 15. Temperatura en el sensor, simulado en Pspice.

La temperatura en el sensor se halla a partir de la potencia de la resistencia. Con el valor del voltaje a la salida del bloque de retardo, hallamos la potencia en el sensor:

$$Ph = \frac{Vh^2}{Rh}$$

La potencia del sensor se obtiene a partir de la tensión de salida del bloque de retardo y de la resistencia ya que si la obtenemos a partir de tensión retardada y la intensidad en la resistencia sin retardar, no podemos ver realmente la potencia del sensor. El bloque de retardo no permite el paso de la corriente, hace una lectura en alta impedancia.

Mediante las tablas que relacionan la potencia con la temperatura, se halla la temperatura:

$$T^a = Ph * 7201 + 235$$

Una vez comprobada que la simulación garantiza que hay un buen control en la potencia del sensor, se realiza la el prototipo de la placa, solo para una membrana.

2.1.3 Montaje y pruebas con el prototipo

Inicialmente se realizó un montaje sobre una placa *protoboard*, utilizando un circuito integrado LM324 con cuatro amplificadores operacionales, además de un filtro en la salida del amplificador proporcional que reduce los picos en el sensor. En ella se comprobaron las tensiones V_h del sensor correspondían a las simuladas. El problema era que no podía hacerse la verificación del correcto funcionamiento del sistema de control para variaciones de temperatura. Así que se decidió hacer el montaje sobre una placa PWB (*Printed Write Board*), que tiene una rigidez óptima para las pruebas.

Una vez realizado el montaje del circuito simulado en una placa “de topos”, se utilizó una segunda placa que contenía el sensor, y que estaba conectada por un cable bipolar a la placa de control.

Las pruebas se realizaron con el sensor dentro del horno de reflujo de aire convectivo TECHNO HA-02. La temperatura variaba desde la ambiente hasta los 140 °C.

Aquí vemos los resultados obtenidos en el laboratorio.

T ^a horno (°C)	V _i (V)	V _h (V)	I _h (mA)	R _h (Ω)	T ^a sensor (°C)
T ^a amb.	0	0	0	635	T ^a amb.
T ^a amb.	2	2	2,98	671	108
60	2	2	2,91	687	145
70	2	2	2,89	692	157
80	2	2	2,87	696	166
100	2	2	2,83	706	189
120	2	2	2,8	714	208
130	2	2	2,78	719	219
140	2	2	2,75	737	238
60	3	3	4,19	715	210
80	3	3	4,13	726	236
100	3	3	4,05	740	268
120	3	3	4,03	744	277
140	3	3	4	750	291

Tabla 2. Temperatura del sensor para diferentes temperaturas de entorno.

Después de analizar los resultados obtenidos, podemos comprobar que el control que se debía hacer sobre la temperatura del sensor es inexistente.

El lazo de control no consigue mantener estable la temperatura para una tensión de entrada fija, por lo tanto algo ha sido erróneo en el planteamiento del sistema.

Si volvemos atrás en el desarrollo del lazo de control, podemos comprobar que lo que mantenemos estable es la potencia en el sensor.

Debido a que en la simulación la resistencia es fija, la potencia se mantiene constante cuando se le aplica la perturbación en forma de voltaje, al contrario que la del sensor que varía con la temperatura.

Cuando la resistencia del sensor varía con las fluctuaciones de la temperatura del entorno, nuestro sistema no sirve porque no gobierna lo que se debe, que es la resistencia del sensor. Es decir, la relación V_h/I_h .

Debido al tipo de control que se desea hacer sobre la resistencia, un control analógico que domine el cociente V_h/I_h es muy complejo de implementar. Tendría que utilizarse un divisor analógico, que resultaría complejo de diseñar. Por lo tanto la opción del control analógico queda descartada.

2.2 Solución adoptada

Después de revisar los requerimientos del sistema, se plantea una nueva forma de realizar el lazo de control. Para ello será necesaria una parte analógica, que comuniqué el microcontrolador con el sensor, y una digital que realice el tratamiento de la información, y que determine la temperatura del *heater*.

El sistema debe disponer de una entrada que determine la temperatura de trabajo, una salida que puede controlar la tensión y corriente que hay en el *heater*, y una entrada más que informe del valor de la resistencia del *heater*.

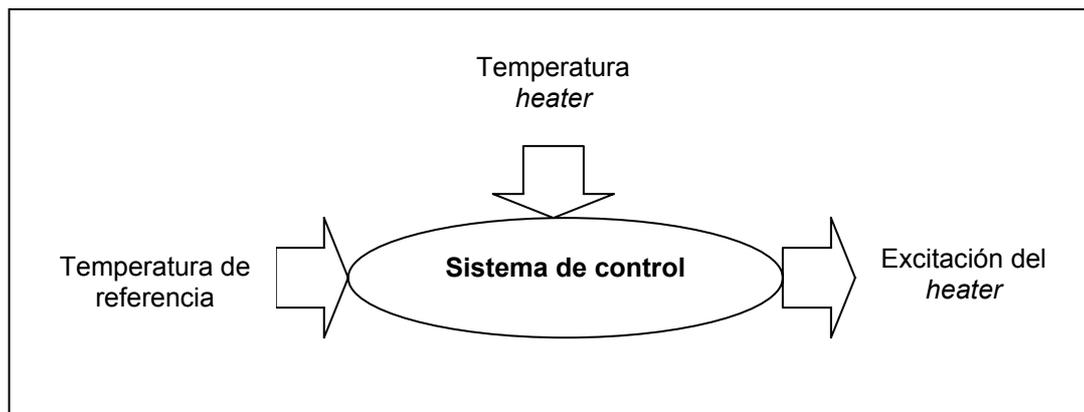


Figura 16. Diagrama del sistema de control.

Para la adquisición de la temperatura de referencia podemos utilizar la relación de la resistencia con la temperatura.

$$R = R_o \cdot (1 + 6,79 \cdot 10^{-4} \cdot (T - T_o))$$

Por lo tanto, conociendo la resistencia del *heater*, podemos conocer la temperatura. La forma de conocer la resistencia es a través de la relación V/I. Así que necesitamos un sistema de medida de intensidad y de voltaje.

Debido a que el *heater* no puede estar como una carga flotante, la intensidad no puede ser medida mediante un conversor I-V.

Hay dos formas de medir la intensidad del *heater*:

La intensidad que fluye por el sensor es controlada por el sistema mediante la tensión de excitación del *heater*. La fuente de corriente que se utilizará tendrá una relación determinada de V-I. Conociendo la tensión de excitación que se aplica a la fuente de corriente, y la relación con su intensidad de salida, podemos obtener la corriente que pasa por el *heater*, sin realizar ninguna medida.

Otra opción sería colocar una resistencia de valor bajo en serie con el *heater*. Haciendo una lectura de la tensión que hay en bornes de esta resistencia, podemos conocer la intensidad que fluye por ella, y por lo tanto por el *heater*.

La excitación del *heater* deberá ser controlada por el microcontrolador. El valor de esta deberá ser variable para poder realizar un control de la temperatura de trabajo. Las salidas del microcontrolador son lógicas, y solo pueden proporcionar los valores de 0 y 5 voltios. No es posible un control con estos valores. Debemos conseguir valores analógicos que varíen entre los 0 y 5 voltios.

La implementación de un control PWM es la solución más apropiada en nuestro caso. Puesto que un conversor digital/ analógico aumentaría el coste y la velocidad del circuito.

Un tren de pulsos de ancho variable, y con una frecuencia fija es la señal que obtendremos para excitar la fuente de corriente. Pero esta señal no se puede aplicar directamente al sensor porque afectaría directamente a la temperatura del mismo produciendo unas fluctuaciones en él, con la misma frecuencia que la señal. La señal deberá ser previamente filtrada para evitar las oscilaciones.

De las dos opciones para medir la intensidad, la primera es descartada debido a el funcionamiento del sistema. Si la salida que se obtiene del microcontrolador es un PWM con un *duty cycle* variable, se deberá filtrar esta señal para conseguir una temperatura constante en el sensor. Esto producirá un retardo de la señal que provocará que la señal media de la señal PWM del microcontrolador no sea la misma que la que está excitando la fuente Howland. Por lo tanto los valores que se calculan con esta intensidad no serán correctos.

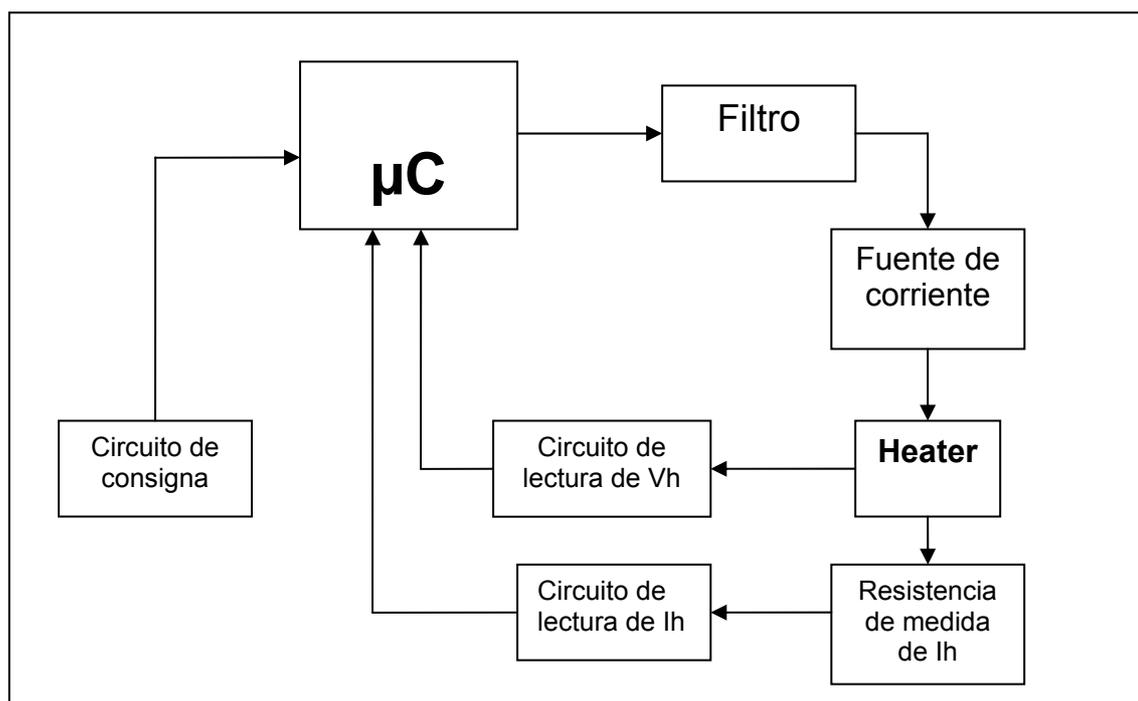


Figura 17. Diagrama de bloques del circuito de control.

El sistema está diseñado de forma modular. Así, cada *heater* dispondrá de un circuito independiente de control de temperatura. Cada sensor podrá trabajar a una temperatura distinta si se desea, para aumentar la versatilidad del sistema.

Las placas de control se conectarán a una placa que las alimentará y hará de interfaz con el ordenador.

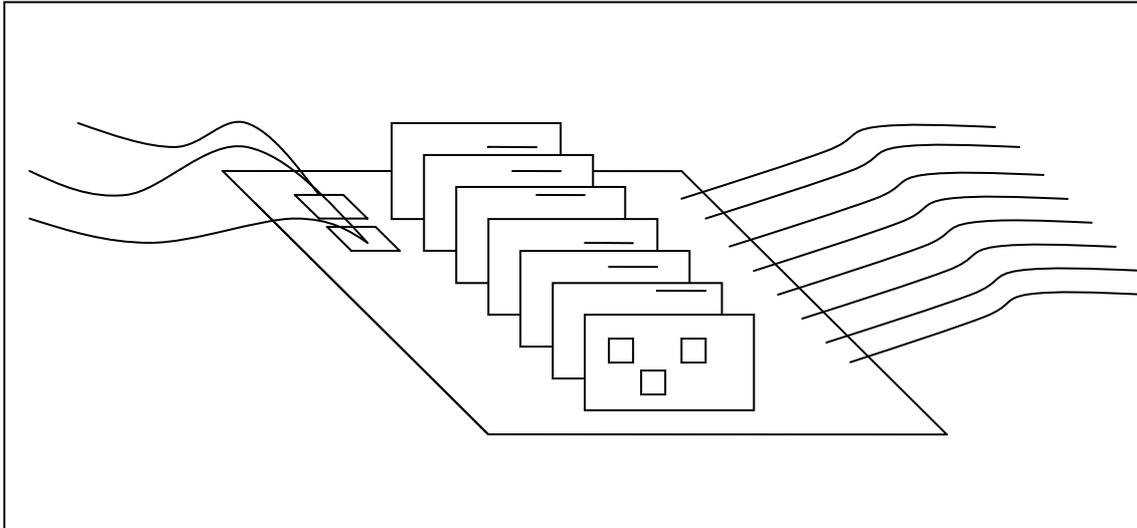


Figura 18. Esquema del sistema modular

2.2.1 Diseño del circuito

La circuitería que conectará el microcontrolador con el sensor deberá tener una precisión elevada. Esto es debido a que va a realizar un control de temperatura y se va a adaptar los valores medidos al rango de valores del microcontrolador. Por lo tanto, los valores que leerá el microcontrolador son valores relativos, y se debe realizar la conversión a valores absolutos. Así que hay que conocer exactamente las constantes proporcionales de los amplificadores, que estas tengan una precisión alta y que no varíen. Por esto todas las resistencias del circuito serán resistencias de un 1 % de tolerancia.

2.2.1.1 Circuito regulador de tensión

El circuito será alimentado a $\pm 15\text{ V}$, para así poder alimentar los amplificadores operacionales que se utilizarán. Un circuito regulador de tensión proporcionará la tensión de alimentación del microcontrolador. Estará formado por un regulador positivo de tensión LM7805

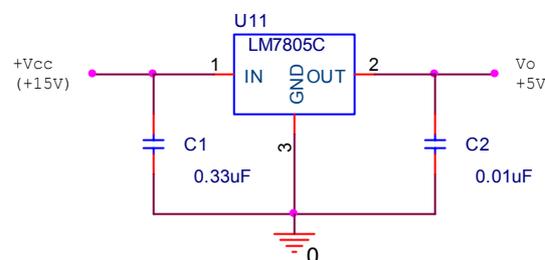


Figura 19. Circuito regulador de tensión.

2.2.1.2 Circuito de consigna

La temperatura de trabajo del sensor será fijada con un voltaje analógico. Su valor deberá variar entre los 0 y los 5 voltios, igual al rango de tensión del conversor A/D del microcontrolador.

Para fijar la temperatura de trabajo de los *heaters*, se implementará una interfaz entre la tarjeta de adquisición de datos, National Instruments PCI6023E, y el circuito de control, de forma que la temperatura de consigna sea fijada mediante un programa. Esto permite variar la temperatura de los sensores en cada sesión de trabajo sin tener que ajustar ningún parámetro en la placa.

Se dispone de 7 salidas lógicas y ninguna analógica en la PCI6023E para conseguir fijar la temperatura de trabajo. De estos 7 bits, 3 serán destinados a la selección de la placa de control a la que hay que fijar la temperatura de trabajo. Un bit se activará una vez se haya realizado la conversión digital/analógico para avisar a los microcontroladores de que pueden adquirir el dato. Quedan 3 bits libres para realizar la conversión.

Se planten dos posibilidades para conseguir un voltaje analógico con el ordenador.

Una de ella es la utilización de un conversor digital-analógico.

Se debería utilizar un conversor serie puesto que solo se dispone de 3 bits de salida libres en la tarjeta del ordenador.

Los convertidores que se han buscado para desarrollar esta tarea son dispositivos que utilizan únicamente dos hilos para la comunicación con el emisor. Su protocolo es I²C. Un protocolo basado en el envío de una dirección correspondiente al dispositivo esclavo, un *byte* de control, y el dato.

Esta solución es más compleja de implementar debido a la dificultad de la programación. Una de las salidas debería ser la de datos y la otra el reloj. El problema se agrava al no disponer de una función en el ordenador que realice este tipo de transmisiones. Si el programa que realiza la transmisión se realiza de forma “artesanal” pueden surgir problemas relacionados con los tiempos de la trama y con la compatibilidad con el conversor.

Para evitar estos problemas se debería estudiar el manual técnico de la tarjeta PCI6023E, y asegurar que no hay incompatibilidades. Todo esto supondría un gran esfuerzo para una sencilla comunicación de inicialización del sistema. Por eso se plantea otra solución más sencilla y fiable.

El objetivo de esta comunicación es obtener unos voltajes analógicos para que los microcontroladores sean inicializados a sus diferentes temperaturas de trabajo.

Una forma sencilla de conseguir un voltaje analógico a partir de una salida digital, sin utilizar más de tres bits de información es la implementación de un PWM.

Si se obtiene una señal procedente de la tarjeta PCI6023E con una frecuencia fija y con un ciclo de trabajo variable, lo que se consigue es una señal con una tensión media variable.

Para obtener la media de esta señal PWM se necesita un filtro paso-bajo que proporcione una señal continua. El filtro será de segundo orden para evitar un rizado de la señal que provoque lecturas erróneas en el conversor A/D del microcontrolador. Su ganancia será unitaria debido a que la tensión de salida del ordenador es de 5 voltios, por lo tanto si el ciclo de trabajo es máximo, su filtrado será de 5 voltios.

Este sistema permite una fácil programación con el ordenador, ya que solo debe realizarse un control de tiempo de la señal de salida del ordenador. La relación entre el *duty cycle* de la señal y el voltaje medio es inmediata. Por lo tanto se puede conocer perfectamente el voltaje de salida.

El PWM puede ser generado por diferentes programas, entre ellos el Labview y el Matlab. Su frecuencia deberá ser lo más alta posible. Esto presenta un problema para la generación por ordenador, debido a que si no es programado correctamente los tiempos no serán exactos y esto puede afectar a la tensión media de la señal de consigna. La frecuencia del PWM inversamente proporcional al rizado de la señal filtrada. Es decir, si su frecuencia es baja el rizado será alto y producirá errores en la lectura.

En las pruebas que se realizarán al laboratorio se utilizará el Labview. Este programa permite realizar un PWM con unos tiempos múltiplos de 1 ms.

Las señales que se pueden realizar con esta base de tiempos son de frecuencia baja. Por lo tanto se debe hacer una selección de la frecuencia del PWM de inicialización dependiendo de la precisión que se desea para el sistema. Puesto que la precisión se verá reducida en cuanto aumente la frecuencia para mejorar la señal filtrada.

A continuación se muestran los errores que se pueden producir para diferentes frecuencias, y el número de temperaturas diferentes que se pueden seleccionar.

f = 40 Hz	rizado = ± 20 mV	error en lectura A/D = 0,4 %	medidas = 35
f = 10 Hz	rizado = ± 50 mV	error en lectura A/D = 1 %	medidas = 100
f = 1 Hz	rizado = ± 500 mV	error en lectura A/D = 10 %	medidas = 1000

Esta es la relación de posibles frecuencias del PWM. Se debe seleccionar la que más convenga para la aplicación del sistema.

Para mejorar la precisión del sistema se puede acotar el rango de temperaturas de trabajo a las realmente útiles. Si se desea trabajar a temperaturas entre los 250 °C y los 400 °C, la señal de consigna seleccionará solo este rango. No lo hará de 0 a 400 °C.

Para definir este rango se deberá modificar la línea del código de programa siguiente:

```
valor1=(unsigned int) (conv0*0.396754);
```

$$\frac{\text{máx_grados}}{\text{bits_conv_A/D}} = \frac{405}{1023} = 0,396$$

En este caso está definido el rango de 0 a 405 °C.

Si se utiliza Matlab para generar el PWM se debe tener en cuenta que si se realiza con la programación común de Matlab el control de tiempo es inexacto. No se puede asegurar que las salidas responden a los tiempos que se le ha fijado. Esto es debido a que las operaciones que realiza el programa pueden retrasar la actuación sobre estas.

Para realizar un control con gran precisión se puede utilizar la opción *real time* de Matlab. Esta permite generar un PWM preciso a frecuencias alta, y que a su vez aumentaría la precisión de selección de temperatura de trabajo y reduciría en error producido por el rizado de la señal.

Esta opción sería la mejor solución para implementar un sistema preciso y fiable, sin embargo la programación del PWM será más costosa.

A continuación se estudia el diseño del filtro que será utilizado para el PWM.

2.2.1.3 Filtro *Butterworth*

El filtrado de la señal PWM se realizará mediante un filtro *Butterworth* de segundo orden. Se trata de un filtro paso-bajo diseñado para impedir oscilaciones en la tensión que excita la fuente de corriente y para el filtrado de la señal de referencia que proviene del ordenador.

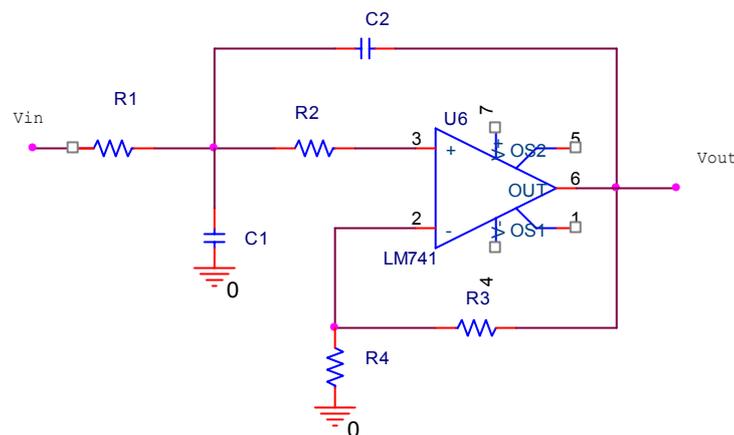


Figura 20. Filtro *Butterworth*

El filtro *Butterworth* permite elegir la posición de los polos y producir una atenuación de 40 dB/década, y también se puede ajustar la ganancia del filtro. En el filtro requerido, la frecuencia de corte debe ser muy baja, del orden de 100 rad/s, unos 16 Hz.

Fijamos el condensador en 100 μF y calculamos la resistencia.

$$\omega = \frac{1}{R \cdot C}$$

$$100 \text{ rad/s} = \frac{1}{R \cdot 10 \mu\text{F}}$$

$$R = \frac{1}{10 \mu\text{F} \cdot 100 \text{ rad/s}} = 1 \text{ k}\Omega$$

Para la señal que produce el ordenador el filtro tendrá una frecuencia de corte más baja que la del filtrado del PWM del microcontrolador. Esto es debido a que la frecuencia del PWM del ordenador puede ser mayor que la del microcontrolador, de esta forma aseguramos una tensión estable.

$$10 \text{ rad/s} = \frac{1}{R \cdot 10 \mu\text{F}}$$

$$R = \frac{1}{10 \mu\text{F} \cdot 10 \text{ rad/s}} = 10 \text{ k}\Omega$$

Una vez determinada la posición de los polos hay que ajustar la ganancia del filtro. Debido a que con el microcontrolador, el máximo que se puede alcanzar son 5 voltios en una de sus salidas, el rango que se puede alcanzar no es muy amplio. En cambio, si se ajusta la ganancia del filtro, se podrá llegar a temperaturas más altas.

Para poder calcular la ganancia del filtro, se debe conocer la relación V/I de la fuente de corriente.

En el PWM que proporciona la tensión de referencia al conversor A/D del microcontrolador la ganancia será unitaria. De esta forma si el *duty cycle* del PWM es el 100% la tensión de salida será de 5 V, como se desea que sea.

2.2.1.4 Fuente de corriente Howland

La fuente de corriente es quién proporciona la intensidad que ha de pasar por el sensor. Una fuente Howland será la que se utilizará para actuar directamente sobre la temperatura del *heater*. La relación entre la intensidad de salida y el voltaje de entrada la determina los siguientes cálculos.

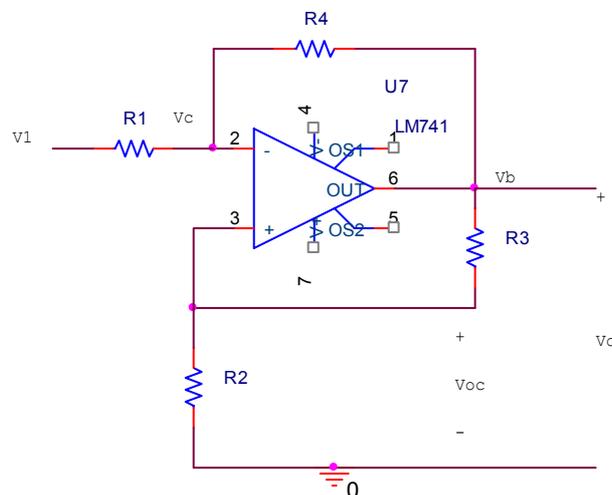


Figura 21. Calculo de la tensión en circuito abierto

$$v_{oc} = v_o \cdot \frac{R_2}{R_2 + R_3}$$

$$\frac{v_1 - v_{oc}}{R_1} = \frac{v_{oc} - v_o}{R_4}$$

$$v_{oc} = \frac{R_2 R_4}{R_2 R_4 + R_1 R_3} \cdot v_1$$

Después de obtener la tensión en circuito abierto, se obtendrá la resistencia equivalente de *Norton* R_{eq} , con el circuito siguiente:

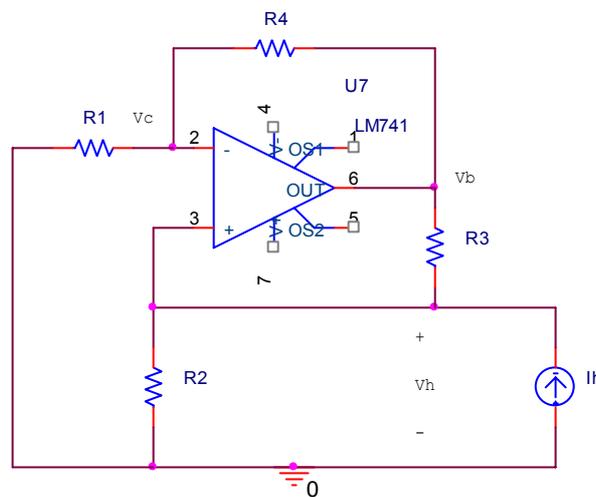


Figura 22. Estructura de la fuente Howland

$$\begin{pmatrix} ih \\ x \\ 0 \end{pmatrix} = \begin{pmatrix} ih \\ x \\ 0 \end{pmatrix} = \begin{pmatrix} G_3 + G_2 & -G_3 & 0 \\ x & x & x \\ 0 & -G_4 & G_1 + G_4 \end{pmatrix} \begin{pmatrix} vh \\ vb \\ vc \end{pmatrix}$$

$$vh = vc$$

$$\begin{pmatrix} ih \\ 0 \end{pmatrix} = \begin{pmatrix} G_3 + G_2 & -G_3 \\ G_1 + G_4 & -G_4 \end{pmatrix} \begin{pmatrix} vh \\ vb \end{pmatrix}$$

El valor de V_h será por lo tanto:

$$v_h = \frac{\begin{vmatrix} i_h & -G_3 \\ 0 & -G_4 \end{vmatrix}}{\begin{vmatrix} G_3 + G_2 & -G_3 \\ G_1 + G_4 & -G_4 \end{vmatrix}} 0 - \frac{-i_h \cdot G_4}{G_3 G_1 - G_2 G_4}$$

El valor de la resistencia equivalente es por lo tanto:

$$R_{eq} = \frac{v_h}{i_h} = \frac{G_4}{G_2 G_4 - G_1 G_3}$$

De las ecuaciones anteriores se halla la corriente de cortocircuito i_{sc} :

$$i_{sc} = \frac{v_{oc}}{R_{eq}} = -\frac{R_4 v_1}{R_1 R_3}$$

La fuente se calcula para valores $R_2 \cdot R_4 = R_1 \cdot R_3$, de esta forma se consigue una variación mínima de la intensidad I_h , para variaciones de R_h .

$$i_{sc} = -\frac{R_4 v_1}{R_1 R_3} = -\frac{v_1}{R_2}$$

$$R_{eq} = \infty$$

Por lo tanto, el circuito equivalente de *Norton* será:

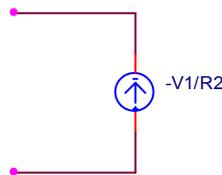


Figura 23. Circuito equivalente de *Norton*

La relación V/I de la fuente Howland es por lo tanto R_2 quién la determina. Así que únicamente tenemos que respetar que se cumpla la relación $R_2 R_4 = R_1 R_3$.

El valor de R_2 será ajustado de forma que permita ajustar el voltaje de salida del filtro con la corriente máxima que deseamos que fluya por el *heater*.

Se puede comprobar que el circuito equivalente de *Norton* tiene la fuente de corriente dirección hacia arriba. Esto provocará voltajes en el *heater* negativos respecto a masa. La temperatura del sensor no se verá afectada por el sentido de la corriente, pero sí será un factor a tener en cuenta a la hora de adquirir los voltajes del *heater* y de la resistencia de medida.

Según las pruebas realizadas con los sensores podemos comprobar que intensidades son las que permiten alcanzar ciertas temperaturas.

Rh=550 Ω

Vh (V)	Ih (mA)	Rh (Ω)	T ^a (°C)
2,01	3,5	574	120
3	5	600	191
4,01	6,2	647	321
5	7,2	694	453
6,01	8,1	742	585
7	8,8	795	733
8	9,6	834	841

Rh=650 Ω

Vh (V)	Ih (mA)	Rh (Ω)	T ^a (°C)
2,01	3	670	96
3	4,3	698	161
4,01	5,5	729	234
5	6,4	781	356
6,01	7,2	835	482
7	7,9	886	602
8	8,5	942	734

Tabla 5. Resultados de pruebas con los *heaters* de los dos tipos de membranas.

Si las temperaturas de trabajo de los sensores estarán entre los 250 y los 400 °C, se puede fijar el máximo de corriente alrededor de los 7,2 mA, es decir, que los *heaters* con resistencia inicial de 550 Ω , pueden llegar a los 453 °C y los que tengan la resistencia inicial de 650 Ω pueden alcanzar los 482 °C.

Una vez definida la máxima intensidad que recorrerá los sensores hay que ajustar la ganancia de las fuentes y del filtro. Podemos ajustar las dos ganancias de forma que obtengamos una tensión máxima en el sensor de 6.01 V, que es la que debe tener el sensor de 650 Ω de resistencia inicial cuando alcance los 482 °C.

La máxima tensión del PWM de microcontrolador será de 5 V y deberá ser aumentada o disminuida por la ganancia del filtro en relación con la ganancia de la fuente Howland.

Definiremos la relación V/I de la fuente de corriente de forma que con los 5 V que proporciona el microcontrolador podamos alcanzar los 7.2 mA de intensidad máxima.

$$I_h = \frac{V}{R_2}$$

$$R_2 = \frac{V}{I_h} = \frac{5V}{7.2mA} = 694.44 \Omega$$

Una resistencia normalizada que se aproxime a 694.4 Ω puede ser la de 680 Ω . De esta forma la intensidad que podrá proporcionar con 5 V será de:

$$I_h = \frac{5V}{680\Omega} = 7.35 mA$$

La máxima temperatura que se podrá alcanzar aumentará ligeramente si la ganancia del filtro es unitaria. Más adelante se comprobará en los ensayos.

Sí la resistencia R_2 es de 680 Ω , y hay que mantener la relación $R_2 \cdot R_4 = R_1 \cdot R_3$, se pueden dar estos valores a la demás resistencias de la fuente:

$$R_1 = 68 \text{ k}\Omega$$

$$R_2 = 680 \Omega$$

$$R_3 = 180 \Omega$$

$$R_4 = 18 \text{ k}\Omega$$

Puesto que con la relación V/I de la fuente de corriente ya se puede alcanzar el rango de temperaturas deseado, el filtro no requiere una ganancia para adaptar las dos etapas. Una amplificación unitaria bastaría para lograr las temperaturas de trabajo.

2.2.1.5 Circuitos adaptadores de señal

Los valores de las tensiones del sensor y de la resistencia en serie de 10Ω , determinarán el valor de la resistencia del *heater*. Las medidas precisas de las mismas son fundamentales para el buen funcionamiento del sistema. Estas tensiones proporcionan el la tensión y la intensidad del *heater*. Una medida en alta impedancia es la más adecuada en este caso.

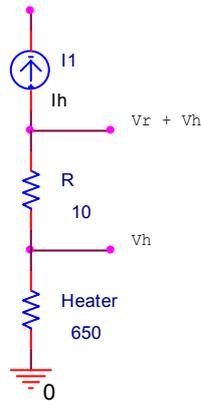


Figura 24. Posición de la resistencia de medida de la corriente del *heater*

La tensión de la resistencia del *heater* tienen un rango de 0 a -6 V, y la de la resistencia de 10Ω , de 0 a -70 mV aproximadamente. Al ser negativos deben invertirse para poder ser adquiridos por el PIC.

El rango de tensiones que puede convertir el microcontrolador en su puerto analógico es de 0 a 5 V. Así que se necesita una adaptación de las tensiones para conseguir abarcar todo el rango y conseguir una mejor resolución respectivamente.

Para realizar la lectura de la tensión del *heater* debemos realizar una reducción de la señal. Debido a que el máximo de V_h está cerca de los 6 voltios:

$$A_1 = \frac{5V}{6V} = 0.83$$

Unos valores normalizados de resistencias que podrían ajustarse a nuestras necesidades son:

$$R_1 = 10 \text{ k}\Omega$$

$$R_2 = 7.4 \text{ k}\Omega$$

$$A_2 = \frac{7.4 \text{ k}\Omega}{10 \text{ k}\Omega} = 0.74$$

El circuito resultante es un amplificador inversor con la ganancia calculada, de esta forma se obtiene un voltaje positivo del rango deseado:

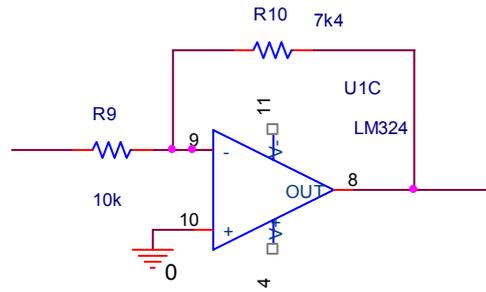


Figura 25. Circuito adaptador de señal de Vh

En el caso de la medida de la tensión de la resistencia de $10\ \Omega$, utilizada para medir la intensidad que fluye por el *heater*, su máximo será de unos $70\ \text{mV}$, es decir, que $7\ \text{mA}$ pueden pasar por el *heater* del sensor.

Para adaptar la tensión de esta resistencia a la entrada analógica del ADC, se calcula la ganancia siguiente:

$$A_2 = \frac{5\ \text{V}}{70\ \text{mV}} = 71.4$$

Si seleccionamos valores comerciales de resistencias:

$$\begin{aligned} R_1 &= 1\ \text{k}\ \Omega \\ R_2 &= 68\ \text{k}\Omega \end{aligned}$$

$$A_2 = \frac{68\ \text{k}\Omega}{1\ \text{k}\Omega} = 68$$

Con estos valores ampliamos ligeramente el margen de valores que podemos medir. El montaje se realizará en un amplificador inversor con la ganancia de 68.

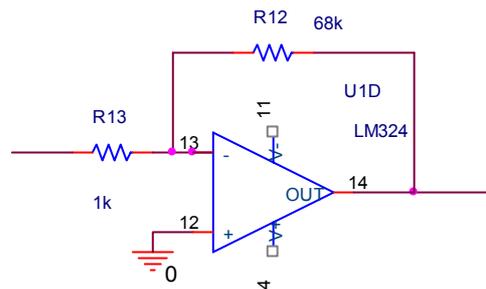


Figura 26. Circuito adaptador de señal de Ih.

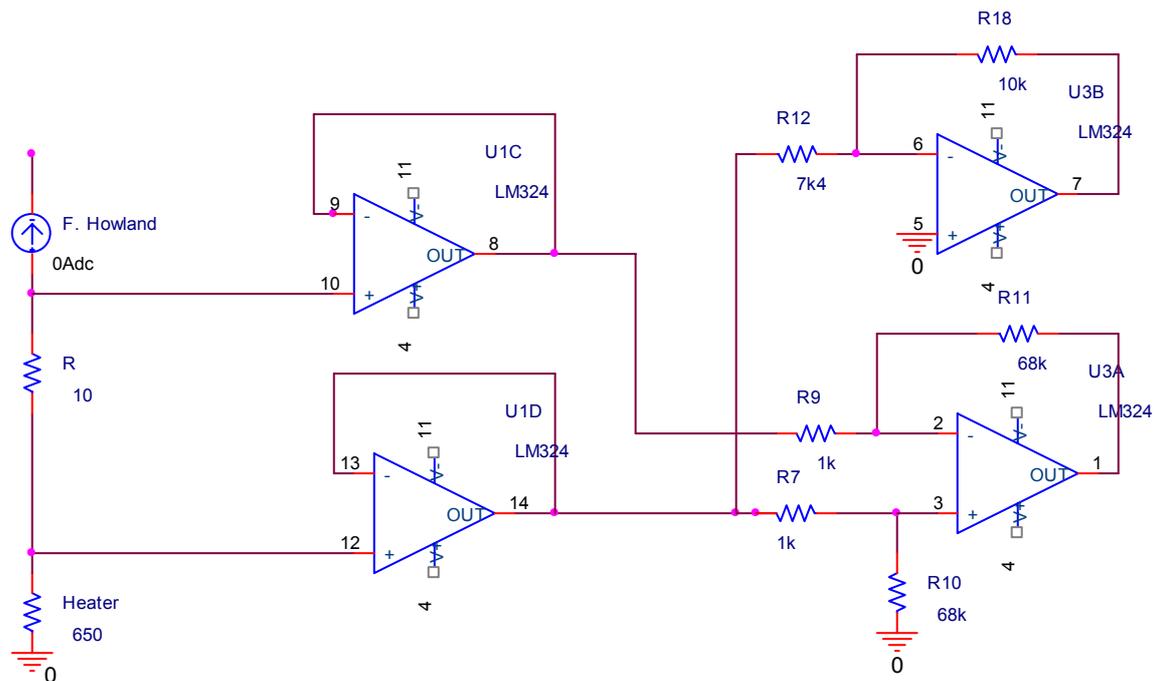


Figura 27. Circuito de adquisición de V_h e I_h .

2.2.2 El microcontrolador

El microcontrolador es la parte más importante del circuito de control. En él, se realizarán todos los cálculos que permitan hacer una regulación de la temperatura de trabajo.

Para elegir el microcontrolador se han planteado los siguientes requerimientos del sistema:

- 3 Canales de conversión analógico-digital
- 2 Entradas digitales
- Interrupciones
- 2 *Timers*
- Dispositivo Flash
- Precio económico

Debido a la modularidad del sistema, el dispositivo que se busca debe ser económico y preferiblemente pequeño, con pocos *pins* de entrada/salida.

El fabricante elegido desde un principio fue Microchip, debido a la disposición de los medios de programación del microcontrolador.

La memoria al igual que la velocidad, no era un requisito importante porque el sistema no ni requiere algoritmos complejos, ni velocidades altas. La temperatura que puede crear interferencias al sistema, no tendrá variaciones violentas en los sensores ya que los flujos de gas que travesarán por los sensores serán continuos.

Los microcontroladores Microchip se dividen en familias lógicas. Las familias 12, 14, 16, 17 y 18. Los de la familia 12 son los más pequeños y con menos prestaciones. Aunque algunos de ellos disponen de conversores A/D, y memoria Flash.

La familia 14 solo cuenta con un PIC. Demasiado equipado y económicamente poco interesante.

En la familia 16 encontramos todo tipo de microcontroladores. Desde los sencillos que no disponen de convertidores A/D, hasta los que disponen de USART de gama alta.

En la familia 17 encontramos dispositivos rápidos con buenas prestaciones, al igual que en la familia 18, donde se encuentra la gama alta de los microcontroladores de Microchip.

Vistos los requisitos del sistema, la gama alta o los PIC's con altas prestaciones no interesan. En la tabla 4 se observa un detalle de los microcontroladores de la familia 12. De ellos, los únicos con memoria Flash son el PIC12F629 y el PIC12F675.

Device	Data RAM	ADC	Words	Speed	Timers	Low Voltage Device
PIC12C671	128	4	1024	10	1+WDT	PIC12LC671
PIC12C672	128	4	2048	10	1+WDT	PIC12LC672
PIC12CE673	128	4	1024	10	1+WDT	PIC12LCE673
PIC12CE674	128	4	2048	10	1+WDT	PIC12LCE674
PIC12F629	64	-	1024	20	2+WDT	PIC12F629
PIC12F675	64	4	1024	20	2+WDT	PIC12F675

Tabla 6. Microcontroladores de la familia 12 de Microchip con memoria Flash

De los dos microcontroladores con memoria Flash, el único que dispone de conversores A/D, además de tener una memoria pequeña, es el PIC12F675.

Vemos una tabla con los PIC's de la familia 16 que tienen memoria Flash.

Device	Data RAM	ADC	Words	Serial I/O	Speed	Timers	Low Voltage Device
PIC16F627	224	-	1024	USART	20	3 + WDT	PIC16LF627
PIC16F627A	224	-	1024	USART	20	3+ WDT	PIC16LF627A
PIC16F628	224	-	2048	USART	20	3 + WDT	PIC16LF628
PIC16F628A	224	-	2048	USART	20	3+ WDT	PIC16LF628A
PIC16F630	64	-	1024	-	20	2+WDT	PIC16F630
PIC16F630	64	-	1024	-	20	2+WDT	PIC16F630
PIC16F648A	256	-	4096	USART	20	3+ WDT	PIC16LF648A
PIC16F676	64	10	1024	-	20	2+WDT	PIC16F676
PIC16F676	64	10	1024	-	20	2+WDT	PIC16F676
PIC16F684	128	8	2048	-	20	3	PIC16F684
PIC16F688	256	8	4096	-	20	2	PIC16F688
PIC16F72	128	5	2048	I ² C, SPI	20	3 + WDT	PIC16LF72
PIC16F73	192	5	4096	I ² C, SPI, USART	20	3+WDT	PIC16LF73
PIC16F737	368	11	4096	MI ² C, SPI, AUSART	20	3+WDT	PIC16LF737
PIC16F74	192	8	4096	I ² C, SPI, USART	20	3+WDT	PIC16LF74
PIC16F747	368	14	8192	MI ² C, SPI, AUSART	20	3+WDT	PIC16LF747
PIC16F76	368	5	8192	I ² C, SPI, USART	20	3+WDT	PIC16LF76
PIC16F767	368	11	8192	MI ² C, SPI, AUSART	20	3+WDT	PIC16LF767
PIC16F77	368	8	8192	I ² C, SPI, USART	20	3+WDT	PIC16LF77
PIC16F777	368	14	8192	MI ² C, SPI, AUSART	20	3+WDT	PIC16LF777
PIC16F818	128	5	1024	I ² C, SPI	20	2/1+WDT	PIC16LF818

PIC16F819	256	5	2048	I ² C, SPI	20	2/1+WDT	PIC16LF819
PIC16F83	36	-	512	-	10	1+WDT	PIC16LF83
PIC16F84	68	-	1024	-	10	1+WDT	PIC16LF84
PIC16F84A	68	-	1024	-	20	1+WDT	PIC16LF84A
PIC16F87	368	-	4096	AUSART	20	3	PIC16LF87
PIC16F870	128	5	2048	USART	20	3+WDT	PIC16LF870
PIC16F871	128	8	2048	USART	20	3+WDT	PIC16LF871
PIC16F872	128	5	2048	I ² C, SPI	20	3+WDT	PIC16LF872
PIC16F873	192	5	4096	USART, I ² C, SPI	20	3+WDT	PIC16LF873
PIC16F873A	192	5	4096	USART, I ² C, SPI	20	3+WDT	PIC16LF873A
PIC16F874	192	8	4096	USART, I ² C, SPI	20	3+WDT	PIC16LF874
PIC16F874A	192	8	4096	USART, I ² C, SPI	20	3+WDT	PIC16LF874A
PIC16F876	368	8	8192	USART, I ² C, SPI	20	3+WDT	PIC16LF876
PIC16F876A	368	5	8192	USART, I ² C, SPI	20	3+WDT	PIC16LF876A
PIC16F877	368	8	8192	USART, I ² C, SPI	20	3+WDT	PIC16LF877
PIC16F877A	368	8	8192	USART, I ² C, SPI	20	3+WDT	PIC16LF877A
PIC16F88	368	7	4096	AUSART	20	3+WDT	PIC16LF88

Tabla 7. Microcontroladores de la familia 16 de Microchip con memoria Flash

Los PIC's de la familia 16 tienen más prestaciones que en la familia 12. La mayoría de ellos posee un puerto de comunicación serie, comparadores, y *timers* adicionales. Las memorias oscilan entre los 1024 *Bytes* y los 8192 *Bytes*, aunque no es una característica que justifique nuestra elección.

El coste de los microcontroladores puede ser concluyente en la elección del dispositivo, puesto que para conservar el principio de modularidad del sistema, se debe utilizar un microcontrolador por placa de control.

A continuación se presenta un tabla con la relación de los precios, según el catálogo de RS 2002/2003, de los PIC's que más se adaptan a las necesidades del sistema.

Familia 12	Precio (€)
PIC12F675-I/P	2,51

Familia 16	Precio
PIC16F627-04/P	5,6
PIC16F628-04/P	6,36
PIC16F84-04/P	6,25
PIC16F870-I/SO	8,11
PIC16F871-I/P	9,6
PIC16F874-04/P	14,41
PIC16F876-04/SP	14,24
PIC16F877-04/P	15,55

Tabla 8. Lista de precios de los posibles microcontroladores

Los precios de los dispositivos con más recursos son evidentemente mayores. Destaca en esta lista el microcontrolador de la familia 12, que tiene el coste más bajo. Las características principales del PIC12F675 se presentan a continuación para determinar si puede ser utilizado en el circuito de control.

- 35 instrucciones.
- Tiempo de instrucción e 200 ns.
- Interrupciones.
- 6 I/O pins
- Conversor digital analógico con 4 canales
- 2 Timer/contadores
- Direccionamiento directo, indirecto y relativo.
- WDT
- *Brown-out-detect* (detección de nivel bajo de alimentación)

Las necesidades de nuestro sistema requieren un dispositivo con estas prestaciones. Su precio es inmejorable para diseñar un sistema modular. Por lo tanto, el PIC12F675 es el microcontrolador que se adaptará para nuestro sistema.

2.2.3 Programa

El programa ha sido realizado en lenguaje de programación C, y traducido a lenguaje ensamblador por el compilador HI-TECH PICC. De esta forma la programación se realiza de una forma más visual, que permite una rápida detección de los errores de código y facilita las posibles modificaciones del programa.

El HI-TECH PICC Compiler se utiliza para microcontroladores de la familia 12, 14, 16 y 17. Dependiendo del dispositivo elegido hay que seleccionar un archivo con extensión “.h” para adaptar el programa al lenguaje ensamblador. En este archivo se indican las nomenclaturas de los registros y sus bits, con sus respectivas direcciones. Utilizándolas se facilita el trabajo de programación y se reduce el margen de error, puesto que no hay que introducir las direcciones de memoria.

El archivo que se introduce en este caso en la cabecera de el programa es el “pic12f6x.h”. En el anexo 1 se puede ver su contenido.

El microcontrolador deberá hacer el tratamiento de la información a fin de realizar el control de la temperatura del *heater*.

En la inicialización del sistema, y a través de los datos enviados por el ordenador, se realizará la adquisición de la temperatura de trabajo del sensor. La consigna se envía en forma de señal analógico al *pin* GPIO0 del microcontrolador, y una vez es estable esta señal, se activa el *pin* GPIO4 para avisar al PIC de que la señal está lista.

El conversor analógico/digital realiza entonces la conversión de la señal, y esta queda fijada para todo el tiempo de trabajo de los sensores. Si se desea variar la temperatura del *heater* deberá ser reinicializado el sistema.

La resistencia que ha de tener el *heater* se calcula con la temperatura que se ha fijado. Una vez realizado este cálculo el programa entra en un bucle infinito. Cada 7,69 ms el programa ejecuta una rutina. Este tiempo es determinado por el *timer* 1, que oscila a una frecuencia de 130 Hz. Dentro de esta rutina hace las lecturas de los valores analógicos provenientes de las tensiones del sensor y de la resistencia que se encuentra en serie con él. A través de estos valores determina la tensión de *heater* y la intensidad que fluye por él, para poder calcular su resistencia.

Los valores de la resistencia real y de consigna se restan para hallar el error del sistema. Este error se multiplica por 10 para aumentar la sensibilidad. El valor resultante será el que adquirirá la variable que determina el *duty cycle*.

El lazo de control queda representado en la siguiente figura.

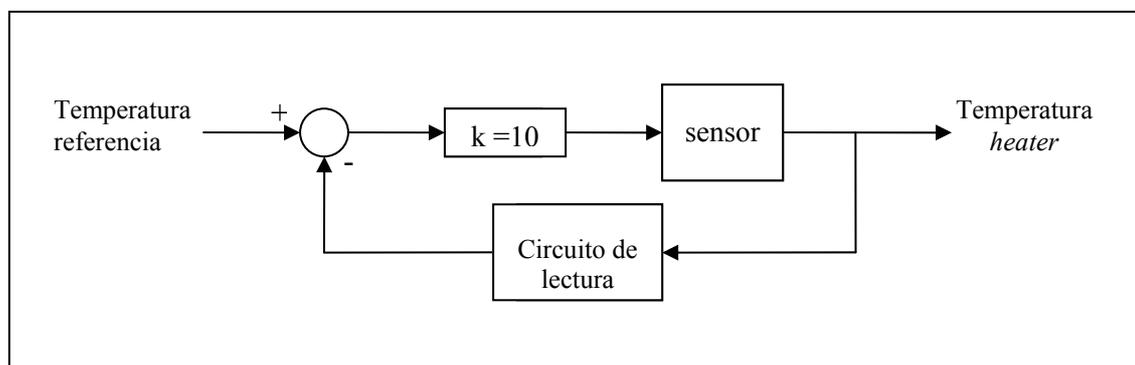


Figura 28. Esquema del circuito de control

Con el aumento del *duty cycle* lo que se consigue es aumentar el ciclo de trabajo del PWM. Este se ha implementado utilizando como base el *timer 0*. Cada 2,04 ms, es decir, a una frecuencia de 488,3 Hz, el *timer 0* provocará una interrupción. A cada interrupción el *timer* será cargado con un dato que corresponderá al nivel alto o al nivel bajo de la señal de pulsos variable que se obtiene en el *pin* GPIO5. La variable “nivel” avisa al programa de cual es el ciclo que acaba de ejecutarse. En el caso de haberse cumplido el ciclo de nivel alto, el tiempo que deberá cargarse será el del ciclo a cero. El valor que se carga en el registro del *timer 0* se tendrá que cargar teniendo en cuenta que el *timer* es incremental, y que produce la interrupción cada vez que se desborda de 0xFF a 0x00. Por lo tanto, si el *duty cycle*, está a 0, el valor cargado para el ciclo positivo de la señal PWM será 0, y el del ciclo a cero será “0xFF-dc”, es decir, 256, el máximo con que se puede cargar el contador, pero que producirá el tiempo más corto en provocar otra interrupción, ya que se desbordará al aumentar una unidad.

El ciclo de lectura de los canales analógicos 1 y 2 se repite cuando el *timer 1* se ha desbordado de nuevo. De forma que el *duty cycle* no varía en cada ciclo del PWM.

El programa está distribuido en el programa principal, y las funciones siguientes:

- void main (void) – En el programa principal se realizan las llamadas a las funciones de inicialización y lectura de canales analógicos. Además de hacer las operaciones para el control de la temperatura dentro de un bucle infinito.

- void inicialitzacions(void) – Es donde se configura los *timers*, las interrupciones, los *pins* de entrada/salida y el convertidor A/D del microcontrolador.

- void leer(char canal) – Aquí se realiza la lectura del canal indicado en la variable “canal”. Hay un bucle en la función que espera a que la conversión se haya realizado.

- void interrupt interrupcions(void) – Función a la que se salta cuando se produce una interrupción. En ella se salta a dos funciones más dependiendo de la fuente de la interrupción.

- void zero(void) – Después de producirse una interrupción provocada por el desbordamiento del *timer 0*, se salta a esta función. Donde se carga de nuevo en *timer* con el valor del ciclo a *ON* o a *OFF*.

- void uno(void) – Después de desbordarse el *timer1*, salta aquí donde se recarga con el valor definido que determina el tiempo que tarda en hacer una revisión del *duty cycle*.

Las variables que han sido utilizadas en el programa tienen tamaños diferentes dependiendo de su función. Las más destacables son las que se utilizan para el cálculo de las resistencias. En este caso se trabaja en coma flotante para lograr una mayor precisión de los cálculos. Las variables de tensión y intensidad del *heater*, al igual que los valores de las resistencias de referencia y real, tienen un tamaño de 32 bits.

Las demás variables dependen de su aplicación, de las magnitudes máximas que deban tratar. Hay de dos tamaños, de 8 y de 16 bits.

3. FABRICACIÓN DEL CIRCUITO

3.1 Fabricación del prototipo de la placa de control

Una vez diseñado el circuito y programado el microcontrolador, se continua el proceso implementando el circuito en una placa *standard* de topes CT-3. Es ella no se hace el montaje del circuito completo. El conversor digital/analógico se sustituye por un potenciómetro que nos proporcione un voltaje de 0 a 5 voltios. El comparador lógico de selección de placa no es necesario puesto que solo se construye una placa y no hay porqué realizar una selección.

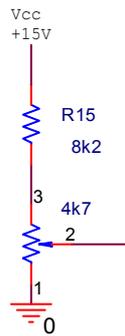


Figura 29. Circuito de provisional de consigna

El regulador de tensión de 5 voltios si que se incluye en esta placa para la alimentación del microcontrolador. La placa de control final no incluirá este regulador ya que un solo regulador situado en la placa base proporcionará la alimentación de 5 voltios a todos los microcontroladores.

Para la implementación de los amplificadores operacionales LM741, se utilizarán circuitos integrados que incluyen 4 A.O. en un chip, como el LM324N. Para proteger el microcontrolador, debido a que se trata de un circuito de pruebas, y se puede producir picos de voltaje mayores de 5 voltios en las entradas analógicas del microcontrolador, se han colocado dos diodos zener que aseguran que no se superarán los 5,1 V en las entradas analógicas.

A continuación vemos el esquema del circuito prototipo.

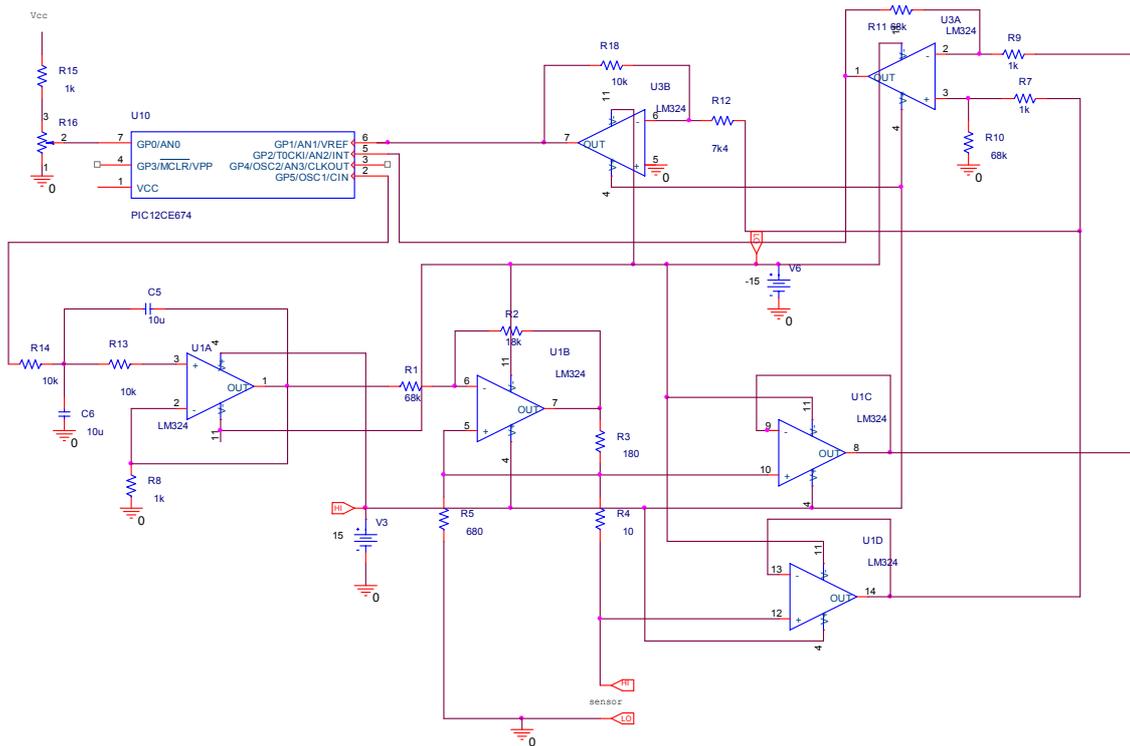


Figura 30. Esquema del circuito prototipo para la placa de topos

Esta placa se utilizará para realizar pruebas sobre funcionamiento del sistema de control. La salida del circuito está conectada a una membrana del sensor, y como se verá en el apartado de ensayos, se comprobará la correcta regulación del sistema para variaciones de la temperatura ambiente de 100 °C.

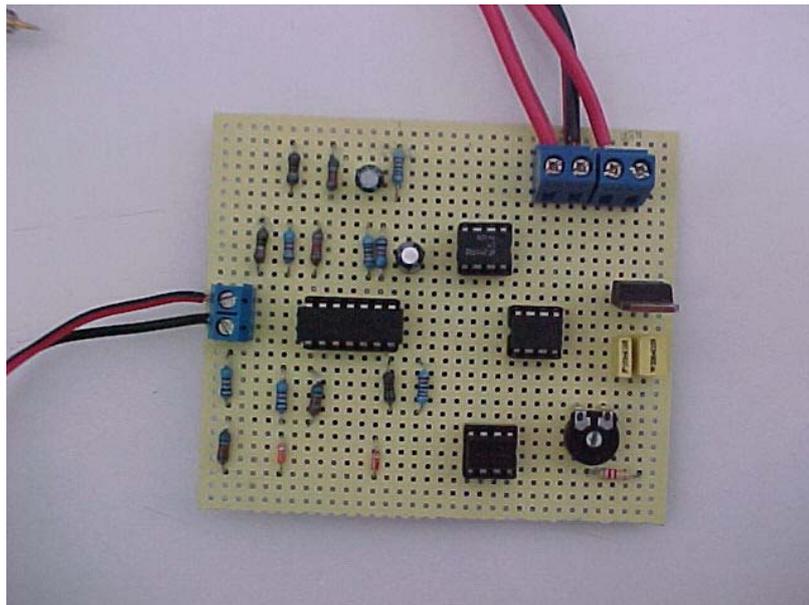


Figura 31. Prototipo de la placa de control

Para conectar el sensor a la placa de control de forma que este pudiese situarse dentro del horno para realizar las pruebas, y la placa de control fuera de él, se fabricó una placa de topos con un patillaje físicamente igual que el del sensor. La placa dispone de dos conexiones para poder utilizar las membranas de 650 Ω o las de 550 Ω .

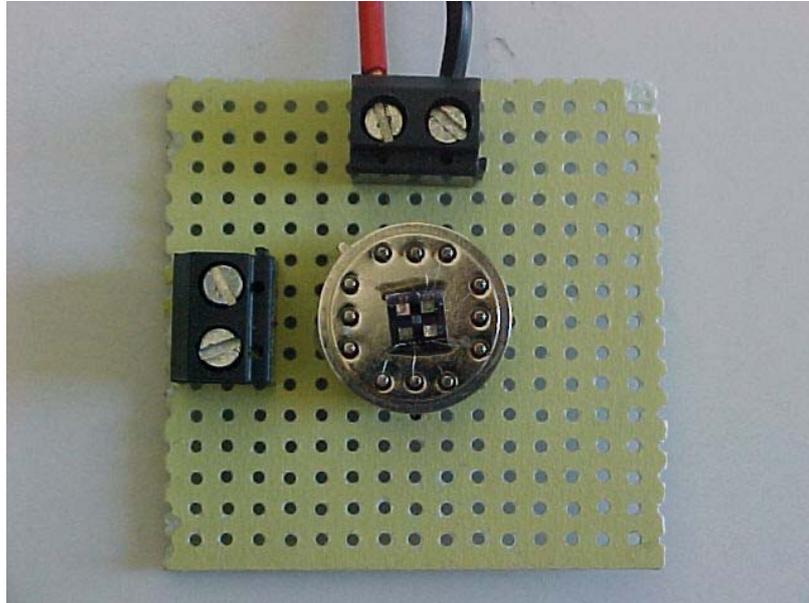


Figura 32. Placa de soporte y conexión del sensor

3.2 Fabricación de la placa base

El diseño de las placas ha sido realizado con el programa Orcad 9.2. Se ha diseñado dos tipos de placa, la de control y la placa base a la que se conectarán estas. El conjunto del sistema se compone de 9 placas. Por lo tanto la placa a la base pueden conectarse 8 placas de control, una para cada membrana de los dos sensores a controlar.

La placa base conectará la tarjeta de adquisición de datos National Instruments PCI6023E del ordenador con los circuitos de control. También proporcionará la alimentación de ± 15 V y de +5 V, además de realizar la conversión digital/analógico del valor de consigna de temperatura.

La placa base está formada por los siguientes componentes:

- 1 Regulador de tensión LM7805C
- 1 Convertidor analógico/digital MAX517
- 7 Terminales verticales de 2 polos
- 1 Conector PCB 20 *pins* para cable plano
- 8 Conectores de borde para PCB de 22 vías
- 1 Condensador de 300nF
- 1 Condensador de 10nF

La placa estará formada por conectores de borde para montaje directo sobre PCB, tipo *slot*. Se trata de conectores de doble cara con un total de 44 contactos, que en el diseño realizado se limita a 22 contactos al estar duplicados los contactos. Es decir, los contactos están unidos de forma que la pista haga contacto independientemente de la cara en la que se encuentre. Los conectores tienen un paso de 3,96 mm y una longitud de 108,7 mm. La placa dispone por lo tanto de un *bus* que tiene la siguiente configuración:

- Línea 1:	+15 V	
- Línea 2:	GND	
- Línea 3:	-15 V	
- Línea 4:	SEL 0	Entrada
- Línea 5:	SEL 1	Entrada
- Línea 6:	SEL 2	Entrada
- Línea 7:	CONFIRM	Entrada
- Línea 8:	+5 V	
- Línea 9:	Vanalog	Entrada
- Línea 10:	SENSOR	Salida

Los bits SEL son los que permiten la selección de la placa que va a ser inicializada con el voltaje de consigna. El bit CONFIRM es el que se activa para avisar de que el dato analógico está disponible para ser leído. El bit 9, Vanalog, es el valor analógico que determina la temperatura de consigna de cada microcontrolador, y finalmente el bit SENSOR es la única salida de la placa de control, que conecta esta con el sensor, al que suministra mediante este *pin* la intensidad que fluirá por él.

Cada placa de control dispone de un bit de salida para excitar el sensor. Lo conjunto de 8 bits están conectados a unos terminales que podrán conectarse con la placa que contiene los sensores. Un total de 8 terminales de salida y 2 de masa son los que conectarán los sensores con la placa base.

Un conector macho PCB para cable plano de 20 *pin* comunicará el ordenador con la placa. Los *pins* de salida del ordenador serán los siguientes.

- <i>Pin</i> 6:	Bit SEL0	Bit D7 de la tarjeta PCI6023E
- <i>Pin</i> 8:	Bit SEL1	Bit D6 “
- <i>Pin</i> 10:	Bit SEL2	Bit D5 “
- <i>Pin</i> 12:	Bit CONFIRM	Bit D4 “
- <i>Pin</i> 14:	Bit SCL	Bit D3 “
- <i>Pin</i> 16:	Bit SDA	Bit D2 “

Con esta configuración de *pins* de salida de la tarjeta PCI6023E y con el protocolo de comunicación del convertidor D/A MAX517, se realizará un programa de inicialización que permita la comunicación con los microcontroladores de cada placa de control.

El circuito que se implementará es el siguiente:

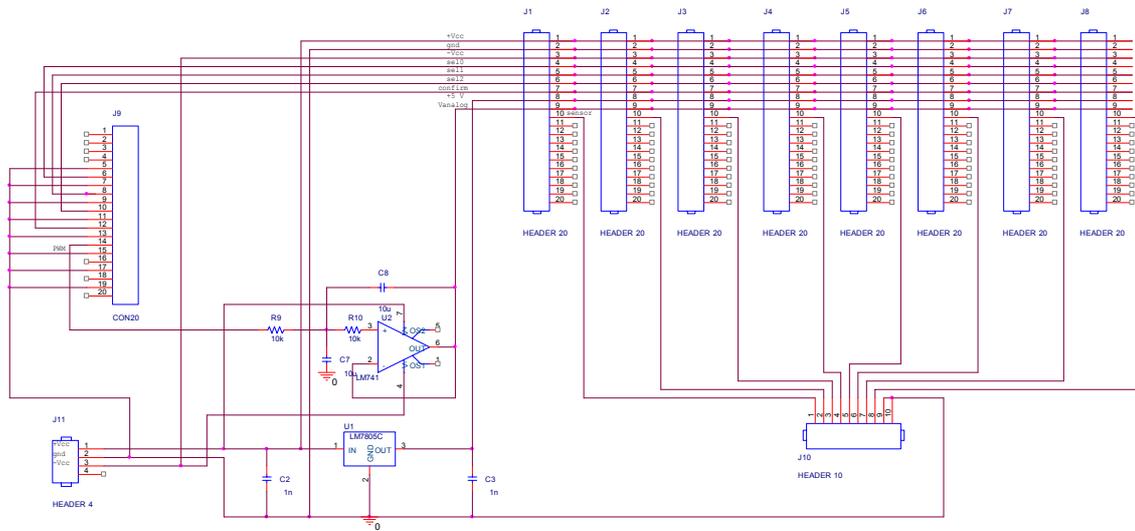


Figura 33. Circuito de la placa base

La placa PCB se realizará a doble cara. Esto permite evitar puentes para que se más sólida. El problema que supone realizar una placa a dos cara es en la soldadura de los componentes. Vemos a continuación el *layout* de la placa base.

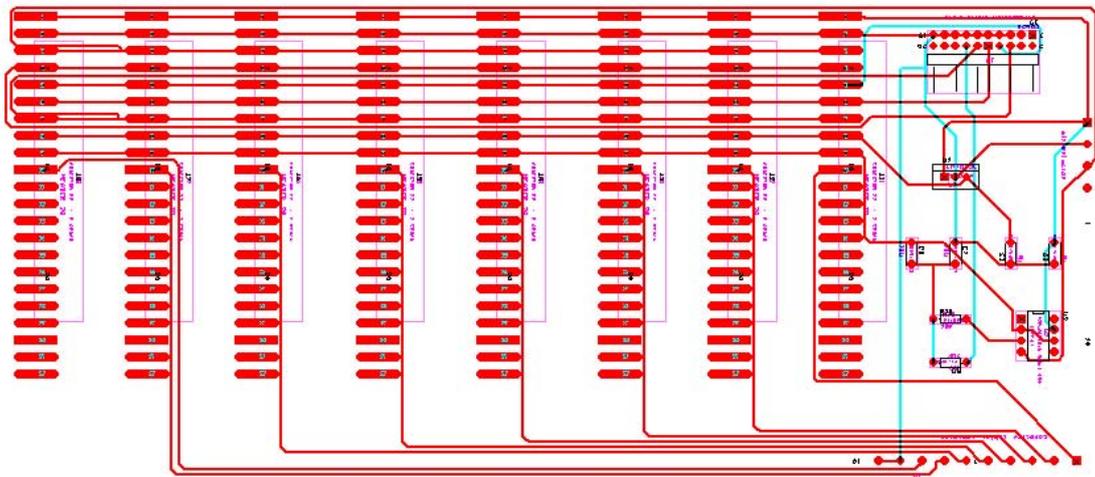


Figura 34. Layout de la placa base con las dos caras superpuestas

El resultado de las pistas de cada capa es el siguiente.

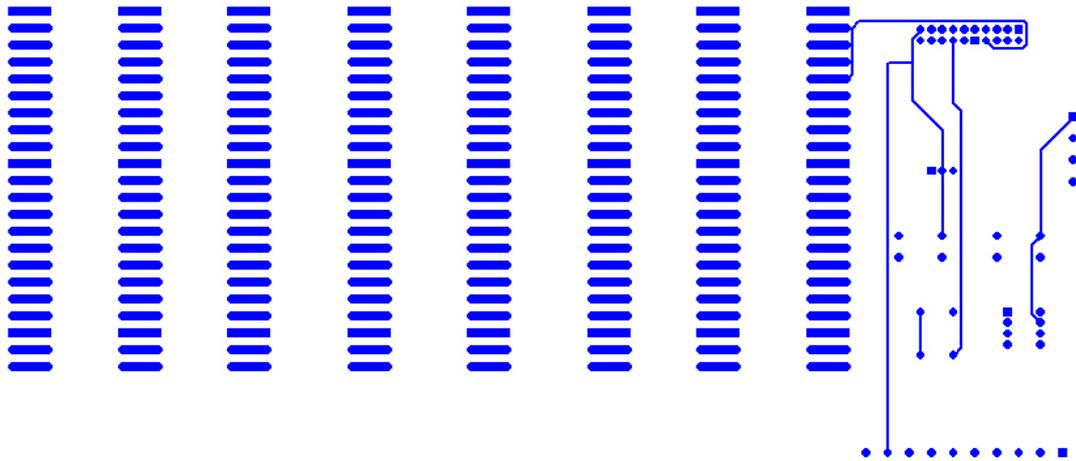


Figura 35. Pistas de la cara superior de la placa base

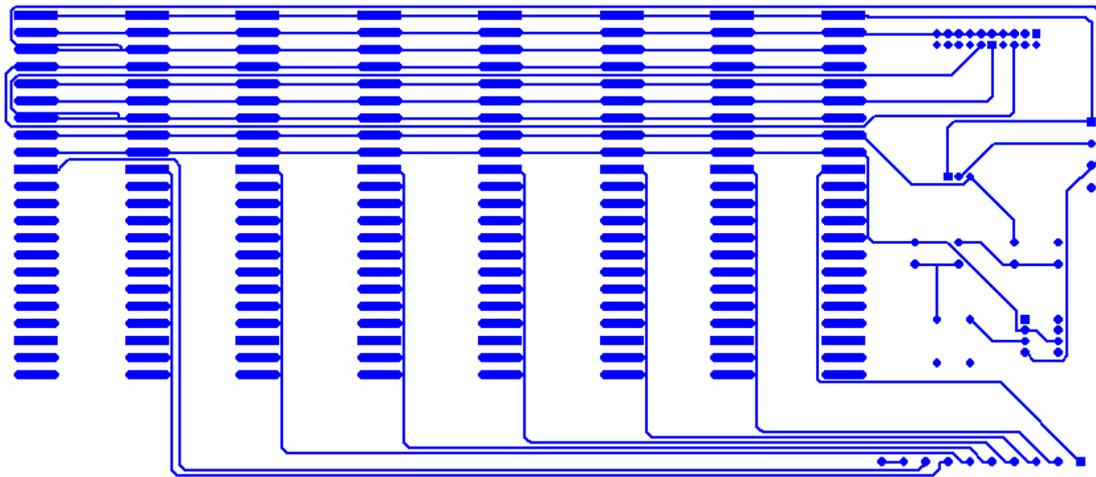


Figura 36. Pistas de la cara inferior de la placa base

3.3 Fabricación de la placa de control

Después de ver como la placa base tiene la circuitería y el conexionado adaptado para la comunicación con el ordenador, hay que ver como se adapta la tarjeta de control a la placa base.

El conexionado con la placa base se realiza mediante el conector de borde de 22 vías. Para esto, en un lateral de la placa se ha diseñado las pistas para que se adapten a los conectores. En al *layout* se aprecia el tipo de conexionado.

La placa de control está formada por los siguientes componentes:

- 1 Microcontrolador 12F675
- 2 C.I. LM324 con 4 Amplificadores Operacionales
- 1 Comparador lógico de 4 bits
- 1 Cuádruple puerta AND de dos entradas
- 3 Microinterruptores
- 2 Condensadores de 10 μ F
- 1 Resistencia de 10 Ω de 1 % tolerancia
- 1 Resistencia de 180 Ω de 1 % tolerancia
- 1 Resistencia de 680 Ω de 1 % tolerancia
- 6 Resistencias de 1 k Ω de 1 % tolerancia
- 1 Resistencia de 7k4 Ω de 1 % tolerancia
- 3 Resistencias de 10 k Ω de 1 % tolerancia
- 3 Resistencias de 68 k Ω de 1 % tolerancia

El circuito de control recibe por el bus los datos de selección de tarjeta, y si coinciden con el número de tarjeta programado el los microinterruptores, se carga la temperatura de consigna que tendrá el *heater*. A parte del circuito de selección, el circuito de control es quién realiza las operaciones adecuadas y que tiene como salida la línea 10 del conector de 22 vías.

El *layout* de las dos caras es el siguiente:

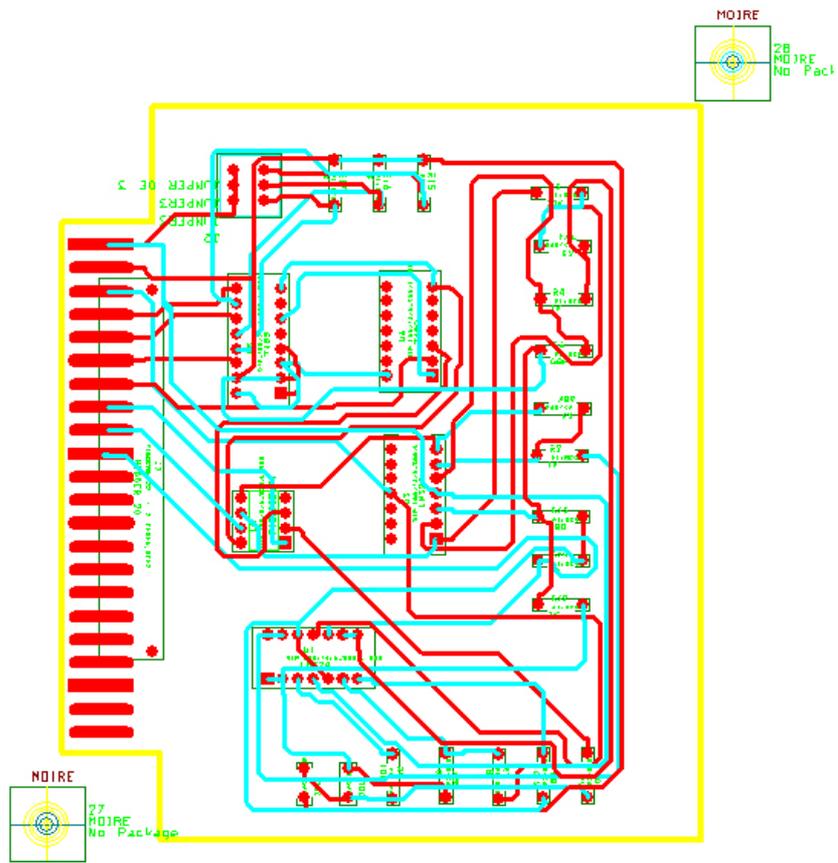


Figura 38. *Layout* de la placa de control con las dos caras superpuestas

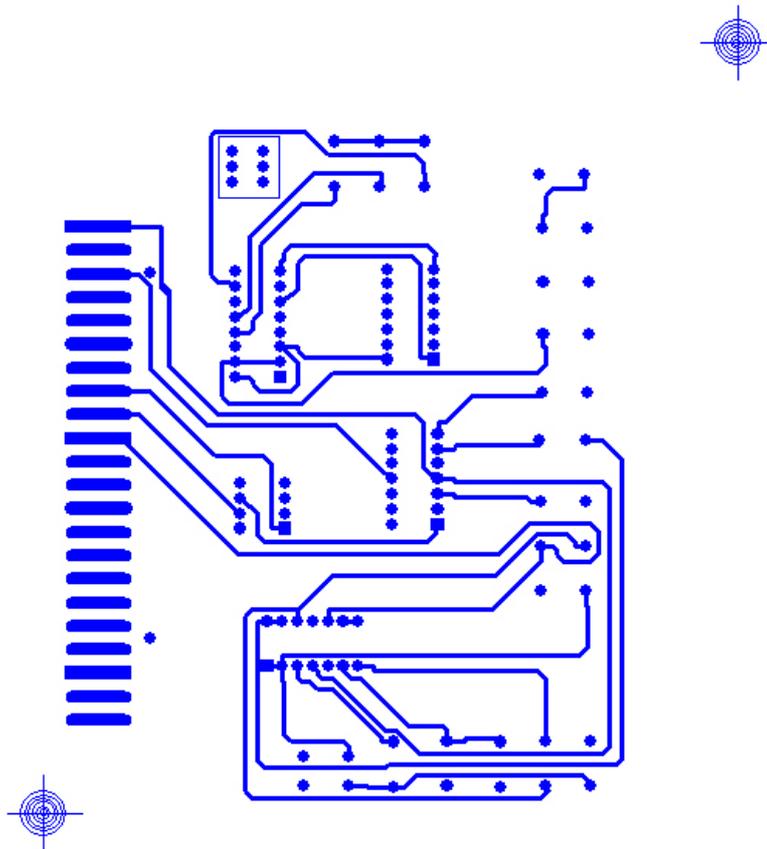


Figura 39. Pistas de la cara superior de la placa de control

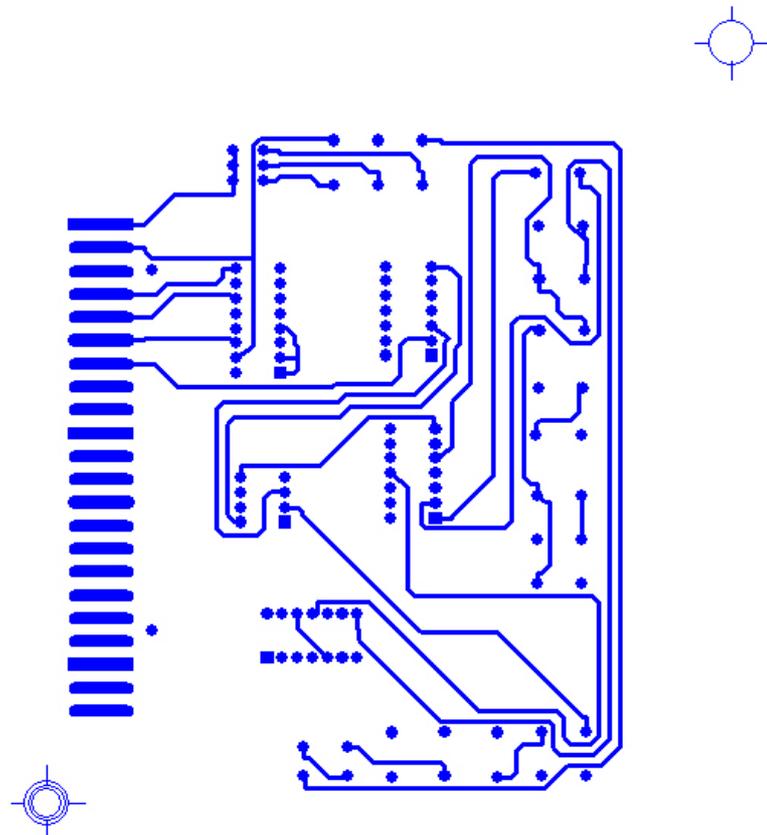


Figura 40. Pistas de la cara inferior de la placa de control

4. ENSAYOS

4.1 Pruebas del control de temperatura

4.1.1 Pruebas con prototipo

Para comprobar que se ha realizado un diseño correcto y que realiza un buen control de temperatura, se realizan unas pruebas con el prototipo de la placa de control. Se utiliza el prototipo para tener una mayor flexibilidad a la hora de comprobar los márgenes de temperatura que puede alcanzar el sistema. El circuito del prototipo cuenta con una tensión de consigna regulable, con lo que se pueden hacer pruebas a diferentes temperaturas fácilmente.

Se realizarán dos tipos de ensayos para comprobar el control de la temperatura. El primero servirá para comprobar si el sistema responde correctamente a las variaciones de la temperatura del entorno. La segunda prueba se realizará para ajustar el microcontrolador a la frecuencia óptima para el control de la temperatura frente a variaciones de la temperatura ambiente.

El primer ensayo se realiza con una frecuencia de muestro de la tensión y la resistencia del *heater* fija de 80 Hz. Las pruebas se realizan en el laboratorio de investigación sobre sensores de gases y nariz electrónica.

Para realizar las pruebas se utiliza un horno de reflujo de aire TECHNO HA-02 igual al utilizado en los ensayos del circuito analógico, una fuente de alimentación, un osciloscopio Tektronics TDS 210 y un voltímetro.

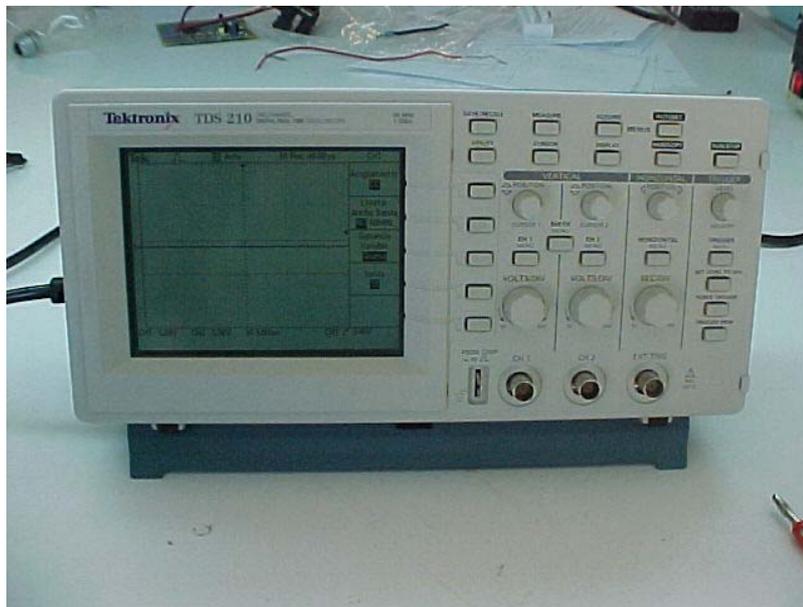


Figura 41. Osciloscopio Tektronics utilizado en las medidas

Las pruebas se realizan a dos tensiones de referencia diferentes de 2 y 3 voltios. La resistencia del sensor medida a temperatura ambiente de unos 25 °C es de 635 Ω. A continuación se presentan los resultados de las mediciones.

Temp. Amb. (°C)	Vi (V)	T ^a Consigna (°C)	Vh (V)	Ih (mA)	Rh (Ω)	Temp. Heater (°C)	Variación %
25	3,56	289	4,5	5,83	771,9	292,4	0,94
60	3,56	289	4,32	5,58	774,2	297,8	-0,89
70	3,56	289	4,25	5,5	772,7	294,4	0,26
80	3,56	289	4,13	5,33	774,9	299,4	-1,41
90	3,56	289	4,02	5,2	773,1	295,2	-0,01
100	3,56	289	3,92	5,08	771,7	291,9	1,11
25	4,51	365	4,92	6,11	805,2	369,8	0,06
60	4,51	365	4,74	5,89	804,8	368,7	0,37
70	4,51	365	4,67	5,79	806,6	372,9	-0,77
80	4,51	365	4,56	5,66	805,7	370,8	-0,20
90	4,51	365	4,48	5,57	804,3	367,7	0,64
100	4,51	365	4,39	5,45	805,5	370,5	-0,11

Tabla 9. Resultados de las pruebas del control de temperatura

Teniendo en cuenta que las medidas han sido realizadas con un osciloscopio y un voltímetro, y no son precisas dadas las oscilaciones de la tensión y las corriente del *heater*, provocadas por la regulación que efectúa el sistema. Se puede apreciar las variaciones de la temperatura del *heater*, y en la última columna se muestra la variación respecto la media de las temperaturas de estas medidas. Los porcentajes no superan el 2 % de variación. Se puede afirmar que el control de temperatura que realiza el sistema es excelente.

El sistema responde correctamente a las variaciones de temperatura. Mantiene la resistencia del *heater* constante mientras la temperatura del entorno varía. El sistema funciona correctamente.

Los siguientes ensayos están dedicados al ajuste del sistema. A su tiempo de muestreo. Este tiempo será determinante para la regulación a cambios de temperatura rápidos. Aunque cambios bruscos en la temperatura ambiente no son esperables. Pueden producirse cambios en la temperatura ambiente cuando hay variaciones en el flujo de aire que llega a la cámara de los sensores. Aunque el flujo que atraviesa la cámara lo mantiene constante el controlador de flujo másico.

Se buscará el tiempo de muestreo más adecuado para responder de una manera rápida sin que el sistema se desestabilice.



Figura 42. Cámara térmica KPK 35

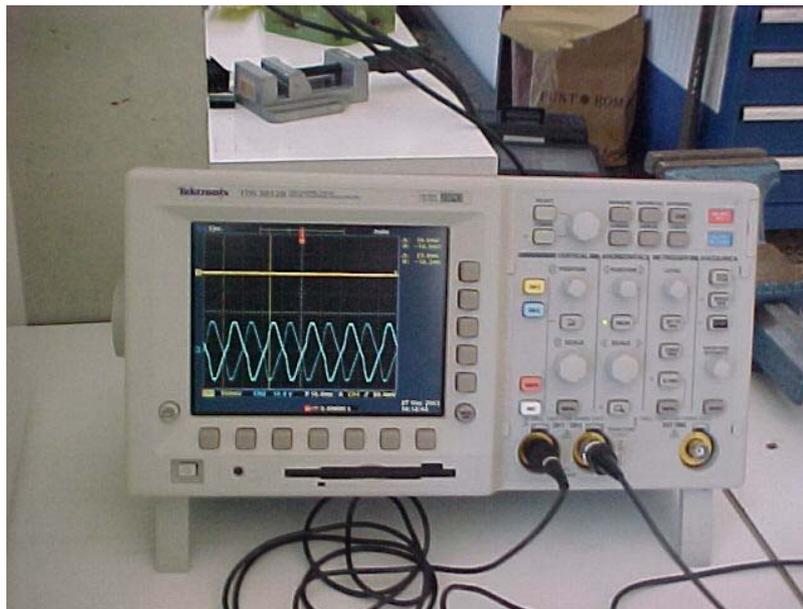


Figura 43. Osciloscopio TDS3012B utilizado en las medidas

Las pruebas se realizan con una cámara térmica KPK 35, que permite el control de la temperatura y la humedad, utilizando una fuente de alimentación, y un osciloscopio Tektronix TDS3012B Digital, que permite la división de las señales de entrada. Esta operación se utilizará para obtener la resistencia del *heater* mediante la división de la tensión del *heater* y la tensión de una resistencia de $10\ \Omega$ colocada en serie con él. Esta resistencia se coloca entre el *heater* y masa. De esta forma alteramos los valores de lectura de la tensión y la intensidad que realiza el microcontrolador. Ya que estamos aumentando en $10\ \Omega$ la resistencia de la membrana. En este caso no importa el ajuste a la temperatura que se ha fijado en la consigna, ya que estas pruebas ya han sido realizadas anteriormente.

Lo que se pretende ajustar ahora es la frecuencia de muestreo del sistema para mejorar su tiempo de respuesta.



Figura 44. Montaje para el ensayo realizado con la cámara térmica

El osciloscopio TDS3012B permite registrar las gráficas que se visualizan. A continuación se muestran las señales de la tensión y la intensidad en el *heater*, además de la división entre estas dos señales.

La tensión del *heater* está en una magnitud correcta, pero la intensidad tiene una escala 10 veces mayor debido a que está multiplicada por la resistencia de $10\ \Omega$. Por lo tanto la resistencia representada por la división de estas dos señales tiene una escala 10 veces menor.

Como se puede ver en las gráficas, la señal correspondiente a la resistencia del *heater* parece tener mucho ruido. Esto se debe a las magnitudes que se están dividiendo. V_h es del orden de los voltios, y la tensión de la resistencia de $10\ \Omega$ es del orden de los mV. Así que esta última puede ser perjudicada por el ruido, ya que es una señal de baja intensidad. Lo que le sucede a la señal de la resistencia es que a pequeñas variaciones de la tensión que hay en la resistencia de $10\ \Omega$, se producen grandes variaciones en esta. Esto se ve afectado además por la velocidad de muestreo del osciloscopio, que no realiza las medidas en el mismo instante, por lo tanto, el rizado de las señales leídas afectan en la división. Los valores que se dividen para hallar la resistencia del *heater* están desplazados en el tiempo uno respecto del otro, puesto que no se pueden realizar la medidas en el mismo instante.

Para todas las señales que se muestran a continuación se representa en azul oscuro el voltaje del *heater*, en azul claro el voltaje de la resistencia de $10\ \Omega$ que hay en serie con él, y en rojo la división de las dos señales. También se indicará la posición de cada señal en la gráfica para facilitar posibles visualizaciones sin color.

La primera medida se realiza a una frecuencia de muestreo de 80 Hz. La temperatura de la cámara es de 30 °C.

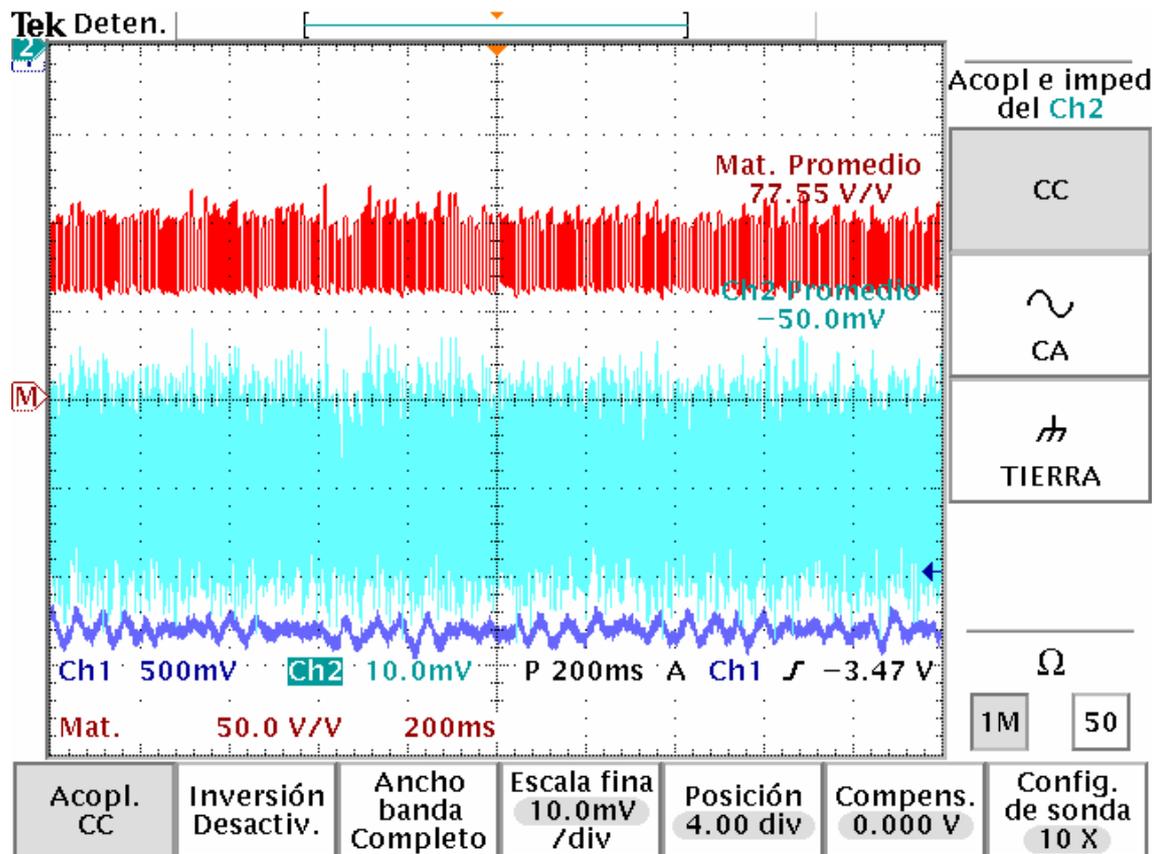


Figura 45. Frecuencia de muestreo de 80 Hz, $V_{ref}=3,19$ V, 200 ms/div

La señal que se encuentra en la parte superior es el resultado de la división entre la señal que representa el voltaje del *heater* que se encuentra en la parte baja de la imagen, y la señal del centro que es diez veces superior a la intensidad del *heater*.

Se puede apreciar en enorme rizado que provoca el ruido causado en la medida. Se puede asegurar que este rizado proviene de la medida, y no se produce en la resistencia del *heater* o la que está en serie con él, ya que el tiempo de respuesta del *heater* es de 20 ms, y como se verá en la gráfica siguiente, este rizado es mucho menor que este tiempo.

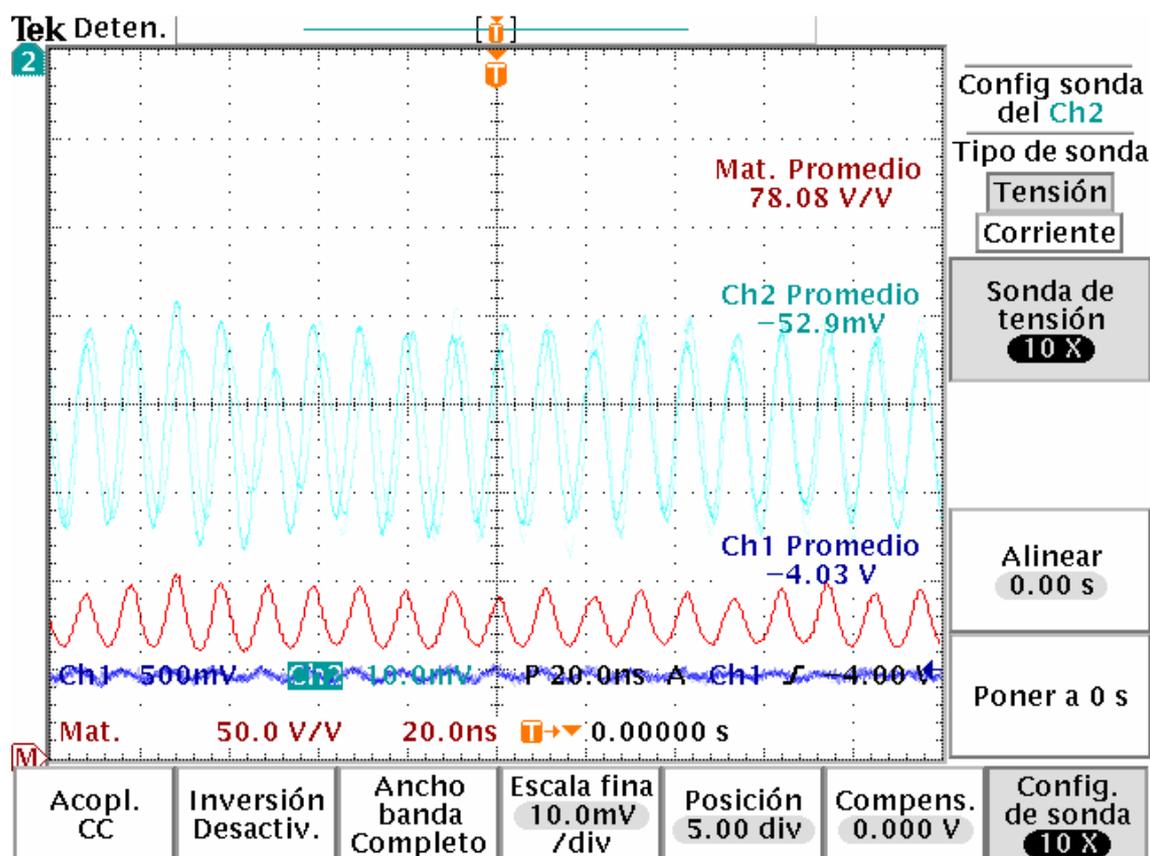


Figura 46. Frecuencia de muestreo de 80 Hz, $V_{ref}=3,19$ V, 20 ns/div

Aquí se aprecia mejor el tiempo el periodo del rizado, que tiene una frecuencia del orden de los 100 MHz. Queda descartada la posibilidad de que se trate de la señal que excita el *heater*.

La señal que representa la resistencia que se sitúa en la parte alta de la figura 53, tiene un valor promedio de 78,08 V/V, esto significa que la resistencia del *heater* es de aproximadamente 780 Ω , ya que se multiplica por 10.

La señal que se aprecia en el centro es el cociente de la tensión del *heater* y la intensidad a una escala 10 veces mayor.

Se han medido los valores de las tensiones en el *heater* y en la resistencia con un voltímetro, los valores son de $V_h=4,18$ V y $V_{r10}=55,8$ mV. La resistencia resultante de la división de estos valores es de $R=749$ Ω , a la que se ha de restar 10 Ω de la resistencia en serie para saber el valor de la R_h , que es de 739 Ω .

Si la temperatura de consigna es de 3,19 V, quiere decir que la temperatura debe ser de 258 $^{\circ}\text{C}$, y para ello la resistencia de *heater* deberá ser de 734 Ω .

Se puede afirmar que el control de temperatura es bueno, y que es el osciloscopio quien nos proporciona una mala lectura debido al ruido.

Ahora la frecuencia de muestreo del *heater* cambia a 60 Hz, la tensión de referencia se mantiene al mismo nivel.

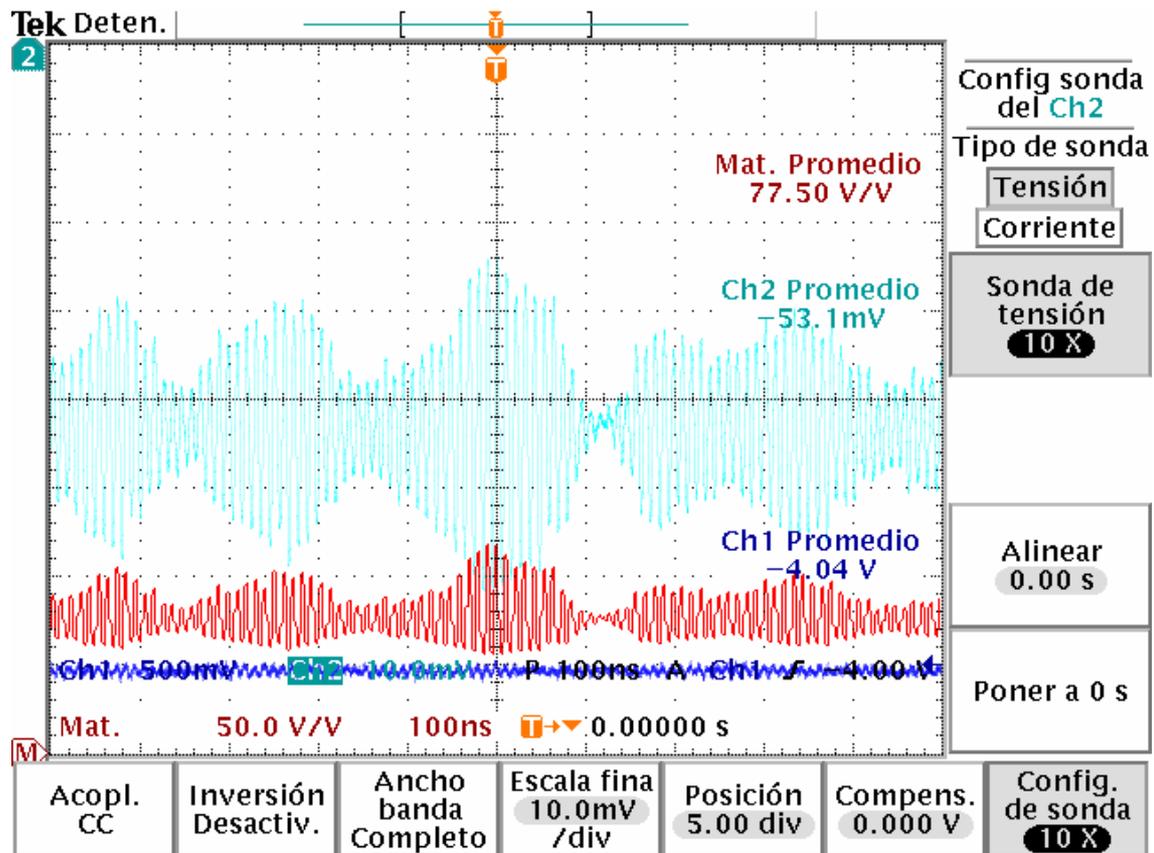


Figura 47. Frecuencia de muestreo de 60 Hz, $V_{ref}=3,19$ V, 100 ns/div

Las señales que se muestran son de arriba abajo, la tensión de la resistencia de 10Ω , el cociente y la tensión de la membrana.

Se aprecian las señales con otra escala de tiempos intermedia a las anteriores. No se aprecian grandes cambios en las señales, ni en los promedios.

La temperatura sigue siendo de $30 \text{ }^\circ\text{C}$ para las medidas realizadas a 60 Hz de frecuencia de muestreo.

A continuación se puede apreciar en la figura 55, la señal superior que muestra el error que se produce en la división de las señales provocado por el osciloscopio. La señal resultante alcanza un máximo que no es coherente con los valores que se pueden alcanzar tanto en el *heater* como en la resistencia en serie.

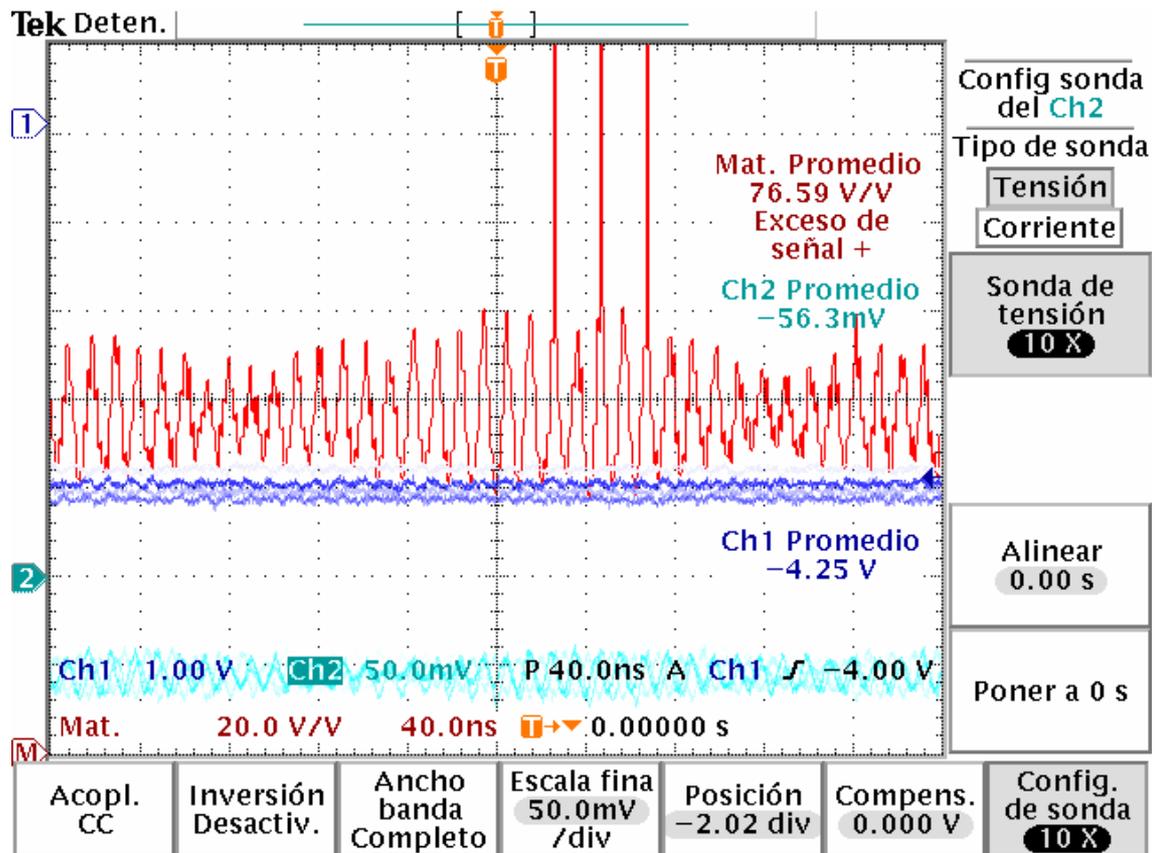


Figura 48. Frecuencia de muestreo de 60 Hz, $V_{ref}=3,19$ V, 40 ns/div

Ahora se aumenta la temperatura de la cámara hasta los 50 °C. Las gráficas siguientes muestran la estabilidad del sistema.

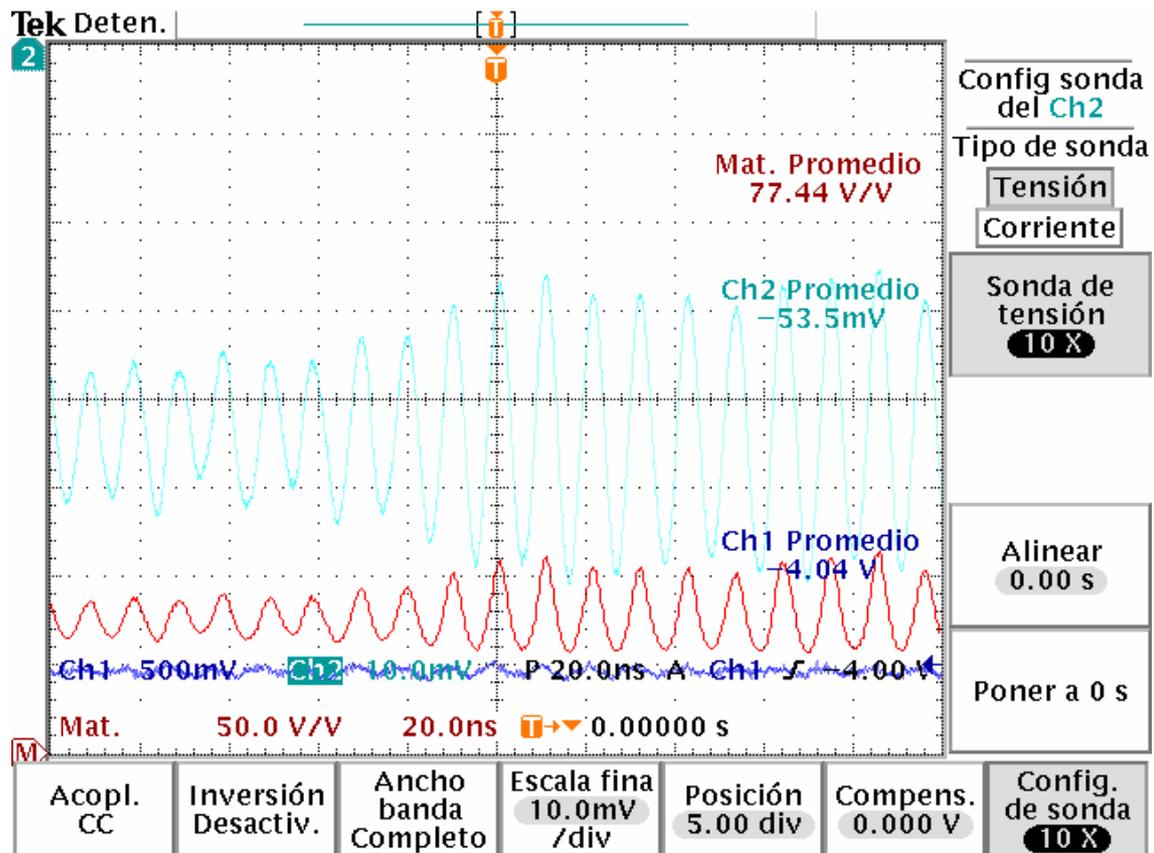


Figura 49. Frecuencia de muestreo de 60 Hz, $V_{ref}=3,19$ V, 20 ns/div

No se producen grandes variaciones en las señales que se obtienen. La señal inferior que tiene un rizado menor es la tensión del *heater*, y la superior es la que representa la intensidad. También se realizan medidas con un voltímetro para conocer la media de la tensión e intensidad del *heater*.

Los valores son $V_h=4.01$ V y $V_{r10}=53,4$ mV, donde la $R_h=740$ Ω . Solo ha variado en un ohmio de la resistencia medida con 30 $^{\circ}\text{C}$, el cual se puede considerar como error de medida.

A continuación se realizará una variación de la frecuencia de muestreo entre los 60 Hz y los 130 Hz para comprobar cual resulta más estable para realizar el control.

Frecuencias mayores de 130 Hz no son posibles puesto que la parte del programa en la que se realizan las operaciones de control tarda aproximadamente 6,5 ms en hacer el tratamiento de los datos.

Para las siguientes gráficas la señal que se encuentra en la parte superior en la tensión de la resistencia de 10 Ω que se encuentra en serie con el *heater*, que representa la intensidad del mismo con una magnitud 10 veces superior, y la otra señal es su tensión.

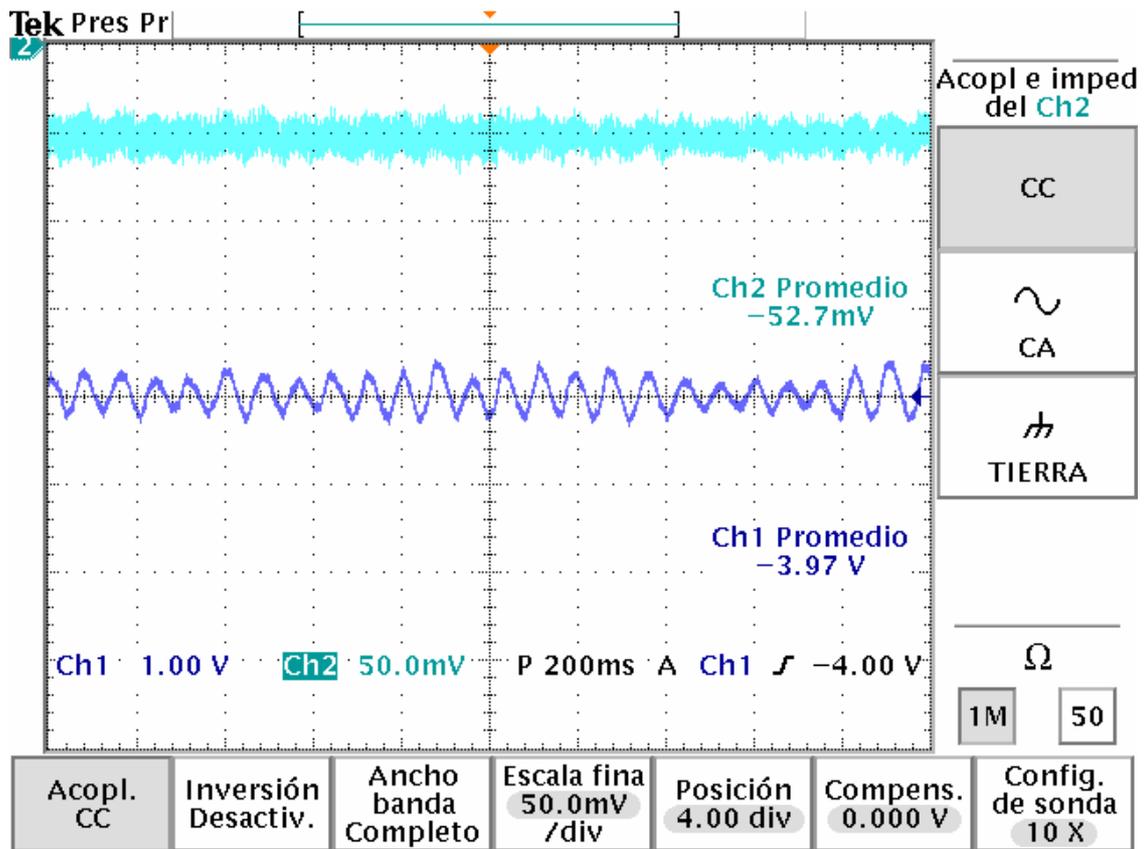


Figura 50. Frecuencia de muestreo de 60 Hz

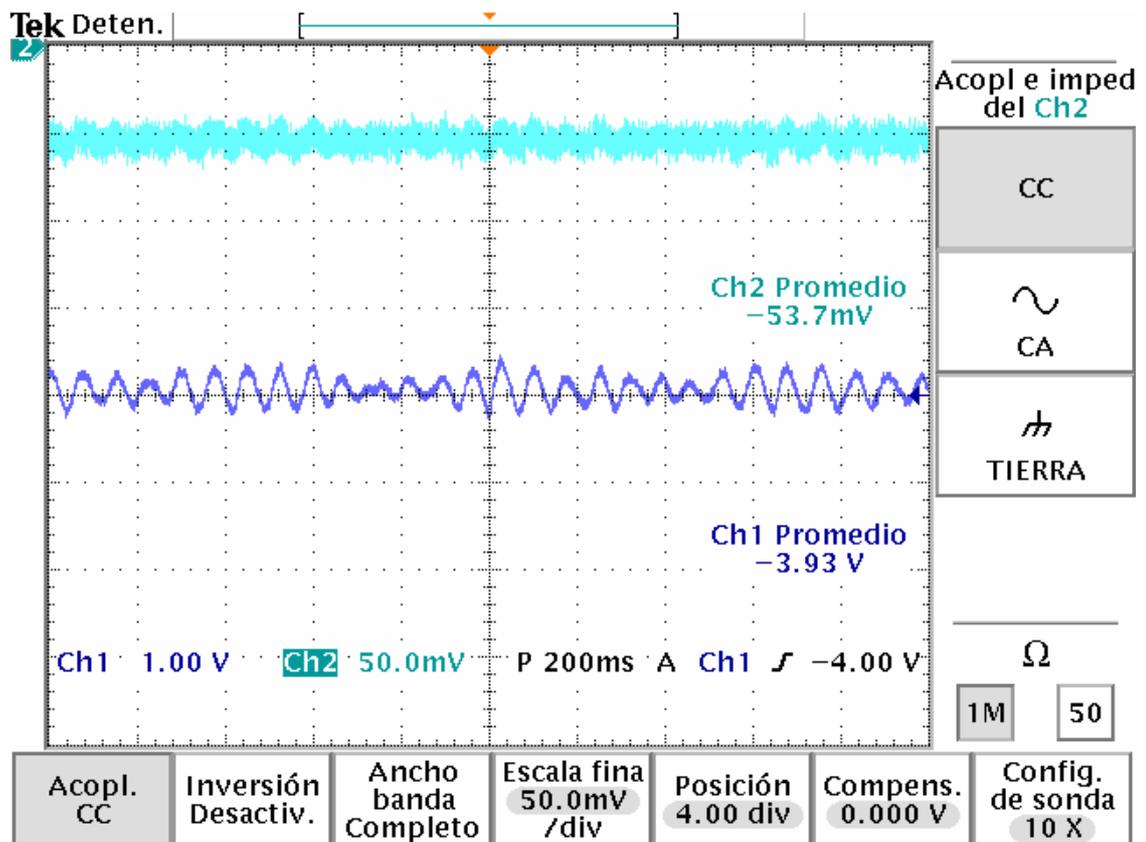


Figura 51. Frecuencia de muestreo de 80 Hz

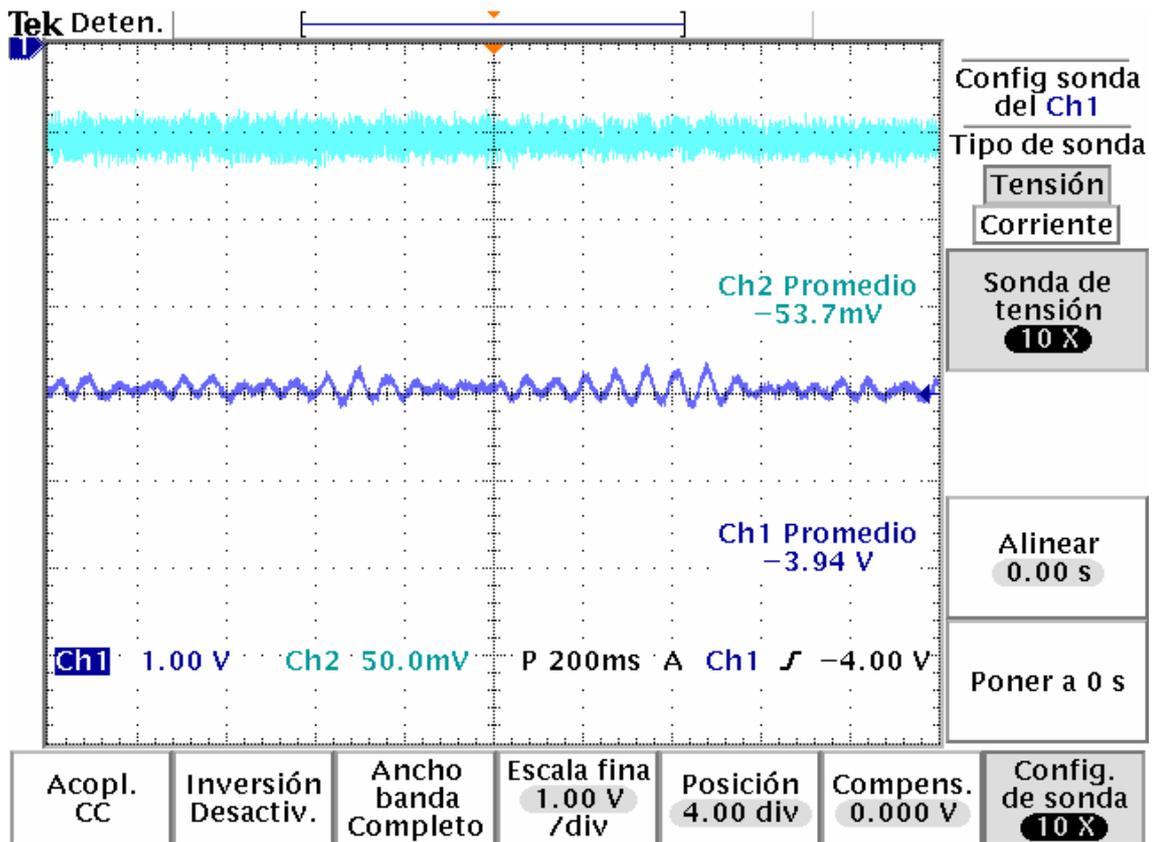


Figura 52. Frecuencia de muestreo de 100 Hz

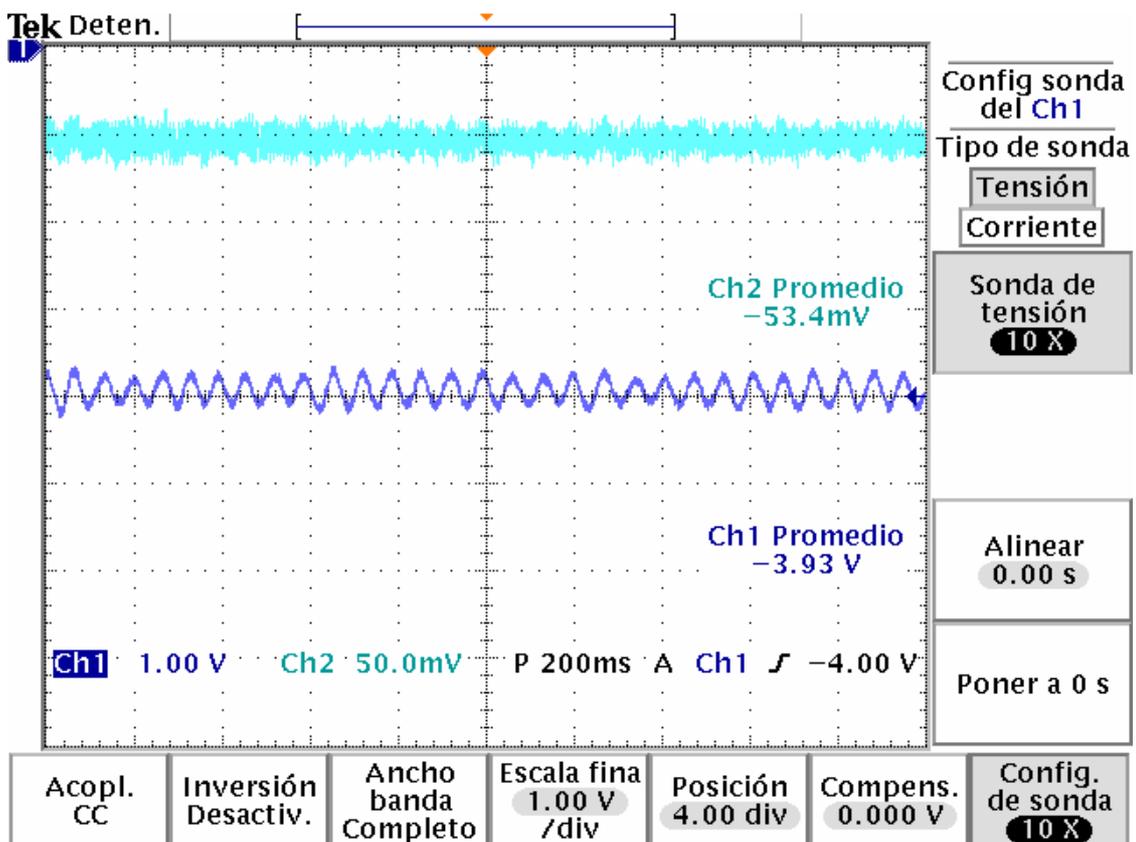


Figura 53. Frecuencia de muestreo de 120 Hz

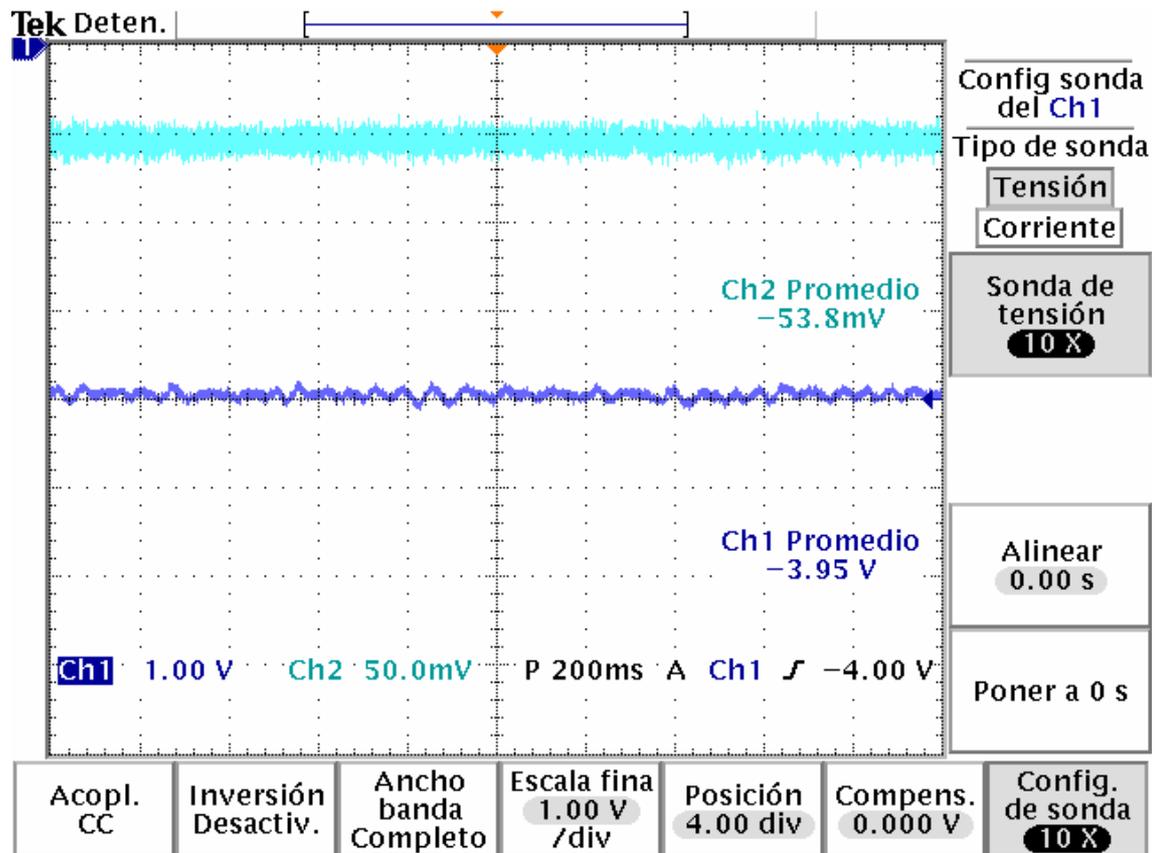


Figura 54. Frecuencia de muestreo de 130 Hz

Si se comparan las cinco gráficas a diferentes frecuencia de muestreo, se puede observar que ninguna de ella es inestable, pero que alguna tiene un rizado mayor que otras. Las señales están tomadas a una escala de tiempos en la que se pueden apreciar las oscilaciones de la señal del *heater*, que coincide con el tiempo de respuesta de éste.

La última gráfica, tomada con una frecuencia de muestreo de 130 Hz es la que presenta un menor rizado en la tensión del *heater*. Es normal que se produzca un rizado puesto que se está realizando el control de la temperatura del sensor, y hay que aumentar o reducir el voltaje de éste cuando lo requiera.

Según las gráficas, la frecuencia de 130 Hz es la más adecuada para el sistema. Todas la medidas que se realizan a partir de este momento son a esta frecuencia, que será la que tendrá el sistema en su funcionamiento sus aplicaciones posteriores.

Las medidas realizadas a continuación muestran los resultados obtenidos al aumentar la temperatura a 60 °C y a 70 °C. Se puede ver que el promedio de la resistencia se mantiene cerca de los 780 Ω . Lo que significa que el sistema responde correctamente a la frecuencia de 130 Hz.

La señal con mayor rizado es la intensidad del *heater* con una magnitud 10 veces mayor, su tensión se sitúa en la parte más baja de la gráfica.

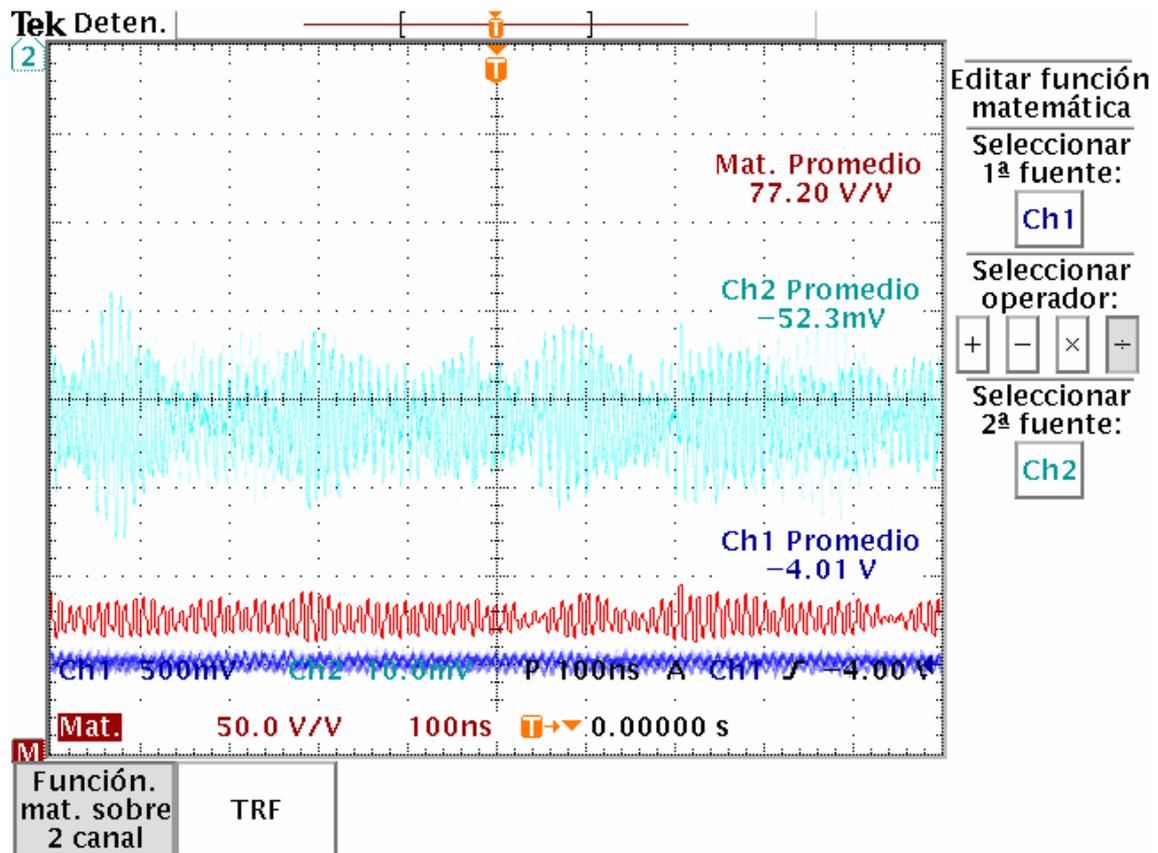


Figura 55. Temperatura de 60 °C. Frecuencia de 130 Hz.

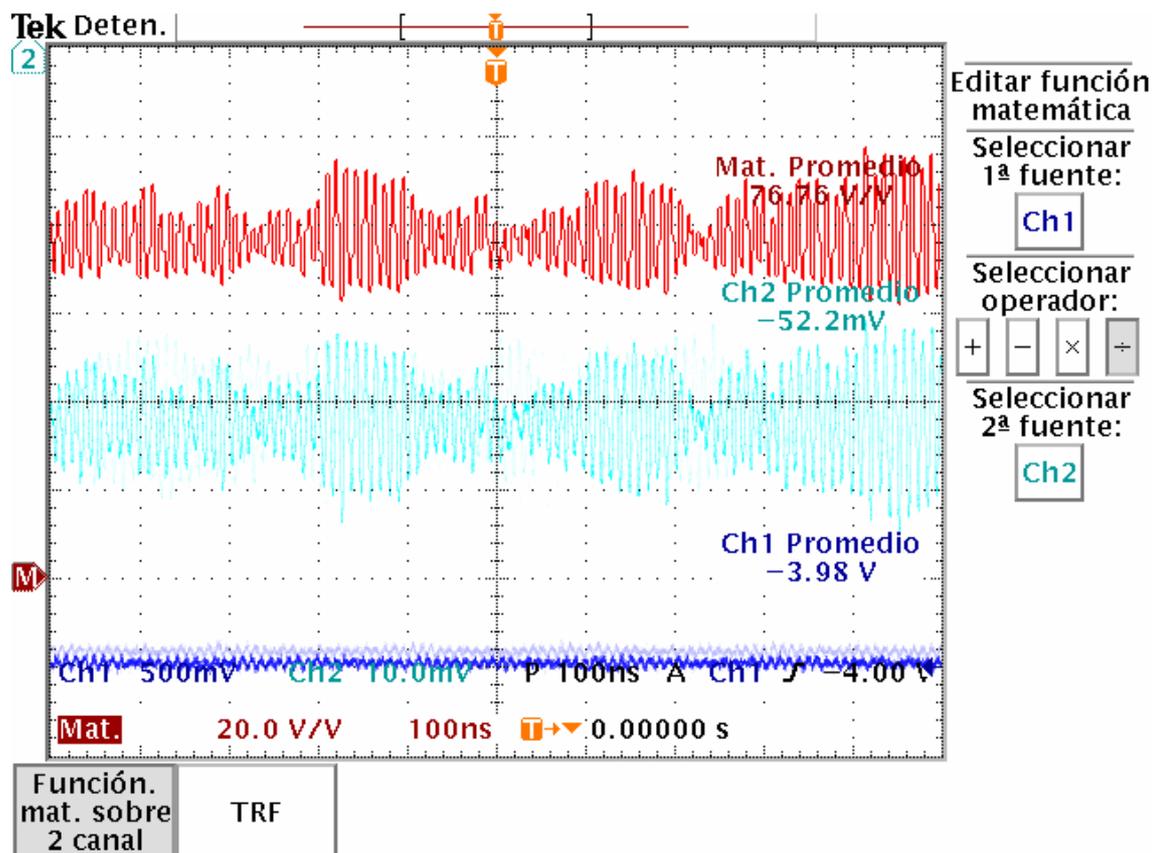


Figura 56. Temperatura de 60 °C. Frecuencia de 130 Hz.

Las señales de la figura 56, 57 y 58 son de arriba abajo, el cociente de las dos señales, la tensión de la resistencia y la del sensor.

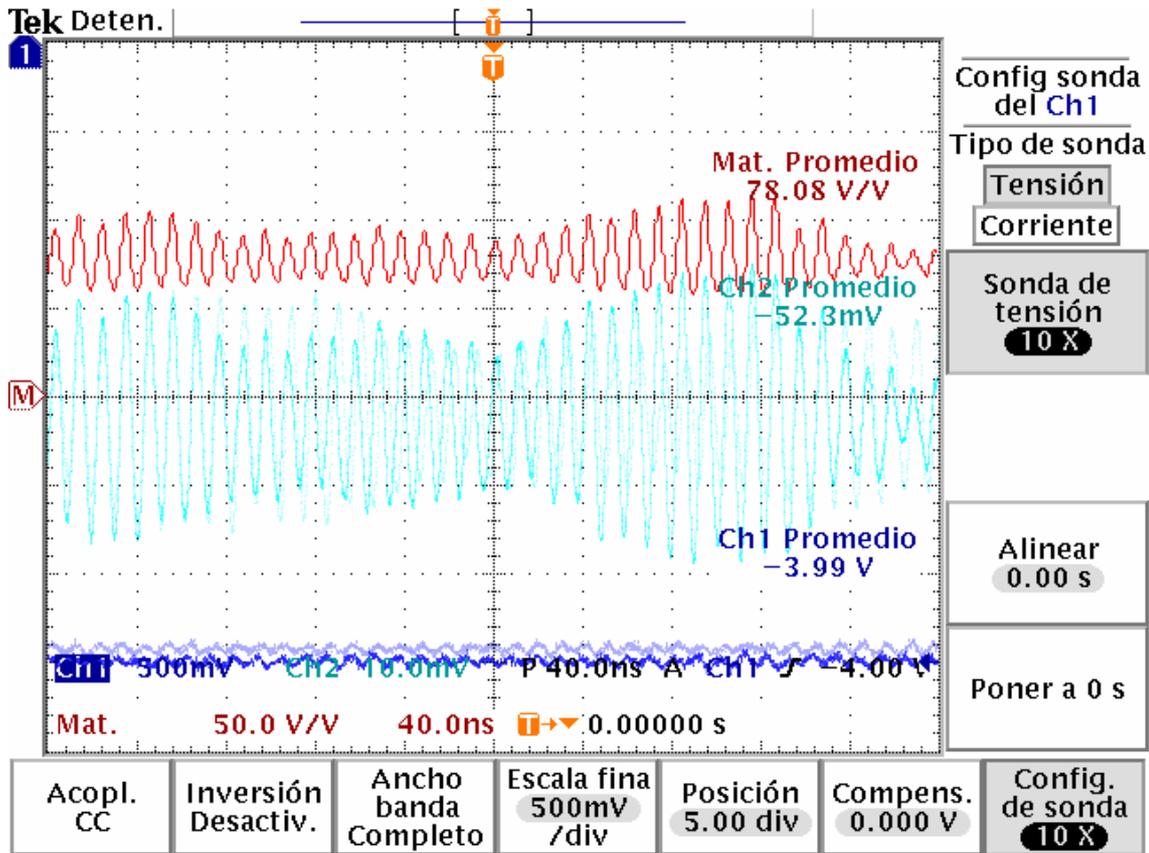


Figura 57. Temperatura de 70 °C. Frecuencia de 130 Hz.

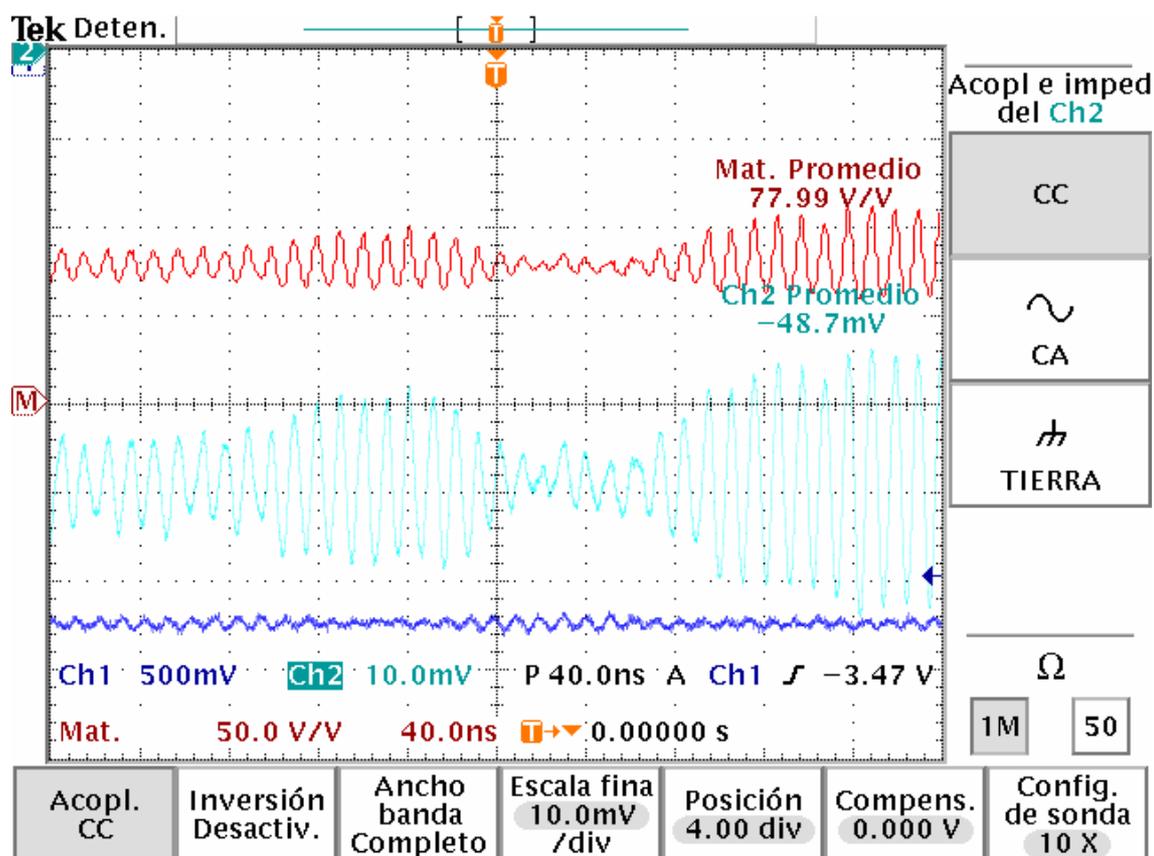


Figura 58. Temperatura de 70 °C. Frecuencia de 130 Hz.

4.1.2 Pruebas con el sistema modular

Las pruebas siguientes se han realizado con el sistema modular compuesto por la placa base y dos placas de control conectadas a dos membranas de un sensor. Las membranas son de resistencias iniciales de 550 y 650 ohmios.

En estos ensayos se pretende verificar el funcionamiento del sistema una vez implementado el sistema de módulos. Los dos primeros *slots* de la placa base son ocupados por las tarjetas de control. En el primer *spot* se sitúa un microcontrolador programado para realizar un control sobre una membrana de 650 Ω , y la segunda tarjeta lo hará sobre la de 550 Ω .

En estas pruebas se ha utilizado el osciloscopio digital TDS3012B y un amperímetro que controlará la intensidad del *heater*.

Las tensiones de referencia se ajustaran a diferentes valores para comprobar los márgenes de acción del sistema. También se puede observar la forma de la señal PWM que genera el microcontrolador.

Todas las señales de color azul oscuro pertenecen a la membrana de 650 Ω y las azul turquesa a la de 550 Ω . Las medias de cada canal que se representan en la pantalla corresponden a las gráficas del mismo color.

Para la primera medida se ajusta la tensión de referencia a 0 voltios

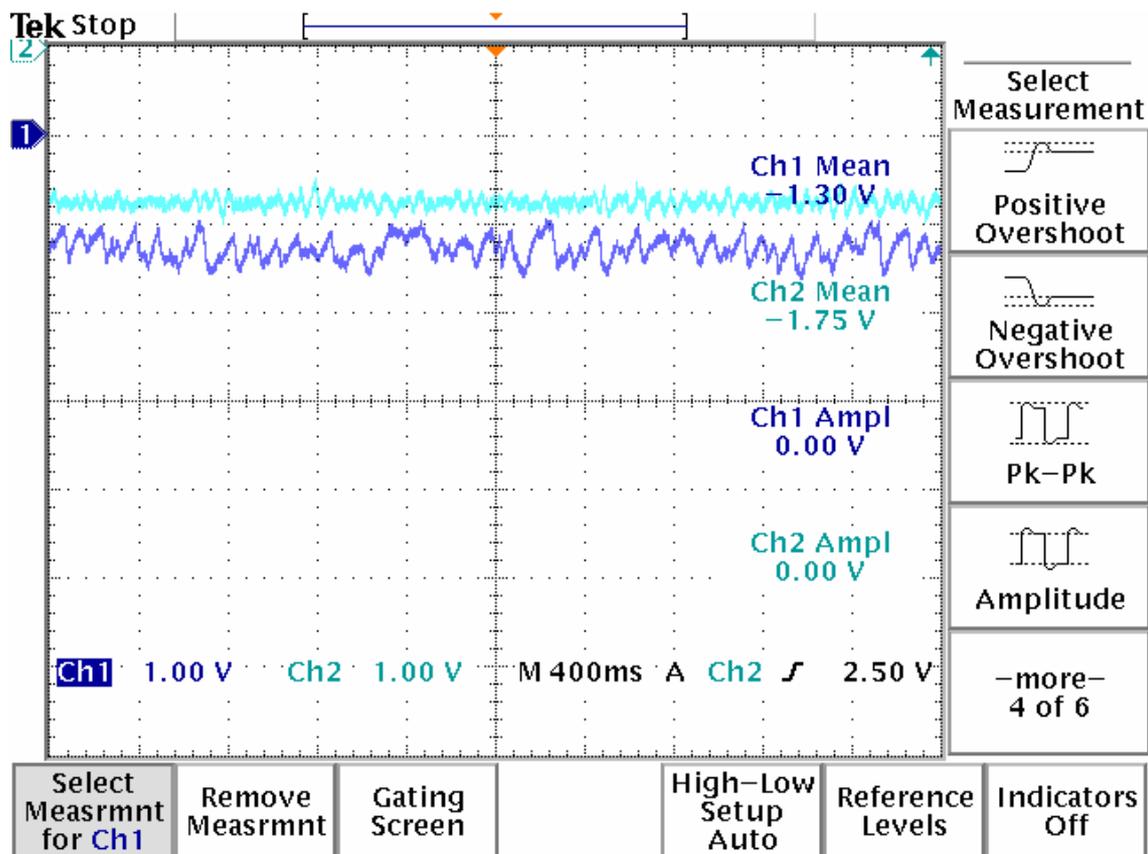


Figura 59. Tensión de los heaters para $V_{ref}=0$ V

La tensión media del heater de 550Ω es de $-1,75$ V, es la señal con menor rizado en la gráfica, y su intensidad es de $-3,01$ mA. Para la membrana de 650Ω la tensión es de $-1,30$ V y la intensidad de $-1,89$ mA.

Al final de estas gráficas se muestra una tabla con las temperaturas que alcanza el heater para las diferentes tensiones de referencia.

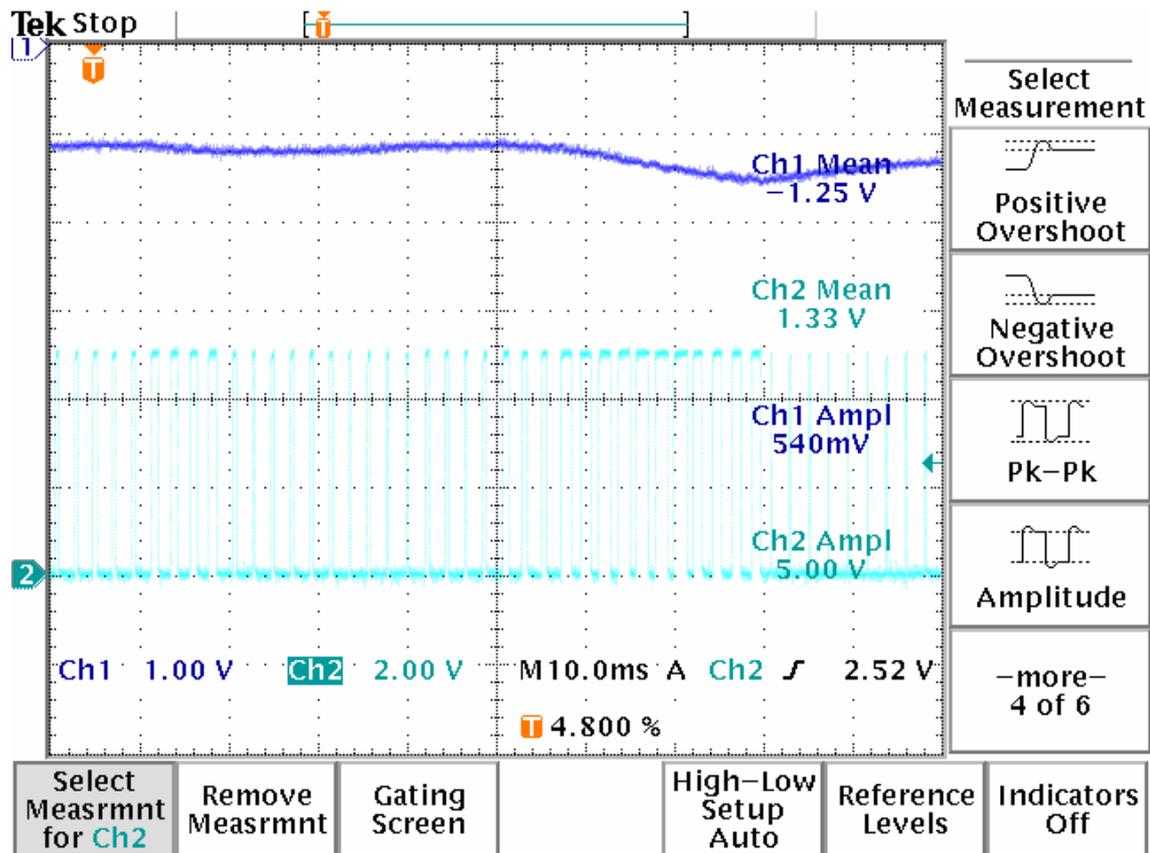


Figura 60. PWM y V_h para $V_{ref}=0$ V para el *heater* de 650Ω

Esta es la forma de PWM que genera el microcontrolador para realizar el control de la corriente que fluye por el *heater*.

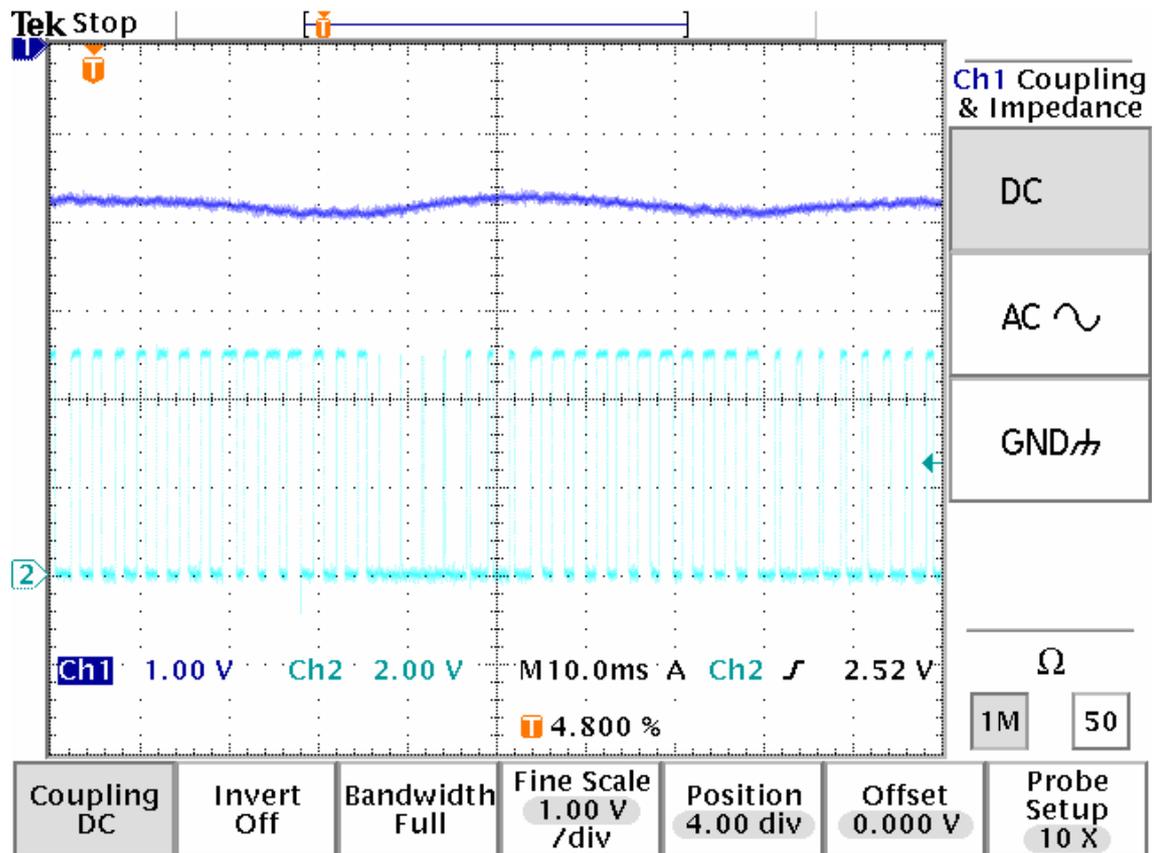


Figura 61. PWM y V_h para $V_{ref}=0$ V para el heater de 550Ω

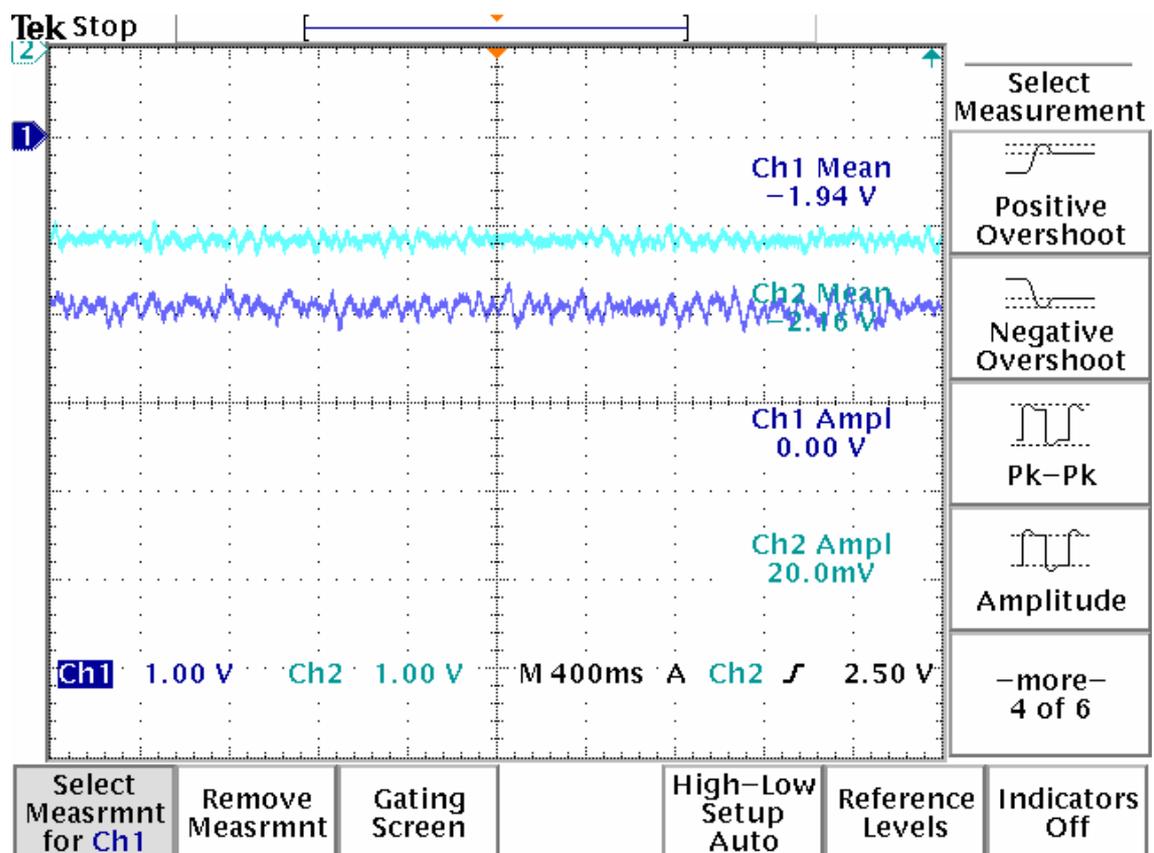


Figura 62. Tensión de los heaters para $V_{ref}=1$ V

Para la tensión de referencia de 1 V la tensión media en el *heater* de 550 Ω es de 2,16 V, representada en la parte superior, y su intensidad es de 3,65 mA. En el de 650 Ω cae una tensión de 1,94 V y pasa una intensidad de 2,77 mA.

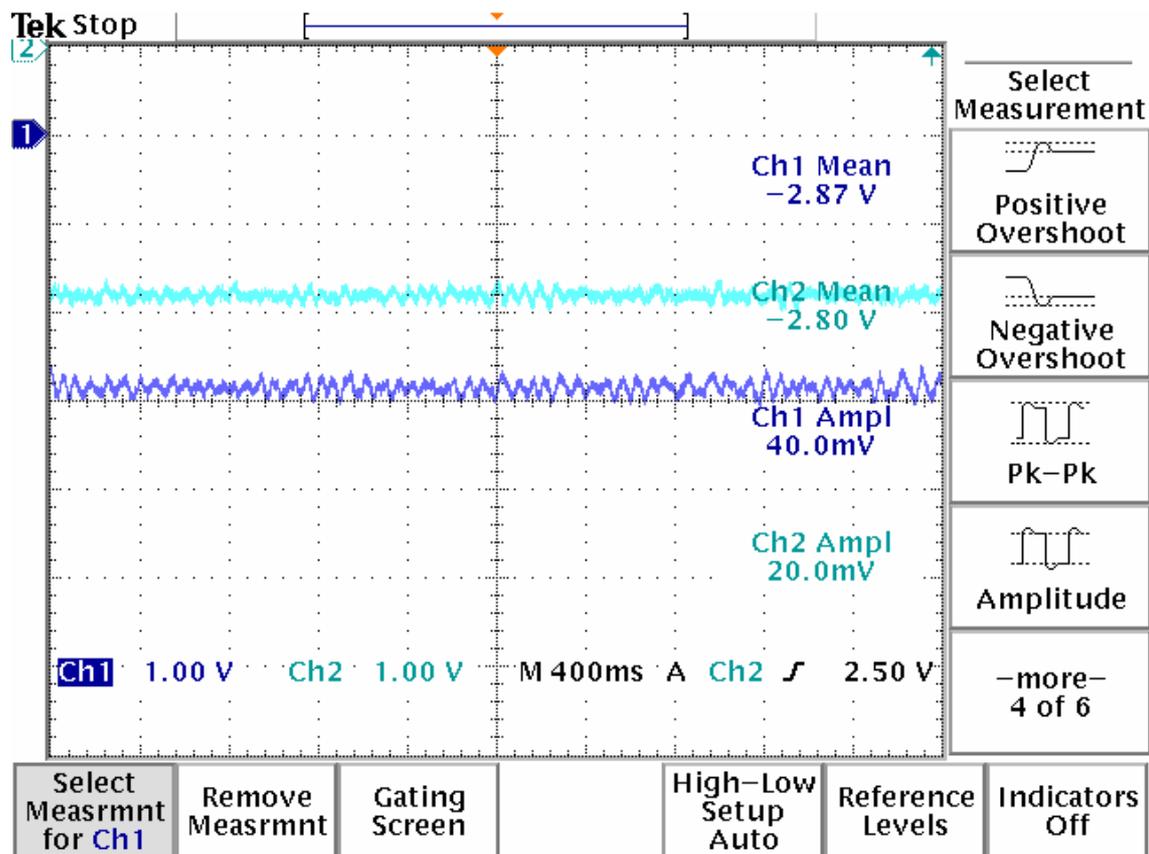


Figura 63. Tensión de los *heaters* para $V_{ref}=2$ V

La membrana de 550 Ω tiene en sus bornes 2,8 V y fluye por ella una corriente de 4,54 mA. Por la de 650 Ω pasan 3,96 mA y caen 2,87 V, se sitúa en la parte inferior de la gráfica.

A continuación se muestran las señales del PWM de los dos microcontroladores.

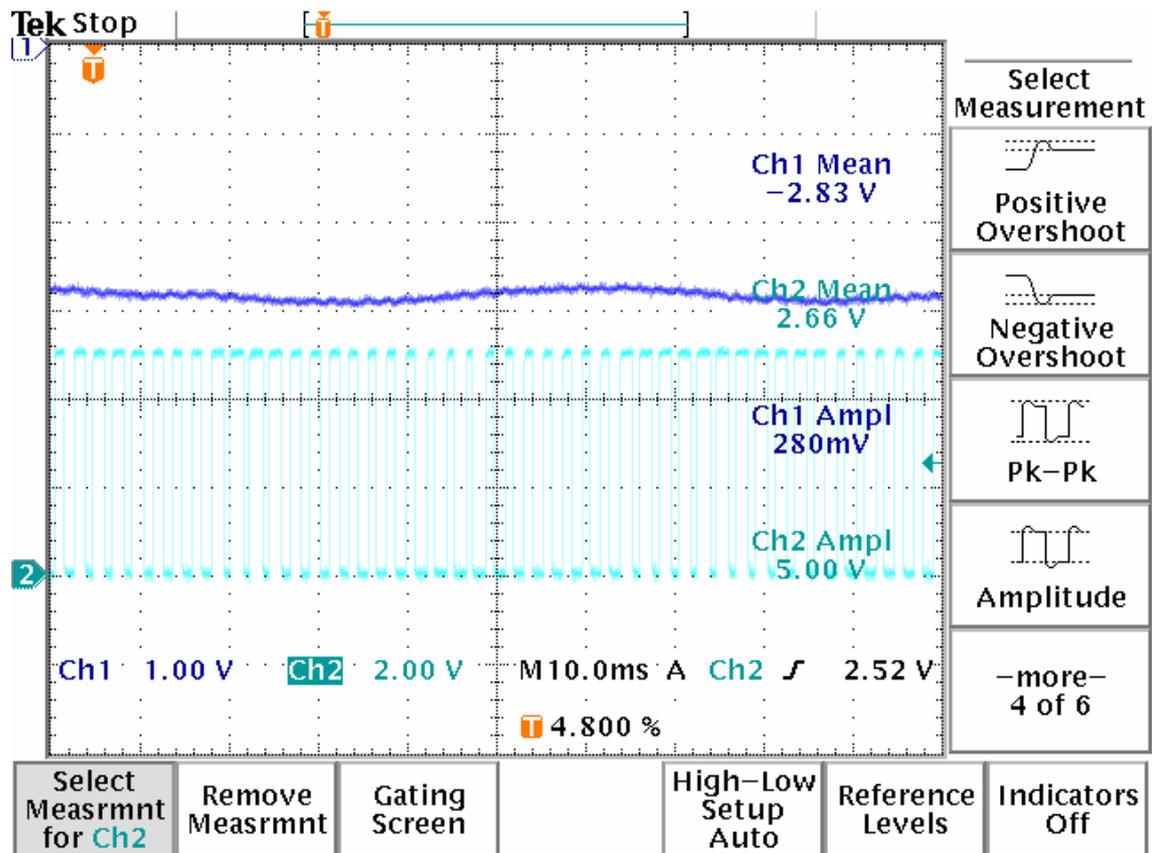


Figura 64. PWM y Vh para Vref=2 V para el heater de 650 Ω

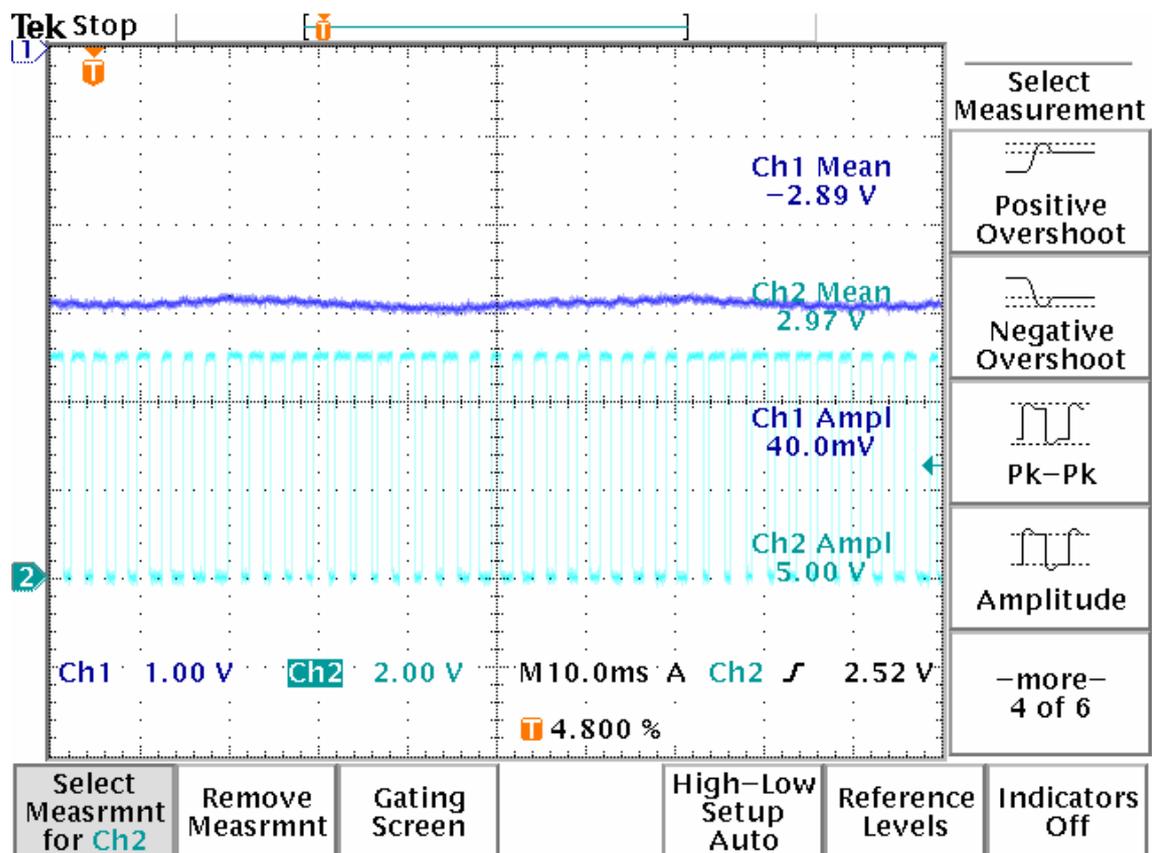


Figura 65. PWM y Vh para Vref=2 V para el heater de 550 Ω

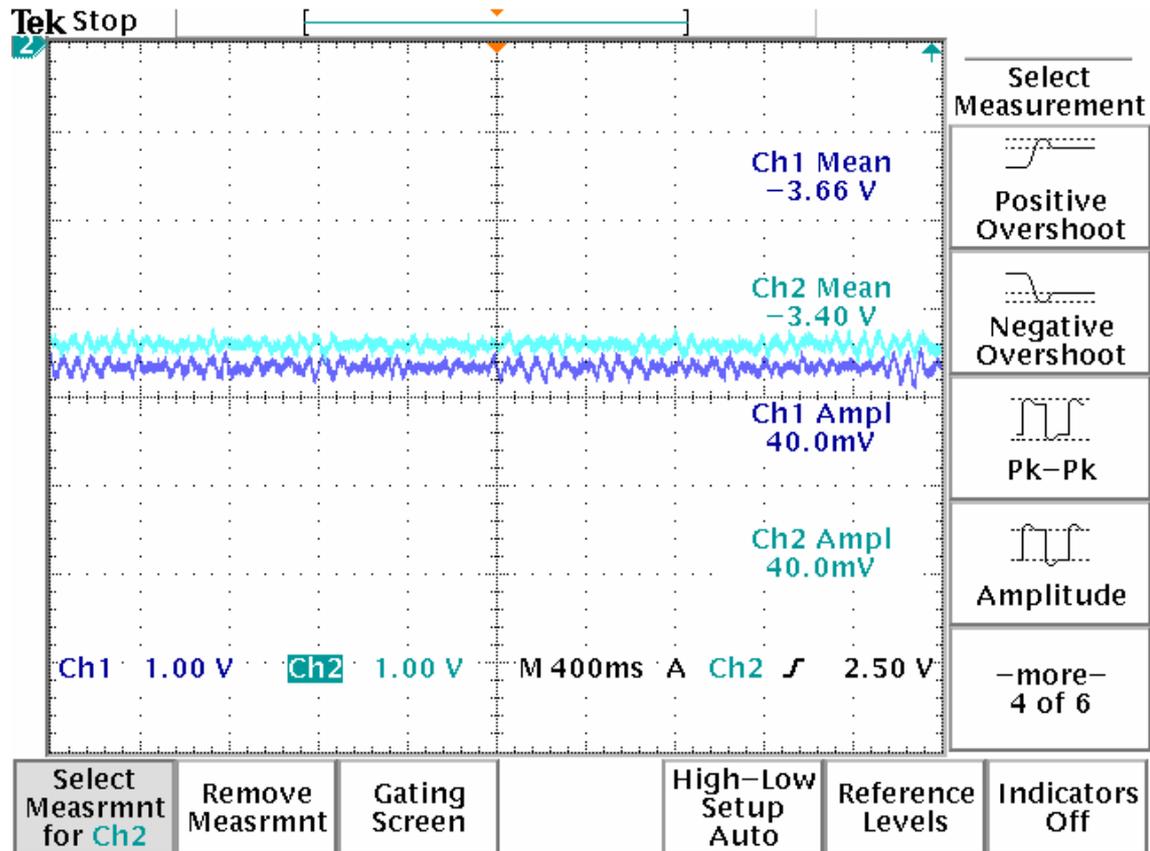


Figura 66. Tensión de los *heaters* para $V_{ref}=3$ V

Cuando la entrada de referencia es de 3 voltios, el voltaje en la membrana de 550Ω y su corriente son de 3,4 V y 5,35 mA respectivamente, es la señal superior. En la membrana de 650Ω son de 3,66 V y 4,88 mA.

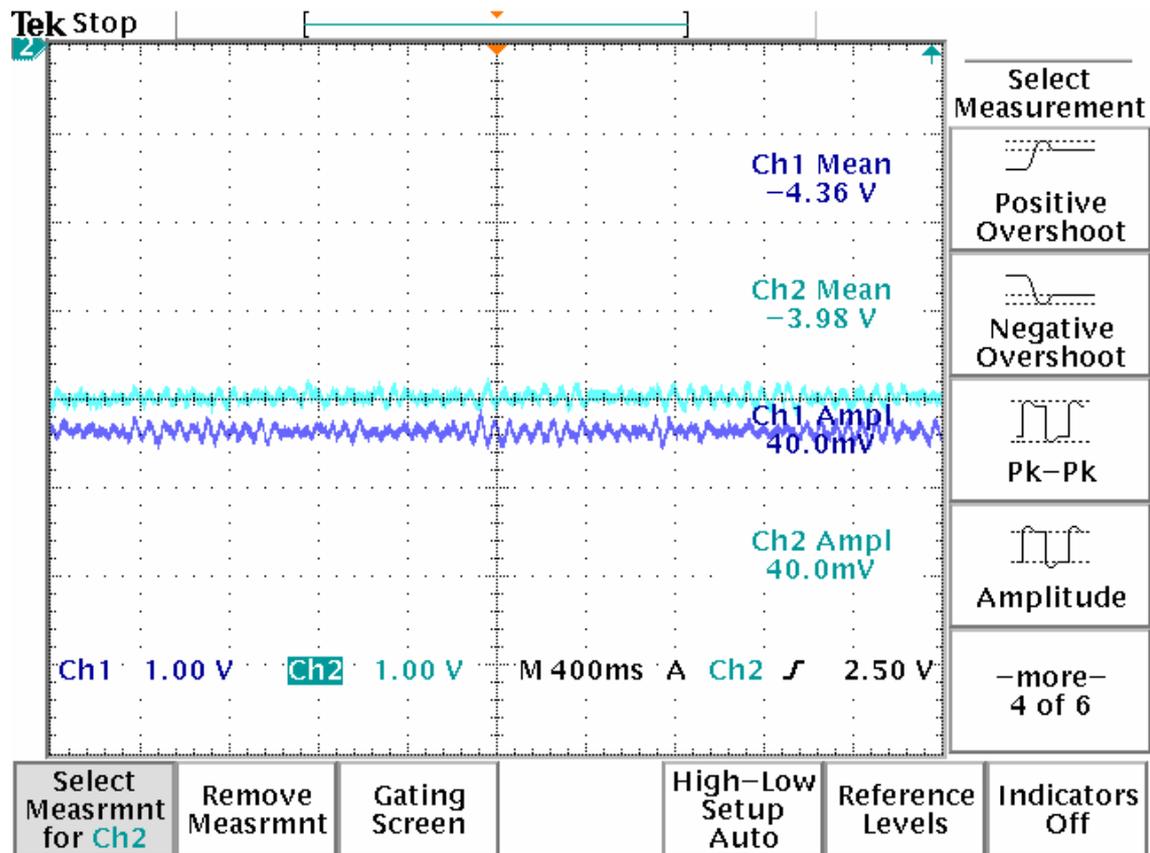


Figura 67. Tensión de los *heaters* para $V_{ref}=4$ V

Para una tensión de referencia de 4 V, se obtienen 3,98 voltios en el *heater* de 550 Ω y una intensidad de 6,03 mA, y 4,36 V y 5,62 mA para la membrana de 650 Ω . Esta última es la señal situada en la parte inferior.

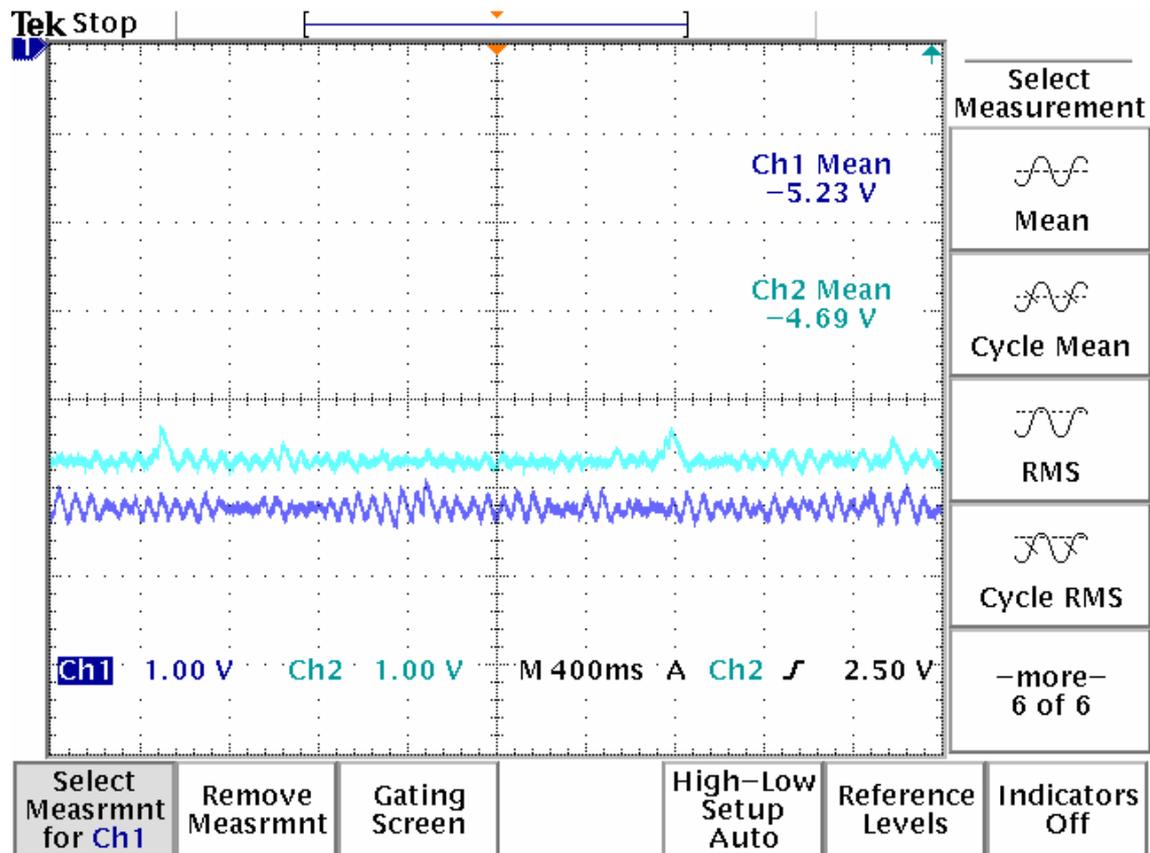


Figura 68. Tensión de los *heaters* para $V_{ref}=5,01$ V

Para el máximo valor en la entrada de referencia se obtienen los valores mayores en cuanto a tensión e intensidad. Para la membrana de 550Ω hay una tensión media de $4,69$ V y una intensidad de $6,78$ mA. Es la señal que se sitúa más arriba. Para la de 650Ω , una tensión de $5,23$ V y una corriente de $6,38$ mA.

La señal del PWM que produce esta entrada de referencia máxima es la siguiente para las dos membranas.

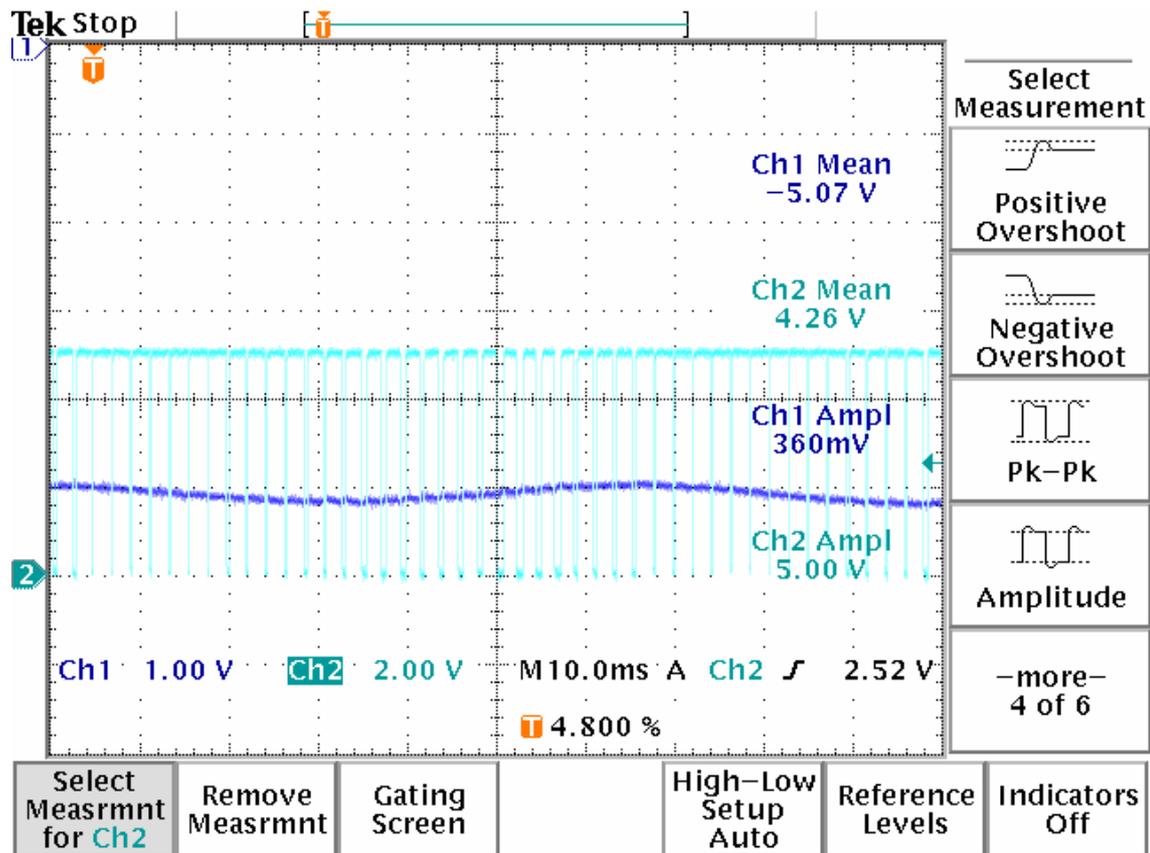


Figura 69. PWM y Vh para Vref=5,01 V para el heater de 650 Ω

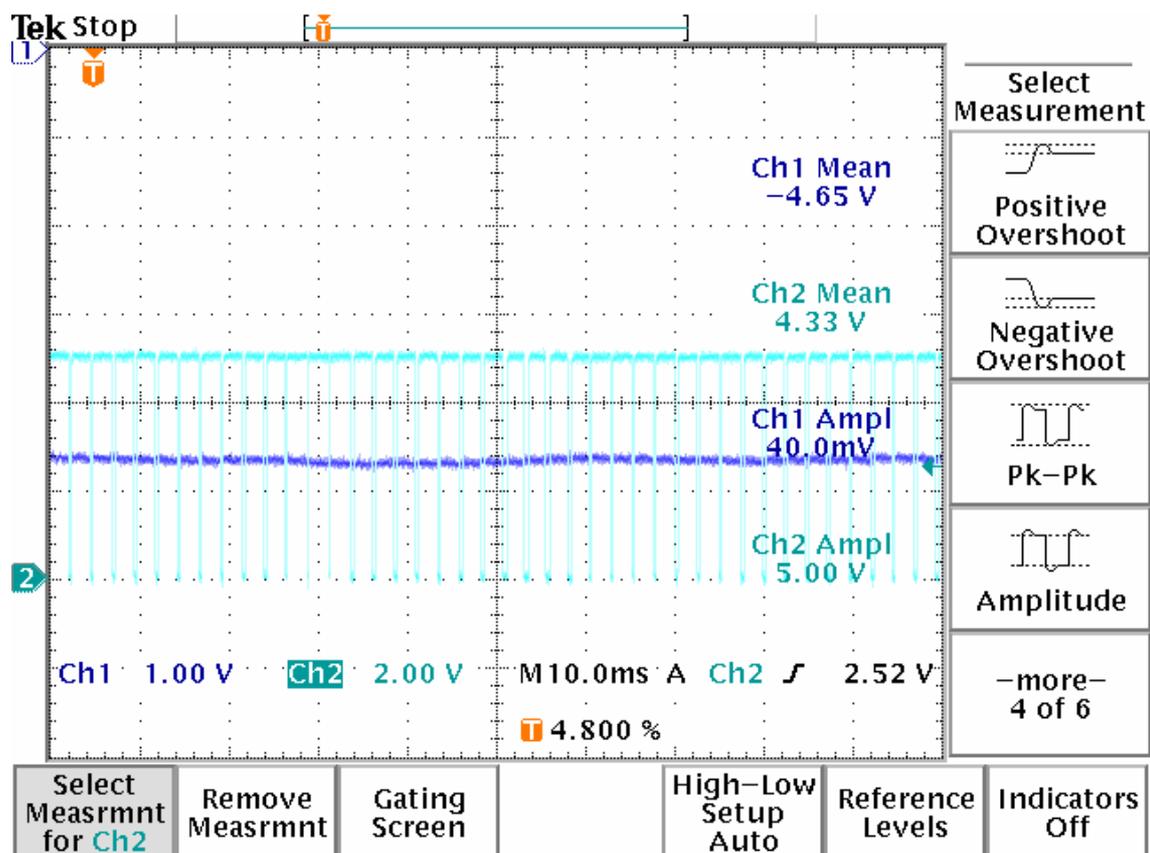


Figura 70. PWM y Vh para Vref=5,01 V para el heater de 550 Ω

Los datos de los ensayos realizados se resumen en esta tabla donde se pueden ver la resistencia del *heater* para las diferentes tensiones de referencia. La temperatura de la membrana está calculada a partir de la siguiente formula:

$$T = \frac{(R_h - R_o)}{\alpha \cdot R_o} + T_o$$

Donde, T es la temperatura del *heater*, R_h su resistencia, T_o y R_o su la temperatura ambiente inicial de 25 °C y su resistencia inicial de 550 Ω o 650 Ω . Y α es la constante de temperatura que tiene un valor de $6,79 \cdot 10E-4$.

Ro=550 Ω				
Vref (V)	Vh (V)	Ih (mA)	Rh (Ω)	T ^a Heater
0	-1,75	-3,01	581,4	109,1
1	-2,16	-3,65	591,8	136,9
2	-2,8	-4,54	616,7	203,7
3	-3,4	-5,35	635,5	254,0
4	-3,98	-6,03	660,0	319,6
5	-4,69	-6,78	691,7	404,5
Ro=650 Ω				
0	-1,3	-1,89	687,8	110,7
1	-1,94	-2,77	700,4	139,1
2	-2,87	-3,96	724,7	194,4
3	-3,66	-4,88	750,0	251,6
4	-4,36	-5,62	775,8	310,0
5	-5,23	-6,38	819,7	409,6

Tabla 10. Relación entre la tensión de referencia y la temperatura de los *heaters* de 550 Ω y 650 Ω .

En esta tabla se puede comprobar que las tensiones de referencia producen unas temperaturas de trabajo en los *heaters* similares, independientemente de cual sea su resistencia inicial. Esto asegura que con la misma temperatura de referencia podemos hacer trabajar a todas las membrana a la misma temperatura sin tener en cuenta el tipo de *heater*.

Las diferencia que se pueden apreciar entre los valores de una membrana y otra pueden ser debidos a que las resistencias iniciales no son exactamente de 550 Ω y de 650 Ω . Se ha comprobado en las distintas pruebas realizadas, que los valores de los *heaters*

pueden oscilar en unos $\pm 7 \Omega$. Estas oscilaciones, trasladadas a temperatura suponen variaciones que pueden alcanza los 18 °C. Por lo tanto las diferencias que se producen entre las temperaturas de los *heater* también se producirían entre *heater* del mismo tipo. Lo que interesa de nuestro sistema es que no asegure una temperatura constante a lo largo del tiempo de trabajo.

4.2 Pruebas con gases

A fin de verificar el sistema en una prueba en el entorno de trabajo real. Se han realizado unas pruebas con gases en el laboratorio de sensores de gases y nariz electrónica. Las medidas han sido realizadas con el HeadSpace Autosampler HP. Es un aparato que se utiliza para inyectar los volátiles de los viales para que puedan ser detectados por los sensores. Estos volátiles se mezclan con aire sintético, que será el gas portador. Un ordenador se utiliza para realizar la adquisición de los datos de la placa de sensores mediante la tarjeta de adquisición de datos PCL-812, y para comunicar con el Headspace y poder sincronizarse.



Figura 71. Headspace Autosampler HP conectado al sistema de control de temperatura.

Las medidas que se han realizado pretenden verificar si el control que se realiza sobre la resistencia del *heater* es correcto cuando se utilizan en medidas con gases. Para ello se ha realizado dos series de medidas en condiciones de temperatura diferentes. La primera serie se ha realizado a una temperatura ambiente de 28 °C, y la segunda serie a 21 °C.

Para cada una de las series se han utilizado 3 viales diferentes; amoníaco, etanol y acetona. Se han realizado tres medidas no consecutivas para cada uno de los viales. De forma que se ha adquirido un total de 9 muestras por cada una de las dos series que se han realizado.

A continuación se muestran para cada tipo de vial, una gráfica que representa el valor de la resistencia del *heater* a lo largo de toda la medida. Los valores de resistencia que se aprecian en las gráficas tienen unos valores que no son correctos. Su escala está en ohmios pero su valor está equivocado debido a que el programa de lectura de datos todavía no está adaptado al nuevo sistema de control. Este error se produce al realizar la lectura de la tensión del *heater* y tratarla como una tensión positiva como sucedía con el antiguo sistema. Para el sistema de control de

temperatura que ha sido diseñado, los voltajes que se producen en el *heater* son negativos, esto produce que al realizar la conversión, no se tenga en cuenta el *bit* de signo, y se produzcan valores con un *offset* respecto el valor real.

A pesar de esto si que se puede apreciar claramente que los valores de la resistencia del *heater* se mantienen constantes en toda la medida.

Aquí se puede ver la medida realizada con una temperatura ambiente de 28 °C, con el vial de amoníaco.

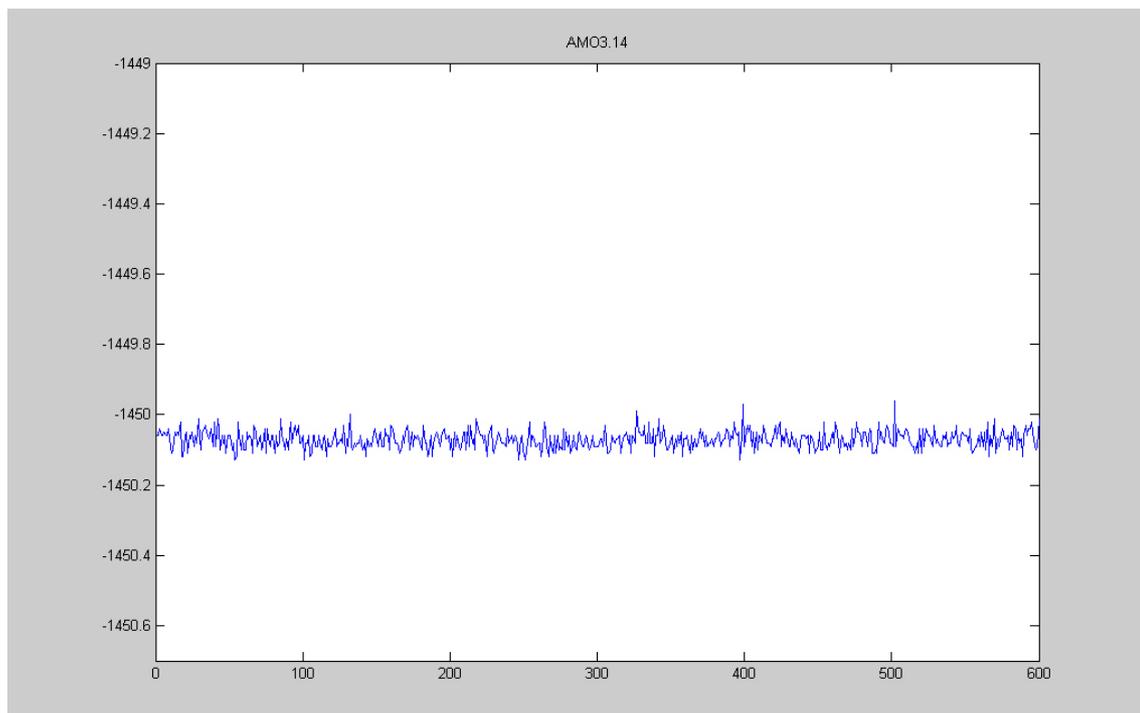


Figura 72. Temperatura de 28 °C, muestra de amoníaco

La muestra de la figura 73 está realizada a una temperatura 7 grados más baja que la anterior, y con el mismo vial.

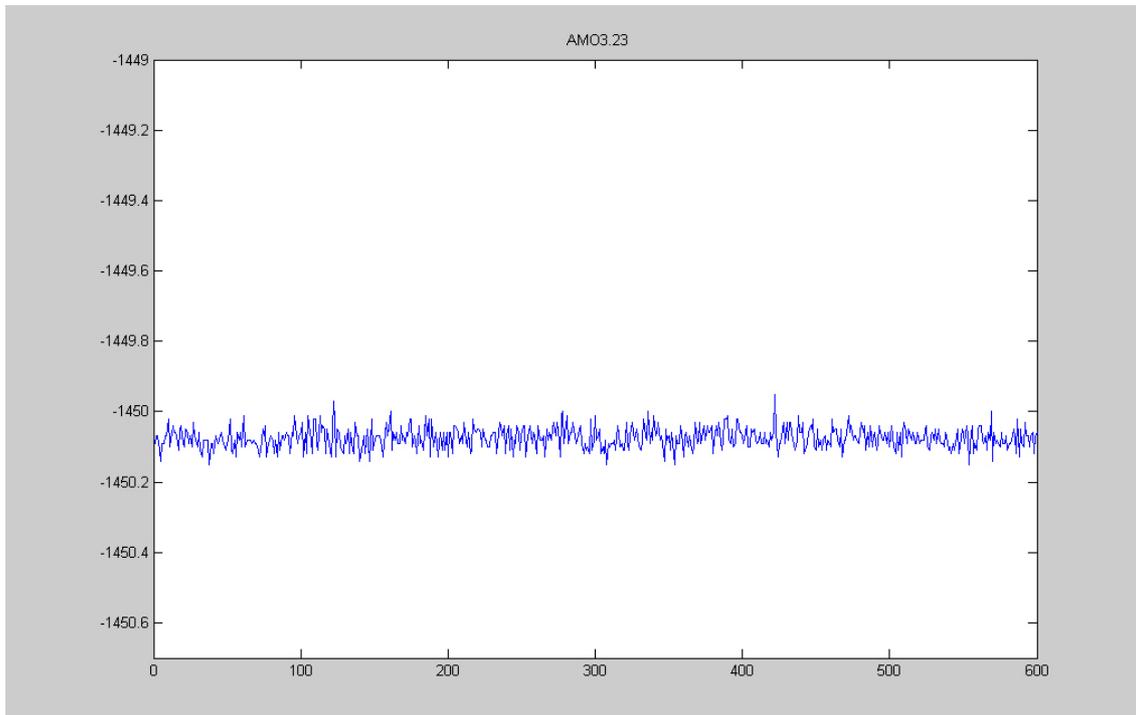


Figura 73. Temperatura de 21 °C, muestra de amoniaco

Se puede apreciar como en las dos muestras de amoniaco la resistencia del *heater* tiene prácticamente la misma temperatura. En las gráficas se aprecian unas oscilaciones en la resistencia. Estas son debidas al control que se realiza sobre ella. Si se observan los valores de resistencia entre los que oscila, se podrá comprobar que el control es realmente preciso, ya que la precisión de la resistencia es de $\pm 100 \text{ m}\Omega$ aproximadamente.

Las oscilaciones de la resistencia del *heater* dentro del rango de $\pm 100 \text{ m}\Omega$ se traducen en una variación de la temperatura de este. Para hallar el margen de oscilación de la temperatura del sensor se utiliza la siguiente formula.

$$R = R_o \cdot (1 + 6,79 \cdot 10^{-4} \cdot (T - T_o))$$

Si $R_o = 1450 \Omega$, $R = 1450,2 \Omega$

$$\Delta T = T - T_o$$

$$R = R_o + R_o \cdot 6,79 \cdot 10^{-4} \cdot (\Delta T)$$

$$\Delta R = R_o \cdot 6,79 \cdot 10^{-4} \cdot (\Delta T)$$

$$\frac{\Delta R}{R_o \cdot 6,79 \cdot 10^{-4}} = \Delta T$$

$$\Delta T = \frac{\Delta R}{R_o \cdot 6,79 \cdot 10^{-4}} = \frac{1450,2 - 1450}{1540 \cdot 6,79 \cdot 10^{-4}} = 0,203 \text{ } ^\circ\text{C}$$

Por lo tanto una variación de 200 mΩ en el valor de la resistencia del *heater* se traduce en una oscilación de 0,2 °C en la temperatura del mismo.

A continuación se muestran las graficas correspondientes a las medidas con acetona, a temperaturas de 28 °C y de 21 °C.

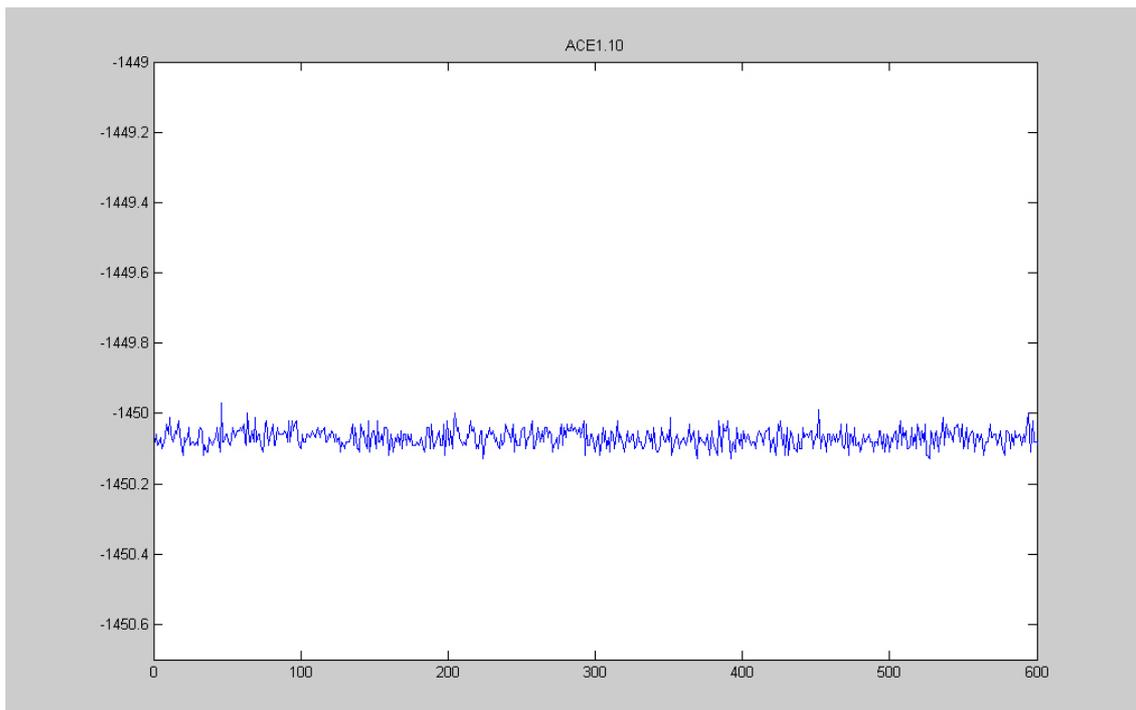


Figura 74. Temperatura de 28 °C, muestra de acetona

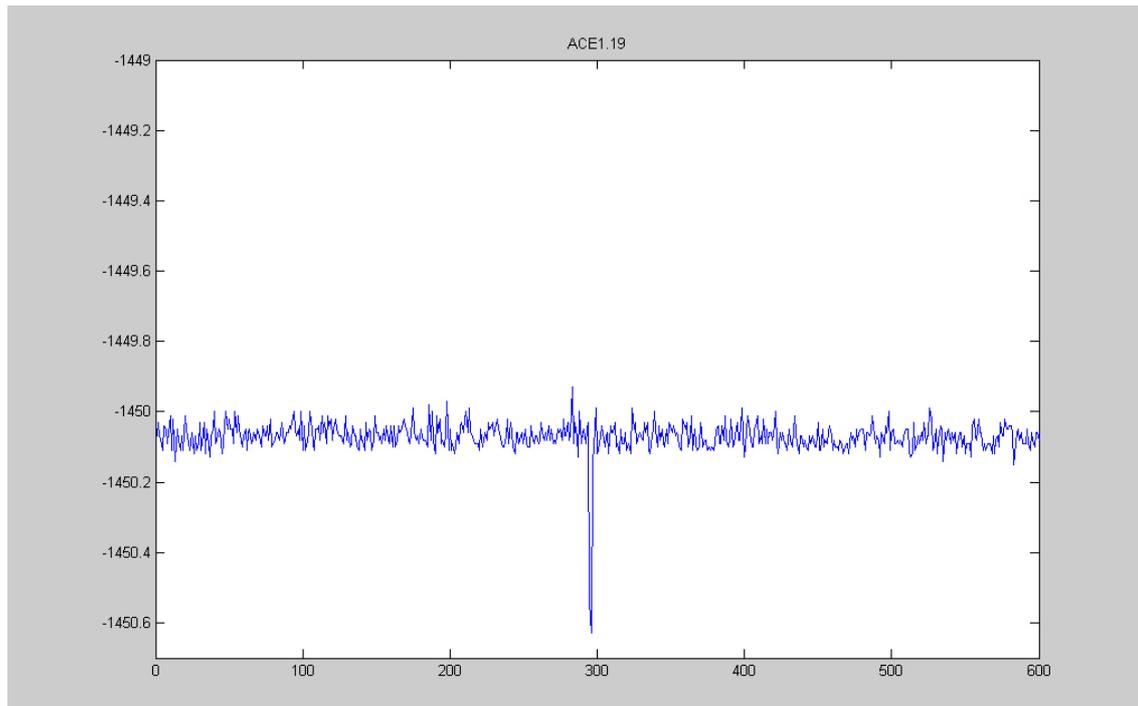


Figura 75. Temperatura de 21 °C, muestra de acetona

En esta última grafica se aprecia un pico negativo. Este pico puede ser debido a una interferencia de las fuentes de alimentación o de algún aparato de medida de la sala. Aunque en esta grafica parece un pico a tener en cuenta, son tiene una variación de 600 mΩ.

Las siguientes dos muestras se han realizado con etanol. Estas presentan una resistencia del *heater* estable y con una media igual a la de las graficas anteriores.

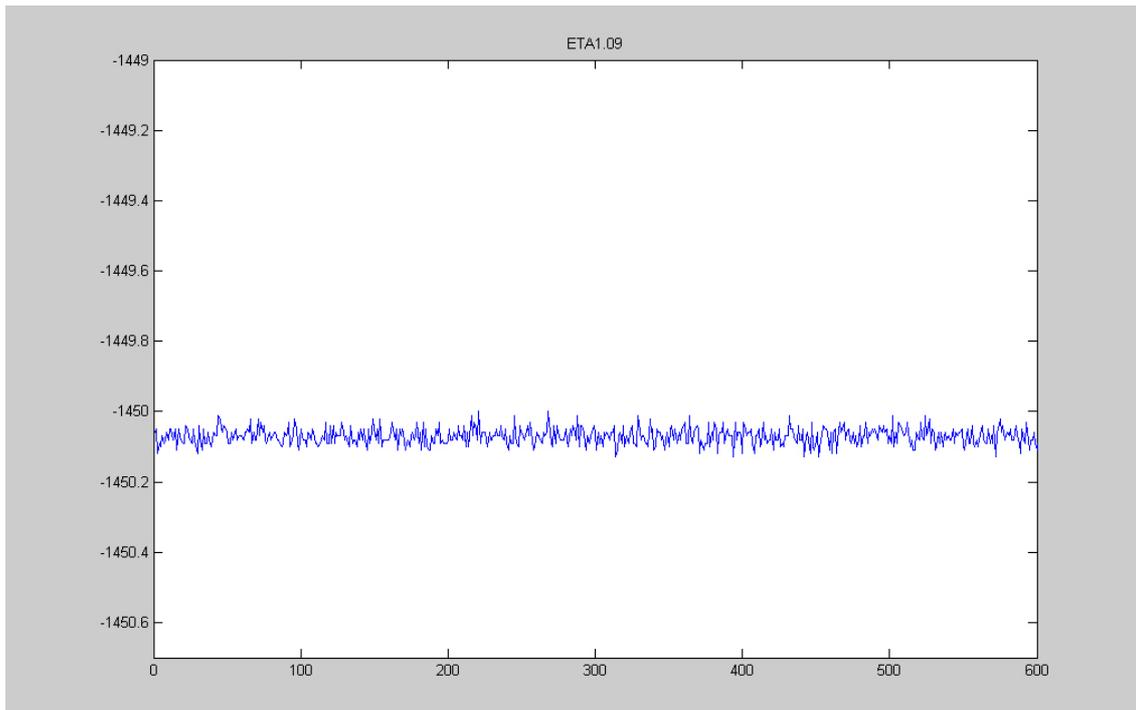


Figura 76. Temperatura de 28 °C, muestra de etanol

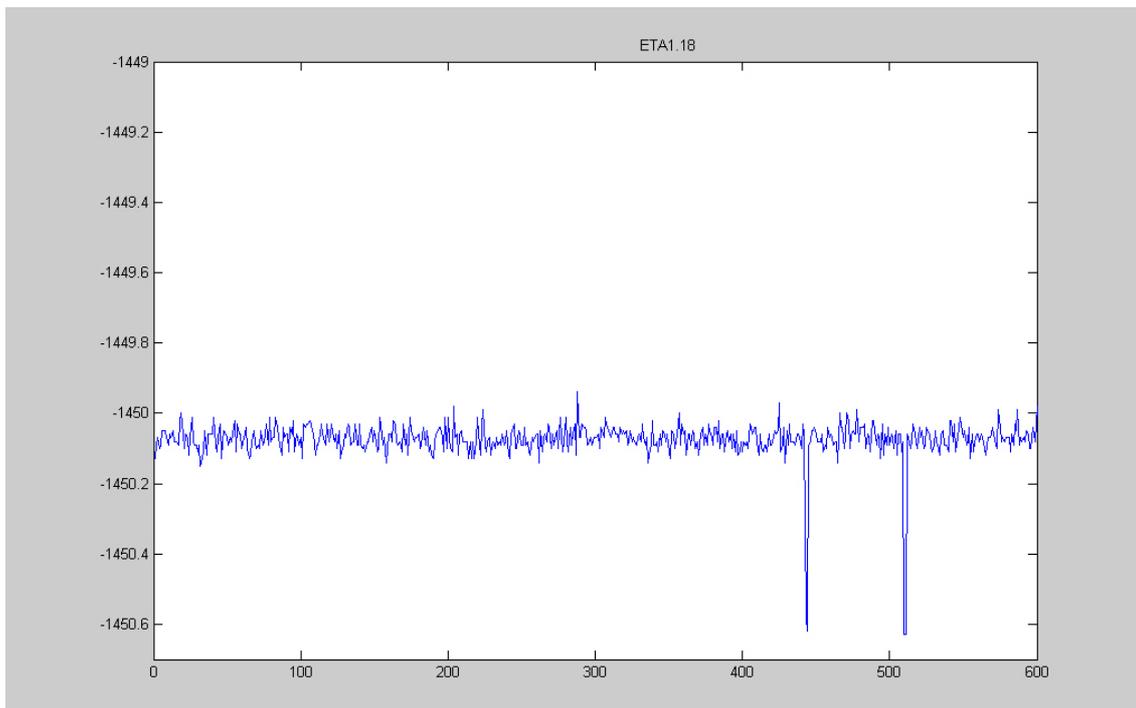


Figura 77. Temperatura de 21 °C, muestra de etanol

Tras revisar los contenidos de las graficas anteriores, se puede confirmar que el control de temperatura es bueno, debido a que la resistencia del *heater* se mantiene estable.

Esto repercutirá en la sensibilidad que tendrá el *heater* cuando se realicen medidas con gases, ya que esta sensibilidad no se verá afectada por las fluctuaciones de la temperatura del entorno.

5. PRESUPUESTO

5.1 Cuadro de precios

5.1.1 Material

Código	Elemento	Cantidad	Tamaño	Precio unitario (€)
01-001	Resistencias 1% tol. 10 Ω	1		0,03
01-002	Resistencias 1% tol. 180 Ω	1		0,03
01-003	Resistencias 1% tol. 680 Ω	1		0,03
01-004	Resistencias 1% tol. 1 K Ω	1		0,03
01-005	Resistencias 1% tol. 7K4 Ω	1		0,03
01-006	Resistencias 1% tol. 10 K Ω	1		0,03
01-007	Resistencias 1% tol. 18 K Ω	1		0,03
01-008	Resistencias 1% tol. 68 K Ω	1		0,03
01-009	LM324	1		0,43
01-010	PIC12F675	1		2,51
01-011	Condensador 10 μ F	1		0,06
01-012	Condensador 10nF	1		0,6
01-013	Condensador 330nF	1		0,81
01-014	Microinterruptores 4 vías	1		2,7
01-015	Cuadruple puerta AND 74HC08N	1		0,22
01-016	Comparador lógico 74HC85N	1		0,46

01-017	Conector de borde para PCB, 22 vías	1		5,11
01-018	LM741CP	1		0,36
01-019	Regulador de tensión L7805CV	1		0,51
01-020	Conector macho cable plano 20 polos	1		3,68
01-021	Conector hembra cable plano 20 polos	1		2,31
01-022	Conectores bipolar	1		0,45
01-023	Zocalo 8 pins	1		0,25
01-024	Zócalo 14 pins	1		0,27
01-025	Zócalo 16 pins	1		0,32
01-026	Placa PCB	1	600x900	144,02
01-027	Cable alimentación 3x0,75	1	1 m	0,73
01-028	Cable plano 20 polos tipo ICD	1	1 m	2,31
01-029	Clavija tipo banana de lámina elástica	1		1,19
01-030	Estaño de núcleo único	1	250 g	6,5

5.1.2 Diseño y montaje

Código	Descripción	Tamaño / Peso	Precio (€ / hora)
02-001	Honorarios técnico	1	20

5.2 Aplicación de precios

5.2.1 Placa base

Código	Elemento	Unidades	Tamaño	Precio unitario (€)	Subtotal (€)
01-006	Resistencias 1% tol. 10 KΩ	2		0,03	0,06
01-011	Condensador 10μF	2		0,06	0,12
01-012	Condensador 10nF	1		0,6	0,6
01-013	Condensador 330nF	1		0,81	0,81
01-020	Conector de borde para PCB, 22 vías	8		5,11	40,88
01-018	LM741CP	1		0,36	0,36
01-019	Regulador de tensión L7805CV	1		0,51	0,51
01-020	Conector macho cable plano 20 polos	1		3,68	3,68
01-021	Conector hembra cable plano 20 polos	2		2,31	4,62
01-022	Conectores bipolar	7		0,45	3,15
01-023	Zocalo 8 pins	1		0,25	0,25
01-026	Placa PCB	1	110x125 mm2	144,02	3,67
01-027	Cable alimentación 3x0,75	1	1 m	0,73	0,73
01-028	Cable plano 20 polos tipo ICD	1	0,4 m	2,31	0,92
01-029	Clavija tipo banana de lámina elástica	3		1,19	3,57
01-030	Estaño de núcleo único	1	125 g	6,5	3,25
				Total placa base	67,18

El capítulo de la placa base asciende a sesenta y siete euros con dieciocho céntimos

5.2.2 Placa de control

Código	Elemento	Unidades	Tamaño / Peso	Precio unitario (€)	Subtotal (€)
01-001	Resistencias 1% tol. 10 Ω	1		0,03	0,03
01-002	Resistencias 1% tol. 180 Ω	1		0,03	0,03
01-003	Resistencias 1% tol. 680 Ω	1		0,03	0,03
01-004	Resistencias 1% tol. 1 K Ω	6		0,03	0,18
01-005	Resistencias 1% tol. 7K4 Ω	1		0,03	0,03
01-006	Resistencias 1% tol. 10 K Ω	3		0,03	0,09
01-007	Resistencias 1% tol. 18 K Ω	1		0,03	0,03
01-008	Resistencias 1% tol. 68 K Ω	3		0,03	0,09
01-009	LM324	2		0,43	0,86
01-010	PIC12F675	1		2,51	2,51
01-011	Condensador 10 μ F	2		0,06	0,12
01-014	Microinterruptores 4 vías	1		2,7	2,7
01-015	Cuadruple puerta AND 74HC08N	1		0,22	0,22
01-016	Comparador lógico 74HC85N	1		0,46	0,46
01-023	Zocalo 8 <i>pins</i>	1		0,25	0,25
01-024	Zócalo 14 <i>pins</i>	3		0,27	0,81
01-025	Zócalo 16 <i>pins</i>	1		0,32	0,32
01-026	Placa PCB	1	265x120	144,02	8,48
01-030	Estaño de núcleo único	1	30 g	6,5	0,78

		Total placa de control	18,02
--	--	------------------------	-------

El capítulo de la placa de control asciende a dieciocho euros con dos céntimos.

5.2.3 Diseño y montaje

Código	Descripción	Horas	Precio / hora	Precio (€ / hora)
02-001	Honorarios técnico	90	20	1800
		Total diseño y montaje		1800

El capítulo de diseño y montaje asciende a mil ochocientos euros.

5.3 Resumen presupuesto

Gasto total de la placa base	67,18 €
Gasto total de las 8 placas de control	144,16 €
	+
Presupuesto de ejecución material	211,34 €
Gastos generales (13%)	27,47 €
Gasto total de diseño y montaje del sistema	1800.00 €
	+
Presupuesto de ejecución de por contrato	2038,82 €
I.V.A. (16%)	326,21 €
	+
TOTAL PRESUPUESTO GENERAL	2365,03 €

El presupuesto general asciende a la cantidad de **DOS MIL TRESCIENTOS SESENTA Y CINCO EUROS CON TRES CÉNTIMOS.**

El ingeniero técnico industrial

Jordi Colomer Campos

6. EVALUACIÓN DEL SISTEMA

6.1 Mantenimiento y ajustes

El mantenimiento del sistema es prácticamente nulo. Una vez se instala en su lugar de funcionamiento, no se modificará su configuración si no se desea otro tipo de funcionamiento.

Estos tipos de funcionamiento se basan en la inicialización de las tarjetas. La configuración de cada una de ellas permitirá su inicialización por individual, por grupos o única para todas las tarjetas.

Esta es la única posibilidad de configuración que permite este sistema al usuario. Si se desea utilizar el sistema par sensores con distintas resistencias iniciales que para las que está diseñado el dispositivo, se debería cambiar algún parámetro del programa de control del microcontrolador.

Los *heater* para los que están programados los microcontroladores son 550 Ω y 650 Ω . Podría utilizarse para otros sensores, siempre que el sistema pueda adaptarse a sus resistencias ambiente y a sus coeficientes de temperatura. Para las resistencias del *heater* que tengan un coeficiente de temperatura de $6,79 \cdot 10^{-4} \text{ C}^{-1}$ se podría mantener el rango de temperaturas que se tiene con este circuito de control. Pero si su coeficiente de temperatura variase, posiblemente el rango que se podría alcanzar no sería el mismo.

Para hacer un ajuste más preciso del control de temperatura a unos sensores concretos se deben realizar variaciones en el programa. Los sensores tienen una resistencia de *heater* definida para una temperatura ambiente de unos 25 $^{\circ}\text{C}$. Esta resistencia no siempre es la que se refleja en las características sino varía según el sensor. Para realizar este ajuste preciso se puede medir la temperatura del entorno y la resistencia del *heater* en el mismo momento, y ajustar el programa de forma que se adapte a este sensor.

En el programa del microcontrolador solo hay que variar una línea del código. Es la línea que define la resistencia que debe alcanzar el sensor para conseguir la temperatura deseada. Es la siguiente para el caso de un *heater* de 650 Ω :

```
resist=((valor1)*(650*0.000679)+650);
```

En ella vemos la ecuación que relaciona la temperatura con la resistencia mediante el coeficiente de temperatura, en este caso de $6,79 \cdot 10^{-4} \text{ C}^{-1}$.

IMPORTANTE: Si se desea adaptar el sistema a un sensor específico podrá hacerse de la forma anteriormente explicada siempre que el coeficiente de temperatura del *heater* sea de $6,79 \cdot 10^{-4} \text{ }^{\circ}\text{C}^{-1}$. Si este coeficiente varía se deberá revisar el programa para comprobar el rango de temperatura que se pueden alcanzar.

6.2 Aplicaciones

El control de temperatura para los sensores se ha realizado con el fin de mejorar las medidas de estos. Por ello este sistema se puede utilizar en cualquiera de las aplicaciones en que se utilicen estos sensores. Debido a que es un sistema que puede trabajar a larga distancia del sensor, ya que realiza una excitación por corriente, puede utilizarse para todas las aplicaciones para las que esté preparado el sensor.

7. CONCLUSIÓN

Los sistemas de control de temperatura requieren un actuador y una realimentación, estos permiten realizar cambios en la temperatura y conocer su valor. En muchos casos, se utilizan sensores para conocer la temperatura a controlar y actuadores que adapten la temperatura a la requerida. Se utilizan de forma separada y son independientes.

Los sensores de gases disponen de un único elemento que deberá realizar la función de actuador y sensor para la realimentación. Estos tienen un comportamiento mejor cuando su temperatura de trabajo es alta. Un sistema que controle su temperatura puede permitir que realice medidas de una forma más fiable.

Este tipo de control que tiene la dificultad de que actuador y sensor son el mismo elemento, requiere un complejo sistema de control. Para conocer la temperatura hay que basarse en las propias características del *heater*. Estas relacionan la resistencia con la temperatura de forma que se puede hablar de un control de la resistencia de *heater*.

Para mantener constante la resistencia de un elemento hay que realizar un control de su tensión y su intensidad. Será el cociente de estas quien determine el valor de su resistencia. La experiencia en la realización de este proyecto demuestra que solo la implementación de un sistema que permita actuar con ambos parámetros hará posible el control de la resistencia.

La realización analógica del sistema de control tiene una solución compleja debido a que se requiere la división de dos señales. Para un sistema que no requiere velocidades elevadas sería innecesario este tipo de implementación.

El microcontrolador demuestra una vez más su adecuación a cualquier sistema. Su forma de programación con el nuevo sistema de compilación en "C", y las prestaciones del propio microcontrolador hacen su fácil utilización.

En esta aplicación se ha utilizado la conversión A/D, se han adaptado las señales analógicas para obtener mejores resultados, flexibilizar el sistema y permitir su uso en otros tipos de sensores.

El PWM es una herramienta que se ha utilizado para las conversiones D/A. Es de uso rápido y sencillo. Es un sistema fiable aunque requiera de una circuitería externa.

La comunicación con el ordenador ha sido implementada para realizar la inicialización se distancia. De esta forma se puede ajustar de forma más precisa y visual las temperaturas de trabajo y evitar el contacto con las placas de control.

Los sensores de gases evitarán que las medidas que realicen se vean afectadas por las variaciones de la temperatura ambiente. Esto permite una mayor fiabilidad del sistema de medida. También pueden realizarse diferentes tipos de medidas en un mismo sensor. Esto se debe a que se ha realizado un sistema modular que permite seleccionar temperaturas de trabajo diferentes en cada membrana, hasta un total de 8 membranas.

Para la realización práctica de un control de temperatura se debe conocer exactamente el funcionamiento de aquello que se desea controlar. La interpretación de los resultados juega un papel muy importante cuando se trata de controlar una resistencia. Los valores que se obtienen pueden tener errores provocados por los equipos de medida o por interferencias. Estos errores solo pueden ser detectados si se tiene un buen conocimiento del sistema.

ANEXO 1

1. Programa en lenguaje C

```

////////////////////////////////////
//
// Programa de control de temperatura para un sensor de gases //
//
// Proyecto final de carrera: Sistema para el control en lazo //
// cerrado de la temperatura de trabajo de una agrupación //
// integrada de microsensores de gases //
//
// Programado por Jordi Colomer Campos //
//
// Escuela Técnica Superior de Ingeniería //
// Titulación: Ingeniería Técnica Industrial //
// Especialidad: Electrónica Industrial //
//
////////////////////////////////////

#include "pic12f6x.h"
#include "conio.h"
#include "stdio.h"
#include "dos.h"
#include "stdlib.h"
#include "intrpt.h"
#include "math.h"

#define canalzero 0
#define canaluno 1
#define canaldos 2
#define ON 1
#define OFF 0

void interrupt interrupcions(void);
void inicialitzacions(void);
void leer(char canal);
void uno(void);
void zero(void);

// Definición de las variables globales

unsigned int conv0=1,conv1=1,conv2=0,dc=1;
unsigned char timer_uno=1,nivell=0,dc1=1;

void main(void)
{

float ih;
float vh;
float rmedida, resist;
unsigned int valor1;

GPIO5=ON;

```

```

nivell=0;
inicialitzacions();

leer(canalzero);

while(GPIO4== 0);

    leer(canalzero);

    valor1=(unsigned int)(conv0*0.396754);
    resist=((valor1)*(650*0.000679)+650);

        //Hallamos la temperatura de consigna con
        //el valor convertido a digital de ADC0
        //"valor" puede tomar un máximo de
        //1023*0.396754=405 °C de temperatura máxima
        //Calculamos ahora el valor de la resistencia
        //de consigna mediante la formula que
        //dependerá del tipo de heater, en este caso
        //la resistencia del heater es de 650 Ω

while(1)                //Bucle infinito en el que se realiza el
                        //control de temperatura

{

if(timer_uno)           //Cada 7,7 ms se activa timer_uno, y se ejecuta
                        //el código que controla la variable duty cycle
    {
    leer(canaluno);      //Lectura de el valor referente a Vh
    leer(canaldos);     //Lectura de el valor referente a Ih

    timer_uno=0;

                        //Cálculo de la intensidad Ih con una magnitud
                        //1000 veces mayor para conseguir una mayor
                        //precisión. Ih=conv1*(1000*5)/(68*1023) Donde
                        //68 es la ganancia del amplificador inversor
                        //que ajusta el voltaje de la R de 10Ω al micro
                        //La Ih máxima será de 7,35 mA

    ih=conv2*0.007187625;

                        //Cálculo del voltaje del heater 1000 veces
                        //mayor. Vh=conv2*(1000*5)/(0,74*1023)
                        //El voltaje máximo será de 6,75 V

    vh=(conv1*6.47); //4.8876);

    rmedida=(vh/ih); //Cálculo de la resistencia del heater

                        //Aquí se compara la resistencia de referencia
                        //y la de consigna para que no producir un duty
                        //cycle negativo a la hora de restar las dos
                        //Se añade una constante proporcional para
                        //aumentar la sensibilidad

    if(rmedida<resist)
        dc=(unsigned int)(10*(resist-rmedida));
    else

```

```

        dc=dc/2;           //Si la resistencia medida supera la de
                           //referencia en lugar de producirse un duty
                           //cycle negativo se reduce este a la mitad
                           //En esta parte es donde se modifica la
                           //variable "dc1" que afecta a los tiempos del
                           //timer 0 que produce el PWM. Por esto se
                           //deshabilitan las interrupciones para esta
                           //operación
    GIE=0;
        if(dc>0xFF) dc=0xFF;
        dc1=(unsigned char)(dc);
    GIE=1;
    }
}

void interrupt interrupcions()
{
    if(TMR1IF== 1) uno(); //Desvía la interrupción a las funciones zero y
                           //uno dependiendo de la fuente que la ha
                           //provocado, mirando los flancos de interrupción
    if(T0IF== 1) zero();
    GIE=1;                //Habilita de nuevo las interrupciones
}

void zero()               //En esta función se hace el tratamiento del
                           //PWM.
{
    //La variable "nivell" depende del estado en
    //que se encontraba la salida GPIO5
    //anteriormente
    if(nivell)
    {
        nivell=0;        //Si "nivell" está a cero es porque la salida
        T0IF=0;          //del PWM está a cero
        TMR0=dc1;        //Carga el contador para el tiempo de OFF
        GPIO5=0;         //Activa la salida a 0
    }
    else
    {
        nivell=1;        //Si "nivell" está a uno es porque la salida
        T0IF=0;          //del PWM está a uno
        TMR0=0xFF-dc1;   //Carga el contador para el tiempo de ON
        GPIO5=1;         //Activa la salida a 1
    }
}

void uno()                //Esta función ligada al timer 1 marca la
                           //frecuencia en que se muestrea el heater
{

```

```

timer_uno=1;           //Activa la variable que permite el acceso al
                       //código de programa que realiza el control
TMR1IF=0;             //Borra el flan de interrupción
TMR1ON=0;             //Timer1 OFF
TMR1H=0xE1;          //Carga el timer 1 para a que cuente 7.7 ms
TMR1L=0xF3;          //por lo tanto cada (130Hz) se desbordará

TMR1ON=1;            //Timer1 ON
}

void inicialitzacions()
{
TMR0=dc;              //Carga el timer 0 con dc que vale 1

OPTION=0xC2;          //prescaler to timer 0, fosc/8, Clock source
                       //extern. Pull-up disabled
                       //el timer 0 tindrà una freq. de 1953.12Hz
INTCON=0xE0;          //Enabled interrupts, enabled peripheral
                       //interrupts, timer0 overflow
PIE1=0x01;           //Enable ADC & timer1 interrupts
TRISIO=0x17;         //bit 0,1,2,4 inputs, las otras outputs

TMR1H=0xE1;          //Carga el timer 1 para que cuente 7,7 ms
TMR1L=0xF3;          //por lo tanto cada (130Hz) se desbordará

T1CON=0x01;          //timer 1 prescaler 1/8, internal oscillator,
                       //timer 1 ON
ANSEL=0x17;          //Fosc/8, AN0-AN1 analog input

}

void leer(char canal)
{
unsigned char i;

ADCON0=0;
ADCON0=ADCON0+((canal<<2)+0x81); //Activa el canal AD indicado sino
//hay otro en marcha
for(i=0;i<30;)        //Bucle de estere para que se realice la
                       //carga del condensador interno
i=i+1;
GODONE=1;

while (GODONE==1);   //Bucle de espera a la conversión A/D

if(canal==1)
{
conv1=0;
conv1=ADRESH;        //Cargamos el dato convertido en una
conv1=conv1<<8;      //variable tipo integer
conv1=conv1+ADRESL;
}
else
{
if(canal== 2)
{
conv2=0;
conv2=ADRESH;        //Cargamos el dato convertido en una

```

```
conv2=conv2<<8;           //variable tipo integer
conv2=conv2+ADRESL;
    }
else
{
    conv0=0;
    conv0=ADRESH;         //Cargamos el dato convertido en una
conv0=conv0<<8;         //variable tipo integer
conv0=conv0+ADRESL;
    }
}
ADON=0;                   //Para el módulo A/D
}
```

2. Programa en lenguaje ensamblador

```

        processor    12F675
indf    equ    0
rtcc    equ    1
pc      equ    2
pcl     equ    2
status  equ    3
fsr     equ    4
porta   equ    5
portb   equ    6
portc   equ    7
pclath  equ    10
psect   __Z39223RS_
opt title "Serial number: HCPIC-39223"
opt subtitle "Licensed to: JORDI ."
;dos.h: 240: int  absread( int  __drive, int  __nsects, long __lsect,

;dos.h: 242: int  abswrite( int  __drive, int  __nsects, long __lsect,

;dos.h: 248: void  ctrlbrk( int  ( *handler )( void ) );

;dos.h: 256: unsigned  _dos_creat( const char *__pathP, unsigned __attr,

;dos.h: 258: unsigned  _dos_creatnew( const char *__pathP, unsigned
__attr,

;dos.h: 260: unsigned  _dos_findfirst( const char *__path,

;dos.h: 261: unsigned  __attrib,

;dos.h: 266: unsigned  _dos_getdiskfree( unsigned __drive,

;dos.h: 269: unsigned  _dos_getfileattr( const char *__filename,

;dos.h: 271: unsigned  _dos_getftime( int  __fd, unsigned *__date,

;dos.h: 275: unsigned  _dos_open( const char *__pathP, unsigned __oflag,

;dos.h: 277: unsigned  _dos_read( int  __fd, void far *__buf, unsigned
__len,

;dos.h: 279: unsigned  _dos_setblock( unsigned __size, unsigned __segx,

;dos.h: 283: unsigned  _dos_setfileattr( const char *__filename,

;dos.h: 287: unsigned  _dos_write( int  __fd, void far *__buf, unsigned
__len,

;dos.h: 296: void  getdfree( unsigned char __drive,

;dos.h: 299: void  getfat( unsigned char __drive,

;dos.h: 311: void  _harderr( void  (far *__fptr)( ) );

;dos.h: 320: void  harderr( int  ( *__handler )( ) );

;dos.h: 334: int  int86( int  __intno,

;dos.h: 335: union REGS *__inregs,

```

```

;dos.h: 337: int  int86x( int  __intno,
;dos.h: 338: union REGS * __inregs,
;dos.h: 339: union REGS * __outregs,
;dos.h: 341: int  intdos( union REGS * __inregs,
;dos.h: 343: int  intdosx( union REGS * __inregs,
;dos.h: 344: union REGS * __outregs,
;dos.h: 360: char *  parsfnm( const char * __cmdline,
;dos.h: 365: void  pokeb( unsigned __segment,
;dos.h: 378: void  unixtodos( long __time, struct date * __d,
;stdlib.h: 122: extern int  atexit(void (*) (void));
;stdlib.h: 126: extern void  qsort(void *, size_t, size_t, int (*) (const
void *, const void *));
;stdlib.h: 127: extern void * bsearch(const void *, void *, size_t,
size_t, int (*) (const void *, const void *));

        global      _main
        signat      _main,88
        psect text0,local,class=CODE,delta=2
        psect text0
        file "prova1.c"
        line 31

_main
;prova1.c: 35: inicialitzacions();

        global      _inicialitzacions
        signat      _inicialitzacions,88
        FNCALL      _main,_inicialitzacions
        line 35

        fcall (_inicialitzacions)
;prova1.c: 38: while(1)

        line 38
        goto 127

128
;prova1.c: 39: {
;prova1.c: 40: if(timer_uno)

        global      _timer_uno
        line 40
        bcf  status,5
        movf (((_timer_uno)))
        btfsc status,2
        goto u11
        goto u10
u11

```

```

        goto 130
u10
;prova1.c: 41: {

;prova1.c: 42: conv++;

        global      _conv
        line 42
        incf (((_conv)))
        btfsc status,2
        incf (((_conv+1)))
;prova1.c: 43: }

;prova1.c: 45: if(timer_zero)

        line 43

130
        global      _timer_zero
        line 45
        bcf  status,5
        movf (((_timer_zero)))
        btfsc status,2
        goto u21
        goto u20
u21
        goto 131
u20
;prova1.c: 46: {

;prova1.c: 47: }

;prova1.c: 48: }

        line 47

131
        line 48

127
        line 38
        goto 128

129
;prova1.c: 50: }

        line 50
126
        global      start
        ljmp start
        FNSIZE      _main,0,0
;prova1.c: 79: void interrupt interrupcions( __CPPARGS)

;prova1.c: 80: {

;        param __CPPARGS assigned to ?_interrupcions+0
_interrupcions$__CPPARGS      set  ?_interrupcions+0
        psect text1,local,class=CODE,delta=2
        psect text1
        global      _interrupcions
        signat      _interrupcions,72

```

```

    line 80

_interrupcions
    psect intcode,class=CODE,delta=2
    psect intcode
    global      int_func
int_func
;prova1.c: 81: GIE = 1;

    line 81
    bsf    (05Fh/8), (05Fh)&7
;prova1.c: 82: if(ADIF==1);

    line 82
    btfss (066h/8), (066h)&7
    goto  u31
    goto  u30
u31
    goto  133
u30
;prova1.c: 83: if(TMR1IF==1) uno();

133
    line 83
    bcf    status,5
    btfss (060h/8), (060h)&7
    goto  u41
    goto  u40
u41
    goto  134
u40
    global      _uno
    signat      _uno,88
    FNCALL      _interrupcions,_uno
    bcf    status,5

    fcall (_uno)
;prova1.c: 84: if(T0IF==1) zero();

134
    line 84
    btfss (05Ah/8), (05Ah)&7
    goto  u51
    goto  u50
u51
    goto  135
u50
    global      _zero
    signat      _zero,88
    FNCALL      _interrupcions,_zero
    bcf    status,5

    fcall (_zero)
;prova1.c: 85: }

135
    line 85
132
```

```

        FNROOT      _interrupcions
        bcf        status,5
        psect intsave_0,ovrld,class=BANK0,space=1
saved_w
        ds         1
        psect intret,class=CODE,delta=2
        psect intret
        movf saved_fsr,w
        movwf fsr ;restore FSR
        movf saved_pclath,w
        movwf pclath
        global     int_restore
int_restore
        movf saved_status,w
        movwf status
        swapf saved_w
        swapf saved_w,w
        retfie
        FNSIZE     _interrupcions,0,2
        global     ?_interrupcions
        psect intentry,class=CODE,delta=2
        psect intentry
        global     int_entry
int_entry
        movwf saved_w
        movf status,w
        clrf status
        movwf saved_status
        psect intsave,class=BANK0,space=1
        psect intsave
saved_status ds 1
        psect intentry
        global     save_btemp0
        global     save_btemp1
        global     save_btemp2
        global     save_btemp3
        global     save_btemp4
        global     save_btemp5
        global     save_btemp6
        global     save_btemp7
        global     save_btemp8
        global     save_btemp9
        global     save_btemp10
        global     save_btemp11
        global     save_btemp12
        global     save_btemp13
        global     save_btemp14
        psect intsave
saved_fsr ds 1
        psect intentry
        movf fsr,w
        movwf saved_fsr
        psect intsave
saved_pclath ds 1
        psect intentry
        movf pclath,w
        movwf saved_pclath
        clrf pclath
        global     save_code_ptr
;proval.c: 87: void covertir()

```

```
;prova1.c: 88: {  
  
    psect intentry  
    global      _covertir  
    signat     _covertir,88  
    psect text2,local,class=CODE,delta=2  
    psect text2  
    line 88  
  
_covertir  
;prova1.c: 92: }  
  
    line 92  
136  
    bcf  status,5  
    return  
    FNSIZE  _covertir,0,0  
;prova1.c: 95: void uno()  
  
;prova1.c: 96: {  
  
    psect text3,local,class=CODE,delta=2  
    psect text3  
    line 96  
  
_uno  
;prova1.c: 98: }  
  
    line 98  
137  
    bcf  status,5  
    return  
    FNSIZE  _uno,0,0  
;prova1.c: 100: void zero()  
  
;prova1.c: 101: {  
  
    psect text4,local,class=CODE,delta=2  
    psect text4  
    line 101  
  
_zero  
;prova1.c: 103: }  
  
    line 103  
138  
    bcf  status,5  
    return  
    FNSIZE  _zero,0,0  
;prova1.c: 105: void inicialitzacions()  
  
;prova1.c: 106: {  
  
    psect text5,local,class=CODE,delta=2  
    psect text5  
    line 106  
  
_inicialitzacions  
;prova1.c: 107: }  
  
    line 107
```

```
139      bcf      status,5
        return
        FNSIZE      _inicialitzacions,0,0
        psect text6,local,class=CODE,delta=2
        psect text6
        _ADCS0      equ      1276
        _ADCS1      equ      1277
        _ADCS2      equ      1278
        _ADFM equ      255
        _ADIE equ      1126
        _ADIF equ      102
        _ADON equ      248
        _ANS0 equ      1272
        _ANS1 equ      1273
        _ANS2 equ      1274
        _ANS3 equ      1275
        _BOD equ      1136
        _CAL0 equ      1154
        _CAL1 equ      1155
        _CAL2 equ      1156
        _CAL3 equ      1157
        _CAL4 equ      1158
        _CAL5 equ      1159
        _CARRY      equ      24
        _CHS0 equ      250
        _CHS1 equ      251
        _CINV equ      204
        _CIS equ      203
        _CM0 equ      200
        _CM1 equ      201
        _CM2 equ      202
        _CMIE equ      1123
        _CMIF equ      99
        _COUT equ      206
        _DC equ      25
        _EEIE equ      1127
        _EEIF equ      103
        _GIE equ      95
        _GODONE      equ      249
        _GPIE equ      91
        _GPIF equ      88
        _GPIO0      equ      40
        _GPIO1      equ      41
        _GPIO2      equ      42
        _GPIO3      equ      43
        _GPIO4      equ      44
        _GPIO5      equ      45
        _GPPU equ      1039
        _INTE equ      92
        _INTEDG      equ      1038
        _INTF equ      89
        _IOCB0      equ      1200
        _IOCB1      equ      1201
        _IOCB2      equ      1202
        _IOCB3      equ      1203
        _IOCB4      equ      1204
        _IOCB5      equ      1205
        _PD equ      27
        _PEIE equ      94
        _POR equ      1137
```

```

_PSO equ 1032
_PS1 equ 1033
_PS2 equ 1034
_PSA equ 1035
_RD equ 1248
_RP0 equ 29
_T0CS equ 1037
_T0IE equ 93
_T0IF equ 90
_T0SE equ 1036
_T1CKPS0 equ 132
_T1CKPS1 equ 133
_T1OSCEN equ 131
_T1SYNC equ 130
_TMR1CS equ 129
_TMR1GE equ 134
_TMR1IE equ 1120
_TMR1IF equ 96
_TMR1ON equ 128
_TO equ 28
_TRIS0 equ 1064
_TRIS1 equ 1065
_TRIS2 equ 1066
_TRIS3 equ 1067
_TRIS4 equ 1068
_TRIS5 equ 1069
_VCFG equ 254
_VR0 equ 1224
_VR1 equ 1225
_VR2 equ 1226
_VR3 equ 1227
_VREN equ 1231
_VRR equ 1229
_WPU0 equ 1192
_WPU1 equ 1193
_WPU2 equ 1194
_WPU4 equ 1196
_WPU5 equ 1197
_WR equ 1249
_WREN equ 1250
_WRERR equ 1251
_ZERO equ 26
_ADCON0 equ 31
_ADRESH equ 30
_ADRESL equ 158
_ANSEL equ 159
_CMCON equ 25
_EEADR equ 155
_EECON1 equ 156
_EECON2 equ 157
_EEDATA equ 154
_FSR equ 4
_GPIO equ 5
_INTCON equ 11
_IOCB equ 150
_OPTION equ 129
_OSCCAL equ 144
_PCL equ 2
_PCLATH equ 10
_PCON equ 142
_PIE1 equ 140

```

```
_PIR1 equ 12
_STATUS equ 3
_T1CON equ 16
_TMR0 equ 1
_TMR1H equ 15
_TMR1L equ 14
_TRISIO equ 133
_VRCON equ 153
_WPU equ 149
    psect rbss_0,class=BANK0,space=1
    global clear_bank0
    psect rbss_0
    global _ref
_ref
    ds 1
_timer_uno
    ds 1
_timer_zero
    ds 1
_conv
    ds 2
end
```

3. Archivo pic12f6x.h

```

/*      Header file for the Microchip
 *      PIC12F629 chip
 *      PIC12F675 chip
 *      Microcontrollers */

static volatile      unsigned char      TMR0   @ 0x01;
static volatile      unsigned char      PCL    @ 0x02;
static volatile      unsigned char      STATUS @ 0x03;
static volatile      unsigned char      FSR    @ 0x04;
static volatile      unsigned char      GPIO   @ 0x05;
static volatile      unsigned char      PCLATH @ 0x0A;
static volatile      unsigned char      INTCON @ 0x0B;
static volatile      unsigned char      PIR1  @ 0x0C;
static volatile      unsigned char      TMR1L @ 0x0E;
static volatile      unsigned char      TMR1H @ 0x0F;
static                unsigned char      T1CON @ 0x10;
static volatile      unsigned char      CMCON @ 0x19;
#if defined(_12F675)
static volatile      unsigned char      ADRESH @ 0x1E;
static volatile      unsigned char      ADCON0 @ 0x1F;
#endif
static                bank1 unsigned char  OPTION @ 0x81;
static                bank1 unsigned char  TRISIO @ 0x85;
static                bank1 unsigned char  PIE1  @ 0x8C;
static volatile bank1 unsigned char  PCON  @ 0x8E;
static                bank1 unsigned char  OSCCAL @ 0x90;
static                bank1 unsigned char  WPU   @ 0x95;
static                bank1 unsigned char  IOCB  @ 0x96;
static volatile bank1 unsigned char  VRCON @ 0x99;
static volatile bank1 unsigned char  EEDATA @ 0x9A;
static volatile bank1 unsigned char  EEADR @ 0x9B;
static volatile bank1 unsigned char  EECON1 @ 0x9C;
static volatile bank1 unsigned char  EECON2 @ 0x9D;
#if defined(_12F675)
static volatile bank1 unsigned char  ADRESL @ 0x9E;
static                bank1 unsigned char  ANSEL @ 0x9F;
#endif

/* STATUS Bits */
static volatile bit   CARRY @ (unsigned)&STATUS*8+0;
static volatile bit   DC    @ (unsigned)&STATUS*8+1;
static volatile bit   ZERO  @ (unsigned)&STATUS*8+2;
static volatile bit   PD    @ (unsigned)&STATUS*8+3;
static volatile bit   TO    @ (unsigned)&STATUS*8+4;
static                bit   RP0 @ (unsigned)&STATUS*8+5;

/* GPIO Bits */
static volatile bit   GPIO0 @ (unsigned)&GPIO*8+0;
static volatile bit   GPIO1 @ (unsigned)&GPIO*8+1;
static volatile bit   GPIO2 @ (unsigned)&GPIO*8+2;
static volatile bit   GPIO3 @ (unsigned)&GPIO*8+3;
static volatile bit   GPIO4 @ (unsigned)&GPIO*8+4;
static volatile bit   GPIO5 @ (unsigned)&GPIO*8+5;

/* INTCON Bits */
static volatile bit   GPIF @ (unsigned)&INTCON*8+0;
static volatile bit   INTF @ (unsigned)&INTCON*8+1;

```

```

static volatile bit    T0IF @ (unsigned) &INTCON*8+2;
static                bit    GPIE @ (unsigned) &INTCON*8+3;
static                bit    INTE @ (unsigned) &INTCON*8+4;
static                bit    T0IE @ (unsigned) &INTCON*8+5;
static                bit    PEIE @ (unsigned) &INTCON*8+6;
static                bit    GIE  @ (unsigned) &INTCON*8+7;

/* PIR1 Bits */
static volatile bit    TMR1IF      @ (unsigned) &PIR1*8+0;
static volatile bit    CMIF        @ (unsigned) &PIR1*8+3;
static volatile bit    ADIF        @ (unsigned) &PIR1*8+6;
static volatile bit    EEIF        @ (unsigned) &PIR1*8+7;

/* T1CON Bits */
static                bit    TMR1ON      @ (unsigned) &T1CON*8+0;
static                bit    TMR1CS      @ (unsigned) &T1CON*8+1;
static                bit    T1SYNC      @ (unsigned) &T1CON*8+2;
static                bit    T1OSCEN     @ (unsigned) &T1CON*8+3;
static                bit    T1CKPS0     @ (unsigned) &T1CON*8+4;
static                bit    T1CKPS1     @ (unsigned) &T1CON*8+5;
static                bit    TMR1GE      @ (unsigned) &T1CON*8+6;

/* CMCON Bits */
static                bit    CM0         @ (unsigned) &CMCON*8+0;
static                bit    CM1         @ (unsigned) &CMCON*8+1;
static                bit    CM2         @ (unsigned) &CMCON*8+2;
static                bit    CIS         @ (unsigned) &CMCON*8+3;
static                bit    CINV        @ (unsigned) &CMCON*8+4;
static volatile bit    COUT             @ (unsigned) &CMCON*8+6;

#if defined(_12F675)
/* ADCON0 Bits */
static volatile bit    ADON @ (unsigned) &ADCON0*8+0;
static volatile bit    GODONE @ (unsigned) &ADCON0*8+1;
static                bit    CHS0 @ (unsigned) &ADCON0*8+2;

static                bit    CHS1 @ (unsigned) &ADCON0*8+3;
static                bit    VCFG @ (unsigned) &ADCON0*8+6;
static                bit    ADFM @ (unsigned) &ADCON0*8+7;
#endif

/* OPTION Bits */
static                bank1 bit    PS0 @ (unsigned) &OPTION*8+0;
static                bank1 bit    PS1 @ (unsigned) &OPTION*8+1;
static                bank1 bit    PS2 @ (unsigned) &OPTION*8+2;
static                bank1 bit    PSA @ (unsigned) &OPTION*8+3;
static                bank1 bit    T0SE @ (unsigned) &OPTION*8+4;
static                bank1 bit    T0CS @ (unsigned) &OPTION*8+5;
static                bank1 bit    INTEDG @ (unsigned) &OPTION*8+6;
static                bank1 bit    GPPU @ (unsigned) &OPTION*8+7;

/* TRISIO Bits */
static                bank1 bit    TRIS0 @ (unsigned) &TRISIO*8+0;
static                bank1 bit    TRIS1 @ (unsigned) &TRISIO*8+1;
static                bank1 bit    TRIS2 @ (unsigned) &TRISIO*8+2;
static                bank1 bit    TRIS3 @ (unsigned) &TRISIO*8+3;
static                bank1 bit    TRIS4 @ (unsigned) &TRISIO*8+4;
static                bank1 bit    TRIS5 @ (unsigned) &TRISIO*8+5;

/* PIE1 Bits */
static                bank1 bit    TMR1IE @ (unsigned) &PIE1*8+0;

```

```

static          bank1 bit   CMIE @ (unsigned) &PIE1*8+3;
static          bank1 bit   ADIE @ (unsigned) &PIE1*8+6;
static          bank1 bit   EEIE @ (unsigned) &PIE1*8+7;

/* PCON Bits */
static volatile bank1 bit   BOD  @ (unsigned) &PCON*8+0;
static volatile bank1 bit   POR  @ (unsigned) &PCON*8+1;

/* OSCCAL Bits */
static          bank1 bit   CAL0 @ (unsigned) &OSCCAL*8+2;
static          bank1 bit   CAL1 @ (unsigned) &OSCCAL*8+3;
static          bank1 bit   CAL2 @ (unsigned) &OSCCAL*8+4;
static          bank1 bit   CAL3 @ (unsigned) &OSCCAL*8+5;
static          bank1 bit   CAL4 @ (unsigned) &OSCCAL*8+6;
static          bank1 bit   CAL5 @ (unsigned) &OSCCAL*8+7;

/* WPU Bits */
static          bank1 bit   WPU0 @ (unsigned) &WPU*8+0;
static          bank1 bit   WPU1 @ (unsigned) &WPU*8+1;
static          bank1 bit   WPU2 @ (unsigned) &WPU*8+2;
static          bank1 bit   WPU4 @ (unsigned) &WPU*8+4;
static          bank1 bit   WPU5 @ (unsigned) &WPU*8+5;

/* IOCB Bits */
static          bank1 bit   IOCB0 @ (unsigned) &IOCB*8+0;
static          bank1 bit   IOCB1 @ (unsigned) &IOCB*8+1;
static          bank1 bit   IOCB2 @ (unsigned) &IOCB*8+2;
static          bank1 bit   IOCB3 @ (unsigned) &IOCB*8+3;
static          bank1 bit   IOCB4 @ (unsigned) &IOCB*8+4;
static          bank1 bit   IOCB5 @ (unsigned) &IOCB*8+5;

/* VRCON Bits */
static          bank1 bit   VR0   @ (unsigned) &VRCON*8+0;
static          bank1 bit   VR1   @ (unsigned) &VRCON*8+1;
static          bank1 bit   VR2   @ (unsigned) &VRCON*8+2;
static          bank1 bit   VR3   @ (unsigned) &VRCON*8+3;
static          bank1 bit   VRR   @ (unsigned) &VRCON*8+5;
static          bank1 bit   VREN  @ (unsigned) &VRCON*8+7;

/* EECON1 Bits */
static volatile bank1 bit   RD    @ (unsigned) &EECON1*8+0;
static volatile bank1 bit   WR    @ (unsigned) &EECON1*8+1;
static          bank1 bit   WREN  @ (unsigned) &EECON1*8+2;
static volatile bank1 bit   WRERR @ (unsigned) &EECON1*8+3;

#ifdef _12F675
/* ANSEL Bits */
static          bank1 bit   ANS0  @ (unsigned) &ANSEL*8+0;
static          bank1 bit   ANS1  @ (unsigned) &ANSEL*8+1;
static          bank1 bit   ANS2  @ (unsigned) &ANSEL*8+2;
static          bank1 bit   ANS3  @ (unsigned) &ANSEL*8+3;
static          bank1 bit   ADCS0 @ (unsigned) &ANSEL*8+4;
static          bank1 bit   ADCS1 @ (unsigned) &ANSEL*8+5;
static          bank1 bit   ADCS2 @ (unsigned) &ANSEL*8+6;
#endif

/* EEPROM related macros */
#define EEPROM_SIZE      128

#define EEPROM_WRITE(addr, value) \
    while(WR) continue; EEADR=(addr); EEDATA=(value); \

```

```

CARRY=0;if (GIE)CARRY=1;GIE=0; \
WREN=1;EECON2=0x55;EECON2=0xAA;WR=1;WREN=0; \
if (CARRY)GIE=1

#define      EEPROM_READ(addr)  (EEADR=(addr), (RD=1), EEDATA)

/* Configuration bit definitions */
#define CONFIG_ADDR      0x2007

/* Protection of data block */
#define CPD 0x3EFF      /* data protect on */
#define UNPROTECT 0x3FFF /* data protect off */

/* Protection of program code */
#define PROTECT 0x3F7F  /* code protection on */
#define UNPROTECT 0x3FFF /* code protection off */

/* Brown out detection enable */
#define BOREN 0x3FFF    /* brown out reset enabled */
#define BORDIS 0x3FBF  /* brown out reset disabled */

/* Memory clear enable */
#define MCLREN 0x3FFF   /* memory clear function enabled */
#define MCLRDIS 0x3FDF /* memory clear function disabled */

/* Power up timer enable */
#define PWRTDIS 0x3FFF  /* power up timer disabled */
#define PWRTEN 0x3FEF  /* power up timer enabled */

/* Watchdog timer enable */
#define WDTEN 0x3FFF    /* watchdog timer enabled */
#define WDTDIS 0x3FF7  /* watchdog timer disabled */

/* Oscillator configurations */
#define RCCLK 0x3FFF    /* GP4=clock out signal/GP5=RC osc */
#define RCIO 0x3FFE    /* GP4=IO/GP5=RC osc */
#define INTCLK 0x3FFD  /* internal osc/GP4=clock out
/* signal/GP5=IO */
#define INTIO 0x3FFC   /* internal osc/GP4=IO//GP5=IO */
#define EC 0x3FFB     /* external clock */
#define HS 0x3FFA     /* high speed crystal/resonator */
#define XT 0x3FF9     /* crystal/resonator */
#define LP 0x3FF8     /* low power crystal/resonator */

```


ANEXO 2:

Manual de consulta rápida para el usuario

1. Introducción

El sistema de control de temperatura está diseñado para actuar sobre sensores de gases con resistencia de 550 Ω y 650 Ω . Este control permite mantener constante la temperatura de las membranas del sensor de forma individual. La configuración modular del sistema permite controlar hasta 8 membranas a diferentes temperaturas.

La regulación de la temperatura permite realizar unas medidas más fiables debido a que estas no se ven afectadas por las variaciones de la temperatura ambiente. La sensibilidad del sensor será pues la misma para todas las medidas tomadas.

El conjunto está formado por una placa base y 8 placas de control interconectadas mediante un bus. Las placas se inicializan por ordenador mediante una tarjeta de comunicaciones. Cada tarjeta de control puede ser inicializada a una temperatura diferente. También pueden inicializarse por grupo o todas a la vez dependiendo de los bits de configuración de placa.

2. Conexión y configuración

El conexionado del sistema se realiza a través de la placa base. Los conectores de la placa son:

- Alimentación. Regletas PCB.
 - + 15 V
 - - 15 V
 - GND

- Comunicación con el ordenador. Conector 20 *pins* para cable plano.
 - Bits de selección
 - PWM de consigna
 - Bit de confirmación

- Conexión de las membranas. Regletas PCB.
 - 8 Regletas para la conexión del *heater*
 - 2 Regletas de GND

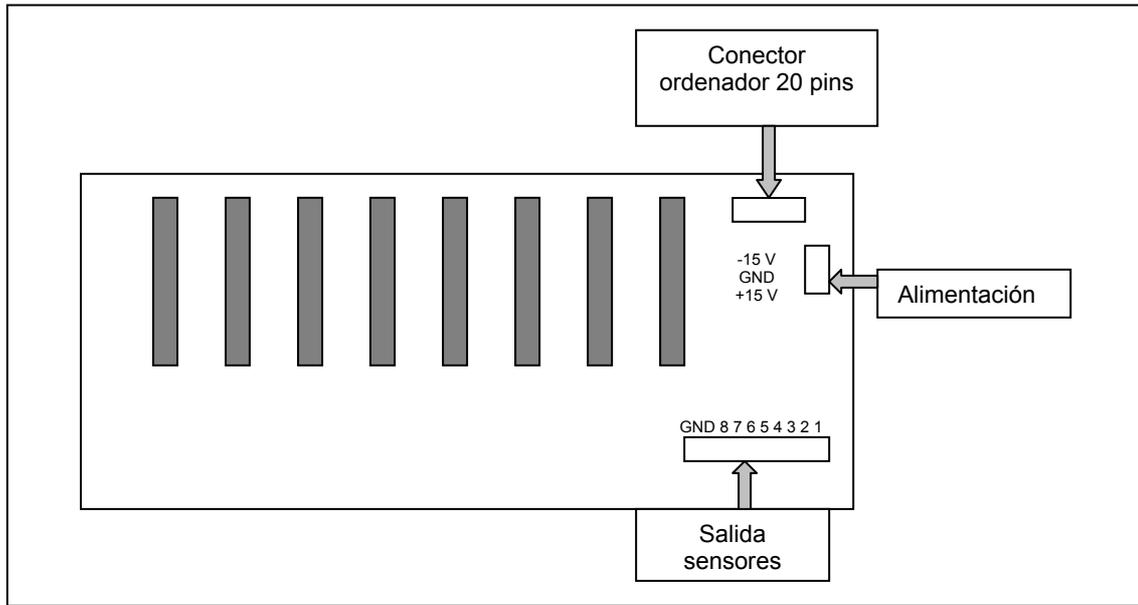


Figura 1. Conectores de la placa base

Las placas de control se colocan en los conectores de 22 *pins* tipo *slot*. Su conexión es sencilla.

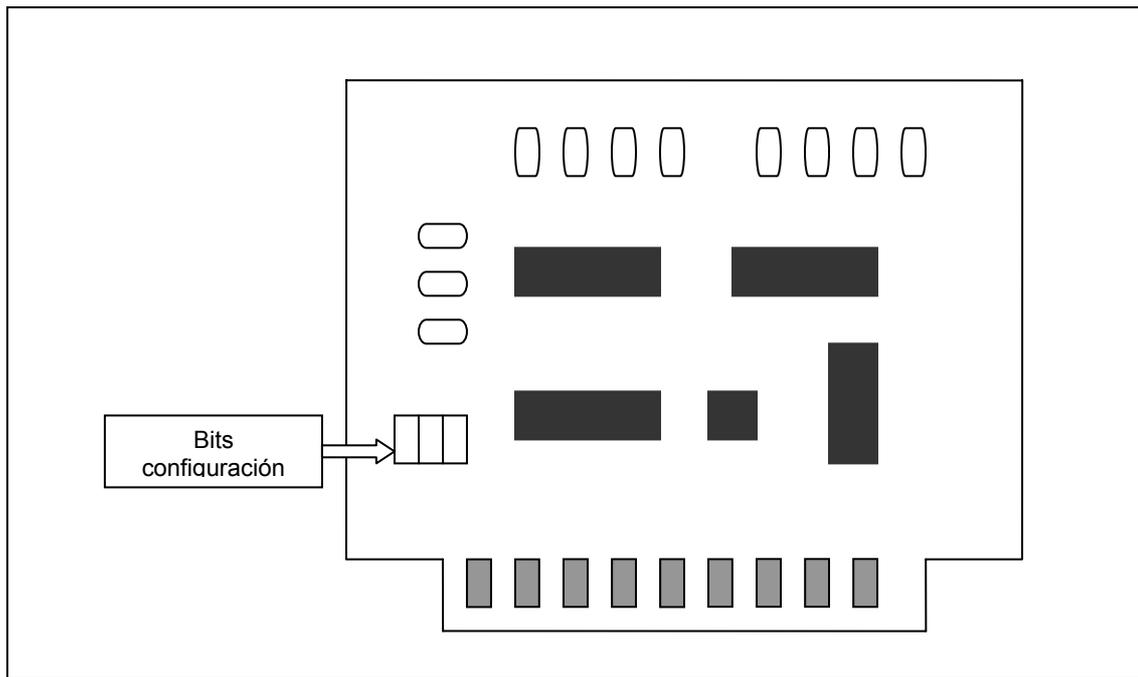


Figura 2. Tarjeta de control con los bits de configuración

Antes de colocarlas en las ranuras debe configurarse el número de placa. Esta configuración dependerá del uso que quiera darse al sistema. El número de placa se utiliza para la inicialización de la placa. Si cada placa tiene un número distinto, estas pueden ser inicializadas de modo independiente seleccionando cada tarjeta. Si se desea una inicialización única para todas las placas estas deberán configurarse con el mismo código binario de tres bits. También se pueden seleccionar por grupos.

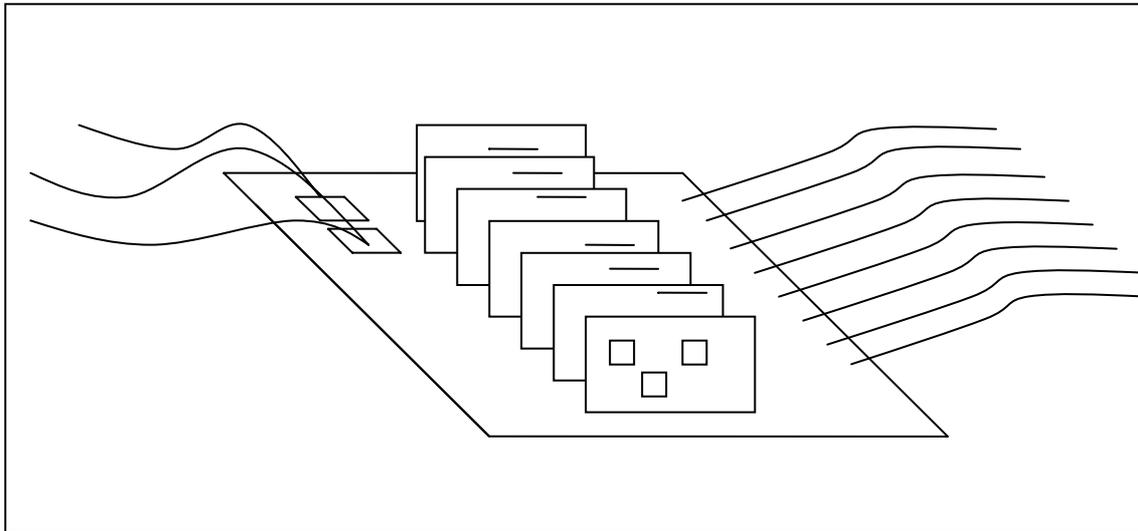


Figura 3. Aspecto del sistema completo

Es importante que se observe bien que la posición que ocupa la placa de control en la placa base corresponde con la salida de corriente del sensor. Y sobretodo que las placas que estén programadas para controlar sensores de 550Ω que se conecten a las membranas de esta impedancia, y al igual con las de 650Ω . Si no fuese de esta forma el control no sería el correcto.

Existe la posibilidad de utilizar sensores con resistencia de *heater* diferentes de los dos valores que se utilizan en este sistema. Únicamente se necesita cambiar una línea del código de programa del microcontrolador. La línea de código es la siguiente:

```
resist=((valor1) * (VALOR_HEATER*0.000679)+VALOR_HEATER);
```

Esta línea de código es equivalente a la ecuación

$$R = T \cdot R_o \cdot 6,79 \cdot 10^{-4} + R_o$$

Donde $R_o = \text{VALOR_HEATER}$ que es igual a la resistencia del *heater* a temperatura ambiente.

Así que si variamos este valor podemos adaptarlo al sensor que se desee.

IMPORTANTE: Si se desea adaptar el sistema a un sensor específico podrá hacerse de la forma anteriormente explicada siempre que el coeficiente de temperatura del *heater* sea de $6,79 \cdot 10^{-4} \text{ } ^\circ\text{C}^{-1}$. Si este coeficiente varía se deberá revisar el programa para comprobar el rango de temperatura que se pueden alcanzar.

3. Funcionamiento

Una vez hecho el conexionado y la configuración de las placas, el sistema puede empezar a funcionar.

Para realizar el control a una temperatura preestablecida hay que inicializar el sistema. La forma de hacerlo se verá en el siguiente apartado.

El circuito de control proporciona la intensidad necesaria al *heater* para que este mantenga su temperatura de consigna.

El sistema realizará este control hasta que se desconecte la alimentación. Por lo tanto para variar la temperatura de trabajo de los sensores se deberá desconectar el sistema y volver a inicializar.

4. Inicialización

Para conseguir que la temperatura del *heater* quede fijada a un determinado valor se establece una comunicación con el ordenador. Este proporciona una señal proporcional a la temperatura de trabajo del sensor. Y el sistema mantiene esta temperatura constante hasta la desconexión del sistema.

El programa del ordenador nos pedirá la temperatura de trabajo a la que se desea trabajar, y también la placa o las placas que se desea inicializar. De esta forma se puede realizar una inicialización para cada placa de control

El PWM pues realizarse de distintas formas con el ordenados. El rango de frecuencias que admite el filtro ha de ser superior a 40 Hz para tener un error inferior al 0,4 %. El programa Matlab permite realizar un PWM en *real time* que proporcionaría al sistema una precisión de menos de 1 °C.