



Diseño y validación de circuito digital para control de temperatura con sensores digitales

Sabrina D. Pryszczuk ^{a*}, Mikaela B. Garcete ^a, Juan P. Gross ^b, Guillermo A. Fernández ^b

^a *Universidad Nacional de Misiones (UNaM). Facultad de Ingeniería. Misiones, Argentina.*

^b *Universidad Nacional de Misiones (UNaM). Facultad de Ingeniería. Instituto de Materiales de Misiones (IMAM).
Grupo de Investigación y Desarrollo en Electrónica (GIDE). Misiones, Argentina.*

sabrinadaianap@gmail.com, garcetemikaela97@gmail.com, gross@fio.unam.edu.ar,
guillermo.fernandez@fio.unam.edu.ar

Resumen

En el presente trabajo se exponen los resultados obtenidos a partir de la elaboración de un sistema electrónico propuesto como actividad integradora en la asignatura Técnicas Digitales 2, de la carrera Ingeniería Electrónica. Dicha experiencia tiene por objetivos la puesta en práctica de los saberes adquiridos durante el curso de la asignatura, así como también su vinculación con los obtenidos en otras materias de la carrera. La actividad consistió en el desarrollo de un prototipo funcional de un sistema embebido cuya finalidad es la supervisión remota de temperaturas, proporcionando también una acción de control determinada en el momento en el que se superan ciertos valores umbrales. Los valores de umbral para cada variable son configurables por el usuario mediante una interfaz conformada por pulsadores y una pantalla. El trabajo realizado cumplió con los requisitos solicitados para la actividad integradora, permitiendo asociar las habilidades y conocimientos adquiridos mediante el desarrollo de un sistema funcional cuya implementación en la vida real resulta factible y práctico. Durante el desarrollo del trabajo se encontraron algunas limitaciones propias del diseño, las cuales constituyen puertas a futuros desarrollos y mejoras para el sistema propuesto.

Palabras Clave – Sensores, microcontroladores, medición de parámetros ambientales, programación.

1. Introducción

El desarrollo del presente trabajo corresponde a la actividad integradora final propuesta en el marco de la asignatura Técnica Digitales 2, correspondiente al cuarto año de la carrera Ingeniería Electrónica. La actividad mencionada tiene como finalidad desarrollar saberes relacionados con el diseño y la implementación de circuitos digitales con microcontroladores, que se ven involucrados en productos de uso masivo o industrial.

El objetivo al que apunta este trabajo es asegurar dichos saberes adquiridos mediante la resolución de una problemática presente en la vida real, tal como un proceso común dentro de la industria, donde es requerido el control de temperatura dentro de cierto rango admisible. La medición y control de una magnitud física en un proceso industrial, tal como la temperatura, es indispensable en la actualidad y constituye la base de muchos sistemas de recolección de datos aplicables al control de producción y calidad.

La propuesta del presente trabajo consiste en el desarrollo de un sistema electrónico basado en el microcontrolador ATmega2560 disponible en la plataforma Arduino MEGA, y mediante el cual se

*Autor en correspondencia.

controlan dos temperaturas medidas a través de sensores del DS18B20. Este tipo de sensor tiene una salida digital que provee la temperatura medida a través de una palabra binaria configurable de 9 a 12 bits. La información es enviada hacia o desde el dispositivo sensor, a través de un cable. Por lo que solamente es necesario conectar un cable el microcontrolador y el sensor mencionado.

En el sistema propuesto, cada sensor de temperatura DS18B20 cuenta con un relé de 12 V (y su indicador luminoso) asociado. Los mismos son accionados por el microcontrolador que posee el sistema, cuando la temperatura supera el umbral correspondiente, configurado por el usuario. La configuración es posible mediante un display, el cual permite visualizar los umbrales de temperatura y la temperatura medida por los dispositivos.

Para el diseño del circuito de control de temperatura, inicialmente se plantean dos soluciones alternativas, de las cuales es seleccionada una de ellas, analizando los pros y contras correspondientes. En la primera solución se propuso un sistema embebido que permite configurar los umbrales de temperatura para cada sensor mediante un resistor variable (potenciómetro). En la segunda solución se propuso la configuración de los umbrales a través de pulsadores, con los cuales se añade o quita 5°C cada vez que uno u otro es pulsado. Esta última opción es la seleccionada para el diseño del circuito de control, ya que evita inconvenientes debido al desgaste que pueden presentar los resistores indicados en la primera solución.

A lo largo del desarrollo del artículo inicialmente se menciona y explican los circuitos propuestos para cada solución, las ventajas y desventajas de cada uno y la selección de la pertinente propuesta. Luego se presenta el diseño del circuito seleccionado y los resultados obtenidos al construirlo.

2. Soluciones propuestas al circuito de control

2.1. Circuito A – Configuración mediante potenciómetro

El diagrama de bloques correspondiente a la primera solución propuesta para el circuito, es el mostrado en la Fig. 1a. En el mismo se aprecian las entradas y salidas del microcontrolador ATmega2560 (uC) en las cuales están conectados los dos sensores DS18B20, un resistor variable para la configuración de los umbrales de temperatura, un pulsador para acceder a la configuración mencionada y un display para visualizar la temperatura medida y los umbrales configurados. El pulsador está conectado al puerto PB0 y cuando el usuario lo acciona, el programa del microcontrolador entra a la configuración donde puede visualizarse los umbrales de temperatura según el valor ajustado con el potenciómetro.

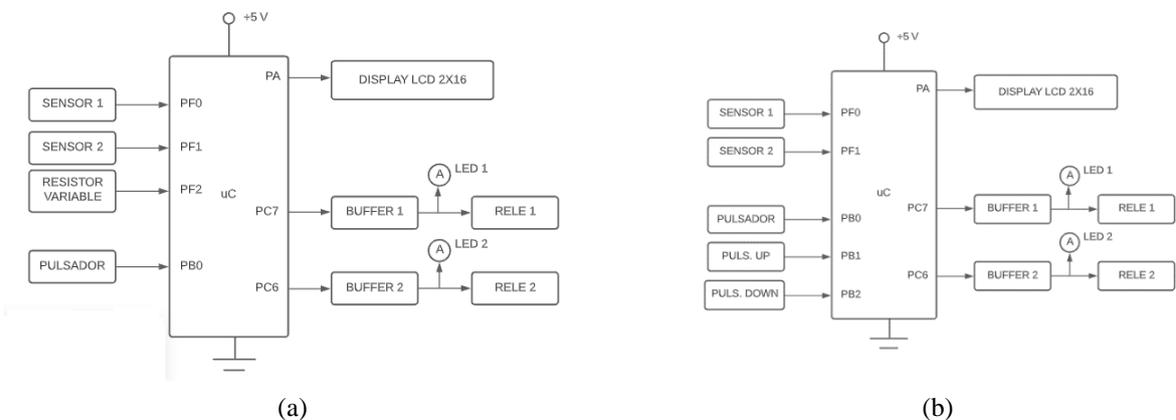


Fig. 1- Diagramas de bloques de las soluciones propuestas: (a) Circuito A, configuración mediante potenciómetros; (b) Circuito B, configuración mediante pulsadores.

En el diagrama de la Fig. 1a, el puerto PF2 del uC funciona como entrada analógica. El convertor analógico-digital (ADC) de este dispositivo permite obtener un valor digital de la tensión aplicada al puerto. El programa ejecutado por el uC usa ecuaciones que primeramente calculan el valor de la tensión en la entrada PF2 y luego determinan el valor equivalente a la temperatura. De esta forma, sería posible configurar cada umbral de temperatura, variando la resistencia del potenciómetro. [1]

En su funcionamiento normal, el programa ejecutado en el uC constantemente mide las dos temperaturas a través de los sensores, compara las mismas con cada umbral configurado por el usuario y al superar el mismo enciende el relé correspondiente. Es decir, si se supera el umbral de temperatura asociado al sensor 1, el relé 1 y su respectivo led son encendidos. De la misma forma, si se supera el umbral de temperatura asociado al sensor 2, el relé 2 y su respectivo led son encendidos.

2.2. Circuito B – Configuración mediante pulsadores

El segundo diseño presentado en la Fig. 1b, funciona de manera similar al anterior, teniendo los dos sensores DS18B20, un pulsador para acceder a la configuración de umbrales de temperatura y las salidas del buffer con el led indicador de encendido para cada relé y un display para visualizar la temperatura medida y los umbrales configurados. Sin embargo, a diferencia del diseño anterior el resistor variable es reemplazado por dos pares de pulsadores. Cada par posee un pulsador para aumentar (pulsador UP) y otro para disminuir (pulsador DOWN) el umbral de temperatura asignado a cada uno de los sensores.

2.3. Selección de la propuesta

A pesar de la ventaja relativa a la precisión en la configuración del umbral de temperatura deseado cuando es variada la resistencia del potenciómetro, el uso frecuente del mismo ocasionaría desgastes prematuros en este dispositivo, provocando fallos en el funcionamiento de la configuración de los umbrales mencionados. Es por esto por lo que se optó por desarrollar el circuito B, indicado en la Fig. 1b. La desventaja que presenta esta propuesta se da en cuestiones de programación relativas a los pulsadores, donde debe considerarse el efecto de rebote de estos y resultando en este aspecto más compleja en comparación con la propuesta de la Fig. 1a.

2.4. Selección de componentes

Los componentes empleados en el circuito correspondiente al diagrama de la Fig. 1b son:

- **DS18B20:** Es el sensor de temperatura usado en el sistema. El mismo posee una salida de datos digital mediante el protocolo 1-Wire, el cual utiliza un sólo cable para la transferencia de datos. Este dispositivo puede efectuar mediciones de temperatura, con un rango admisible entre $-55\text{ }^{\circ}\text{C}$ y $150\text{ }^{\circ}\text{C}$. [2]
- **Arduino® MEGA:** Esta placa de desarrollo es encargada de administrar tanto la lógica de control de temperatura como la interfaz del usuario, a través de un microcontrolador ATmega2560. Dicho dispositivo cuenta con características de velocidad y memoria que lo hacen adecuado para llevar a cabo la función principal de control en el sistema embebido desarrollado [3].
- **Display LCD (Liquid Crystal Display):** El misma forma parte de la interfaz de usuario del sistema, es del tipo estándar con 2 líneas y 16 caracteres. Estas características permiten la visualización de los valores de temperatura medidos, así como también del menú de configuración desarrollado para el sistema. [4]
- **Relés:** Estos dispositivos permiten implementar un control del tipo ON/OFF para cualquier tipo de dispositivo compatible que se conecte al sistema, de acuerdo con las necesidades del usuario [5].

Seguidamente se desarrolla el diseño del circuito propuesto para el sistema de control de temperatura con sensores digitales.

3. Diseño del circuito

3.1. Diseño de la etapa de potencia y circuitos de indicación

El circuito de la etapa de potencia que permite al microcontrolador accionar cada relé y su led indicador asociado, es el mostrado en la Fig. 2.

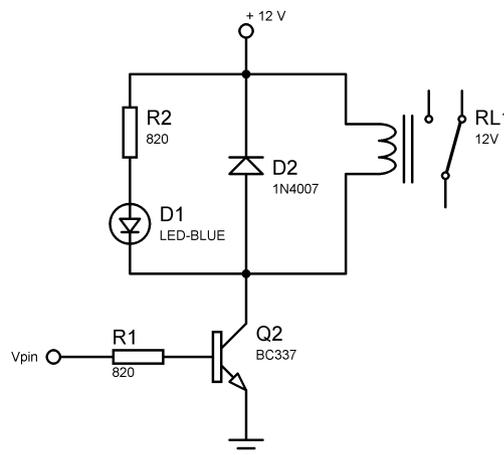


Fig. 2 - Circuito de la etapa de potencia para el accionamiento de cada relé y su led indicador.

En base a la corriente de colector máxima que debe operar y la tensión de bloque que debe soportar el transistor, como lo muestra la Fig. 2, se ha seleccionado el BC337 para este dispositivo en la etapa de potencia [6]. A su vez, para protección de dicho transistor, se conectó un diodo 1N4007 en paralelo con el relé [7]. Las ecuaciones (1) y (2) permiten el dimensionamiento del resistor limitador de corriente utilizado para el led indicador que posee la etapa de potencia.

$$R_{LED} = \frac{V_{CC} - V_{CEsat} - V_{LED}}{I_{LED}} \quad (1)$$

$$P_{R_{LED}} = I_{LED}^2 \cdot R_{LED} \quad (2)$$

El dimensionamiento de la resistencia de base para la polarización del transistor de la Fig. 2, se hace mediante las ecuaciones (3) y (4).

$$R_B = \frac{V_{pin} - V_{BEon}}{I_B} = \frac{V_{pin} - V_{BEon}}{10 \cdot (I_{LED} + I_{RELE}) / \beta} \quad (3)$$

$$P_{R_B} = I_B^2 \cdot R_B \quad (4)$$

Para hallar los valores de los resistores mencionados se tuvieron en cuenta las especificaciones de los componentes involucrados. Estas especificaciones están en la Tabla 1.

Tabla 1 - Valores involucrados en los cálculos de los resistores.

REFERENCIA	VALOR	OBSERVACIONES
V_{CC}	12 V	Tensión de alimentación
$V_{CE(sat)}$	0,7 V	Tensión colector emisor en saturación [6]
V_{LED}	2 V	Tensión de LED
I_{LED}	11,3 mA	Corriente de LED
V_{PIN}	4,2 V	Tensión de salida de pin [3]
V_{BE}	1,2 V	Tensión base emisor en saturación [6]
I_{Bmin}	3,65 mA	Corriente de base mínima
I_C	41,3 mA	Corriente de colector
β	100	Ganancia de corriente del transistor [6]
I_{RELE}	30 mA	Corriente del relé [5]

De esta forma, el valor obtenido para el resistor limitador del led R_{LED} fue de 930 Ω adoptándose el valor comercial de la serie E12 de 820 Ω con una potencia de 1/4 W. Mientras que el resistor base R_B resultó en 726,4 Ω eligiéndose también uno de 820 Ω pero con potencia de 1/8 W. Para las resistencias *pull-up* de los pulsadores se adoptó el valor de 10 k Ω con una potencia de 1/4 W.

La resistencia *pull-up* utilizada en la conexión de los sensores es escogida teniendo en cuenta que su valor depende de la longitud del cable de la sonda que lo une al microcontrolador. Según el fabricante sensor, los valores recomendados según la longitud son los indicados en la Tabla 2.

Tabla 2 - Tabla de valores de resistencia según la longitud del cable de la sonda [1]

<i>Resistencia pull-up</i>	<i>Distancia del cable en metros</i>
4,7 k Ω	De 0 m a 5 m
3,3 k Ω	De 5 m a 10 m
2,2 k Ω	De 10 m a 20 m
1,2 k Ω	De 20 m a 50 m

En este trabajo la longitud de los cables de ambos sensores es menor a 5 m, por lo que se utilizó una resistencia *pull-up* de 4,7 k Ω con una potencia de 1/4 W.

Cabe mencionar, el circuito completo del sistema de control de temperatura con sensores digitales desarrollado está presentado en el Anexo 1 de este artículo.

3.2. *Diseño del firmware*

El *firmware* es el programa que ejecuta el microcontrolador (uC) del sistema embebido desarrollado. A continuación, es explicada la operación del programa diseñado, el cual comanda el funcionamiento del circuito del sistema de control de temperatura.

Los sensores de temperatura DS18B20 usados en este trabajo, son aptos para líquidos y gases debido a que están dentro de una sonda impermeable. Una de las características importantes de estos sensores, es que poseen una salida digital con resolución programable de 9 a 12 bits. En este trabajo, los sensores operan con la resolución por defecto de 12 bits. Otra característica importante de los sensores DS18B20 es que utilizan el protocolo de comunicación serie “1-Wire”. Este protocolo, diseñado por la firma “Dallas Semiconductor”, utiliza una sola línea de datos para comunicarse y permite que se pueda conectar más de un sensor en el mismo bus, siempre y cuando éstos tengan una referencia a tierra común.

Para poder utilizar el protocolo 1-Wire en el *firmware* del uC, primeramente, debió incorporarse al software de programación la librería “OneWire.h”. A partir de las funciones propias de esta librería, en el uC del sistema se ejecutó un programa que permite identificar la dirección única que cada dispositivo sensor trae incorporada en su memoria de 64 bits. Esta dirección es necesaria para identificar a cada uno de los sensores conectados al bus de comunicación que posee el sistema de control de temperatura construido. Una vez obtenidas las direcciones de ambos sensores, también es incorporada la librería “DallasTemperature.h”, con el fin de utilizar las funciones que permiten convertir en grados centígrados los datos enviados por los sensores. [8]

A partir de lo anterior, se desarrolla el *firmware* para el sistema de control. La rutina principal del mismo es repetida permanentemente mientras el circuito permanezca energizado. Esta rutina verifica constantemente si el pulsador P1 para acceso a la configuración es presionado, de ser así puede

accederse a la etapa de configuración del *firmware*. Esta etapa muestra el primer umbral de temperatura “Tu1” (correspondiente al sensor 1), que el usuario debe configurar respetando un rango establecido entre 20 °C y 100 °C. Si el usuario supera estos límites, el sistema muestra el rango admisible en su display y vuelve a indicar el valor del umbral de temperatura por defecto (20 °C). El umbral “Tu1” puede configurarse accionando los pulsadores P2 (sube umbral) y P3 (baja umbral). Una vez fijado el valor para Tu1, accionando el pulsador P1, el usuario puede configurar el umbral “Tu2” para el sensor 2 de forma similar con P2 y P3. Luego de configurar los umbrales, con un nuevo accionamiento de P1 se sale de la etapa de configuración.

Una vez configurados los umbrales de temperatura Tu1 y Tu2, es ejecutada la rutina principal donde el programa realiza la comparación constante entre las temperaturas medidas por los sensores y los umbrales configurados por el usuario. En caso de superarse una de estos, el programa enciende el relé correspondiente, junto al led indicador asociado. Si la temperatura medida con algunos de los sensores vuelve a ser menor al umbral correspondiente, el respectivo relé es apagado por el programa.

Mientras se ejecuta lo anterior, en el display se muestran las temperaturas medidas por los sensores y las correspondientes temperaturas de umbral.

3.3. Diagramas de flujo

En la Fig. 3 puede apreciarse el diagrama de la rutina principal correspondiente al *firmware* desarrollado para el sistema de control de temperatura. En esta figura puede verse que en su parte inicial son incluidas las librerías necesarias y también se definen las variables y constantes utilizadas en el programa. Además, es inicializado el display que posee el sistema, por el bucle de la rutina principal, que como ya se explicó, es repetido en tanto el circuito del sistema permanezca energizado, permitiendo así el control constante de la temperatura a través de los sensores, junto al consecuente accionamiento de cada relé cuando sea necesario.

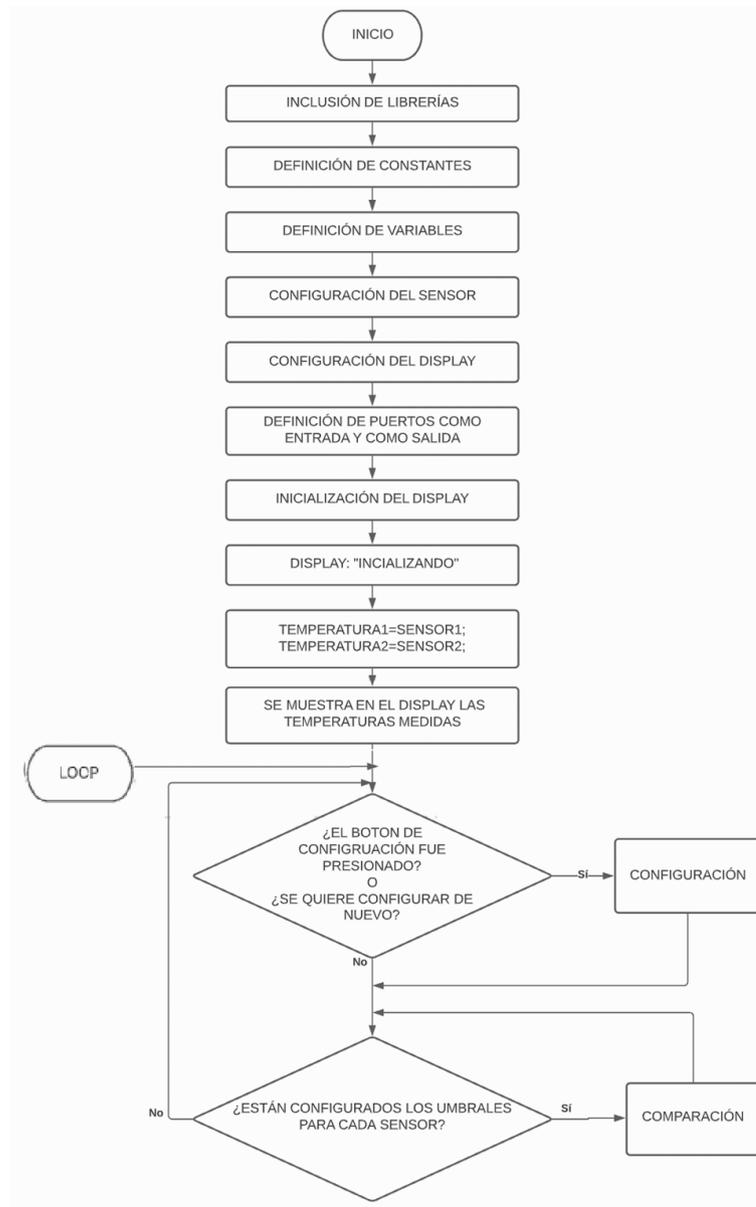


Fig. 3 - Diagrama de flujo de la rutina principal del *firmware*.

La Fig. 4 muestra el diagrama de flujo del *firmware* desarrollado, correspondiente a la función “Configuración” indicada en la Fig. 3, la cual permite al usuario configurar los umbrales de temperatura para el control de esta variable.

Primeramente, el usuario debe elegir la temperatura de umbral Tu_1 correspondiente al sensor 1. Para ello mediante los pulsadores P2 y P3 puede bajar o subir, respectivamente, 5°C de temperatura. Solamente puede seleccionar una temperatura dentro del rango 20°C a 100°C , por lo que, si se intenta salir de ese rango, el proceso de configuración del umbral se reinicia en 20°C . Una vez que el usuario selecciona la temperatura de umbral deseada, debe presionar el pulsador P1 para proceder a elegir la temperatura de umbral Tu_2 , correspondiente al sensor 2. Para configurar Tu_2 , el procedimiento es el

temperaturas medidas se muestran en pantalla junto a las temperaturas de umbral elegidas. Luego, se procede a comparar las temperaturas medidas con las temperaturas de umbral, si alguna de las medidas es mayor a su correspondiente umbral se encienden sus respectivos relé y led indicador. Si el usuario presiona el pulsador P1, se apagan ambos relés con sus respectivos leds y se vuelve al proceso de configuración de las temperaturas de umbral

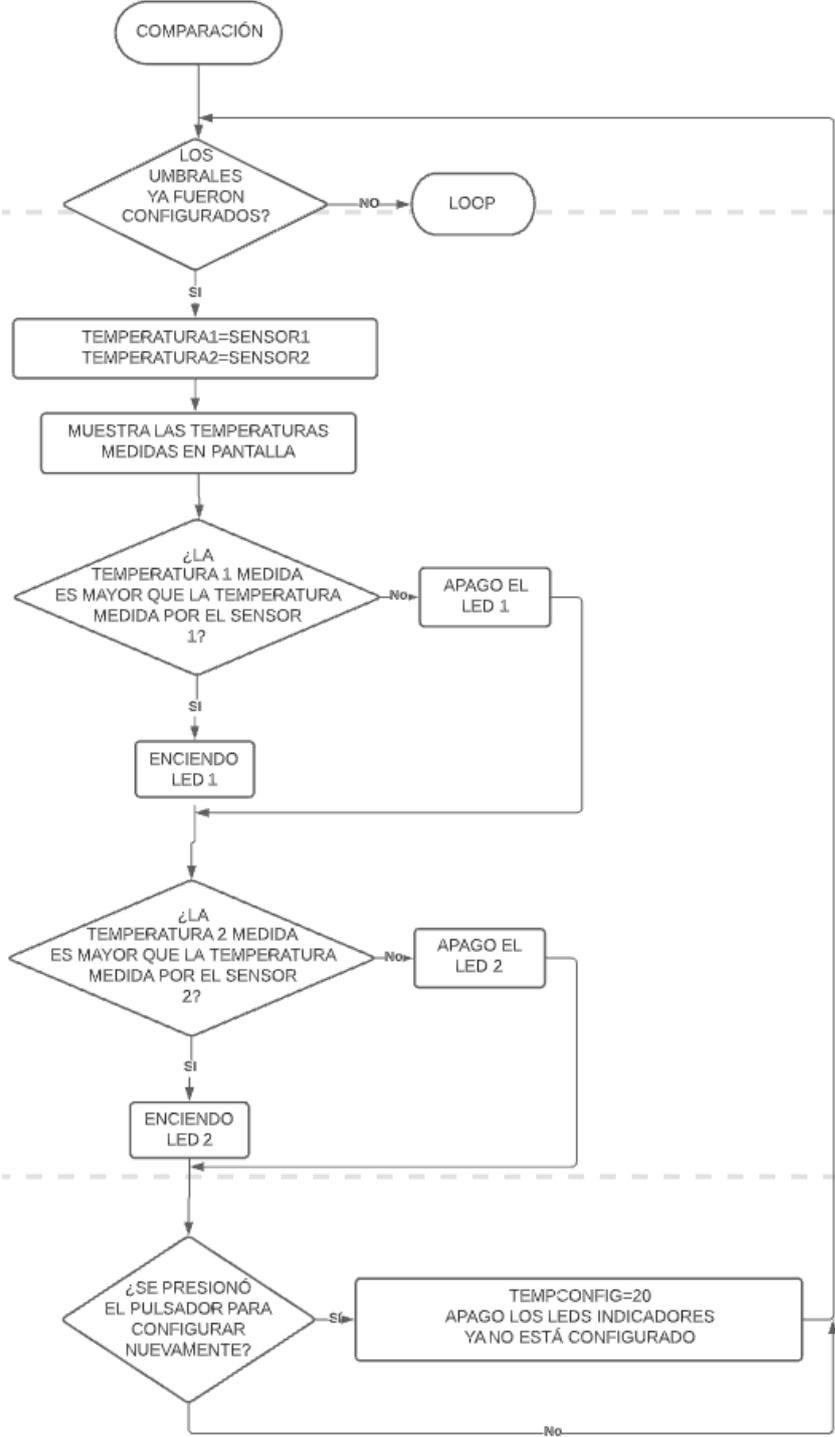


Fig. 5 - Diagrama de flujo de la etapa de comparación de temperaturas.

4. Resultados

Para el desarrollo del firmware se trató de utilizar el software de simulación Proteus, pero se presentaron dificultades, al cargar el programa el mismo no reflejaba ninguna acción. Debido a esto, la depuración del *firmware* y las pruebas de funcionamiento fueron realizadas directamente mediante el circuito montado en un *protoboard*. Estas pruebas fueron realizadas por partes. Primeramente, se verificó el funcionamiento de cada sensor, comprobando que las direcciones obtenidas de los mismos fueran las correctas y que los valores de temperatura medidos fueran distintos de -127°C , ya que este es el valor que entregan los sensores cuando hay alguna falla en la conexión. Luego, se procedió a medir algunas temperaturas conocidas como la temperatura ambiente y la temperatura humana. Los resultados para estas mediciones fueron los esperados, sin embargo, inicialmente hubo dificultades ya que uno de los conectores empleados en la conexión de los sensores no hacía buen contacto, provocando errores en la obtención de la temperatura.

Una vez realizadas estas pruebas iniciales, se procedió a realizar un ensayo de varios puntos de temperatura de un volumen de agua. Para corroborar la exactitud en la medición de los sensores, se compararon las temperaturas obtenidas con estas y las de un termómetro patrón (Fluke 50S con termocupla tipo K). Los resultados obtenidos se muestran en la fig. 6. Como puede observarse en esta figura, en las mediciones no hay diferencias mayores a 1°C , por lo que se comprobó el correcto funcionamiento de los sensores.

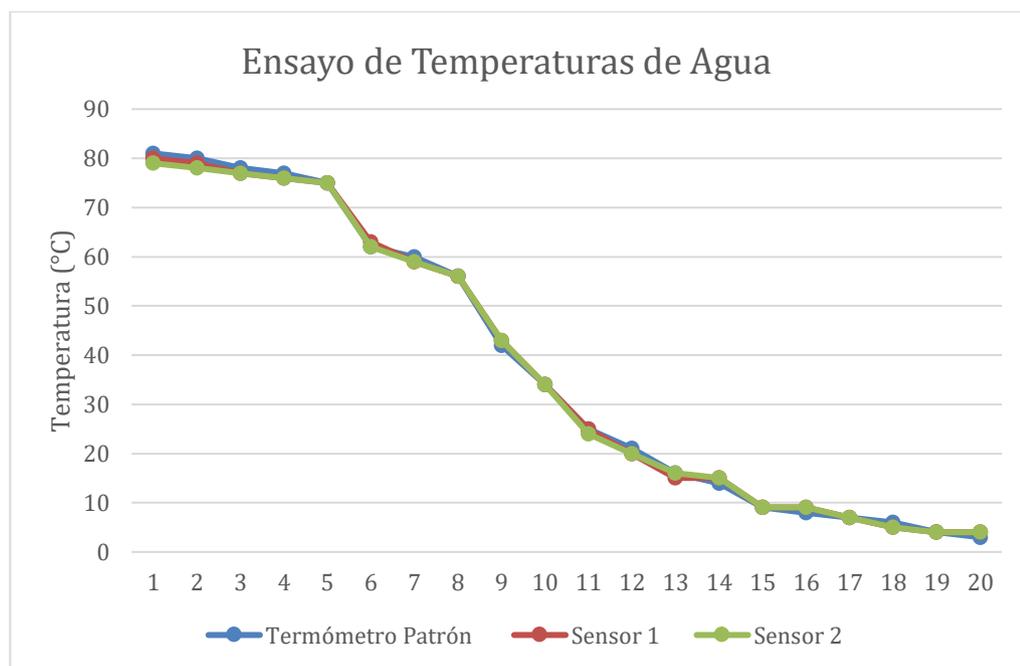


Fig. 6 - Gráfica comparativa entre temperaturas medidas por los sensores y un termómetro patrón.

Luego fue testado el funcionamiento de la interfaz de usuario conformada por el display y los pulsadores P1, P2 y P3. Para esto se comprobó la operación de la etapa de configuración de los

umbrales de temperatura, realizándose varias modificaciones hasta lograr el correcto funcionamiento del menú implementado en el *firmware* del sistema de control.

Una vez alcanzados los resultados deseados de *hardware* y *software* en el *protoboard*, quedó establecido el circuito definitivo para el sistema de control propuesto. El mismo puede apreciarse en el Anexo 1 de este trabajo y a partir del mismo fue diseñado el circuito impreso (PCB, *Printed Circuit Board*).

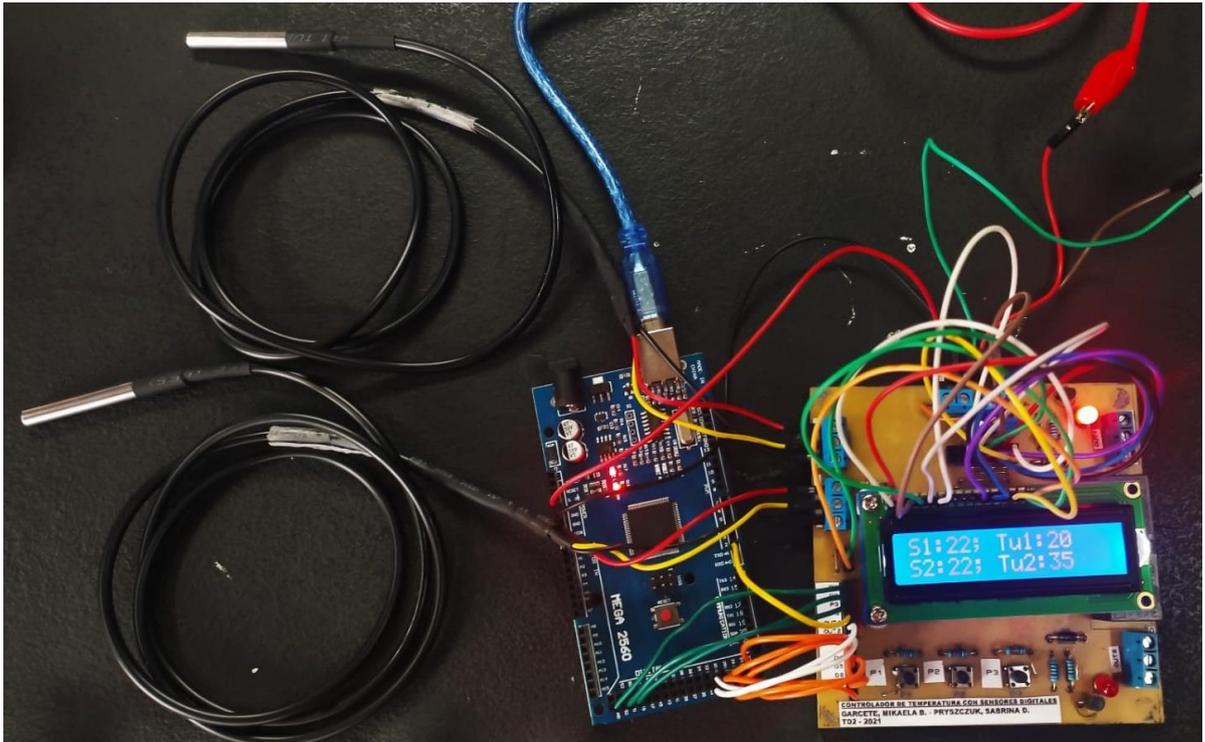


Fig. 7- Prototipo del Sistema de Control de Temperatura con sensores digitales.

En la Fig. 7 puede apreciarse el prototipo de circuito ya ensamblado en la placa, donde se observa en pantalla la información arrojada por los sensores y los umbrales de temperatura umbral configurados por el usuario. La Tabla 3 resume los componentes utilizados en la construcción del prototipo mencionado

Para comprobar la operación del sistema desarrollado, se realizaron los ensayos pertinentes verificándose el funcionamiento de cada parte del sistema de control de temperatura diseñado. En primer lugar, se probó el funcionamiento de la interfaz de usuario, configurando los umbrales de temperatura para cada sensor, luego fueron conectados los sensores verificándose las mediciones correspondientes. Finalmente, fue comprobado el accionamiento de los relés y el encendido de los leds asociados, al superar los umbrales de temperatura configurado para cada sensor. Estos ensayos permitieron detectar y corregir algunos errores de software, como por ejemplo que se debía presionar al menos dos veces el pulsador P1 para acceder a la configuración de los umbrales de temperatura. También se realizaron algunas mejoras en la interfaz para que ésta sea más comprensible para el usuario, como por ejemplo mostrar mensajes de instrucciones desplazándolos por la pantalla. [9]

Tabla 3 - Componentes empleados en la confección del circuito

REFERENCIA	CANTIDAD	OBSERVACIONES
LCD	1	Display LCD 1602
R1, R2, R3, RV2	4	Resistor 10 k Ω
RL1, RL2, RB1, RB2	4	Resistor 820 Ω
PULSADORES	3	Tipo <i>touch</i>
Q1, Q2	2	Transistor BC337
RELE 1, RELE 2	2	Relé 12 VDC HJR-3FF-S-ZH
LED1, LED2	2	LED Rojo 5mm
D1, D2	2	Diodo 1N4007
C2	1	Capacitor de 10 μ F
ARD1	1	Placa Arduino® MEGA
U1, U2	2	Sensor DS18B20

5. Conclusiones

En este trabajo se construyó un sistema electrónico que permite la medición y control de temperaturas mediante un microcontrolador y el uso de dos sensores que poseen comunicación digital de datos. La configuración del sistema puede realizarse a través de pulsadores y un display LCD, este último muestra los datos de las temperaturas medidas y de los umbrales de control correspondiente. Si uno de los sensores presenta una medición que supera el umbral correspondiente, el microcontrolador lleva a cabo una determinada acción de control encendiendo el relé asignado al sensor. A través de los resultados obtenidos, fue posible comprobar que el sistema de control de temperatura implementado opera según lo mencionado, cumpliendo de esta forma con los objetivos propuestos de la actividad integradora.

A través de las actividades realizadas en este trabajo, puede concluirse que la verificación de un programa no siempre podrá ser llevada a cabo mediante un software de simulación; por otra parte, las pruebas realizadas en el armado del protoboard no se corroboran completamente con los resultados obtenidos en la placa diseñada; el entorno de desarrollo de la plataforma Arduino facilita la programación del *firmware*, permitiendo la integración de funciones específicas para el manejo de dispositivos tales como los sensores digitales DS18B20 y el display LCD.

Con la realización de este proyecto, también fue posible distinguir ciertas limitaciones respecto a los dispositivos empleados. En el caso de los sensores de temperatura DS18B20, según la descripción técnica provista por el fabricante, son dispositivos de alta rapidez de sensado, pudiendo medir temperaturas instantáneamente. Sin embargo, en la práctica las mediciones entregadas por los sensores aumentan o disminuyen progresivamente hasta establecerse luego de varios segundos en el valor medido. Es decir, que la temperatura medida no es entregada por el sensor de manera instantánea. Otra limitación encontrada, es que debido al tamaño del display usado (dos líneas y dieciséis caracteres), las instrucciones mostradas en el mismo para que el usuario pueda operar el sistema, no resultaban claras. Para solucionar esto, mediante funciones provistas por el IDE de Arduino, se logró mostrar mensajes más largos desplazándolos en la pantalla. Sin embargo, la desventaja de esto es que al ser más extensos los mensajes se demora más tiempo mostrándolos en

pantalla y durante ese tiempo el sistema no puede ejecutar otras acciones como leer sus entradas (pulsadores y sensores), por ejemplo.

A partir de lo mencionado, fue posible diseñar, programar, construir y verificar un sistema embebido como el ya descrito, aplicando los saberes adquiridos en la asignatura Técnicas Digitales 2 y además integrando los mismos con lo desarrollado en otras asignaturas de la carrera. Estos resultados reflejan que tales experiencias logran fijar conceptos, al poner a prueba las habilidades y la competencia de cada alumno, herramientas de gran importancia en el futuro profesional.

Referencias

- [1] «ADC Arduino – Entradas Analógicas» [En línea].
Disponible en: <https://urlshortner.org/EqWCX>. [Último acceso: 21-21-2021].
- [2] «DS18B20 Programmable Resolution 1-Wire Digital Thermometer» [En línea].
Disponible en: <https://urlshortner.org/VvTOe>. [Último acceso: 12-12-2021].
- [3] «MEGA2560 Datasheet (PDF) - ATMEL Corporation» [En línea].
Disponible en: <https://urlshortner.org/DFRQu>. [Último acceso: 21-12-2021].
- [4] «LM016L Datasheet (PDF) - Hitachi Semiconductor» [En línea].
Disponible en: <https://urlshortner.org/aNVZZ>. [Último acceso: 21-12-2021].
- [5] «HJR-3FF datasheet» [En línea].
Disponible en: <https://urlshortner.org/PNIVe>. [Último acceso: 12-12-2021].
- [6] «BC337 Datasheet (PDF) - Fairchild Semiconductor» [En línea].
Disponible en: <https://urlshortner.org/PdGFi>. [Último acceso: 10-12-2021].
- [7] «1N4007 Datasheet(HTML) - Vishay Siliconix» [En línea].
Disponible en: <https://urlshortner.org/gTsia>. [Último acceso: 10- 08-2021].
- [8] «DS18B20 sensor de temperatura para líquidos con Arduino» [En línea].
Disponible en: <https://urlshortner.org/dARCM>. [Último acceso: 21-12-2021]
- [9] «Tutorial Arduino: pantalla LCD» [En línea].
Disponible en: <https://urlshortner.org/wSobw>. [Último acceso: 21-21-2021].

ANEXO 1

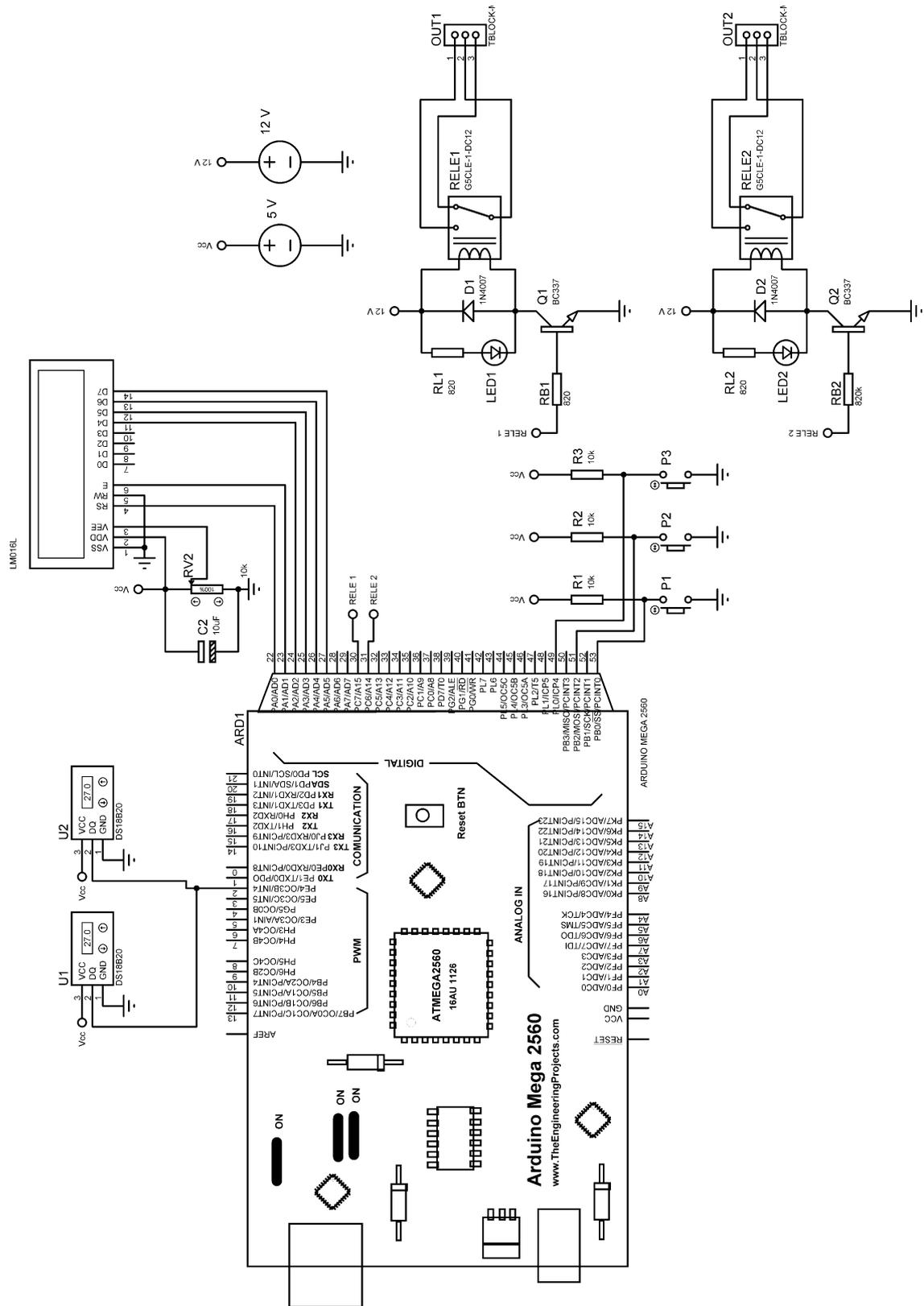


Fig. 8 - Circuito del sistema de control de temperatura con sensores digitales.