



Universidad de Valladolid



**ESCUELA DE INGENIERÍAS
INDUSTRIALES**

UNIVERSIDAD DE VALLADOLID

ESCUELA DE INGENIERIAS INDUSTRIALES

Grado en Ingeniería en Electrónica Industrial y Automática

Termómetro digital

Autor:

Olivar Ruiz, Marta

Tutor:

**González de la Fuente, José Manuel
Dpto. Tecnología Electrónica**

Valladolid, Julio de 2015

Resumen

En este trabajo de fin de grado se ha realizado un termómetro digital utilizando Arduino, plataforma hardware y software basada en microcontroladores comerciales integrados en una PCB y un display digital. Como transductor de temperatura se emplea el termopar de tipo K. Se abarcan todas las etapas, desde el diseño del circuito electrónico hasta la fabricación del prototipo. El circuito electrónico diseñado realiza el acondicionamiento de las señales, tanto del termopar como del sensor de temperatura de la unión fría, así como la implementación del resto de funciones del termómetro. También se muestra la labor de programación realizada en la plataforma Arduino para la compensación de la unión fría del termopar y ejecución del resto de funcionalidades. Finalmente se realiza la construcción del prototipo que engloba la fabricación de la placa de circuito impreso e integración con el resto de componentes del dispositivo dentro de una carcasa.

Palabras clave:

Arduino, termómetro, termopar, compensación, acondicionamiento

Índice general

| | |
|--|-----------|
| Termómetro digital..... | 1 |
| Resumen..... | 3 |
| Índice general | 5 |
| Índice de figuras | 7 |
| 1. Introducción | 9 |
| 2. Teoría | 11 |
| 2.1. Termopar | 11 |
| 2.2. Compensación de la unión fría | 12 |
| 2.3. Compensación de la unión fría por Hardware | 13 |
| 2.4. Compensación de la unión fría por Software..... | 13 |
| 3. Diseño..... | 15 |
| 3.1. Especificaciones..... | 15 |
| 3.2. Arduino..... | 16 |
| 3.3. Display..... | 19 |
| 3.4. Acondicionamiento de la señal del termopar | 20 |
| 3.4.1. Amplificación | 20 |
| 3.4.2. Circuito generador de la tensión V_{ref} | 21 |
| 3.4.3. Circuito de detección de rotura de termopar | 22 |
| 3.4.4. Filtrado de la señal amplificada del termopar | 23 |
| 3.4.5. Limitación de la tensión en las entradas analógicas | 23 |
| 3.5. Acondicionamiento de la señal del LM35 | 24 |
| 3.5.1. LM35..... | 24 |
| 3.5.2. AMPLIFICACIÓN | 24 |
| 3.5.3. Limitación de la tensión en las entradas analógicas | 25 |
| 3.6. Alimentación asimétrica-tierra virtual..... | 26 |
| 3.7. Circuito de detección de batería baja..... | 27 |
| 3.8. Circuito de encendido y apagado..... | 28 |
| 3.9. Conmutación entre batería y adaptador de corriente | 30 |
| 3.10. Diseño de PCB..... | 31 |
| 3.11. Construcción del prototipo | 35 |
| 4. Programación..... | 39 |
| 4.1. Descripción del software de Arduino | 39 |
| 4.2. Descripción del programa implementado | 39 |

| | | |
|----------------|--|-----------|
| 4.3. | Diagrama de bloques del programa..... | 40 |
| 5. | Cálculos..... | 45 |
| 5.1. | Cálculos de diseño | 45 |
| 5.1.1. | Divisor de tensión del circuito Vref..... | 45 |
| 5.1.2. | Resistencia y condensador del filtro pasa-bajo | 45 |
| 5.1.3. | Resistencia del limitador de tensión de entradas analógicas..... | 46 |
| 5.1.4. | Calibración del acondicionamiento de la señal del termopar..... | 46 |
| 5.1.5. | Ganancia del amplificador del LM35..... | 47 |
| 5.1.6. | Calibración del acondicionamiento de la señal del LM35 | 47 |
| 5.1.7. | Divisor de tensión en la Alimentación asimétrica..... | 48 |
| 5.1.8. | Divisor de tensión para detección de batería baja | 48 |
| 5.1.9. | Resistencia para alimentación de la bobina del relé..... | 48 |
| 5.1.10. | Consumo del dispositivo..... | 49 |
| 5.2. | Cálculo de errores | 50 |
| 5.2.1. | Error debido al convertidor analógico-digita | 50 |
| 5.2.2. | Error en el acondicionamiento de la señal del termopar | 51 |
| 5.2.3. | Error en el acondicionamiento de la señal del LM35 | 52 |
| 6. | Resultados | 55 |
| 7. | Conclusiones..... | 57 |
| 8. | Bibliografía | 59 |
| Anexo A | | 63 |
| Anexo B | | 65 |
| Anexo C | | 75 |
| Anexo D | | 81 |

Índice de figuras

| | |
|--|-----------|
| <i>Figura 1. Representación de un termopar</i> | <i>11</i> |
| <i>Figura 2. Representación del termopar con la unión fría a 0 °C.....</i> | <i>12</i> |
| <i>Figura 3. Vista frontal y trasera de la placa Arduino UNO.....</i> | <i>17</i> |
| <i>Figura 4. Display PC 1602-H de Powertip</i> | <i>19</i> |
| <i>Figura 5. Esquemático del circuito de acondicionamiento de la señal del termopar</i> | <i>20</i> |
| <i>Figura 6. Esquema típico de un filtro pasabajos</i> | <i>23</i> |
| <i>Figura 7. Circuito diseñado para el acondicionamiento de la señal del sensor LM35</i> | <i>24</i> |
| <i>Figura 8. Configuración no inversora del amplificador operacional</i> | <i>25</i> |
| <i>Figura 9. Esquemático del circuito de tierra virtual para la alimentación asimétrica</i> | <i>27</i> |
| <i>Figura 10. Esquemático del circuito de detección de batería baja.....</i> | <i>28</i> |
| <i>Figura 11. Esquemático del circuito de apagado y encendido.....</i> | <i>29</i> |
| <i>Figura 12. Imagen y esquemático de conexiones del conector del adaptador de corriente.....</i> | <i>30</i> |
| <i>Figura 13. Esquemático del circuito de conmutación entre batería y adaptador de corriente.....</i> | <i>31</i> |
| <i>Figura 14. Esquemático completo del circuito diseñado.....</i> | <i>32</i> |
| <i>Figura 15. Situación de los componentes en la PCB</i> | <i>33</i> |
| <i>Figura 16. Footprints de la cara superior (izquierda) e inferior (derecha) de la PCB.....</i> | <i>33</i> |
| <i>Figura 17. Cara superior de la PCB fabricada</i> | <i>34</i> |
| <i>Figura 18. Cara inferior de la PCB fabricada</i> | <i>34</i> |
| <i>Figura 19. Imágenes de la conexión de la PCB con la placa de Arduino....</i> | <i>35</i> |
| <i>Figura 20. Parte frontal del termómetro con el display y los botones pulsadores.....</i> | <i>35</i> |
| <i>Figura 21. Interior del termómetro</i> | <i>36</i> |
| <i>Figura 22. Conectores para termopar y adaptador de corriente</i> | <i>36</i> |
| <i>Figura 23. Parte posterior del termómetro</i> | <i>37</i> |
| <i>Figura 24. Diagrama de bloques del programa principal: inicialización y rutina de T^a máxima, mínima y media.....</i> | <i>40</i> |
| <i>Figura 25. Diagrama de bloques del programa principal: rutina Hold</i> | <i>41</i> |
| <i>Figura 26. Diagrama de bloques del programa principal: rutina Clear</i> | <i>41</i> |

| | |
|---|-----------|
| <i>Figura 27. Diagrama de bloques del programa principal: rutina de retorno a pantalla principal por tiempo y rutina de apagado y auto_apagado.....</i> | <i>42</i> |
| <i>Figura 28. Diagrama de bloques de la interrupción: inicialización de variables, lectura de señales y rutina de detección de batería baja.....</i> | <i>43</i> |
| <i>Figura 29. Diagrama de bloques de la interrupción: tratamiento de las señales del termopar y LM35.....</i> | <i>43</i> |
| <i>Figura 30. Diagrama de bloques de la interrupción: incremento de contadores temporales.....</i> | <i>44</i> |
| <i>Figura 31. Esquema de un divisor de tensión.....</i> | <i>45</i> |
| <i>Figura 32. Esquema del circuito limitador para el cálculo de $R7$.....</i> | <i>46</i> |
| <i>Figura 33. Imagen de la caja aislada en la que se toman las medidas (izquierda) y detalle de la colocación de los termopares en la resistencia cerámica (derecha).....</i> | <i>55</i> |

1. Introducción

Existen multitud de tipos de termómetros, aunque en definitiva todos se basan en la lectura de una magnitud medible (tensión, presión, resistencia, etc.) que cambia con la temperatura.

Hoy en día con el avance de la tecnología, multitud de termómetros electrónicos integran funciones adicionales a la simple medida de la temperatura: registros de temperatura, comunicación, alarmas (algunas asociadas a salidas para actuadores...), utilizando para ello un pequeño microcontrolador.

El objeto de este TFG es la realización de un termómetro digital basado en la plataforma comercial Arduino y utilizando el termopar de tipo K como transductor de temperatura. Para mostrar la temperatura se utilizará un display digital. La fabricación del termómetro parte del diseño electrónico del circuito, pasando por la realización física de la placa, la programación para el control del dispositivo y por último la construcción del prototipo.

El objetivo final es que las prestaciones del termómetro se aproximen a los de un termómetro comercial y cumpla con las especificaciones planteadas al inicio del desarrollo. La principal prioridad del diseño es la precisión de la medida, y por lo tanto la minimización de los errores, con el menor consumo posible de energía y construyendo un dispositivo manejable.

2. Teoría

2.1. Termopar

El termopar es uno de los transductores de temperatura más frecuentemente utilizados. Está formado por dos hilos de diferente metal unidos en un extremo. Este punto es llamado unión caliente o punto de medida de la temperatura, mientras que el otro extremo del termopar forma la unión fría o unión de referencia (ver Figura 1). Si estos dos puntos se encuentran a diferente temperatura, se produce una pequeña diferencia de tensión en la unión fría (del orden de milivoltios) y por tanto, una pequeña corriente que circulará por el circuito. Este efecto termoeléctrico es llamado *Efecto Seebeck* [1]. De esta forma, se obtiene una tensión en función de la diferencia de temperatura entre la unión caliente (T_j) y la unión fría (T_{ref}).

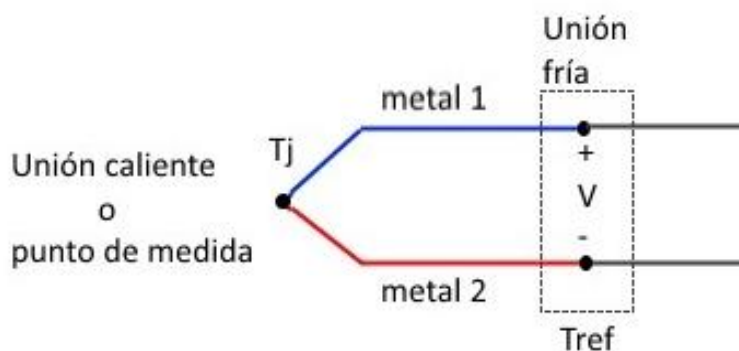


Figura 1. Representación de un termopar

Como ventajas que presentan, los termopares son económicos, robustos, tienen conexiones estándar, miden un amplio rango de temperatura y proporcionan una respuesta rápida. Por otro lado, su principal desventaja es que la relación entre tensión y temperatura no es lineal.

Existen distintos tipos de termopares en función de la combinación de metales de que estén fabricados. En la Tabla 1 se muestran los principales tipos y sus características.

| Tipo de termopar | Conductor | | Rango T^a (°C) | Rango tensión (mV) |
|------------------|---------------|-----------------|------------------|--------------------|
| | Positivo | Negativo | | |
| B | Platino (30%) | Platino (6%) | 0 a 1820 | 0 a 13.82 |
| | Rodio | Rodio | | |
| E | Níquel-Cromo | Constantán | -270 a 1000 | -9.835 a 76.373 |
| J | Hierro | Constantán | -210 a 1200 | -8.095 a 69.553 |
| K | Níquel-Cromo | Níquel-Aluminio | -270 a 1372 | -6.548 a 54.871 |
| N | Nicrosil | Nisil | -270 a 1300 | -4.345 a 47.502 |

| | | | | |
|----------|------------------------|------------|------------|-----------------|
| R | Platino (13%) Rodio | Platino | -50 a 1768 | -0.226 a 21.103 |
| S | Platino (10%) Rodio | Platino | -50 a 1768 | -0.235 a 18.693 |
| T | Cobre | Constantan | -270 a 400 | -5.603 a 20.872 |

Tabla 1. Tipos de termopares

(Fuente: http://srdata.nist.gov/its90/tables/table_i.html)

El termómetro digital diseñado en este TFG está creado para usarlo con termopar tipo K, que es uno de los más comunes.

2.2. Compensación de la unión fría

Como ya se explicó anteriormente, un termopar va a generar una tensión en función de la diferencia de temperatura entre la unión fría y la unión caliente: $V = f(T_j - R_{ref})$. Por ello, es necesario conocer la temperatura de la unión de referencia para poder conocer la temperatura a medir.

Existen unas tablas muy precisas [23] creadas por *The National Institute of Standards and Technology* (NIST), donde se relaciona el voltaje producido en el termopar y la temperatura de la unión caliente, suponiendo la unión fría a 0 °C. Además estos valores de la tabla se pueden obtener con bastante exactitud mediante la utilización de cálculos polinómicos.

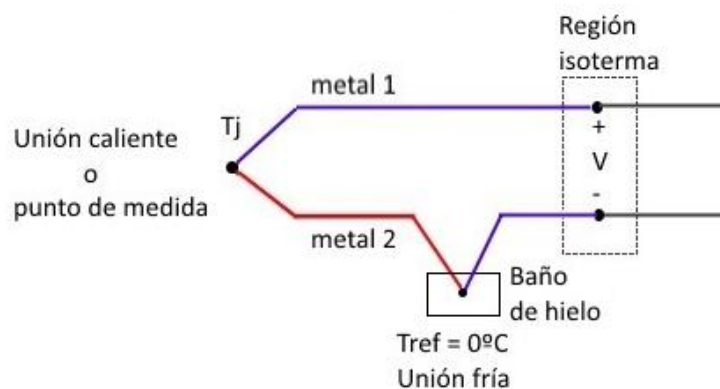


Figura 2. Representación del termopar con la unión fría a 0 °C

Tradicionalmente, para conseguir que la unión fría se mantuviera a 0 °C, se introducía en un baño de agua y hielo. Por supuesto, este método no es nada práctico, por lo que actualmente se utiliza otro, llamado *compensación de la unión fría*. Consiste en obtener la temperatura de la unión fría a través de otro sensor, para después calcular la tensión que produciría el termopar suponiendo que la unión de referencia estuviese a cero grados centígrados y la unión caliente a la temperatura medida con ese otro sensor (a través de las tablas del NIST). Calculada esa tensión, se añadiría a la tensión producida por

el propio termopar. Con esta tensión final calculada se podría obtener la temperatura absoluta medida por el termopar, utilizando las tablas del *NIST*.

2.3. Compensación de la unión fría por Hardware

En la compensación por hardware se agrega a la propia tensión producida por el termopar, una tensión proporcional a la temperatura de referencia, que es la que produciría el termopar si su unión fría estuviera a cero grados centígrados y su unión caliente a la temperatura de referencia. Así, la tensión que se obtiene es independiente de la temperatura de referencia y representa la temperatura que mide el termopar de forma absoluta.

Como ya se explicó, la relación tensión-temperatura del termopar no es lineal aunque en ciertos tramos de valores se puede aproximar a una relación lineal. Esta proporcionalidad lineal aproximada es la que se utiliza en la compensación por hardware, de manera que se obtiene menor precisión que en la compensación por software, además de necesitar una red específica de compensación diferente para cada tipo de termopar. Por otro lado, este tipo de compensación resulta útil debido a que no se necesita de ningún sistema de cálculo y la mayoría de las aproximaciones suelen ser válidas.

Un circuito integrado muy típico que realiza esta función de compensación, es el AD595 de *Analog Devices* [11]. Además de trabajar como compensador de unión fría, también realiza la función de amplificador, suministrando una tensión de salida de 10 mV/°C. Es un valor de tensión directamente proporcional a la temperatura medida, lo cual resulta muy útil, aunque presenta un error de hasta 4.25 °C para un rango de temperatura ambiente de 0 °C a 50 °C [4].

El objetivo del presente proyecto es obtener la medida de temperatura a través de otro método de compensación, para intentar minimizar el error en las mediciones. Ese otro método que se utilizará se explica a continuación.

2.4. Compensación de la unión fría por Software

La compensación se realiza mediante cálculos una vez es conocida la tensión producida por el termopar (V_{term}) y la temperatura de la unión fría (T_{ref}). El proceso para conocer la temperatura absoluta medida sigue los siguientes pasos:

Se mide la temperatura de la unión fría con un sensor independiente (T_{ref}).

Se convierte esta temperatura en un voltaje equivalente (V_{ref}), que será el que mediría un termopar con la unión fría a 0 °C y la unión caliente a T_{ref} . Para ello, se utiliza, o bien el cálculo polinómico, o las tablas de equivalencias.

En este diseño se contempla el uso del cálculo polinómico, que en este caso realizará la conversión de temperatura a tensión. El polinomio de cálculo se muestra en la ecuación (1):

$$V = \sum_{i=0}^n c_i \cdot T^i \quad (1)$$

Donde c_i son coeficientes que dependen del tipo de termopar que se utilice.

Para el termopar tipo K, si la temperatura es mayor que cero, la anterior ecuación no se deberá usar, siendo la ecuación (2) la correcta:

$$V = a_0 \cdot (a_1(T - a_2)^2) + \sum_{i=0}^n c_i \cdot T^i \quad (2)$$

Donde a_0 , a_1 y a_2 son también coeficientes específicos para termopar de tipo K.

Una vez se obtiene esa tensión V_{ref} , se añade ésta a la tensión producida por el termopar, obteniendo el verdadero voltaje que produciría el termopar con la unión fría a 0 °C:

$$V_{total} = V_{term} + V_{ref} \quad (3)$$

Por último, sólo hay que transformar ese voltaje en temperatura, a través del polinomio de conversión de tensión a temperatura según la ecuación (4):

$$T = \sum_{i=0}^n d_i \cdot V^i \quad (4)$$

Donde d_i son coeficientes que dependen del tipo de termopar que se utilice.

Todos los coeficientes d_i , c_i , a_0 , a_1 y a_2 necesarios para realizar la compensación de la unión fría por software se pueden obtener de la página web del *NIST* [24]. En el Anexo A se muestran los coeficientes para el termopar de tipo K.

Para medir la temperatura de la unión fría se va a utilizar el sensor LM35 de Texas Instruments [25]. Es un sensor frecuentemente utilizado, bastante preciso y fácil de encontrar en el mercado.

3. Diseño

3.1. Especificaciones

El termómetro a diseñar en este TFG ha de utilizar como transductor de temperatura el termopar de tipo K y la placa Arduino UNO como plataforma hardware y software para controlar su funcionamiento.

La temperatura medida se ha de mostrar cada segundo a través de un display de tipo LCD.

El rango de temperatura a medir se establece desde $-100\text{ }^{\circ}\text{C}$ hasta $630\text{ }^{\circ}\text{C}$, mientras que el rango de temperatura de uso que soportará el termómetro va desde $0\text{ }^{\circ}\text{C}$ hasta $50\text{ }^{\circ}\text{C}$, limitado por las especificaciones del display.

Se dispondrá de un único pulsador para el apagado y encendido del dispositivo: *ON/OFF*.

Para alimentar el termómetro existirán dos alternativas: se puede utilizar de manera portátil empleando una pila de 9 V o se puede utilizar un adaptador de corriente, que al igual que la pila, ha de ser de 9 V. La conmutación entre un modo y otro de alimentación no ha de afectar al funcionamiento del termómetro, esto es, que el dispositivo no se apague durante la conmutación.

Respecto a la alimentación a través de baterías, el termómetro ha de incluir un sistema que detecte si la carga de la batería es inferior a un límite, a partir del cual, no se asegura un correcto funcionamiento del dispositivo. En este caso, el termómetro continuará funcionando pero se mostrará el mensaje *LOW BATT* por pantalla, que indicará que esas mediciones pueden no ser del todo fiables, ya que se está en un modo de trabajo en el que no se garantiza un correcto funcionamiento.

Es muy importante realizar un diseño que permita un intercambio de termopar de manera fácil ya que en el mercado existen con una gran variedad de características diferentes, necesitando en cada momento unos u otros. Además, los termopares están formados por la unión de dos hilos muy finos de dos metales distintos, por lo que pueden romperse con cierta facilidad, necesitando así un reemplazo rápido para continuar con las mediciones.

Otra característica que ha de tener el termómetro es que sea capaz de detectar la rotura del termopar, en cuyo caso se mostrará el mensaje *ERROR* en el display para evitar lecturas erróneas que puedan confundir al usuario.

Una función interesante a incluir es un registro de los valores máximo, mínimo y medio de una serie de medidas tomadas. Para mostrar estos datos se dispondrá de un pulsador que se llamará "MMA" (*max-min-avg*). Pulsándolo una vez se mostrará por pantalla el valor máximo, pulsándolo una segunda vez se mostrará el valor mínimo y pulsándolo una tercera vez se mostrará el valor medio. Pulsándolo una vez más se volverá a la pantalla

principal en la que se muestra la temperatura medida en cada momento. Si al cabo de 30 segundos no se ha regresado a la pantalla principal, el dispositivo lo hará automáticamente. Cada vez que se apague el dispositivo o se active otro pulsador que se llamará “*CLEAR*”, el registro se borrará.

Una opción más a incluir, muy típica en dispositivos comerciales de este tipo, es la de mantener el dato que se está leyendo en un momento dado fijo en pantalla durante un cierto tiempo. Para activar esta opción se dispondrá de un pulsador que se llamará “*HOLD*”. Si se pulsa, el dato mostrado en ese instante quedará fijo en pantalla hasta que se vuelva a activar el pulsador o transcurra un tiempo de 30 segundos, para retornar a la pantalla principal.

Destacar que durante este tiempo en que se muestra un dato fijo o los valores máximo, mínimo y medio, el termómetro seguirá registrando datos internamente aunque no los pueda mostrar por pantalla.

Ha de implementarse también un sistema que apague el dispositivo automáticamente al cabo de 5 minutos desde la última vez que se activó alguno de los pulsadores del termómetro. Así se pretende ahorrar energía en los casos en que el usuario olvide que dejó el dispositivo encendido, sobre todo cuando se está trabajando con baterías. Además se ha de contemplar una opción en la que se desactive ese modo de apagado automático para casos donde, por ejemplo, se quieran hacer registros de datos durante un tiempo mayor a esos 5 minutos. Esta desactivación se realiza entrando en modo *HOLD* y a continuación pulsando el botón *CLEAR*. Inmediatamente se mostrará un mensaje de desactivación del apagado automático y se volverá a la pantalla principal. Se ha de tener en cuenta que, cada vez que se apague el termómetro, el modo de auto-apagado volverá a ser activado.

El dispositivo ha de ser manejable y fácil de usar, con un aspecto similar a los termómetros comerciales de este tipo. Con todos los componentes incluidos en una caja, ha de tener un lugar accesible en el que se pueda intercambiar la pila si ésta se agota. También ha de tener accesible la lectura de datos en el display, así como los distintos pulsadores para todas las funciones planteadas: *ON/OFF*, *MMA*, *HOLD* y *CLEAR*. Por último, ha de incluir el *jack* para el adaptador de corriente y un conector que permita el intercambio de termopares.

3.2. Arduino

Arduino es una plataforma hardware y software basada en los microcontroladores comerciales *ATMEL AVR* que se presenta en una placa *PCB*, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

Permite desarrollar programas interactivos, leyendo señales de una gran variedad de sensores y pulsadores, controlando gran variedad de luces, motores, actuadores, etc. Los proyectos desarrollados con Arduino pueden funcionar solos, o comunicarse con distintos tipos de software (ej. *Flash*, *Processing*). Las placas se pueden montar manualmente o comprarlas ya

montadas. El software de código abierto puede descargarse de manera gratuita.

De todas las placas de Arduino, la que se va a usar en este diseño es la Arduino UNO revisión 3 [13]. Está basada en el microcontrolador ATmega328 [17] de 8 bits. Tiene 14 entradas/salidas digitales de las cuales 6 se podrían usar como salidas PWM. También contiene 6 entradas analógicas.



Figura 3. Vista frontal y trasera de la placa Arduino UNO

(Fuente: <http://www.arduino.cc>)

Además, tiene un oscilador de cristal de cuarzo de 16 MHz, una conexión USB, un *jack* de alimentación y botón de *reset*. Contiene todo lo necesario para dar soporte al microcontrolador.

Alimentación

Arduino UNO puede ser alimentado a través de la conexión USB o a través de una fuente externa que bien puede ser un adaptador de corriente o una batería. Esta fuente de tensión externa puede ir conectada al *jack* de alimentación de la placa o entre los pines V_{in} y GND del conector *POWER*.

La placa se puede alimentar externamente con tensiones entre 6 V y 20 V. Si se alimenta por debajo de 7 V se corre el riesgo de que el regulador interno de la placa suministre menos de 5 V, lo cual hace que el funcionamiento no sea estable. Si se sobrepasan los 12 V, el regulador de tensión se puede sobrecalentar y podría dañar la placa.

Para este diseño, se alimentará la placa a través de los pines V_{in} y GND del conector *POWER*. Además, se aprovechará la tensión de 5 V que suministra la propia placa (pin 5V) para alimentar otros componentes del resto del circuito.

Entradas y salidas digitales

Cada uno de los 14 pines digitales en Arduino UNO puede ser usado como entrada o salida, configurándolos de forma adecuada. Funcionan a una tensión de 5 V y cada pin puede proporcionar o recibir una corriente máxima de 40 mA. A mayores, algunos de estos pines tienen funciones especiales:

- Comunicación serie: 0 (RX) y 1 (TX): usado para recibir y enviar datos por comunicación serie.
- Interrupciones externas: pines 2 y 3.
- PWM: pines 3, 5, 6, 9, 10 y 11. Proporcionan una salida PWM de 8 bits.
- SPI: 10, 11, 12 y 13. Utilizados para comunicación SPI.
- LED: 13. Incluye un LED en la placa conectado a este pin.

En el caso del termómetro diseñado en este TFG, se utilizarán los pines para la comunicación serie, ya que resulta muy útil a la hora de programar y depurar código. Por otro lado, el resto de pines, desde el 2 hasta el 13 se utilizarán para la comunicación con el display y otras funciones como el encendido y apagado.

Entradas analógicas

Arduino UNO tiene 6 entradas analógicas, desde A0 hasta A5, con 10 bits de resolución (1024 valores diferentes). Por defecto, miden tensiones desde 0 V hasta 5 V, aunque es posible cambiar el límite superior de este rango utilizando el pin llamado *AREF* de la placa.

Para el caso del termómetro que se está diseñando, se utilizarán cuatro de las seis entradas analógicas de la placa, que se explicarán con posterioridad.

Otros pines

Existen otro par de pines en la placa:

- *AREF*: voltaje de referencia para las entradas analógicas.
- *RESET*: señal activa a nivel bajo que reinicia el microcontrolador.

En este caso, se dará un voltaje de referencia a las entradas analógicas a través del pin *AREF*. Esto viene motivado por la intención de minimizar el error en la lectura de la señales. La tensión aplicada a través de este pin vendrá suministrada por una referencia de tensión, más precisa que el regulador de tensión que proporciona la tensión de referencia por defecto.

Como dato ilustrativo, la referencia de tensión que se utilizará es la AD780BN de *Analog Devices* [12], que para una tensión de salida de 3 V tiene un error máximo de ± 0.001 V. Mientras que el regulador de tensión que trae Arduino por defecto, LP2985 de *Texas Instruments* [26], tiene un error de hasta ± 0.2 V. Como se muestra en el apartado 5.2.1, el error en la lectura de la señal del termopar, suponiendo ésta lineal, para esta última es de ± 1.49 °C, mientras que para la referencia de 3 V es de ± 1.426 °C. El error apenas difiere 0.07 °C, pero la segunda opción es más estable, es decir, el error del convertidor varía menos respecto del error sin tener en cuenta las variaciones de la tensión de referencia.

3.3. Display

El display utilizado es el PC1602-H de *Powertip* [21]. Es un display LCD de dos filas y 16 columnas o caracteres por fila. Tiene un consumo de corriente bajo (1 mA) que favorece el ahorro de energía, sobre todo si se usa el termómetro de manera portátil.



Figura 4. Display PC 1602-H de Powertip

(Fuente: <http://kamami.com>)

Utiliza el conocido controlador para displays LCD Hitachi HD44780 [22]. La interfaz de control es estándar para este tipo de displays e incluye caracteres ASCII, caracteres japoneses Kana y algunos símbolos, en dos líneas de hasta 28 caracteres por línea.

El uso de este controlador está limitado a displays de texto monocromo. Los displays compatibles pueden tener diferentes configuraciones: desde una fila de ocho caracteres (8x1) hasta cuatro filas de cuarenta caracteres cada una (40x4) si se utilizan dos módulos con chip de expansión. Algunos displays pueden incorporar iluminación que puede ser de tipo LED, fluorescente o electroluminiscente. El display usado en este diseño no dispone de esta última opción.

Este tipo de displays utiliza una interfaz de conexiones de 16 contactos. Los que no disponen de iluminación incluirán solo 14 conexiones:

- GND (0 V)
- Vcc (+5 V)
- Ajuste de contraste de los caracteres (VO): GND.
- Selección de registro (RS).
- Lectura/escritura (R/W).
- Enable.
- Bit 0
- Bit 1
- Bit 2
- Bit 3
- Bit 4
- Bit 5
- Bit 6
- Bit 7

- Iluminación (+)
- Iluminación (-)

La interface del *HD44780* soporta dos tipos de operación según el número de conexiones utilizadas, 8-bits y 4-bits. Utilizar el modo de 4-bits es más complejo, ya que necesita dos operaciones para mandar la misma información, aunque reduce a la mitad el número de conexiones necesarias. En este caso se utilizarían las conexiones desde el Bit 4 al Bit 7.

Existe un juego de instrucciones para controlar el display, aunque en este TFG se va a controlar a través de Arduino, que incorpora una librería específica con funciones mucho más sencillas de manejar. El modo de operación va a ser el de 4-bits ya que se necesitarán el resto de entradas y salidas digitales de Arduino para otras funciones.

3.4. Acondicionamiento de la señal del termopar

El diseño realizado para la tarea del acondicionamiento de la señal del termopar se muestra en el esquemático de la Figura 5:

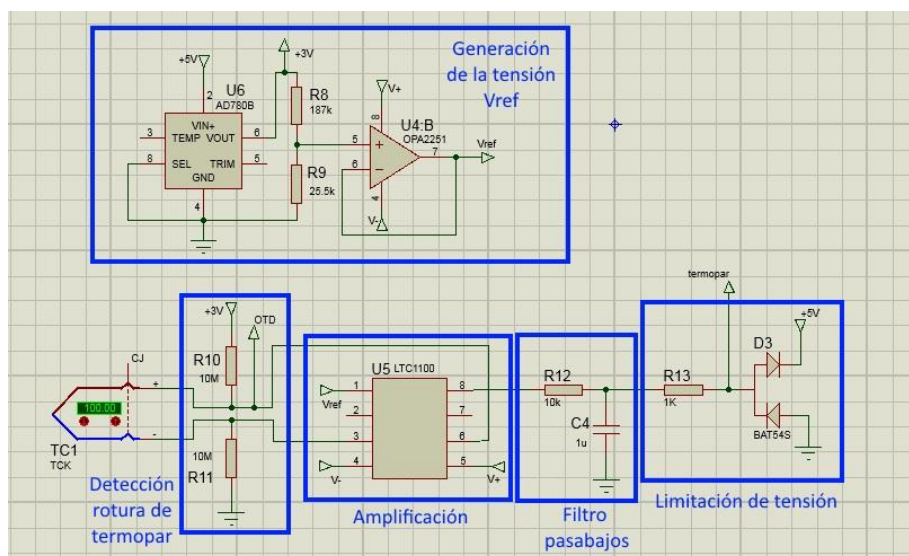


Figura 5. Esquemático del circuito de acondicionamiento de la señal del termopar

En este esquemático podemos ver que el acondicionamiento se puede dividir en varias partes, recuadradas en azul, que se explicarán con más detalle a continuación.

3.4.1. Amplificación

Para el rango de temperatura especificado a medir, el termopar, para $-100\text{ }^{\circ}\text{C}$, suministrará $-3,554\text{ mV}$, mientras que para $630\text{ }^{\circ}\text{C}$ suministrará $26,19\text{ mV}$. De manera global, la tensión a medir será en total $29,744\text{ mV}$.

Debido a que el rango de tensión para las entradas analógicas de Arduino va a ser de 0 V a 3 V , se amplificará esta tensión del termopar con una

ganancia de valor 100. Al suministrar el termopar una tensión diferencial muy pequeña, no se puede amplificar su señal con un amplificador operacional al uso, si no que se ha de usar un amplificador de instrumentación [2] con el objetivo de minimizar errores.

El amplificador seleccionado para este efecto es el LTC1100CN de *Linear Technology* [19]. Entre sus características más importantes se encuentran:

- Ganancia diferencial de valor 100 fijada internamente, sin uso de resistencias externas para minimizar errores debidos tanto a la propia tolerancia de las resistencias, como a la diferencia de temperatura entre la resistencia externa y las internas que configuran la ganancia.
- Baja tensión de offset de entrada de $\pm 1 \mu\text{V}$ y baja deriva de la tensión de offset con la temperatura $\pm 5 \text{ nV}/^\circ\text{C}$, valores típicos a 25°C .
- Bajo error de ganancia de $\pm 0.01\%$ y baja no linealidad de la misma de 12 ppm, valores típicos.
- Alto coeficiente de rechazo de modo común de 110 dB.
- Amplio rango de tensión de alimentación (de V+ a V-): 18 V.
- Consumo de corriente: 2 mA.
- Rango de tensión de entrada: (V+ +0.3 V, V- -0.3 V).
- Rango de tensión de salida: (V+ -0.3 V, V- +0.3 V).
- Rango de tensión de entrada de modo común: (V+ -2 V, V-)

Al tener el amplificador una ganancia de valor 100, con el rango ya indicado de la tensión suministrada por el termopar, a la salida se tendrán tensiones entre -355,4 mV y 2,619 V. Esta señal se leerá desde una de las entradas analógicas de Arduino (AO). Como bien se ha explicado anteriormente, estas entradas sólo leen valores entre 0 V y 3 V, por lo que todos los valores negativos contemplados quedarían fuera del rango de lectura. Para solucionar este inconveniente, el amplificador de instrumentación cuenta con el pin llamado V_{ref} . Su función es sumar la tensión conectada a ese pin a la tensión esperada a la salida del amplificador. Por tanto, necesitaremos generar una tensión de alrededor de 356 mV, que sea precisa y estable para intentar evitar errores en la señal de salida, y así evitar esas tensiones negativas que Arduino no podrá leer.

Para ello, se utilizará el mismo circuito de referencia de tensión AD780BN mencionado en la configuración de los límites del rango de tensión del convertidor ADC de Arduino.

3.4.2. Circuito generador de la tensión Vref

Para la generación de esta tensión de referencia que tiene que ser estable y precisa se va a utilizar un circuito de referencia de tensión que suministra 3 V. A continuación se utilizará un divisor de tensión para obtener los 356 mV deseados.

La referencia de tensión es la AD780BN [12] ya mencionada en el apartado 3.2 de este TFG. Se alimentará con la tensión de 5 V que suministra Arduino, como se puede observar en la Figura 5.

La salida de la misma se conecta a un divisor de tensión del que se extrae la tensión final de 360 mV. Los valores seleccionados para las resistencias (apartado 5.1.1) son de 187 K Ω y 25.5 K Ω , ambas con una tolerancia no mayor de $\pm 0.1\%$.

Para que esa tensión no se vea modificada se conecta a continuación un amplificador operacional en configuración de *buffer*, aislando así el divisor de tensión para que no se vea afectado. El amplificador operacional utilizado para este propósito es el OPA2251PA de *Texas Instruments* [27]. Sus principales características son:

- Consumo de corriente muy bajo: 25 μ A
- Doble canal
- Amplio rango de tensión de alimentación: ± 15 V
- Baja tensión de offset de entrada de ± 100 μ V y baja deriva de la tensión de offset con la temperatura ± 0.6 μ V/ $^{\circ}$ C, valores típicos.
- Alto coeficiente de rechazo de modo común de 106 dB.
- Rango de tensiones de modo común: (V+ -0.8 V, V- -0.2 V).
- Salida *rail-to-rail*.

3.4.3. Circuito de detección de rotura de termopar

Dentro de las especificaciones del termómetro se encuentra la detección de rotura del termopar, *OTD* (*open thermocouple detection*). En la Figura 5 se muestra el diseño realizado.

Consiste en colocar una resistencia de un valor muy alto (10 M Ω) entre la entrada diferencial positiva del amplificador y una tensión determinada, en este caso de 3 V, y otra resistencia de igual valor entre la entrada diferencial negativa y tierra. Estas resistencias tienen que tener un valor alto para que la señal del termopar no se vea afectada.

Con esta configuración se genera una tensión en ambas entradas diferenciales del amplificador, de valor la mitad de los 3 V aplicados, y que hay que tener en consideración ya que el amplificador debe tolerar una tensión de modo común de ese valor. En el caso del amplificador de instrumentación LT1100, el rango de tensiones de modo común aceptables para los valores de alimentación del circuito +8 V y -1 V, (+6.2 V y -0.77 V con la batería en su límite de descarga), va desde +6 V hasta -1 V (+4.2 V hasta -0.77 V), quedando la tensión de modo común de 1.5 V dentro del mismo.

Para detectar si el termopar se ha roto se lee la tensión de la entrada diferencial positiva a través de una entrada analógica de Arduino, de manera que si el termopar está en perfecto estado, la tensión esperada será de 1.5 V, mientras que si el termopar se rompe, la tensión leída será de 3 V.

3.4.4. Filtrado de la señal amplificada del termopar

La señal generada por un termopar, al ser tan pequeña, es muy susceptible al ruido por lo que es conveniente añadir un filtrado que elimine señales de alta frecuencia. Para ello es muy común utilizar un filtro pasabajos, que se configura como se muestra en la Figura 6:

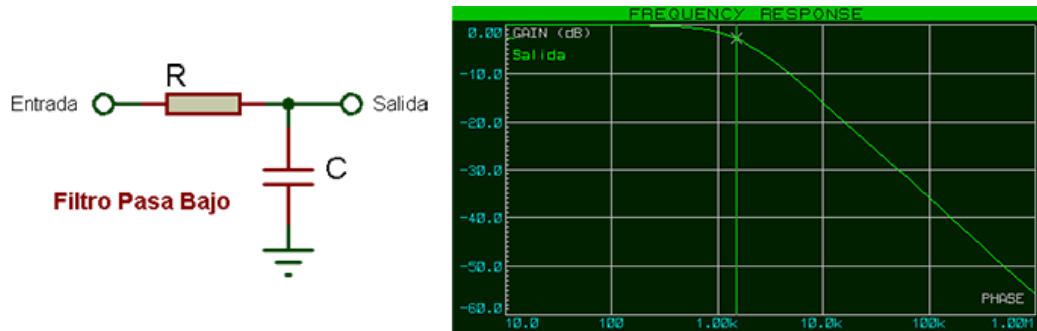


Figura 6. Esquema típico de un filtro pasabajos

(Fuente: <http://gzalo.com/filtrorc>)

Se ajustarán los valores, tanto de la resistencia como del condensador, para obtener una frecuencia de corte, rechazando las señales con frecuencias superiores a ésta. Los valores obtenidos en el apartado 5.1.2 son de 10 K Ω para la resistencia y 1 μ F para el condensador.

3.4.5. Limitación de la tensión en las entradas analógicas

Según las especificaciones del microcontrolador utilizado en Arduino UNO, ATmega328, las tensiones límite admitidas en las entradas analógicas son su tensión de alimentación ± 0.5 V. Como el microcontrolador se alimenta con los 5 V regulados en la propia placa de Arduino, los valores límite serán -0.5 V y 5.5 V. Para evitar que por saturación de los distintos amplificadores se sobrepasen esos límites, se ha configurado el pequeño circuito limitador que se muestra en la Figura 5.

Consiste en una resistencia en serie con la señal a limitar, que además es común al ánodo del diodo conectado a +5 V, y al cátodo del diodo conectado a tierra. Si la tensión de entrada al circuito limitador es mayor de 5 V, el diodo conectado a +5 V conducirá, haciendo que a la salida del circuito se tengan 5.4 V (suma de la tensión +5 V y la caída de tensión directa del diodo). Por el contrario, si la tensión de entrada cae por debajo de los 0 V, será el diodo conectado a tierra el que comience a conducir, haciendo que a la salida haya -0.4 V (0 V menos la tensión de caída directa del diodo). La resistencia de 1 K Ω sirve para limitar la corriente que se le pide al amplificador cuando los diodos entran en funcionamiento. Los cálculos necesarios para seleccionar el valor de esta resistencia se muestran en el apartado 5.1.3.

Se utilizan diodos *schottky* ya que tienen una caída en directa menor que la de los de propósito general. Además el modelo elegido, BAT54S de Vishay

[30], ya viene con los dos diodos en un mismo integrado y conectados de la forma que se indica en el esquemático.

3.5. Acondicionamiento de la señal del LM35

Este componente será el encargado de medir la temperatura de la unión de referencia del termopar para después realizar la compensación por software de la unión fría y obtener la temperatura absoluta medida.

El esquemático del diseño propuesto se muestra en la Figura 7.

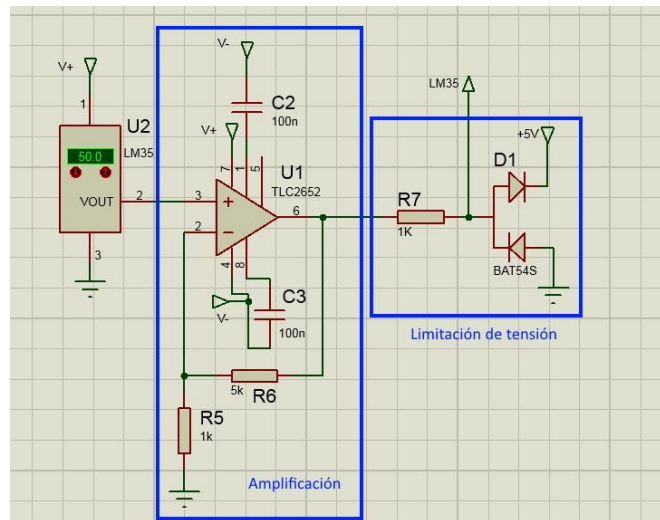


Figura 7. Circuito diseñado para el acondicionamiento de la señal del sensor LM35

A continuación se detalla cada una de las partes marcadas en el esquemático.

3.5.1. LM35

El circuito integrado LM35 de *Texas Instruments* [25] es un sensor de temperatura de precisión que genera una salida en tensión linealmente proporcional a la temperatura medida: 10 mV/°C. El error máximo de precisión a 25 °C es de ± 0.5 °C.

A diferencia de otros integrados con la misma función, éste tiene la ventaja de que está calibrado en °C en vez de en °K, evitando complicaciones en la cambio de unidades.

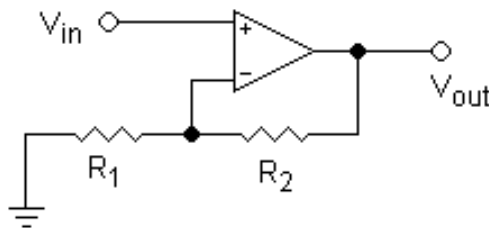
Puede medir temperaturas desde -55 °C y 150 °C, aunque en este caso sólo se necesitará medir entre 0 °C y 50 °C.

3.5.2. AMPLIFICACIÓN

Tal y como se ha explicado, el rango de temperaturas a medir con el sensor LM35 será entre 0 °C y 50 °C, obteniendo a la salida una tensión entre 0 V y 500 mV. Esta tensión ha de ser leída en una de las entradas analógicas de

Arduino (A2). Como se pueden leer tensiones desde 0 V hasta 3 V podremos amplificar esta señal en un factor de 6.

Para ello, se utilizará un amplificador operacional en la configuración no inversora [5] como el mostrado en la Figura 8. Los valores de las resistencias R_1 y R_2 se seleccionan de tal forma que la ganancia sea 6, como se muestra en el apartado 5.1.5. La tolerancia de las mismas no ha de ser mayor de $\pm 0.1\%$.



$$V_{out} = V_{in} \cdot \left(1 + \frac{R_2}{R_1}\right) \quad (5)$$

Figura 8. Configuración no inversora del amplificador operacional

(FUENTE: https://es.wikipedia.org/wiki/Amplificador_operacional)

En este caso se ha seleccionado el amplificador operacional TLC2652AC de *Texas Instruments* [28]. Las características más importantes de éste son:

- Baja tensión de offset de entrada de $\pm 1 \mu\text{V}$ y baja deriva de la tensión de offset con la temperatura $\pm 0.03 \mu\text{V}/^\circ\text{C}$, valores máximos.
- Alto coeficiente de rechazo de modo común de 140 dB.
- Amplio rango de tensión de alimentación: $\pm 8 \text{ V}$
- Consumo de corriente: 2,5 mA, valor máximo.
- Rango de tensión de modo común: $[(V+) - 1.9 \text{ V}, (V-)]$.
- Rango de tensión de salida: $[(V+) - 0.3 \text{ V}, (V-) + 0.3 \text{ V}]$.

Las especificaciones del integrado recomiendan el uso de dos condensadores de valor entre $0.1 \mu\text{F}$ y $1 \mu\text{F}$. Como se puede observar en la Figura 7, los condensadores seleccionados tienen un valor de 100 nF .

3.5.3. Limitación de la tensión en las entradas analógicas

Al igual que para la lectura de la señal del termopar, se añadirá un circuito limitador de tensión en la entrada analógica. En este caso, se repite el mismo diseño con el objetivo de evitar tensiones fuera del rango permitido por Arduino.

3.6. Alimentación asimétrica-tierra virtual

Dentro de las especificaciones del termómetro está la de alimentar el mismo a una tensión de 9 V, a través de una batería o de un adaptador de corriente.

Según el diseño realizado para la adquisición de las señales, se necesita alimentar los amplificadores operacionales con tensiones positiva y negativa. Los valores definidos han sido, para la tensión positiva +8 V, y para la negativa -1 V, al alimentar con el adaptador de corriente o con la pila cargada, y +6.2 V y -0.77 V en el límite de descarga de ésta. Estos valores se han seleccionado de forma que el uso que se da de los amplificadores esté dentro de las especificaciones de los mismos.

En el caso del amplificador de la señal del termopar, se van a tener señales positivas y negativas en la entrada debido a que se van a medir temperaturas por encima y por debajo de cero, desde -3,554 mV hasta 26.19 mV. Según el rango de tensiones de entrada permitido expuesto en el apartado 3.4.1 [(V+) + 0.3 V, (V-) - 0.3 V)], se cumplen las especificaciones. Por otro lado, a la salida se tendrán tensiones entre 0 V y 3 V, valores que cumplen las especificaciones ya que el rango de tensiones de salida permitido es [(V+) - 0.3 V, (V-) + 0.3 V].

En el caso del amplificador de la señal del LM35, la señal de entrada de modo común podrá tomar valores desde 0 V hasta 500 mV, cumpliéndose las especificaciones ya que el rango permitido es [(V+) -1.9 V, (V-)]. La señal de salida podrá tomar valores de 0 V a 3 V, cumpliéndose también las especificaciones para este amplificador [(V+) - 0.3 V, (V-) + 0.3 V] según el apartado 3.5.2.

En el caso del amplificador utilizado como buffer en la generación de la tensión V_{ref} , la tensión de entrada de modo común y la tensión de salida será de 360 mV. En este caso también se cumplen las especificaciones del amplificador al ser el rango de tensión de modo común a su entrada [(V+) - 0.8 V, (V-) - 0.2 V] y tener una salida rail-to-rail (apartado 3.4.2).

Otra restricción a la hora de seleccionar las tensiones de alimentación es la de alimentar Arduino por encima de los 6 V que marcan sus especificaciones. Además, teniendo en cuenta que con la descarga de la batería la tensión positiva irá disminuyendo, se ha optado por un rango en el que esta tensión positiva sea la máxima posible considerando una tensión negativa suficiente para el correcto funcionamiento de los amplificadores.

El diseño realizado para obtener esta tensión asimétrica es un circuito de tierra virtual, como se muestra en la Figura 9:

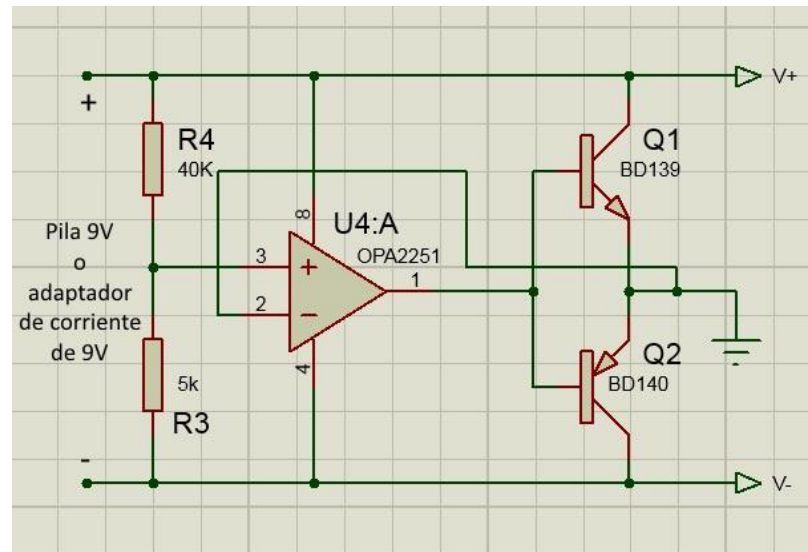


Figura 9. Esquemático del circuito de tierra virtual para la alimentación asimétrica

La tensión de alimentación general de todo el circuito se divide a través del divisor de tensión, generando una tensión de 8 V entre los extremos de la resistencia R4 y 1 V en la resistencia R3 (suponiendo alimentado a 9 V) como se explica en el apartado 5.1.7. Para que el resto del circuito no afecte al divisor de tensión, se aísla éste a través de un amplificador operacional en configuración de buffer. Para ello se utilizará el otro canal disponible del amplificador OPA2251 de la generación de la tensión V_{ref} .

El termómetro al menos va a consumir 50 mA (consumo de Arduino) lo que hace necesaria la utilización de los dos transistores [9] a continuación del buffer. Los amplificadores operacionales a su salida no proporcionan en general tanta corriente. En el caso del OPA2251 elegido, la máxima corriente que proporciona es de 10 mA.

3.7. Circuito de detección de batería baja

Se considerará que la carga de la batería está en un nivel bajo cuando la tensión $V+$ del circuito tome valores por debajo de 6.2 V aproximadamente. Este valor límite de tensión viene restringido por la placa de Arduino al no poder alimentarlo con tensiones por debajo de los 6 V.

Se leerá esta señal $V+$ a través de una entrada analógica de Arduino (A3) para determinar si está por encima o debajo del límite establecido. Arduino sólo acepta valores de tensión entre 0 y 3 V, por lo que se habrá de adaptar esta señal a ese rango. Para ello se dispone de un divisor de tensión tal y como se muestra en la Figura 10. La selección de las resistencias R15 y R16 se explica en el apartado 5.1.8

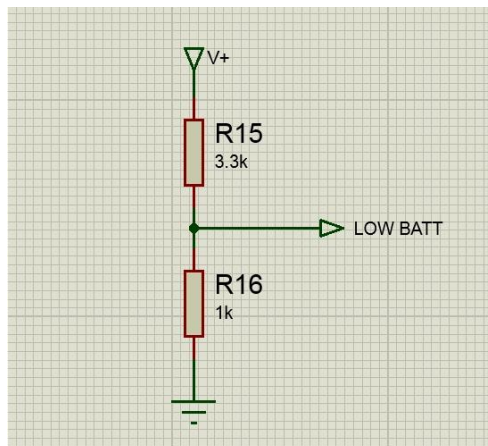


Figura 10. Esquemático del circuito de detección de batería baja

3.8. Circuito de encendido y apagado

Como ya se ha explicado, el encendido y apagado se ha de poder realizar pulsando un mismo botón. Además se ha de incorporar la posibilidad de un apagado automático del dispositivo para ahorrar energía.

El corte de la alimentación durante el apagado del dispositivo se llevará a cabo directamente en la conexión de la pila o el adaptador de corriente, es decir, en la conexión de entrada al circuito de alimentación asimétrica-tierra virtual explicado en el apartado 3.6.

También se podría realizar el corte de la alimentación después de ese circuito, directamente en las tensiones de alimentación de +8 V y -1 V, sobre todo porque estas dos últimas están referenciadas a la tierra virtual de todo el circuito, lo que haría la manipulación de las señales de apagado y encendido más sencilla. Sin embargo, en este caso, el amplificador operacional que hace la función de buffer al generar la alimentación asimétrica, aunque no tuviera carga, sí tendría un pequeño consumo, lo que en sus características es llamado *quiescent current*. En el caso del amplificador seleccionado, este consumo es de 30 μA [27].

Se opta, por tanto, por la primera opción ya que se pretende que el consumo durante el tiempo en que el termómetro esté apagado, sea nulo para así agotar la capacidad de la pila lo más tarde posible.

Para la conexión y desconexión de la alimentación se utiliza un relé de enclavamiento de dos bobinas aisladas, el G6CK-1117P de *Omron* [20]. La razón del uso de un relé es el aislamiento entre la señal de mando del mismo (tensión en la bobina) y sus contactos. Se va a necesitar generar una señal de apagado desde Arduino, bien porque se detecte el pulsador de apagado, bien porque actúe el apagado automático. Esta señal de apagado desde Arduino, referenciada a la tierra virtual del sistema, será la que se aplique a la bobina del relé para controlarlo, mientras que la conexión o desconexión de los 9 V, no referenciados a la tierra virtual, se tendrá que hacer de forma aislada, lo cual permite el relé a través de sus contactos.

Se selecciona un relé de enclavamiento para que las bobinas sólo consuman en el momento de la conmutación y así evitar consumos constantes añadidos a los de los dispositivos ya involucrados en el funcionamiento, que además en el caso de las bobinas de los relés, suelen ser de varias decenas de miliamperios.

El relé es de dos bobinas aisladas. La bobina *SET* cierra el contacto y la bobina *RESET* lo abre. Se utilizará la primera en el encendido del dispositivo y la segunda en el apagado. El contacto del relé es de tipo *SPST*, un único polo y una única vía.

En la Figura 11 se muestra el circuito diseñado:

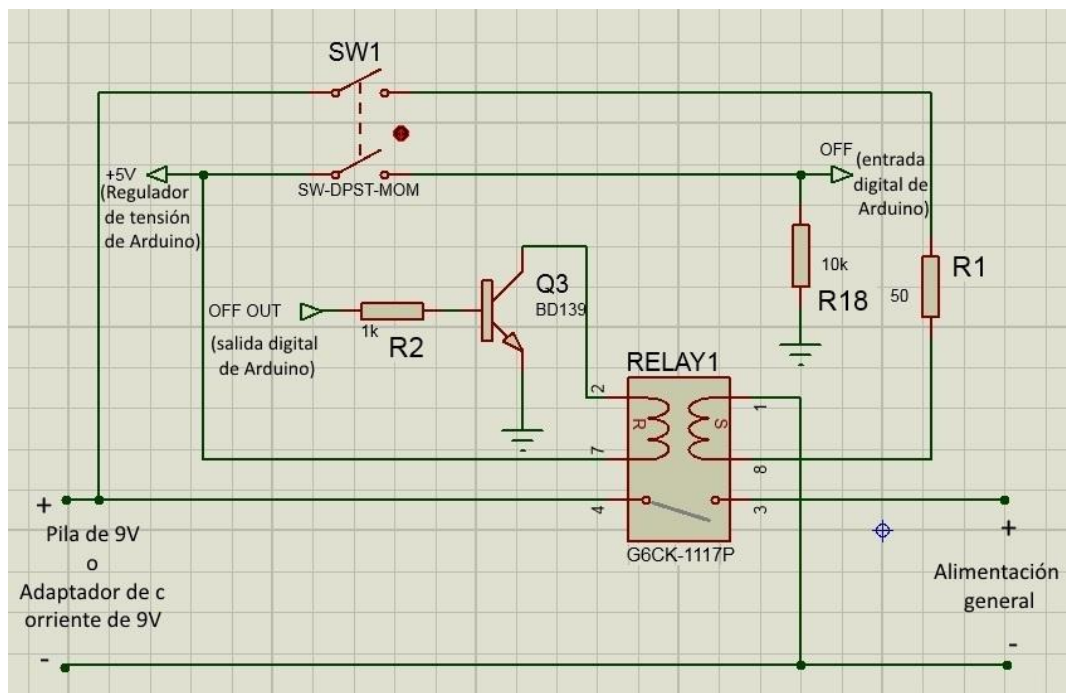


Figura 11. Esquemático del circuito de apagado y encendido

El pulsador de encendido y apagado será de tipo *DPST*, dos polos y única vía. Uno de los polos se usará durante el encendido y el otro durante el apagado.

Para encender el dispositivo se conectará la tensión de la batería o del adaptador de corriente, a través del pulsador, a la bobina *SET* del relé. Como se puede observar se añade una resistencia *R1* para adaptar esa tensión a la tensión permitida en la bobina del relé. La selección del valor de esta resistencia se muestra en el apartado 5.1.9. El contacto del relé se cerrará y por tanto quedará el circuito alimentado.

Para apagar el dispositivo se detectará la activación del pulsador a través de la entrada digital *OFF* de Arduino. Para ello, se conectará al otro polo del pulsador una tensión de 5 V, en este caso obtenida del regulador de tensión que tiene Arduino y que se leerá a través de la entrada digital correspondiente, añadiendo una resistencia *R18* para que la corriente pueda

circular al activar el pulsador. Arduino a través del software tratará esta señal para generar una señal de salida, también digital, correspondiente a la señal *OFF OUT* de la Figura 11. Debido a que las salidas digitales de Arduino sólo pueden suministrar hasta 40 mA y la bobina del relé consume 56 mA a 5 V, se utiliza un transistor para poder amplificar la corriente y activar dicha bobina.

Para apagar el dispositivo de forma automática se utilizará la misma salida digital *OFF OUT*, pero será Arduino internamente el que decida en qué momento activarla.

A través del software se discriminará la detección de la señal *OFF* durante el encendido, ya que se podría dejar pulsado el botón *ON/OFF* durante un cierto tiempo, provocando si no el apagado del termómetro.

3.9. Conmutación entre batería y adaptador de corriente

Como especificación en el diseño del termómetro se indicó que el dispositivo podría ser alimentado a través de una pila de 9 V o a través de un adaptador de corriente, también de 9 V. Además durante la conmutación, el dispositivo no se apagaría ni experimentaría ningún cambio en su funcionamiento.

Para realizar esta conmutación se utiliza un conector para el adaptador de corriente que incorpora una conexión para la batería. El conector internamente conmuta su conexión de polaridad negativa entre la batería o el adaptador de corriente, si éste está desconectado o conectado, respectivamente.



Figura 12. Imagen y esquemático de conexiones del conector del adaptador de corriente

(Fuente: <http://docs-europe.electrocomponents.com/webdocs/0bbc/0900766b80bbc7ce.pdf>)

Según el esquemático de la Figura 13, las conexiones positivas de la pila y del adaptador de corriente son comunes, realizando la conmutación en las conexiones negativas. El polo negativo de la pila se conectará al pin 3 del conector del adaptador de corriente. Mientras el adaptador esté desconectado, internamente el conector une el pin 3 con el pin 1, quedando alimentado el resto del circuito a través de la pila. Si por el contrario se

conecta el adaptador, la conexión entre los pines 1 y 3 se abre, quedando alimentado el circuito a través del adaptador.

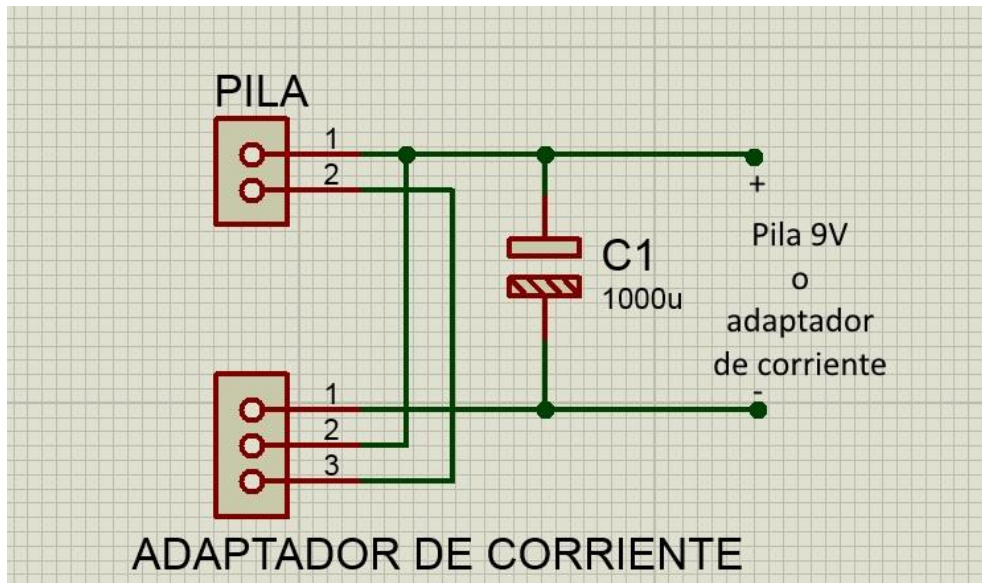


Figura 13. Esquemático del circuito de conmutación entre batería y adaptador de corriente

Se incorpora un condensador entre los extremos de la alimentación del circuito para asegurar que éste no llegue a quedarse sin tensión de alimentación en el transitorio entre la desconexión de la pila y la conexión del adaptador y viceversa. La capacidad del condensador se elige con un valor alto, teniendo en cuenta su tamaño dentro de las dimensiones que permiten la PCB y la caja del termómetro.

3.10. Diseño de PCB

Tanto para el diseño del esquemático y simulación del circuito, como para la creación de la propia PCB se ha utilizado el software *Proteus 8*. *Proteus* es una recopilación de programas de diseño y simulación electrónica desarrollados por *Labcenter Electronics* que consta de dos programas principales, *ISIS* y *ARES*, además del módulo *VSM*.

ISIS, (*Intelligent Schematic Input System*) permite diseñar planos eléctricos de circuitos con componentes muy variados, desde resistencias hasta microcontroladores, incluyendo fuentes de alimentación o generadores de señal.

Una de las prestaciones de *Proteus* es *VSM*, (*Virtual System Modeling*), una extensión integrada en *ISIS*, a través de la cual se pueden simular todas las características de multitud de componentes, entre ellos varias familias de microcontroladores, introduciendo el mismo usuario el programa de control. Además, se pueden simular circuitos con microcontroladores conectados a distintos dispositivos, como motores, LCD's, teclados en matriz, etc. Incluye, entre otras, las familias de PIC's PIC10, PIC12, PIC16, PIC18, PIC24 y dsPIC33.

ARES, (*Advanced Routing and Editing Software*), es la herramienta de ubicación, edición y rutado de componentes. Se utiliza para la fabricación de placas de circuito impreso *PCB*, permitiendo la edición de multitud de parámetros en el diseño de las mismas.

En la Figura 14 se muestra el esquemático completo del diseño realizado. Se puede visualizar con mayor resolución en el Anexo D. A mayores de los esquemáticos ya mostrados anteriormente, se han añadido todos los conectores necesarios para el funcionamiento del termómetro, como son la conexión del termopar, de los distintos botones o la propia conexión con Arduino. Todos los pulsadores, *ON/OFF*, *MMA*, *HOLD* y *CLEAR* van a ir situados fuera de la placa, en una placa pre-perforada sujeta a la carcasa del termómetro, por lo que se han añadido unos conectores para poder conectar sus señales. También se ha incluido un conector para la pila, uno para el display, además de los distintos conectores para Arduino, ya que ambas placas irán conectadas de forma directa a través de unos pines en la *PCB* fabricada que encajarán directamente en los conectores de Arduino.

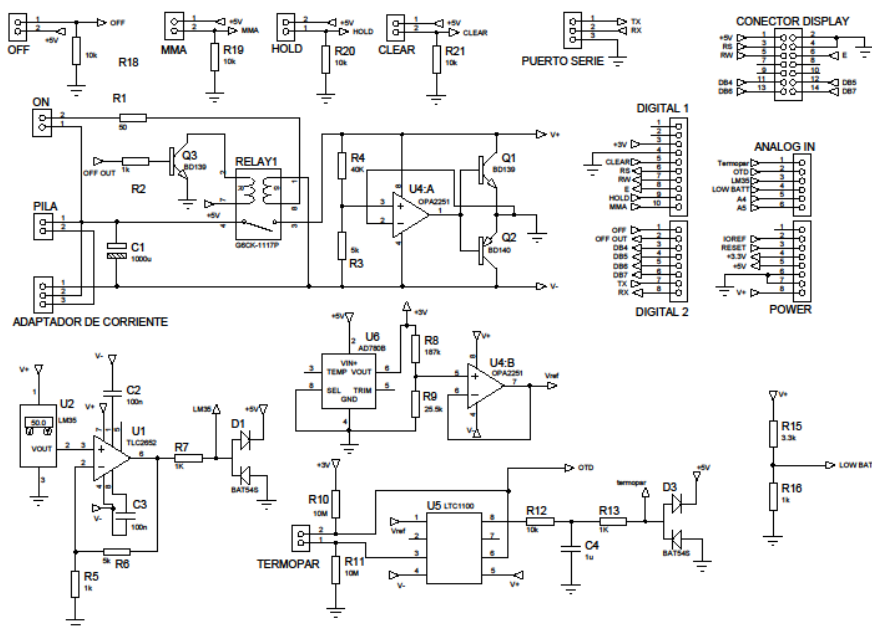


Figura 14. Esquemático completo del circuito diseñado

La *PCB* se diseña con unas medidas determinadas (90 mm x 72 mm) limitadas por la carcasa del termómetro. Será una placa con componentes y rutado en ambas caras.

Los conectores para la placa de Arduino se disponen en las posiciones marcadas por el fabricante. La placa de Arduino irá conectada a la cara inferior de la *PCB*, por lo que los conectores se deberán situar también en esa cara, con los *pad* (área alrededor de cada pin) para la soldadura en la cara superior. Serán los únicos componentes colocados en la cara inferior.

En la Figura 15 se muestra la situación de los componentes en la *PCB*.

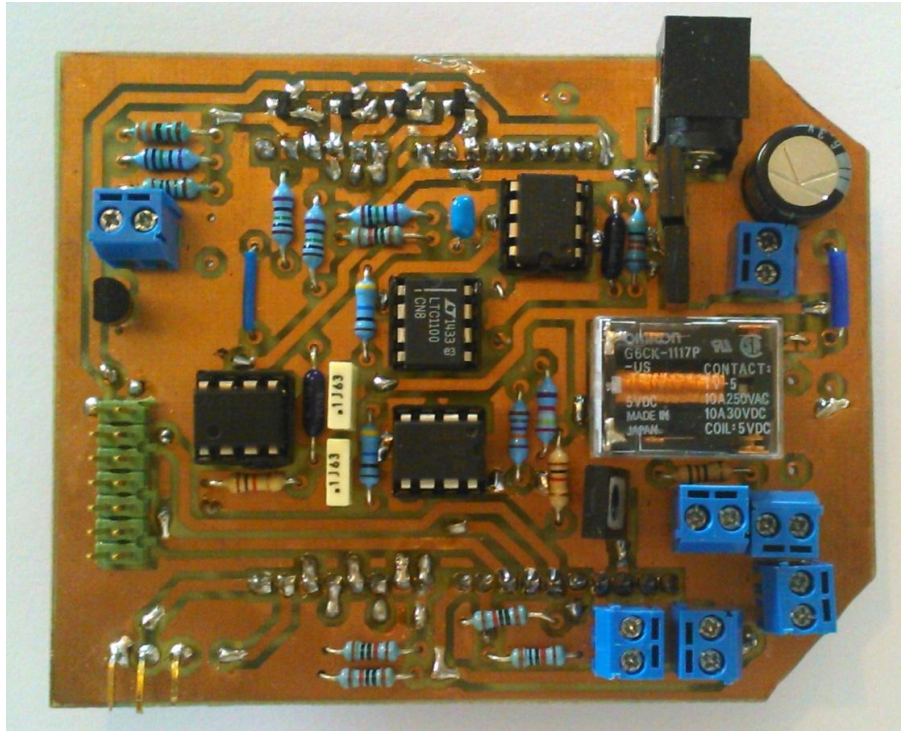


Figura 17. Cara superior de la PCB fabricada

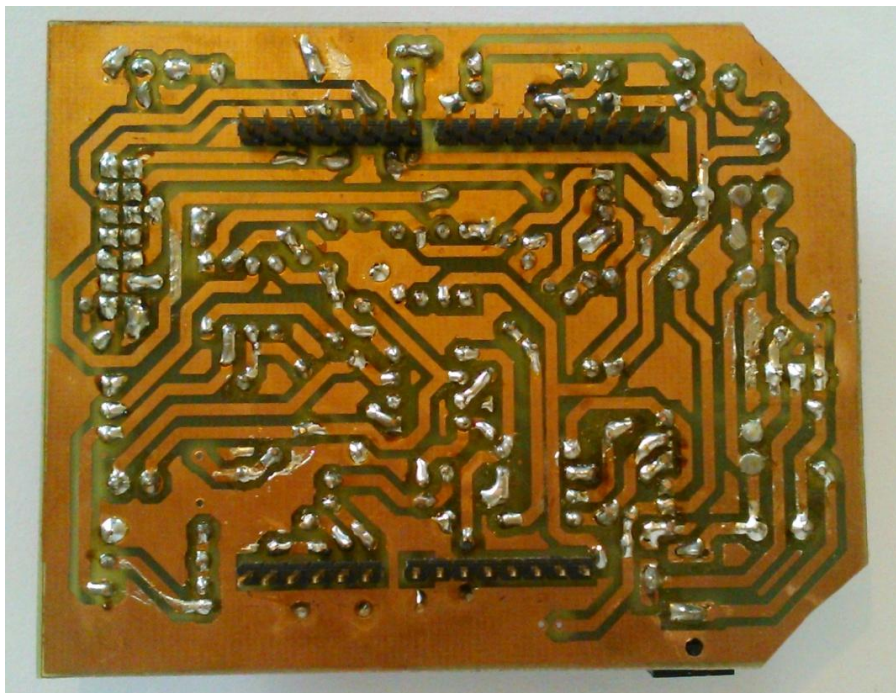


Figura 18. Cara inferior de la PCB fabricada

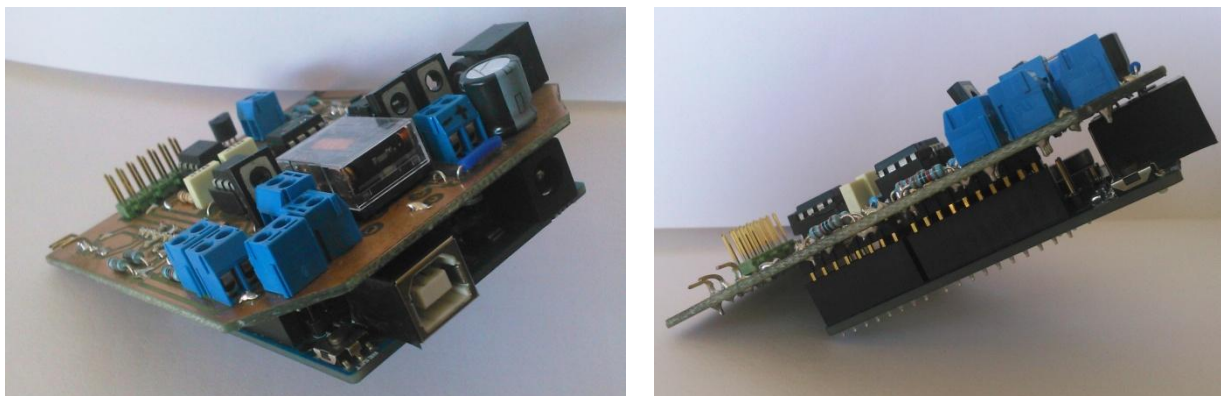


Figura 19. Imágenes de la conexión de la PCB con la placa de Arduino

3.11. Construcción del prototipo

Todos los componentes que forman el termómetro se integran en una caja de material plástico. La cara frontal de la caja (Figura 20), a través de la cual el usuario puede interactuar con el dispositivo, incluye el display y los botones pulsadores *ON/OFF* (encendido y apagado), *CLEAR* (reiniciar valores máximo, mínimo y medio de temperatura y desactivar el apagado automático), *HOLD* (mantener un valor fijo de temperatura en pantalla) y *MMA* (mostrar valores máximo, mínimo y medio de temperatura).



Figura 20. Parte frontal del termómetro con el display y los botones pulsadores

Los botones pulsadores van situados en una pequeña placa que a su vez irá conectada a la principal. El display irá conectado mediante cable plano (Figura 21).

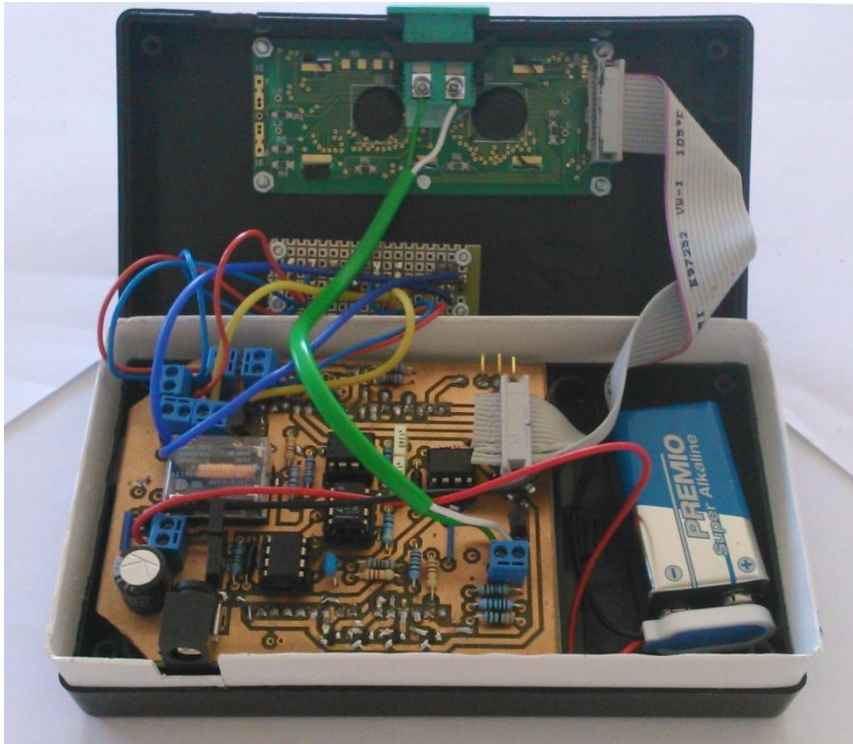


Figura 21. Interior del termómetro

En la parte superior de la caja (Figura 22) se incluye el conector de termopar para facilitar el intercambio del mismo y la conexión para el adaptador de corriente.



Figura 22. Conectores para termopar y adaptador de corriente

Por último, en la parte posterior (Figura 23) se incluye una pequeña tapa que se retira para acceder a la pila y cambiarla con facilidad.



Figura 23. Parte posterior del termómetro

4. Programación

4.1. Descripción del software de Arduino

Para desarrollar aplicaciones con Arduino se utiliza el entorno de programación desarrollado por los mismos creadores de Arduino basado en el entorno de código abierto *Processing* adaptado al lenguaje *Wiring*.

Wiring [32] es una plataforma para prototipos electrónicos de código abierto formada por un lenguaje de programación, un entorno de desarrollo integrado (IDE) y una placa de microcontrolador. El código en *Wiring* se escribe en lenguaje C/C++, aunque el usuario sólo necesita de dos funciones para conseguir un programa que se pueda ejecutar:

- `setup()`: función que sólo se ejecuta una vez al comienzo del programa y que suele ser utilizada para definir la configuración inicial del entorno
- `loop()`: función que se repite continuamente hasta que se deja de alimentar la placa

El entorno de programación de Arduino es de fácil uso para principiantes y lo suficientemente flexible para usuarios avanzados. Contiene un editor de texto para escribir el código, una consola de mensajes, una barra de herramientas con botones con distintas funciones y una serie de menús.

Los archivos de código escrito con Arduino son llamados *SKETCHES* y tienen extensión *.ino*. Incluyen funciones como cortar/pegar y buscar/reemplazar texto. La consola muestra los mensajes de error y otra información durante la compilación o carga del programa.

Para cargar los *SKETCHES* en la placa se utiliza el *bootloader* de Arduino, o gestor de arranque. Éste es un pequeño programa cargado en una parte de la memoria del microcontrolador de la placa de Arduino que permite programar el resto de la memoria utilizando la comunicación serie y por lo tanto sin necesidad de un programador externo.

4.2. Descripción del programa implementado

El programa realizado se estructura en dos partes diferenciadas:

Por un lado, en el programa principal, dentro de la función *loop ()*, que como se explicó en el apartado 4.1 se repite indefinidamente hasta que se desconecta la alimentación de la placa, se leerán las señales procedentes de los distintos pulsadores del termómetro y se llevarán a cabo las tareas asociadas a las funciones de cada uno de ellos.

Por otro lado, se leerán las señales del termopar y del LM35 cada 100 ms aproximadamente, para que al mostrar el valor de temperatura por pantalla cada segundo, se hayan tomado 10 lecturas de cada señal y así trabajar con un valor medio. Para realizar estas lecturas sin afectar a la detección de las

señales de los pulsadores, se utiliza un temporizador interno del microcontrolador (*Timer1*), para generar interrupciones, en este caso, cada 100 ms. Para la configuración y utilización del *timer1* del microcontrolador existe una librería específica creada para Arduino que incluye funciones sencillas que facilitan su uso [15].

Además, Arduino también dispone de otras librerías, como la llamada *LiquidCrystal* [16] que permite controlar displays LCD basados en el controlador *Hitachi HD44780*, que es precisamente el controlador del display utilizado en este TFG (apartado 3.3).

A través del programa implementado también se configurará la tensión de referencia del convertidor analógico-digital para que tome una referencia externa, en vez de la referencia por defecto (apartado 3.2).

4.3. Diagrama de bloques del programa

A continuación se muestra el diagrama de bloques correspondiente al programa principal. Se divide por tramos con diferentes tareas cada uno:

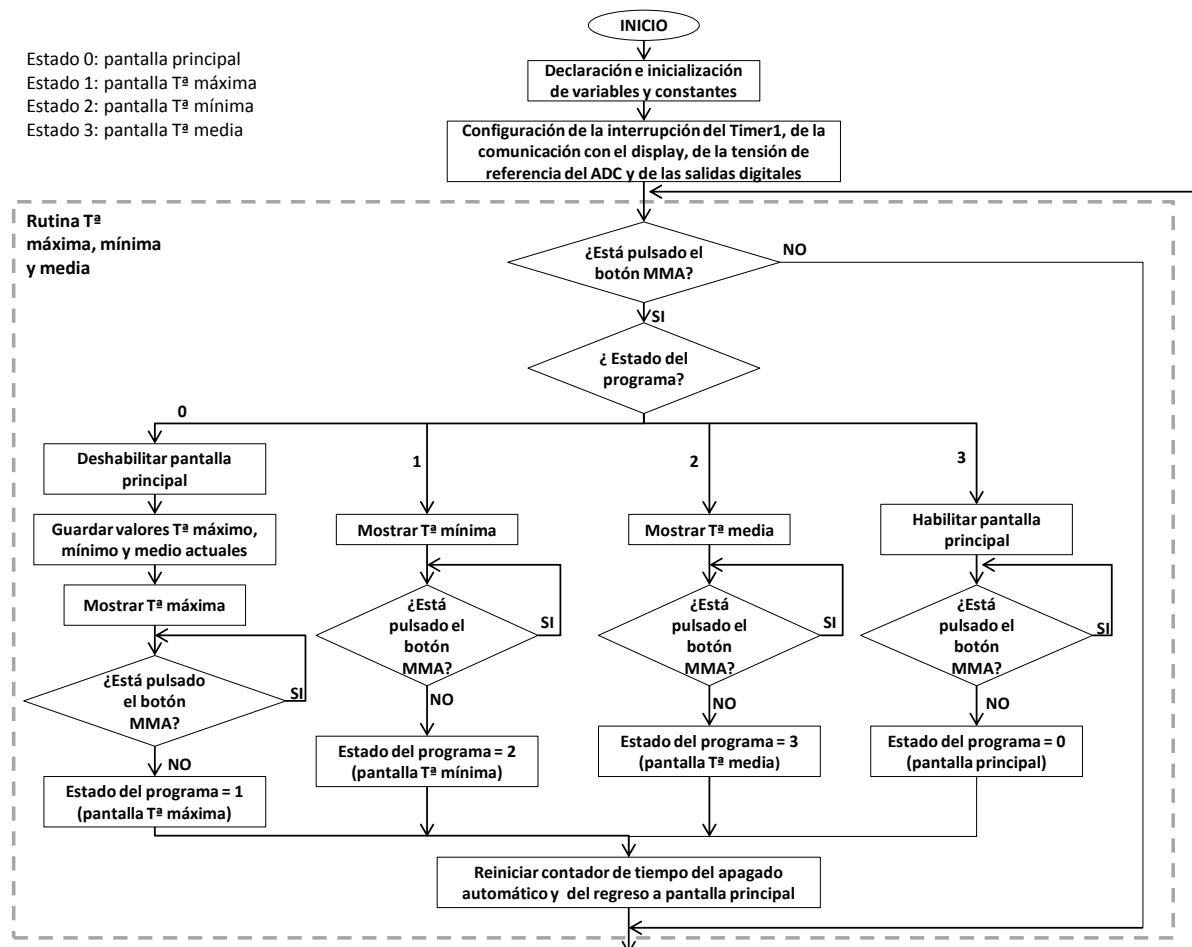


Figura 24. Diagrama de bloques del programa principal: inicialización y rutina de Tª máxima, mínima y media

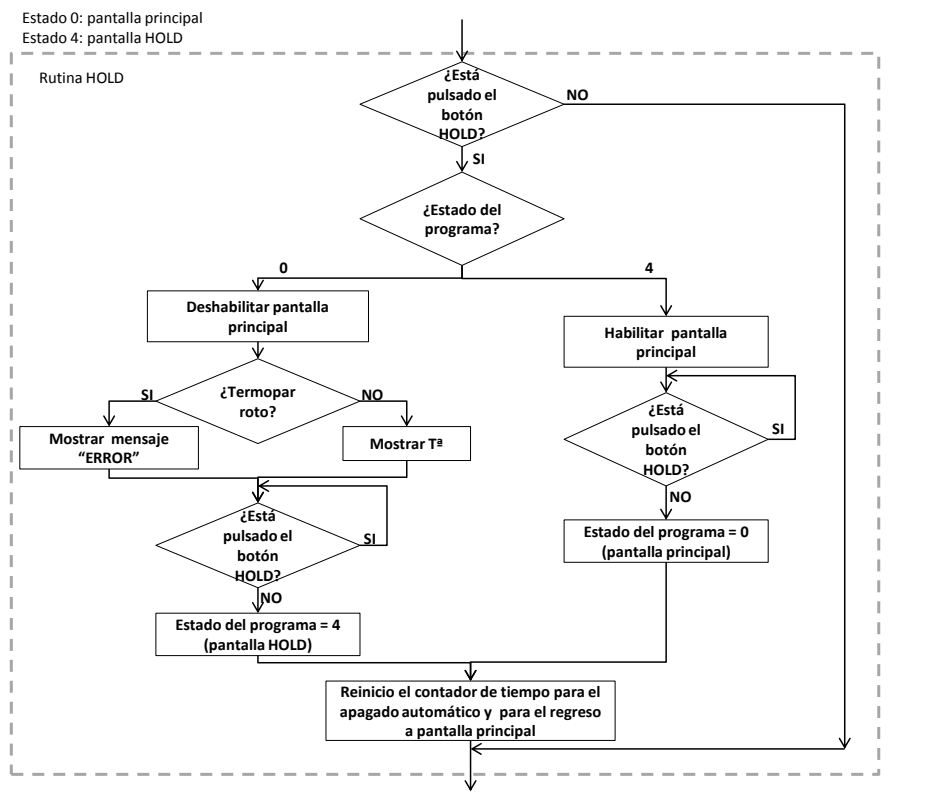


Figura 25. Diagrama de bloques del programa principal: rutina Hold

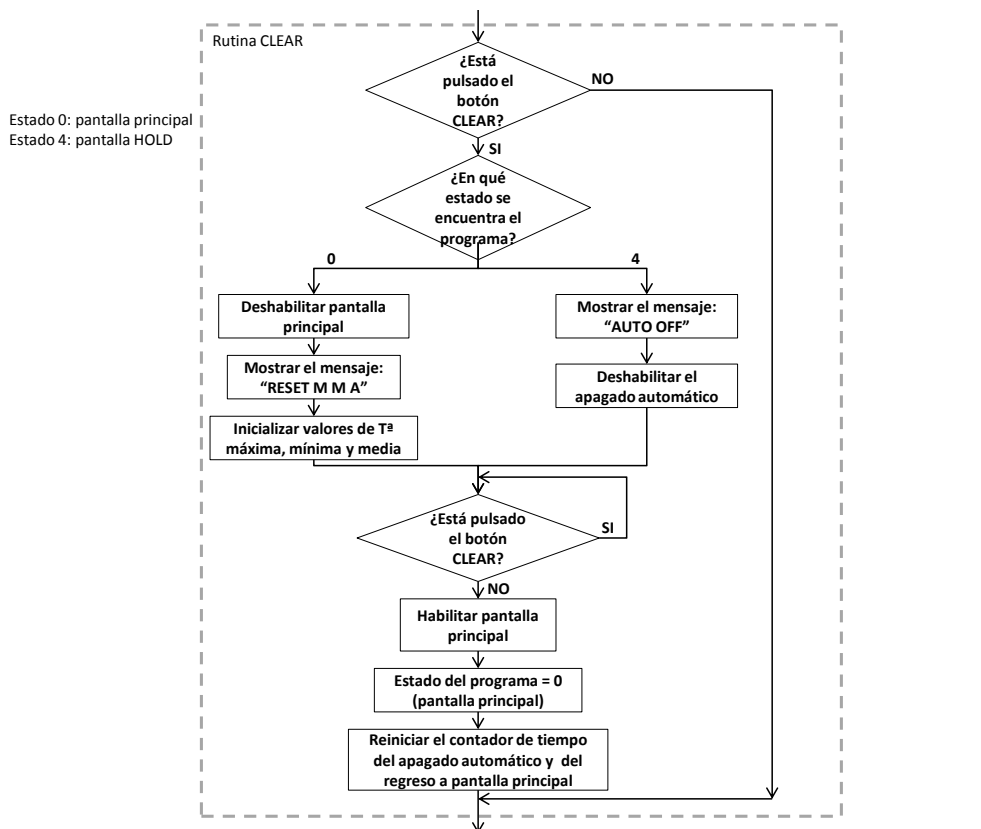


Figura 26. Diagrama de bloques del programa principal: rutina Clear

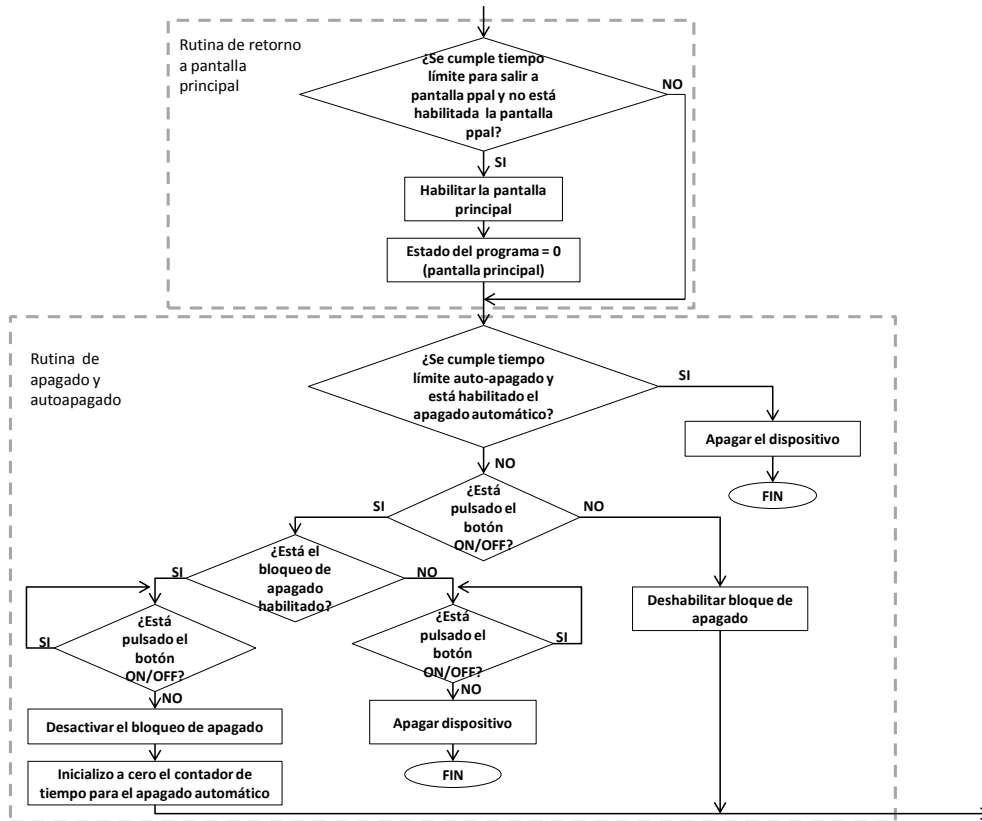


Figura 27. Diagrama de bloques del programa principal: rutina de retorno a pantalla principal por tiempo y rutina de apagado y auto_apagado

A continuación se muestra el diagrama de bloques correspondiente a la interrupción del *Timer1* de Arduino. Se divide por tramos con diferentes tareas cada uno:

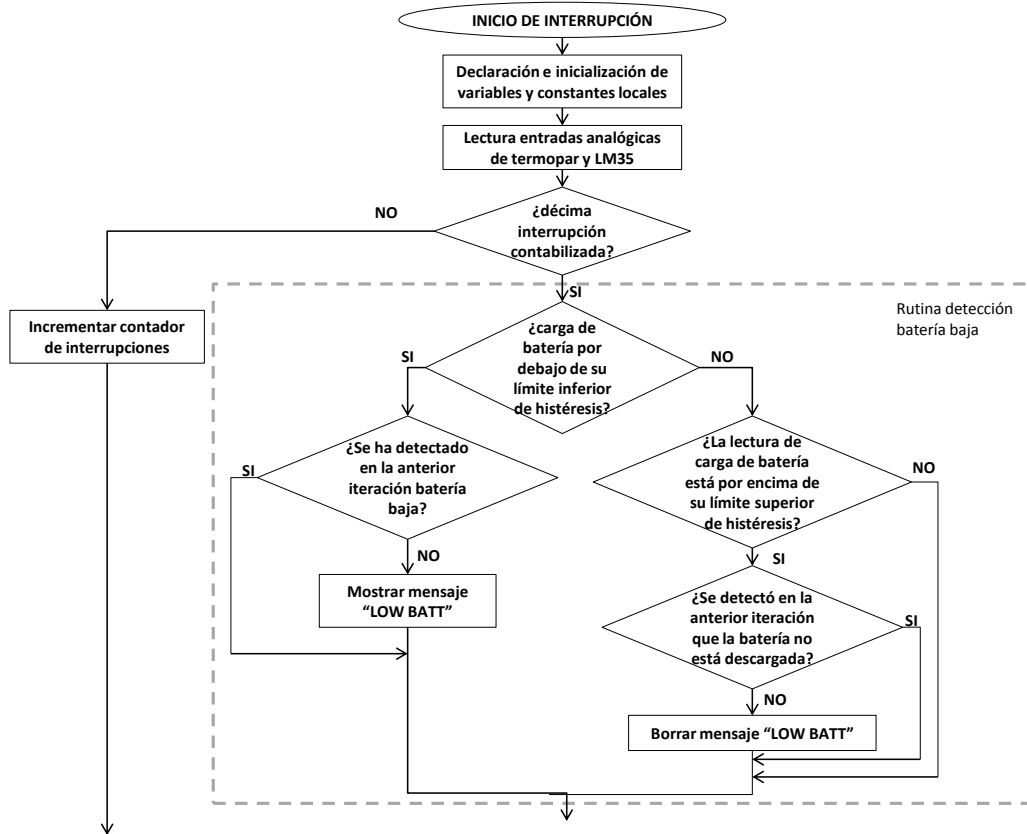


Figura 28. Diagrama de bloques de la interrupción: inicialización de variables, lectura de señales y rutina de detección de batería baja

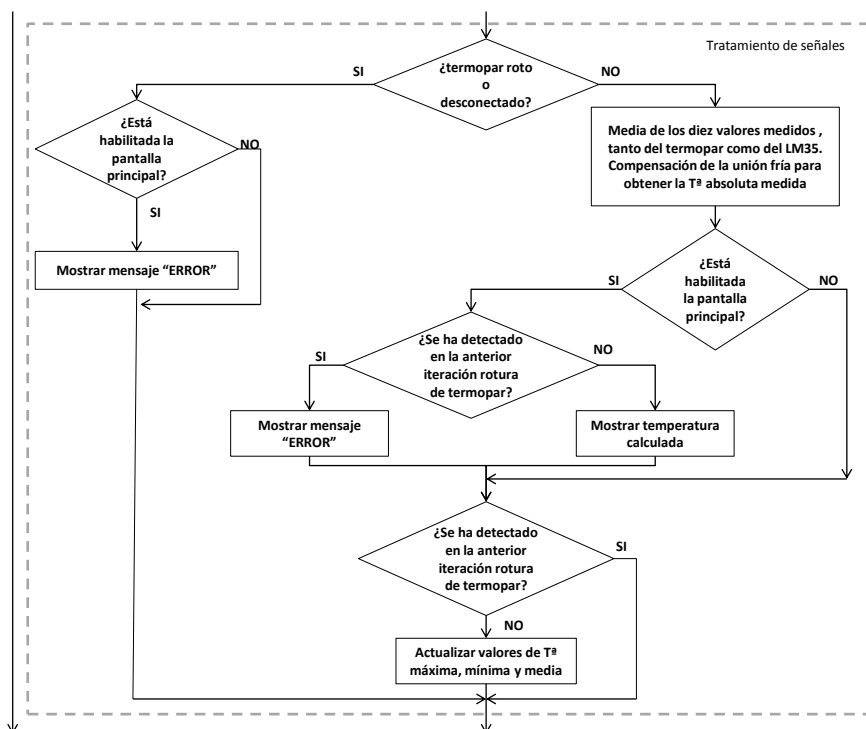


Figura 29. Diagrama de bloques de la interrupción: tratamiento de las señales del termopar y LM35

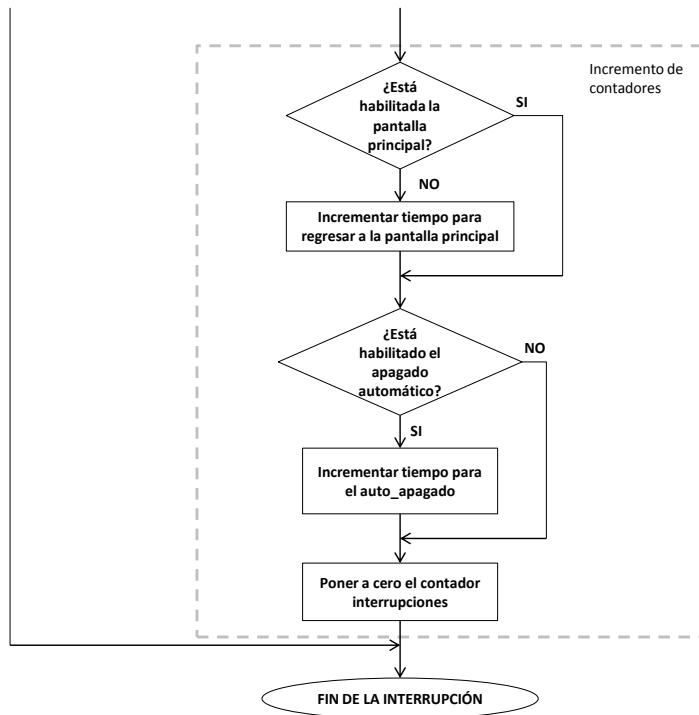


Figura 30. Diagrama de bloques de la interrupción: incremento de contadores temporales.

El código programado en Arduino se puede consultar en el Anexo B de esta memoria.

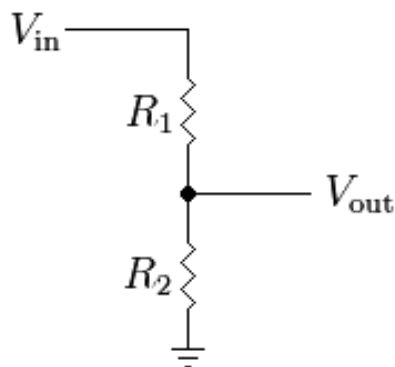
5. Cálculos

5.1. Cálculos de diseño

En este apartado se explicarán todos los cálculos necesarios para el diseño desarrollado en el apartado 3 de este TFG.

5.1.1. Divisor de tensión del circuito Vref

Para obtener esta tensión a partir de los 3 V de la referencia de tensión se utiliza un divisor, como el mostrado en la Figura 31 . La tensión a la salida ha de tener un valor de 360 mV:



$$V_{out} = \frac{R_2}{(R_2 + R_1)} \cdot V_{in} \quad (6)$$

Figura 31. Esquema de un divisor de tensión

(Fuente: https://es.wikipedia.org/wiki/Divisor_de_tensi%C3%B3n)

Se han seleccionado las resistencias de manera que R1 tendrá un valor de 187 KΩ y R2 de 25.5 KΩ para, según la ecuación (6), obtener la tensión de salida necesaria:

$$V_{out} = \frac{25.5 \text{ K}\Omega}{(187 \text{ K}\Omega + 25.5 \text{ K}\Omega)} \cdot 3 \text{ V} = 0.36 \text{ V} \quad (7)$$

5.1.2. Resistencia y condensador del filtro pasabajo

Como ya se mostró en la Figura 6, el filtro pasabajo se configura a través de una resistencia y un condensador. Dependiendo del valor de éstos, se obtendrá una frecuencia de corte determinada, definida como aquella donde la amplitud de la señal entrante al filtro cae hasta un 70.7% de su valor máximo. Así, las señales con frecuencias superiores a la de corte dejarían de afectar a la señal de interés. El cálculo de R y C se realiza a través de la ecuación (8):

$$f_c = \frac{1}{2 \cdot \pi \cdot R \cdot C} \quad (8)$$

El valor de R seleccionado ha sido de 10 K Ω y el de C de 1 μ F, para obtener una frecuencia de corte aproximadamente de 15 Hz, muy baja, ya que se van a medir tensiones en continua.

$$f_c = \frac{1}{2 \cdot \pi \cdot 10 \text{ K}\Omega \cdot 1 \mu\text{F}} = 15.91 \text{ Hz} \quad (9)$$

5.1.3. Resistencia del limitador de tensión de entradas analógicas

Para los casos del acondicionamiento de las señales del termopar y del LM35, se ha diseñado un pequeño circuito para limitar la tensión en las entradas analógicas de Arduino. Dentro del circuito se ha de dimensionar la resistencia en serie con la señal a medir. Ha de tener un valor tal que permita limitar la corriente que se le pide al amplificador cuando los diodos entran en funcionamiento.

En la Figura 32 se muestra el esquema que se tendría si los amplificadores se saturasen a la tensión positiva de alimentación +8 V, que sería la situación más desfavorable ya que si se saturasen a la tensión negativa -1 V, la caída de tensión en la resistencia sería menor.

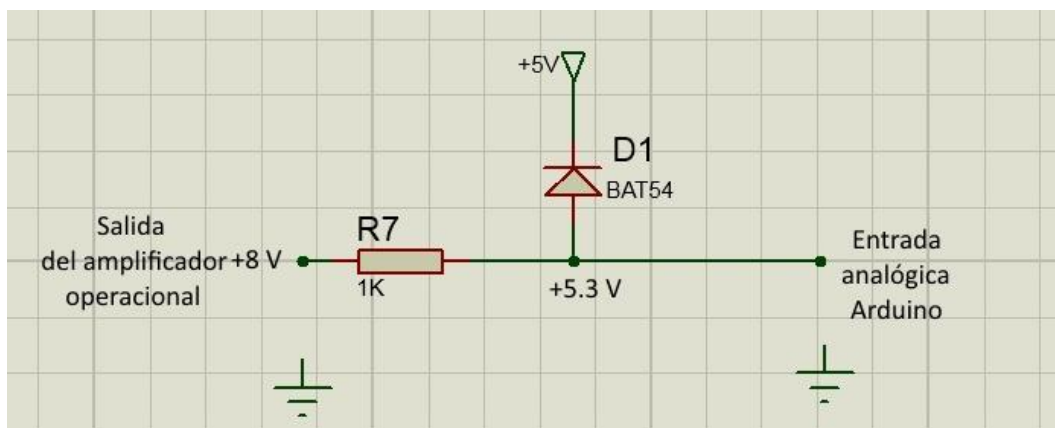


Figura 32. Esquema del circuito limitador para el cálculo de R7

La corriente que es capaz de suministrar un amplificador suele ser pequeña, del orden de pocas decenas de miliamperios. Para el cálculo de la resistencia se considera una corriente de 2 mA, para la que el diodo tendrá una tensión en directa de unos 300 mV según sus especificaciones [30], y que es un valor por debajo de la corriente máxima que los amplificadores son capaces de proporcionar. Así, el valor de la resistencia será, para una corriente de 2 mA y una tensión en sus extremos de 2.6 V, de 1 K Ω aproximadamente.

5.1.4. Calibración del acondicionamiento de la señal del termopar

Para el acondicionamiento de la señal del termopar se utiliza la señal V_{ref} , generada a partir de un divisor de tensión a la salida de una referencia de tensión. Esta tensión se aplica a un amplificador operacional en configuración de *buffer*. A la salida de este *buffer* se tendrá un error u *offset* debido a la

tensión de *offset* de entrada del amplificador y la tensión de modo común, que se podrá corregir a través del software. Para la tensión de entrada al buffer de 360 mV, en la salida se tiene una tensión de 359.2 mV. Ese error de -0.8 mV se compensará a través del software a la hora de obtener la tensión real del termopar.

Por otro lado, el amplificador de instrumentación también generará un error a su salida debido a su tensión de *offset* de entrada y a la tensión de modo común. Para una tensión diferencial de entrada de 0 V y una tensión V_{ref} de 0 V también, se tiene un salida de -0.9 mV. Ese error es un *offset* que también se ajustará a través del software.

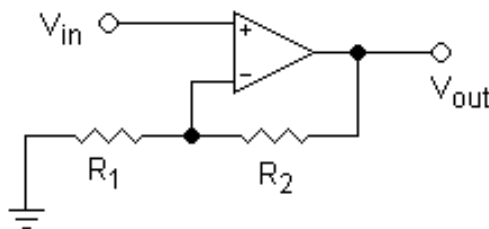
Por tanto, la calibración de este acondicionamiento queda como en la siguiente expresión:

$$V_{out} = 100 \cdot V_{in} + 358.3 \quad (10)$$

Donde V_{out} es la tensión a la salida del amplificador de instrumentación que se leerá a través de Arduino, y V_{in} es la tensión que genera el termopar.

5.1.5. Ganancia del amplificador del LM35

Como se indicó en el apartado 3.5.2, la ganancia necesaria para amplificar la señal del LM35 tiene que ser 6. A partir de la ecuación (5), se obtiene el valor de la ganancia G:



$$G = 1 + \frac{R_2}{R_1} \quad (11)$$

Los valores seleccionados son, para R_1 , 1 K Ω y para R_2 , 5 K Ω , para obtener a través de la ecuación (11) el valor de ganancia requerido:

$$G = 1 + \frac{5 \text{ K}\Omega}{1 \text{ K}\Omega} = 6 \quad (12)$$

5.1.6. Calibración del acondicionamiento de la señal del LM35

El acondicionamiento de la señal del LM35 se realiza a través de un amplificador operacional en configuración no inversora. La tensión a la salida de este amplificador tendrá un error debido a la tensión de *offset* de entrada del mismo. Para una tensión de entrada de 0 V se obtiene una salida de 0 V (con resolución hasta la primera décima de mV). Por lo que no se hará ningún ajuste de *offset*, quedando la expresión de calibración:

$$V_{out} = 6 \cdot V_{in} \quad (13)$$

Donde V_{out} es la tensión a la salida del amplificador operacional que se leerá a través de Arduino y V_{in} es la tensión que genera el LM35.

5.1.7. Divisor de tensión en la alimentación asimétrica

Según la explicación del apartado 3.6, para obtener la tensión asimétrica necesaria para la alimentación del circuito se utiliza un divisor de tensión. A partir de 9 V se ha de obtener una tensión positiva de 8 V y una negativa de 1 V.

Los valores de las resistencias seleccionadas, según el esquema eléctrico genérico de un divisor de tensión de la Figura 31, son, para R1, 40 K Ω y para R2, 5 K Ω . Según la ecuación (6), la tensión en extremos de la resistencia R2 será:

$$V_{R2} = \frac{5 \text{ K}\Omega}{(5 \text{ K}\Omega + 40 \text{ K}\Omega)} \cdot 9 \text{ V} = 1 \text{ V} \quad (14)$$

Por lo que en extremos de R1, la tensión será de 8 V.

5.1.8. Divisor de tensión para detección de batería baja

Como se indica en el apartado 3.7, para detectar si la batería está por debajo de un límite establecido, se implementa un divisor de tensión para poder leer la señal a través de la entrada analógica correspondiente de Arduino.

El divisor de tensión diseñado, según el esquema genérico de la Figura 31, está formado por una resistencia R1 de 3.3 K Ω y una resistencia R2 de 1 K Ω . La tensión medida puede tomar valores desde +6.2 V hasta +8 V, obteniéndose un rango de tensiones de salida desde +1.44 V hasta +1.86 V:

$$V_{OUT+8V} = \frac{1 \text{ K}\Omega}{(1 \text{ K}\Omega + 3.3 \text{ K}\Omega)} \cdot 8 \text{ V} = 1.86 \text{ V} \quad (15)$$

$$V_{OUT+6.2V} = \frac{1 \text{ K}\Omega}{(1 \text{ K}\Omega + 3.3 \text{ K}\Omega)} \cdot 6.2 \text{ V} = 1.44 \text{ V} \quad (16)$$

Los valores medidos en las entradas analógicas van de 0 V a 3 V, que corresponden a un valor digital del ADC entre 0 y 1023. Para la tensión límite medida de 1.44 V se tendrá un valor de 491 y para la tensión de 1.86 V de 634, diferenciando así el valor límite de descarga de la pila.

5.1.9. Resistencia para alimentación de la bobina del relé

Para alimentar la bobina SET del relé de encendido y apagado se ha añadir en serie una resistencia (apartado 3.8).

Las especificaciones del relé marcan que la bobina *SET* ha de ser alimentada con valores entre el 70% y el 130% de la tensión nominal [20]. Como la bobina es de 5 V, esto es, entre 3.5 V y 6.5 V.

La tensión aplicada a la resistencia y bobina en serie tomará valores desde 9 V, usando adaptador de corriente o pila cargada, hasta 7 V con la pila en el nivel límite de descarga. Por tanto se ha de elegir una resistencia que permita alimentar la bobina dentro de estos límites.

La bobina del relé tiene una resistencia de 89Ω , formando un divisor de tensión con la resistencia a seleccionar, que tendrá un valor de 50Ω .

Para el caso en que se alimente el circuito a través del adaptador de corriente o con la pila cargada:

$$V_{bobina} = \frac{89 \Omega}{(89 \Omega + 50 \Omega)} \cdot 9 V = 5.76 V \quad (17)$$

Para el caso en que se alimente el circuito a través de la pila en el límite de descarga:

$$V_{bobina} = \frac{89 \Omega}{(89 \Omega + 50 \Omega)} \cdot 7 V = 4.48 V \quad (18)$$

Para ambos casos, la resistencia seleccionada hace cumplir las especificaciones de alimentación de la bobina.

5.1.10. Consumo del dispositivo

A continuación se explica la relación de consumos de corriente de los diferentes componentes que forman el termómetro, así como una estimación del tiempo aproximado de uso de la pila antes de su descarga.

Los siguientes componentes listados estarán en funcionamiento, de manera continua, durante el uso del termómetro, por tanto, serán los que más directamente contribuyan al consumo general del dispositivo:

- Placa de Arduino: 50 mA
- Amplificador de la señal del termopar: 2 mA
- Amplificador buffer en la generación de la tensión de alimentación asimétrica y V_{ref} : 25 μ A
- LM35: 60 μ A
- Amplificador de la señal del LM35: 2.5 mA
- Display: 1 mA

Otros componentes que contribuirán con consumos puntuales son:

- Bobina *SET* y *RESET* del relé utilizado para el encendido y apagado: 56 mA
- Diodos *schottky* limitadores de tensión en las entradas analógicas: 2 mA

El tipo de pila que se va a usar, de 9 V, normalmente tiene una capacidad de unos 300 mAh. Suponiendo que se tiene un consumo promedio de 60 mA, el tiempo estimado de uso de la pila será de 5 horas.

5.2. Cálculo de errores

En este apartado se realizará un estudio de los errores causados por distintos componentes del diseño y como afectan a la precisión en la lectura final de temperatura.

5.2.1. Error debido al convertidor analógico-digital

El convertidor analógico-digital contribuye con un error de resolución en la conversión. El rango de temperaturas a medir va desde -100 °C hasta 630 °C, en total 730 °C, para la lectura de la señal del termopar y de 0 a 50°C para la lectura de la señal del LM35. Como el convertidor es de 10 bits, se podrán medir 1024 (2^{10}) valores distintos. El error de resolución en cada caso será:

$$Error_{resolución_termopar} = \frac{730\text{ °C}}{1023} = 0.71\text{ °C} \quad (19)$$

$$Error_{resolución_LM35} = \frac{50\text{ °C}}{1023} = 0.049\text{ °C} \quad (20)$$

Nota: para los cálculos de errores en la lectura de la señal del termopar se toma la señal de éste como lineal.

Por otro lado, el convertidor tiene un error de exactitud que engloba errores de linealidad, ruido, resolución, etc. Para el caso del microcontrolador ATmega328 este error es de $\pm 2\text{LBS}$ (*least significant bit*) [17]. Esto quiere decir que el error es dos veces el valor del bit menos significativo, es decir:

$$Error_{exactitud_termopar} = \pm 2 \cdot 0.71\text{ °C} = \pm 1.42\text{ °C} \quad (21)$$

$$Error_{exactitud_LM35} = \pm 2 \cdot 0.049\text{ °C} = \pm 0.098\text{ °C} \quad (22)$$

El convertidor A/D admite diferentes modos de tensión de referencia. El de 5 V utiliza directamente la tensión proporcionada por el regulador de tensión de Arduino, que no es demasiado exacta tal y como se indicó en el apartado 3.2. El error en esta tensión de 5 V es de $\pm 0.2\text{ V}$. Se tendrá el mayor error de exactitud cuando la tensión de referencia sea la mayor, es decir, 5.2 V. Para esta tensión, cada LBS corresponderá a:

$$LBS_{termopar} (mV) = \frac{5200\text{ mV}}{1023} = 5.083\text{ mV} \quad (23)$$

Que en grados centígrados corresponde a:

$$Error_{REF\ 5V_termopar} = \pm 2 \cdot \frac{730\ ^\circ C}{5000\ mV} \cdot 5.083\ mV = \pm 1.49\ ^\circ C \quad (24)$$

$$Error_{REF\ 5V_LM35} = \pm 2 \cdot \frac{50\ ^\circ C}{5000\ mV} \cdot 5.083\ mV = \pm 0.101\ ^\circ C \quad (25)$$

Utilizando la referencia de tensión externa más precisa de 3 V con un error de ± 0.001 V. El error, finalmente, en la temperatura será:

$$Error_{REF\ 5V_termopar} = \pm 2 \cdot \frac{3001\ mV}{1023} \cdot \frac{730\ ^\circ C}{3000\ mV} = \pm 1.426\ ^\circ C \quad (26)$$

$$Error_{REF\ 5V_LM35} = \pm 2 \cdot \frac{3001\ mV}{1023} \cdot \frac{50\ ^\circ C}{3000\ mV} = \pm 0.097\ ^\circ C \quad (27)$$

Por tanto, el error total debido al convertidor analógico/digital será de $\pm 1.523\ ^\circ C$.

5.2.2. Error en el acondicionamiento de la señal del termopar

Por un lado se tendrán los errores debidos a las imperfecciones del amplificador de instrumentación y por otro estará el error en la generación de la V_{ref} , que igualmente afectará a la lectura final de la temperatura.

Errores debidos al amplificador de instrumentación

Uno de los errores en el amplificador es el debido a la deriva de la tensión de offset con la temperatura. Como ya se indicó en el apartado 3.4.1, este valor es de $\pm 5\ nV/^\circ C$ a $25\ ^\circ C$. Así el error debido a la tensión de *offset* en microvoltios es:

$$Error_{V_{OUT\ offset}} = (\pm 5\ nV/^\circ C \cdot 25\ ^\circ C) \cdot 100 = \pm 12.5\ \mu V \quad (28)$$

También, existirán errores por la precisión y no linealidad de la ganancia del amplificador. El error de precisión típico es de $\pm 0.01\%$. Este indica que la ganancia del amplificador podría tomar valores en el rango 100 ± 0.01 . El mayor error se dará para el mayor valor de tensión a la entrada esperado, que es de $26.19\ mV$. Así el mayor error esperado en la tensión a la salida del amplificador debido a la precisión en la ganancia es:

$$Error_{V_{precision_G}} = (\pm 0.01 \cdot 26.19\ mV) = \pm 262\ \mu V \quad (29)$$

La no linealidad de la ganancia tiene un valor típico de $3\ ppm$. Esto equivale a una variación en el valor de la ganancia de 100 ± 0.0003 . El error en la tensión a la salida será:

$$Error_{V_{no\ lin.G}} = (\pm 0.0003 \cdot 26.19\ mV) = \pm 7.8\ \mu V \quad (30)$$

El error total debido a las imperfecciones del amplificador de instrumentación será la suma de todos los anteriores errores expuestos: $\pm 268.9\ \mu V$.

Errores en la generación de la tensión V_{ref}

El primer error que tendrá influencia en la tensión de referencia, será el error de la propia referencia de tensión de 3 V. Como se ha indicado anteriormente tiene un error de $\pm 0.001\ V$. Este error a través del divisor de tensión se traducirá en:

$$Error_{ref_tension} = \frac{25.5\ K\Omega}{(25.5\ K\Omega + 187\ K\Omega)} \cdot \pm 0.001\ V = \pm 120\ \mu V \quad (31)$$

Otro error esperado vendrá provocado por las tolerancias de las resistencias. Las que se han seleccionado para este diseño tienen una tolerancia de $\pm 0.1\%$. Los valores de las resistencia que más error generarán serán los opuestos, es decir, para una de ellas será su valor nominal $+0.1\%$ y para la otra será su valor nominal -0.1% . Así el error final será:

$$Error_{tol.R} = 360\ mV - \frac{25.5\ K\Omega \cdot (1 \pm 0.001)}{(25.5\ K\Omega \cdot (1 \pm 0.001) + 187\ K\Omega \cdot (1 \mp 0.001))} \cdot 3V = \pm 634\ \mu V \quad (32)$$

Por último, los errores derivados de las imperfecciones del amplificador operacional utilizado como *buffer*. Por un lado, la tensión de offset de salida, que con ganancia de valor 1 y una deriva con la temperatura de la misma de $\pm 0.6\ \mu V/^{\circ}C$, genera un error de:

$$Error_{V_{OUT\ offset}} = (\pm 0.6\ \mu V/^{\circ}C \cdot 25\ ^{\circ}C) \cdot 1 = \pm 15\ \mu V \quad (33)$$

El error en tensión total debido a la generación de la tensión de referencia V_{ref} será la suma de los tres anteriores: $\pm 769\ \mu V$.

El error total en el acondicionamiento será el equivalente a la suma del error del amplificador y del error en la generación de V_{ref} : $\pm 1.037\ mV$. Si se pretende medir un rango de $730\ ^{\circ}C$ en un intervalo de 3V, cada grado equivaldrá a $4.10\ mV$. Con esta equivalencia, el error típico total en el acondicionamiento de la señal del termopar será entonces de $\pm 0.25\ ^{\circ}C$.

5.2.3. Error en el acondicionamiento de la señal del LM35

El primer error a destacar es el del propio sensor de temperatura LM35, que como ya se indicó en el apartado 3.5.1, es de $\pm 0.5\ ^{\circ}C$.

Los errores debidos a las imperfecciones del propio amplificador utilizado vendrán provocados por la deriva de la tensión de *offset* con la temperatura y la tensión de modo común. La tensión de *offset* a la entrada del amplificador tiene una deriva térmica de $0.03 \mu\text{V}/^\circ\text{C}$, y que con una ganancia de valor 6, generan un error a la salida de:

$$Error_{V_{OUT_offset}} = (\pm 0.03 \mu\text{V}/^\circ\text{C} \cdot 25^\circ\text{C}) \cdot 6 = \pm 4.5 \mu\text{V} \quad (34)$$

Otro error importante es el debido al rechazo del modo común del amplificador. El coeficiente de rechazo de modo común (CMRR) para este amplificador tiene un valor típico de 140 dB. Este coeficiente se puede definir como en la ecuación (35):

$$CMRR(\text{dB}) = 20 \cdot \log_{10} \left(\frac{G \cdot V_{CM}}{V_{OUT}} \right) \quad (35)$$

Donde, G es la ganancia del amplificador, V_{CM} es la tensión de modo común a la entrada del mismo y V_{OUT} es la tensión a la salida debida a la tensión de modo común a la entrada.

La tensión a la entrada del amplificador va a ser de entre 0 y 500 mV. Como el valor mitad de la alimentación asimétrica es 3.5 V, el mayor valor de tensión de modo común que se tendrá será de 3.5 V. El error a la salida del amplificador debido al modo común será:

$$Error_{V_{OUT_VCM}} = \frac{6 \cdot 3.5 \text{ V}}{\frac{140}{10^{20}}} = 2.1 \mu\text{V} \quad (36)$$

También se producirá un error en el ajuste de la ganancia debido a la tolerancia de las resistencias utilizadas para ello. Ambas resistencias se han seleccionado con una tolerancia de $\pm 0.1\%$. El mayor error en la ganancia se dará cuando el valor de una de las resistencias sea su valor nominal $+0.1\%$ y el de la otra sea su valor nominal -0.1% :

$$Error G = 6 - \left(1 + \frac{5 \text{ K}\Omega \cdot (1 \pm 0.001)}{1 \text{ K}\Omega \cdot (1 \mp 0.001)} \right) = \pm 0.009 \quad (37)$$

Para la mayor señal de entrada, 500 mV, la tensión de error a la salida para este error en la ganancia será:

$$Error = 500 \text{ mV} \cdot \pm 0.009 = \pm 4.50 \text{ mV} \quad (38)$$

El error total en tensión en la adquisición de la señal del LM35 será la suma de todos los anteriores: $\pm 4.5 \text{ mV}$. Si se mide un rango de 50°C en un intervalo de tensiones de 0 V a 3 V, cada grado equivale a 60 mV. Así este

error de ± 4.5 mV equivale a una temperatura de $\pm 0.075^\circ\text{C}$. Este error, unido al error del propio integrado LM35, dará un error total de $\pm 0.575^\circ\text{C}$.

6. Resultados

Una vez diseñado el termómetro es momento de estudiar si las mediciones de temperatura que realiza están dentro del rango esperable, teniendo en cuenta los errores calculados en el apartado 5.2.

A continuación se muestra un resumen de los errores calculados:

| | |
|---|-------------------------------------|
| Error convertidor analógico/digital: | $\pm 1.523\text{ }^{\circ}\text{C}$ |
| Error acondicionamiento señal termopar: | $\pm 0.25\text{ }^{\circ}\text{C}$ |
| Error acondicionamiento LM35: | $\pm 0.575\text{ }^{\circ}\text{C}$ |

El error total del termómetro será de $\pm 2.34\text{ }^{\circ}\text{C}$.

Para comprobar que el error en las medidas está dentro de este rango, se comparará el termómetro diseñado con otro comercial del que se conoce su precisión. Este termómetro patrón es el modelo MM2000 de *TME electronics* [29], cuyas características más importantes son:

| | |
|-----------------------------------|--|
| Precisión | $\pm 0.1\%$ de la lectura $\pm 0.02\text{ }^{\circ}\text{C}$ |
| Rango de temperatura medida | $-200\text{ }^{\circ}\text{C}$ a $1372\text{ }^{\circ}\text{C}$ para termopar tipo K |
| Rango de temperatura de operación | $-30\text{ }^{\circ}\text{C}$ a $50\text{ }^{\circ}\text{C}$ |
| Tensión de alimentación | Batería de 9 V |

Para realizar la comparación, se medirán una serie de temperaturas de manera simultánea con ambos termómetros. Las mediciones se tomarán en la superficie de una resistencia cerámica dentro de una caja aislada, como se muestra en la Figura 33:

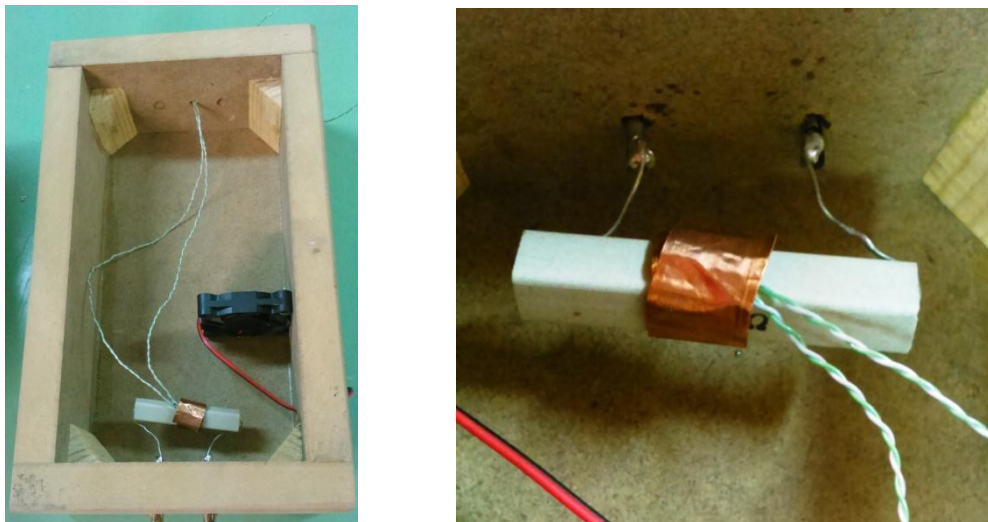


Figura 33. Imagen de la caja aislada en la que se toman las medidas (izquierda) y detalle de la colocación de los termopares en la resistencia cerámica (derecha)

En la Tabla 2 se muestran las medidas de temperatura tomadas con ambos termómetros y la diferencia de temperatura que existe entre ambas:

| Temperatura (°C) Termómetro Patrón TMM 2000 | Temperatura (°C) Termómetro diseñado | Diferencia (°C) |
|---|--|-----------------|
| 29.5 | 29.3 | 0.2 |
| 66.3 | 66.3 | 0 |
| 87.7 | 87.9 | -0.2 |
| 102.9 | 103.2 | -0.3 |
| 115.3 | 115.5 | -0.2 |
| 129.2 | 129.3 | -0.1 |
| 142.6 | 142.4 | 0.2 |
| 155.3 | 155 | 0.3 |
| 167 | 166.7 | 0.3 |
| 179.8 | 179.3 | 0.5 |

Tabla 2. Comparación de las medidas tomadas con el termómetro diseñado y el termómetro patrón

7. Conclusiones

En este TFG se ha diseñado y construido el prototipo de un termómetro digital basado en la plataforma Arduino cumpliendo con todas las especificaciones planteadas.

Gracias a un meticuloso diseño electrónico de los circuitos de acondicionamiento de las señales se ha conseguido una lectura final de temperatura bastante precisa, siendo el propio convertidor analógico/digital de Arduino el que más penaliza con diferencia sobre esta característica (apartado 5.2.1). Por ello, sería interesante sopesar la posibilidad de utilizar otro tipo de plataforma o microcontrolador que incluyera un convertidor analógico/digital con mayor precisión y resolución.

El uso de las placas de Arduino facilita enormemente la tarea de configuración y programación de los microcontroladores, haciendo del desarrollo de un dispositivo como el de este TFG una tarea mucho menos tediosa. Sin embargo, placas como la utilizada aquí (Arduino UNO) perjudican en otro tipo de tareas o funciones de los dispositivos creados, como es el caso de la precisión y resolución en la lectura de señales, ya descrito antes, su consumo de corriente o su tamaño. Es el caso de este termómetro, en que el uso de pilas para alimentarlo no resulta la mejor de las opciones ya que su consumo se ve enormemente encarecido debido a la placa de Arduino (apartado 5.1.10). Por ello, una posible mejora futura sería su sustitución por otro tipo de plataforma o microcontrolador en la que el consumo se reduzca de manera considerable.

A nivel personal este proyecto ha servido para ampliar los conocimientos sobre el uso de termopares como transductores de temperatura, sus características y el acondicionamiento de sus señales, siendo, además, un elemento tan comúnmente utilizado en la industria. Además, ha servido para conocer las características y funcionamiento de las placas Arduino, con más importancia cada vez dentro del ámbito tecnológico, así como para ampliar los conocimientos y mejorar la destreza en las tareas de diseño de circuitos electrónicos y placas de circuito impreso.

8. Bibliografía

- [1] ATMARAM MORE, S. (2014). *Application Report SLOA204: Thermocouple, Cold-junction Compensation*. Texas Instruments.
<<http://www.ti.com.cn/cn/lit/an/sloa204/sloa204.pdf>> [Consulta: 30 de junio de 2015]
- [2] COUNTS L., KITCHIN C. (2006). *A Designer's Guide to Instrumentation Amplifiers*. Analog Devices
<http://www.analog.com/media/en/training-seminars/design-handbooks/5812756674312778737Complete_In_Amp.pdf> [Consulta: 30 de junio de 2015]
- [3] FERGUSON, T. (2001). *Application Note 043: Measuring Temperature with Thermocouples*. National Instruments.
<<http://www.engr.sjsu.edu/sanjay/courses/ME120/me120pdf/ThermocoupleMeas.pdf>> [Consulta: 30 de junio de 2015]
- [4] MARCIN, J. *Application Note 369: Thermocouple Signal Conditioning using AD594/AD595*. Analog Devices.
<<http://www.analog.com/media/en/technical-documentation/application-notes/AN-369.pdf>> [Consulta: 30 de junio de 2015]
- [5] NATIONAL SEMICONDUCTOR (2002). *Application Note 31: Op Amp Circuit Collection*.
<<http://www.ti.com/ww/en/bobpease/assets/AN-31.pdf>> [Consulta: 30 de junio de 2015]
- [6] NASH, E. *Application Note 539: Errors and Error Budget Analysis in Instrumentation Amplifier Applications*. Analog Devices.
<<http://www.cypress.com/?docID=46842>> [Consulta: 30 de junio de 2015]
- [7] NASH, E (1998). *Common mode and Instrumentation Amplifiers*. Analog Devices.
<<http://www.analog.com/media/en/technical-documentation/technical-articles/25406877Common.pdf>> [Consulta: 30 de junio de 2015]
- [8] PALMER, R. (2001) *Application Report SLOA050. DC Parameters: Input Offset Voltage*. Texas Instruments.
<<http://www.ti.com/lit/an/sloa059/sloa059.pdf>> [Consulta: 30 de junio de 2015]
- [9] WILLIAMS, J. (1986). *Op Amp Booster Stages Part-1*. Linear Technology Corp.
<http://www.ka-electronics.com/images/pdf/Op_Amp_Booster_Stages_pt1.pdf> [Consulta: 30 de junio de 2015]

- [10] COUGHLIN, R. y DRISCOLL, F. (1999) *Amplificadores Operacionles y Circuitos Integrados Lineales*. México: Prentice Hall.
- [11] ANALOG DEVICES. *AD595 Datasheet. Monolithic Thermocouple Amplifiers with Cold Junction Compensation*.
<http://www.analog.com/media/en/technical-documentation/data-sheets/AD594_595.pdf> [Consulta: 30 de junio de 2015]
- [12] ANALOG DEVICES. *AD780 Datasheet. High Precision Reference*.
<<http://www.analog.com/media/en/technical-documentation/data-sheets/AD780.pdf>> [Consulta: 30 de junio de 2015]
- [13] ARDUINO. *Información Arduino UNO R3*.
<<https://www.arduino.cc/en/Main/ArduinoBoardUno>> [Consulta: 30 de junio de 2015]
- [14] ARDUINO. *Librería TimerOne*.
<<http://playground.arduino.cc/Code/Timer1>> [Consulta: 30 de junio de 2015]
- [15] ARDUINO. *Arduino Development Environment*.
<<https://www.arduino.cc/en/Guide/Environment>> [Consulta: 30 de junio de 2015]
- [16] ARDUINO. *Librería Liquid Crystal*.
<<https://www.arduino.cc/en/Reference/LiquidCrystal>> [Consulta: 30 de junio de 2015]
- [17] ATMEL. *ATMEL 8-BIT Microcontroller with 4/8/16/32Kbytes in-system programmable flash datasheet*.
<<http://www.atmel.com/images/doc8161.pdf>> [Consulta: 30 de Junio de 2015]
- [18] LABCENTER ELECTRONICS. *ISIS. Intelligent Schematic Input System User Manual. Issue 6.0*.
<<http://faculty.qu.edu.qa/malmeer/506471/Proteus%20All-in-One%20Manual%20v6%5B1%5D.3.pdf>> [Consulta: 30 de junio de 2015]
- [19] LINEAR TECHNOLOGY. *LTC1100 Precision, zero-drift Instrumentation Amplifier datasheet*.
<<http://cds.linear.com/docs/en/datasheet/1100fc.pdf>> [Consulta: 30 de junio de 2015]
- [20] OMRON. *G6C PCB Power Relay. Miniature Hight Capacity Relays*.
<<http://www.omron.com/ecb/products/pdf/en-g6c.pdf>> [Consulta: 30 de junio de 2015]
- [21] POWERTIP. *PC 1602-H LCD Display Datasheet*.

- <<http://www.powertipusa.com/pdf/pc1602h.pdf>> [Consulta: 30 de junio de 2015]
- [22]STANFORD UNIVERSITY. *HD44780 LCD Starter Guide*.
<http://web.stanford.edu/class/ee281/handouts/lcd_tutorial.pdf>
[Consulta: 30 de junio de 2015]
- [23]THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). Tablas de equivalencias tensión-temperatura para termopares.
<<http://srdata.nist.gov/its90/download/download.html>> [Consulta: 30 de junio de 2015]
- [24]THE NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST). *Coeficientes de polinómicos para varios tipos de termopar*.
<<http://srdata.nist.gov/its90/download/allcoeff.tab>> [Consulta: 30 de junio de 2015]
- [25]TEXAS INSTRUMENTS. *LM35 Datasheet. Precision Centigrade Temperature Sensors*.
<<http://www.ti.com/lit/ds/symlink/lm35.pdf>> [Consulta: 30 de junio de 2015]
- [26]TEXAS INSTRUMENTS. *LP2985-N Micropower Low-Noise Ultra-Low Dropout Regulator*.
< <http://www.ti.com/lit/ds/symlink/lp2985-n.pdf> > [Consulta: 30 de junio de 2015]
- [27]TEXAS INSTRUMENTS. *OPA2251 Single-Supply, Micro-Power Operational Amplifiers datsheet*.
<<http://www.ti.com/lit/ds/sbos075/sbos075.pdf>> [Consulta: 30 de junio de 2015]
- [28]TEXAS INSTRUMENTS. *TLC2652 Advanced LinCMOS Precision Chopper-stabilized Operational Amplifiers*.
<<http://www.ti.com/general/docs/lit/getliterature.tsp?genericPartNumber=tlc2652&fileType=pdf> > [Consulta: 30 de junio de 2015]
- [29]TME ELECTRONICS. *Technical Data Model MM2000*.
<http://www.tmelectronics.co.uk/downloads/dl/file/id/17/mm2000_data_sheet.pdf > [Consulta: 30 de junio de 2015]
- [30]VISHAY. *BAT54, BAT54A, BAT54C, BAT54S Small Signal Schottky Diodes, Single and Dual datasheet*.
<<http://www.vishay.com/docs/85508/bat54v.pdf>> [Consulta: 30 de junio de 2015]
- [31]WIKIPEDIA. *Filtro pasa bajo*.
<https://en.wikipedia.org/wiki/Low-pass_filter> [Consulta: 30 de junio de 2015]

[32]WIRING.

< <http://wiring.org.co/> > [Consulta: 30 de junio de 2015]

[33]“*Thermocouple 101 tutorials. Analog Devices*”. Youtube

<<https://www.youtube.com/playlist?list=PL36A224E609F3F7C9>>

[Consulta: 30 de junio de 2015]

Anexo A

Coefficientes para la compensación de la unión fría por software:

```

*****
* This section contains coefficients for type K thermocouples for
* the two subranges of temperature listed below. The coefficients
* are in units of °C and mV and are listed in the order of
constant
* term up to the highest order. The equation below 0 °C is of the
form
*  $E = \sum_{i=0}^n c_i t^i$ .
*
* The equation above 0 °C is of the form
*  $E = \sum_{i=0}^n c_i t^i + a_0 \exp(a_1 (t - a_2)^2)$ .
*
* Temperature Range (°C)
* -270.000 to 0.000
* 0.000 to 1372.000
*****
name: reference function on ITS-90
type: K
temperature units: °C
emf units: mV
range: -270.000, 0.000, 10
0.000000000000E+00
0.394501280250E-01
0.236223735980E-04
-0.328589067840E-06
-0.499048287770E-08
-0.675090591730E-10
-0.574103274280E-12
-0.310888728940E-14
-0.104516093650E-16
-0.198892668780E-19
-0.163226974860E-22
range: 0.000, 1372.000, 9
-0.176004136860E-01
0.389212049750E-01
0.185587700320E-04
-0.994575928740E-07
0.318409457190E-09
-0.560728448890E-12
0.560750590590E-15
-0.320207200030E-18
0.971511471520E-22
-0.121047212750E-25
exponential:
a0 = 0.118597600000E+00
a1 = -0.118343200000E-03
a2 = 0.126968600000E+03

*****
* This section contains coefficients of approximate inverse
* functions for type K thermocouples for the subranges of

```

* temperature and voltage listed below. The range of errors of
 * the approximate inverse function for each subrange is also
 given.
 * The coefficients are in units of °C and mV and are listed in
 * the order of constant term up to the highest order.
 * The equation is of the form $t_{90} = d_0 + d_1 * E + d_2 * E^2 + \dots$
 * $+ d_n * E^n$,
 * where E is in mV and t_{90} is in °C.

| Temperature range (°C) | Voltage range (mV) | Error range (°C) |
|------------------------|--------------------|------------------|
| -200. to 0. | -5.891 to 0.000 | -0.02 to 0.04 |
| 0. to 500. | 0.000 to 20.644 | -0.05 to 0.04 |
| 500. to 1372. | 20.644 to 54.886 | -0.05 to 0.06 |

 Inverse coefficients for type K:

| | | | |
|-------------|----------------|---------------|---------------|
| Temperature | -200. | 0. | 500. |
| Range: | 0. | 500. | 1372. |
| Voltage | -5.891 | 0.000 | 20.644 |
| Range: | 0.000 | 20.644 | 54.886 |
| | 0.0000000E+00 | 0.0000000E+00 | -1.318058E+02 |
| | 2.5173462E+01 | 2.508355E+01 | 4.830222E+01 |
| | -1.1662878E+00 | 7.860106E-02 | -1.646031E+00 |
| | -1.0833638E+00 | -2.503131E-01 | 5.464731E-02 |
| | -8.9773540E-01 | 8.315270E-02 | -9.650715E-04 |
| | -3.7342377E-01 | -1.228034E-02 | 8.802193E-06 |
| | -8.6632643E-02 | 9.804036E-04 | -3.110810E-08 |
| | -1.0450598E-02 | -4.413030E-05 | 0.0000000E+00 |
| | -5.1920577E-04 | 1.057734E-06 | 0.0000000E+00 |
| | 0.0000000E+00 | -1.052755E-08 | 0.0000000E+00 |
| Error | -0.02 | -0.05 | -0.05 |
| Range: | 0.04 | 0.04 | 0.06 |

Anexo B

Programa principal:

```

#include <LiquidCrystal.h>
#include <TimerOne.h>

//Control de pantallas
/*Variables utilizadas para el control de las distintas pantallas mostradas*/
int READY = 0;
int screen = 0;
int time = 0;
/*Constantes utilizadas para el control de las distintas pantallas mostradas*/
int time_lim = 30;

//Variables para el modo MMA
float Tmin = 650;
float screenMin = 0;
float Tmax = -150;
float screenMax = 0;
float avg = 0;
unsigned long avg_counter = 0;
float screenAvg = 0;

//Variables para el modo HOLD
float Tscreen = 0;

//Variables para el modo OTD
int SET_OTD = 0;
//Constante para el modo OTD (límite detección de termopar roto)
int lim_OTD = 600;

//Variables para el modo LOWBATT
int SET_LOWBATT = 0;
//Constante para el modo LOWBATT (límite detección de pila agotada)
int lim_inf_LOWBATT = 490;
int lim_sup_LOWBATT = 510;

//Apagado y autoapagado
/*Variables utilizadas para el apagado y autoapagado*/
int LOCK_OFF = 1;
int t_AUTO = 0;
int AUTO = 1;
/*Constantes utilizadas para el apagado y autoapagado*/
int t_AUTO_lim = 300;//segundos

//-----Variables para la adquisición de datos-----//
int DATA_counter = 0;
//Vectores que almacenan los diez valores adc del LM35 y termopar, respectivamente, uno
cada 100ms

```

```

int a0[10] = { -1, -1, -1, -1, -1, -1, -1, -1, -1, -1};
int a1[10] = { -1, -1, -1, -1, -1, -1, -1, -1, -1, -1};
//Variables valores medios de elementos de los vectores a0 y a1
float adc0 = 0;
float adc1 = 0;

//-----Constantes setup entradas y salidas-----//
/*Entradas analógicas*/
int in_termocouple = 0;      //Entrada del termopar en pin A1
int OTD = 1;                //Entrada de la detección de termopar roto
int in_LM35 = 2;           //Entrada del LM35 en pin A0
int LOWBATT = 3;          //Entrada de detección de batería baja

/*Entradas y salidas digitales*/
int OFF_OUT = 6;          //Salida digital señal de apagado
int OFF = 7;              //Entrada digital pulsador ON/OFF
int MMA = 8;             //Entrada digital pulsador máximo, mínimo y valor medio de la temperatura
int HOLD = 9;            //Entrada digital pulsador modo HOLD
int CLEAR = 13;          //Entrada digital pulsador CLEAR

/*Entradas y salidas digitales para el display*/
LiquidCrystal lcd( 12, 10, 11, 4, 5, 2, 3);

void setup() {

  lcd.begin(16,2); //Inicialización de filas y columnas del display
  analogReference(EXTERNAL); //Selección de la referencia para el convertidor ADC externa
  pinMode(OFF_OUT,OUTPUT); //Configuración del pin OFF_OUT como salida digital
  pinMode(CLEAR,INPUT); //Configuración del pin CLEAR como entrada digital
  Timer1.initialize(100000); //Configuración del tiempo de la interrupción a 100ms
  Timer1.attachInterrupt(acquisition); //Configuración del nombre de la función a la que
  llama la interrupción
}

void loop() {
//-----//
//-----MODO MMA-----//
//-----//
  if (digitalRead(MMA) == 1)
  {
    switch (READY)
    {
      case 0:
        screen = 1;
        screenMax=Tmax;
        screenMin=Tmin;
        screenAvg=avg/avg_counter;
        lcd.setCursor(0,1);
        lcd.print (" ");
        lcd.setCursor(0,1);
        lcd.print("MAX");
        lcd.setCursor(9,1);

```

```
lcd.print(screenMax,1);
lcd.setCursor (14,1);
lcd.write(char(223));
lcd.setCursor (15,1);
lcd.print ("C");
while (digitalRead(MMA) == 1)
{
}
READY = 1;
break;

case 1:
lcd.setCursor(0,1);
lcd.print (" ");
lcd.setCursor(0,1);
lcd.print("MIN");
lcd.setCursor(9,1);
lcd.print(screenMin,1);
lcd.setCursor (14,1);
lcd.write(char(223));
lcd.setCursor (15,1);
lcd.print ("C");
while (digitalRead(MMA) == 1)
{
}
READY = 2;
break;

case 2:
lcd.setCursor(0,1);
lcd.print (" ");
lcd.setCursor(0,1);
lcd.print("AVG");
lcd.setCursor( 9,1);
lcd.print(screenAvg,1);
lcd.setCursor (14,1);
lcd.write(char(223));
lcd.setCursor (15,1);
lcd.print ("C");
while (digitalRead(MMA) == 1)
{
}
READY = 3;
break;

case 3:
screen = 0;
while (digitalRead(MMA) == 1)
{
}
READY = 0;
break;
}
time = 0;
t_AUTO = 0;
```

```
}
//-----//
//-----//
//-----//

//-----//
//-----Modo HOLD-----//
//-----//

if (digitalRead(HOLD)==1)
{
  switch (READY)
  {
    case 0:
      screen = 1;
      if(SET_OTD ==1)
      {
        lcd.setCursor(0,1);
        lcd.print(" ");
        lcd.setCursor(0,1);
        lcd.print("HOLD");
        lcd.setCursor(10,1);
        lcd.print ("ERROR");
      }
      else
      {
        lcd.setCursor(0,1);
        lcd.print(" ");
        lcd.setCursor(0,1);
        lcd.print("HOLD");
        lcd.setCursor(9,1);
        lcd.print (Tscreen,1);
        lcd.setCursor (14,1);
        lcd.write(char(223));
        lcd.setCursor (15,1);
        lcd.print ("C");
      }
      while (digitalRead (HOLD) == 1)
      {
      }
      READY = 4;
      break;

    case 4:
      screen = 0;
      while (digitalRead (HOLD) == 1)
      {
      }
      READY = 0;
      break;
    }
  time = 0;
  t_AUTO = 0;
}
```

```

//-----//
//-----//
//-----//

//-----//
//-----MODO CLEAR-----//
//-----//
if (digitalRead (CLEAR) == 1)
{
  switch (READY)
  {
    case 0:
      screen = 1;
      lcd.setCursor (0,1);
      lcd.print(" ");
      lcd.setCursor (0,1);
      lcd.print("RESET M M A");
      Tmax=-150;
      Tmin=650;
      avg=0;
      avg_counter=0;
      break;

    case 4:
      lcd.setCursor(0,1);
      lcd.print(" ");
      lcd.setCursor(0,1);
      lcd.print("AUTO OFF");
      AUTO =0;
      break;
  }
  while (digitalRead (CLEAR) == 1)
  {
  }
  delay(1000);
  screen = 0;
  READY = 0;
  time = 0;
  t_AUTO = 0;
}

//-----//
//-----//
//-----//

//-----Regreso a pantalla principal-----//
if (time > time_lim && screen != 0)
{
  screen = 0;
  READY = 0;
}
//-----//

```

```
//-----//
//-----APAGADO Y AUTOAPAGADO-----//
//-----//

if (t_AUTO > t_AUTO_lim && AUTO == 1)
{
  digitalWrite(OFF_OUT,1);
}
else
{
  if (digitalRead(OFF) == 1)
  {
    if(LOCK_OFF == 0)
    {
      while(digitalRead (OFF) ==1 )
      {
      }
      delay(50);
      digitalWrite(OFF_OUT,1);
    }
    else
    {
      while(digitalRead (OFF) == 1)
      {
      }
      LOCK_OFF = 0;
      t_AUTO = 0;
    }
  }
  else
  {
    LOCK_OFF = 0;
  }
}

//-----//
//-----//
//-----//
}
```

Programa de la interrupción:

```
//-----//
/*Función a la que se llama cada 100 milisegundos, desde la interrupción del timer1.
Esta función hace la media de los valores medidos de temperatura de la unión fría
y del termopar, realiza la compensación y muestra el valor de la temperatura por pantalla.*/
//-----//

void acquisition() {
//Coeficientes termopar tipo k
float C[10]={-1.760041e-2, 3.892121e-2, 1.855877e-5, -9.945759e-8, 3.1840945719e-10, -
5.607285e-13, 5.6075059059e-16, -3.202072e-19, 9.715115e-23, -1.210472e-26};
float a[3]={1.185976e-1, -1.183432e-4, 126.9686};
float d1[10]={ 0, 25.17346,-1.166288, -1.083364, -8.977354e-1, -3.734238e-1, -8.663264e-2, -
1.04506e-2, -5.1920577e-4, 0};
float d2[10]={ 0, 25.08355, 7.860106e-2, -2.503131e-1, 8.31527e-2, -1.228034e-2, 9.804036e-
4, -4.41303e-5, 1.057734e-6,-1.052755e-8};
float d3[10]={ -131.8058, 48.30222, -1.646031, 5.464731e-2, -9.650715e-4, 8.802193e-6, -
3.11081e-8, 0, 0, 0};

//Variables de adquisición
float V0 = 0; //Tensión amplificada del LM35
float T0 = 0; //Temperatura leída por el LM35
float VOT = 0; //Tensión equivalente de termopar relativa a la temperatura leída por el LM35
float V = 0; //Tensión del termopar
float T = 0; //Temperatura después de la compensación de la unión fría

//Declaración de Constantes
float m0 = 50/3; //Pendiente recta calibrado LM35
int K = 100; //ganacia del amplificador del termopar
float Vref = 0.358; //Tensión offset "Vref" del amplificador del termopar

//Lectura de las entradas analógicas del LM35 y termopar
a0[DATA_counter]= analogRead (in_LM35);
a1[DATA_counter]= analogRead (in_termocouple);

if (DATA_counter == 9)
{
/*-----Detección de batería baja cada segundo-----*/
if (analogRead (LOWBATT) < lim_inf_LOWBATT)
{
if (SET_LOWBATT == 0)
{
lcd.setCursor(0,0);
lcd.print(" ");
lcd.setCursor (0,0);
lcd.print ("LOW BATT");
SET_LOWBATT = 1;
}
}
else
{
if (analogRead (LOWBATT) > lim_sup_LOWBATT)
{

```

```

    if (SET_LOWBATT == 1)
    {
        lcd.setCursor(0,0);
        lcd.print("      ");
        SET_LOWBATT = 0;
    }
}
}
/*-----*/
if (analogRead(OTD) > lim_OTD)
{
    SET_OTD=1;
    if(screen==0)
    {
        lcd.setCursor (10,0);
        lcd.print (" ");
        lcd.setCursor(0,1);
        lcd.print("      ");
        lcd.setCursor(10,1);
        lcd.print("Error");
    }
}
else
{
    //media de los valores leidos del LM35
    adc0=0;
    for (int i=0; i < 10; i++)
    {
        adc0 = adc0 + a0[i];
    }
    adc0 = adc0/10;

    //conversión a voltaje
    V0 = 3 * adc0 / 1023;
    //conversión a temperatura
    T0 = m0 * V0;

    // Transformacion a voltaje de termopar tipo k
    /*Se aplica el polinomio de cambio de temperatura a voltaje, para el termopar
    de tipo k, para temperaturas mayores de cero */
    for (int i=9; i>0; i--)
    {
        V0T=(V0T+C[i])*T0;
    }
    V0T=C[0]+V0T+a[0]*exp(a[1]*pow(T0-a[2],2));

    //media de los valores leidos del termopar
    adc1=0;
    for (int i=0; i < 10; i++)
    {
        adc1 = adc1 + a1[i];
    }
    adc1 = adc1/10;

    //conversión a voltaje

```



```

V = 3.0 * adc1 / 1023.0;
V = V - Vref;
V = V / K;

//Suma del voltaje del LM35
V= V*1000 + V0T;

//conversión a temperatura
/*Se aplica el polinomio de cambio de voltaje a temperatura, para el termopar
de tipo k, para diferentes intervalos de temperatura */

if (V < 0)
{
  for (int i=9; i>0; i--)
  {
    T=(T+d1[i])*V;
  }
  T=d1[0] + T;
}
else
{
  if (V < 20.644)
  {
    for (int i=9; i>0; i--)
    {
      T=(T+d2[i])*V;
    }
    T=d2[0] + T;
  }
  else
  {
    for (int i=9; i>0; i--)
    {
      T=(T+d3[i])*V;
    }
    T=d3[0] + T;
  }
}
T=T+0.9;
//-----Mostrar en display la temperatura-----//
/*Si estamos en la screen 0, muestro la temperatura por pantalla. Si no, continúo
tomando datos, pero sin mostrarlos.
También modifico con la palabra ERROR, si el termopar se ha desconectado */
if (screen == 0)
{
  if (SET_OTD == 1)
  {
    lcd.setCursor(0,1);
    lcd.print("      ");
    lcd.setCursor(10,1);
    lcd.print("Error");
  }
  else
  {
    lcd.setCursor(0,1);

```

```
    lcd.print("      ");
    lcd.setCursor(9,1);
    lcd.print(T,1);
    lcd.setCursor (14,1);
    lcd.write(char(223));
    lcd.setCursor (15,1);
    lcd.print ("C");
    Tscreen=T;
  }
}

if (SET_OTD == 0)
{
  if (T>Tmax)
  {
    Tmax=T;
  }
  if (T<Tmin)
  {
    Tmin=T;
  }
  avg=avg+T;
  avg_counter ++;
}
else
{
  SET_OTD =0;
}
}
//-----//

if(screen == 1)
{
  time ++;
}
if (AUTO ==1)
{
  t_AUTO ++;
}
DATA_counter = 0;
}
else
{
  DATA_counter ++;
}
}
```

Anexo C

Manual de uso del termómetro digital.

Este termómetro permite mostrar a través de una pantalla LCD la temperatura medida a través de un termopar de tipo K. Puede medir un amplio rango de temperatura además de implementar otras funciones como el registro de los valores máximos, mínimos y medios medidos, el apagado automático para ahorrar energía o la utilización del dispositivo de manera portátil alimentándolo a través de una pila.

| Características | |
|-----------------------------------|---------------------------------------|
| Alimentación | Pila 9 V o adaptador de corriente 9 V |
| Consumo de corriente | 60 mA |
| Rango de temperatura de operación | 0 °C a 50 °C |
| Tipo de sonda | Termopar tipo K |
| Rango de medida de temperatura | -100 °C a +630 °C |
| Precisión | ±2.34 °C |

Conexiones del termómetro:

Conector de termopar: en la parte superior del termómetro se dispone un conector de termopar tipo K para facilitar el intercambio del mismo.

Conector de adaptador de corriente: en la parte superior derecha se dispone un conector para adaptador de corriente.



Detalle de los conectores de termopar y adaptador de corriente.

Conexión para la pila: en la parte trasera, a través de una tapa de la carcasa del termómetro, se tiene acceso a la conexión para la pila de 9 V.



Detalle de la conexión para la pila.

Funcionamiento del termómetro:

Para encender el termómetro basta con accionar el pulsador marcado como ON/OFF. Inmediatamente se mostrará por pantalla la temperatura leída en ese momento y que se actualizará cada segundo. La temperatura leída se mostrará con una resolución de un decimal y siempre en grados centígrados.



Pantalla principal y pulsador ON/OFF

Para apagarlo bastará con accionar ese mismo pulsador ON/OFF.

Registro de valores máximo, mínimo y medio:

El termómetro internamente registrará los valores máximo, mínimo y medio de una serie de lecturas realizadas. Para poder visualizarlos se acciona el pulsador marcado como MMA (max, min, average) desde la pantalla principal. Al pulsarlo una primera vez, se mostrará el valor máximo, al pulsarlo una segunda vez se mostrará el valor mínimo y al pulsarlo una tercera vez se mostrará el valor medio. Pulsándolo una vez más, se regresa a la pantalla principal en la que se muestra la temperatura leída en cada momento.



Pantallas de valores máximo, mínimo y medio y pulsador MMA

Estos valores se reiniciarán cada vez que se encienda el termómetro. Además, existe la posibilidad de reiniciarlos manualmente accionando el pulsador marcado como CLEAR desde la pantalla principal, indicándolo por pantalla con el mensaje "RESET M M A".



Mensaje de reinicio de valores máximo, mínimo y medio y pulsador CLEAR

Nota: aunque por pantalla no se esté mostrando el valor de temperatura leído, el termómetro internamente sigue registrando los valores de temperatura medidos.

Función HOLD:

Esta función sirve para mantener el valor mostrado por pantalla fijo durante un tiempo determinado. Activando el pulsador marcado como HOLD desde la pantalla principal, se muestra la temperatura que se estaba mostrando en el momento en que se pulsó el botón. Si se pulsa otra vez se regresará a la pantalla principal.



Pantalla en modo HOLD y pulsador HOLD

Nota: aunque por pantalla no se esté mostrando el valor de temperatura leído, el termómetro internamente sigue registrando los valores de temperatura medidos.

Nota: tanto en la función HOLD como en la función de valores máximo, mínimo y medio, si al cabo de 30 segundos no se ha regresado manualmente a la pantalla principal lo hará el termómetro automáticamente.

Desconexión o rotura del termopar:

Siempre que el termopar sea desconectado o se haya roto, el termómetro mostrará por pantalla el mensaje "Error".



Mensaje de desconexión o rotura de termopar

Batería baja:

Si la carga de la batería es inferior al valor mínimo en el que se asegura un buen funcionamiento del dispositivo, se mostrará en la esquina superior izquierda de la pantalla el mensaje "LOW BATT". En ese momento se recomienda sustituir la pila o utilizar el adaptador de corriente.



Mensaje indicador de batería baja

Función de Auto-apagado:

Para ahorrar energía, sobre todo si se está alimentando el termómetro a través de la pila, el termómetro se apagará automáticamente al cabo de 5 minutos desde la última vez que se haya pulsado cualquier botón.

Para los casos en que se quiera registrar temperaturas por un tiempo superior a esos cinco minutos, existe la posibilidad de desactivar este apagado automático. Para ello, se entrará en el modo HOLD (activando el pulsador HOLD) y a continuación se activará el pulsador CLEAR. Se mostrará por pantalla el mensaje "Auto off" que indica que el apagado automático está desactivado. Hay que tener en cuenta que cada vez que se encienda el termómetro, el auto-apagado volverá a ser activado.



Desactivación del apagado automático

Anexo D

Esquemático:

