

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

SISTEMA ELECTRÓNICO PARA MEDIR Y EVALUAR LA DISTRIBUCIÓN DE TEMPERATURAS EN INCUBADORAS NEONATALES

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

JORGE LUIS COELLO DURAND

Asesor: Msc. Bruno Castillon Levano

Lima, abril del 2011

RESUMEN

La mayoría de termómetros en el mercado presentan deficiencias para la medición de cambios de temperatura en incubadoras neonatales, como: lento tiempo de respuesta, dificultad para medir con precisión, difícil instalación, e inapropiados para registrar cambios de temperatura en varios puntos a la vez. Para superar estas limitaciones se desarrolló un sistema electrónico (hardware y software) especializado para incubadoras neonatales que permite realizar mediciones simultáneas de temperatura en cinco puntos, el rango de medición va desde los 0°C a $+100^{\circ}\text{C}$ con un error de medición de $\pm 0.015^{\circ}\text{C}$ en el rango de $+20^{\circ}\text{C}$ y $+55^{\circ}\text{C}$, una resolución de 0.01°C , con capacidad de visualizar las mediciones en tiempo real y registrar hasta 24 horas de evaluación en una computadora personal, que forma parte del sistema.

El sistema desarrollado está compuesto por cinco unidades de medición independientes, cada una de ellas se compone de módulos que cuentan con hardware y software dedicados, las mediciones se aprecian una computadora personal mediante el software aplicativo propio.

El presente trabajo ha logrado incrementar la precisión y confiabilidad de las evaluaciones de los prototipos de incubadoras neonatales que se desarrolla en nuestra universidad, y diagnosticar incubadoras comerciales en centros de salud públicos [1]. También ha sido utilizado en el desarrollo de trabajos de tesis dentro del GIMDES [2].

FACULTAD DE
CIENCIAS E
INGENIERÍA

 PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título : Sistema electrónico para medir y evaluar la distribución de temperaturas en incubadoras neonatales
 Área : Electrónica # 645
 Asesor : Bruno Castillon Lévano
 Alumno : Jorge Luis Coello Durand
 Código : 20014109
 Fecha : 22 de octubre de 2008


Descripción y Objetivos

El presente tema tesis tiene como fin ayudar al personal encargado a evaluar el desempeño térmico de incubadoras neonatales, a través de un instrumento especializado. Este instrumento contará con un hardware y software dedicados, cada uno de ellos estará conformado por una etapa de adquisición de datos, procesamiento y transporte. El software poseerá una etapa adicional, la cual permite apreciar las mediciones en una computadora personal.

El objetivo principal es:

Diseñar e implementar un sistema electrónico que permita medir y evaluar la distribución de temperaturas en incubadoras neonatales.

Los objetivos específicos son:

- *Diseñar e implementar un hardware que permita la adquisición, procesamiento y transporte de datos de temperatura.
- *Diseñar e implementar un software que permita la adquisición, procesamiento, transporte y visualización de datos de temperatura.
- *Integrar el sistema electrónico de medición a partir de las distintas etapas de software y hardware.

 MÁXIMO 100 PÁGINAS

 PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU
SECCION ELECTRICIDAD Y ELECTRONICA


 Ing. ANDRES FLORES ESPINOZA
Coordinador de la Especialidad de Ingeniería

FACULTAD DE
CIENCIAS E
INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título : Sistema electrónico para medir y evaluar la distribución de temperaturas en incubadoras neonatales.

Índice

Introducción

1. PLANTEAMIENTO DEL PROBLEMA
2. MARCO CONCEPTUAL
3. DESARROLLO DEL SISTEMA
4. RESULTADOS

Conclusiones

Recomendaciones

Bibliografía

Anexos

PONTIFICIA UNIVERSIDAD CATOLICA DEL PERU
SECCION ELECTRICIDAD Y ELECTRONICA



Ing. ANDRÉS FLORES ESPINOZA
Coordinador de la Especialidad de Ingeniería Electrónica

MÁXIMO 100 PÁGINAS



Agradecimientos.



Son muchas las personas especiales a las que me gustaría agradecer por su amistad, apoyo, ánimo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo, otras en mis recuerdos y en el corazón. Sin importar en dónde estén o si alguna vez llegan a leer esta dedicatoria quiero darles las gracias por formar parte de mí, por todo lo que me han brindado y por todas sus bendiciones.

Mami, no me equivoco si digo que eres la mejor mamá del mundo, gracias por todo tu esfuerzo, tu apoyo y por la confianza que depositaste en mí. Gracias porque siempre has estado a mi lado. Te quiero mucho.

Papá, éste es un logro que quiero compartir contigo, gracias por ser mi papá y por creer en mí, quiero que sepas que eres el mejor y ocupas un lugar especial en mi corazón.

A mi gran hermana, ¿adivina qué? acabé y también itienes cara de panqué! Eres la mejor gran hermana que una gran hermano puede tener, igracias!.

A todos mis amigos, sin excluir a ninguno, pero en especial a mi hermano Fredy Pillaca igracias por darme tu amistad y apoyo Fredy!. A José Quenta, Raúl Chipana, Ana Moreno, Otto Velarde, Arlon Castillon, Jorge de la Torre, Anderson Lezameta, Johnny Liñan, mil gracias por todos los momentos que hemos pasado juntos y porque han estado conmigo siempre. A Liz Leyla que sin su apoyo incondicional no hubiera podido alcanzar esta meta, gracias por enseñarme a ser mejor persona. ¡Lección aprendida!.

A todos mis profesores no sólo de la carrera sino de toda la vida, mil gracias porque de alguna manera forman parte de lo que ahora soy. Especialmente a Bruno que está en esto conmigo, sin tu apoyo poco o nada de este trabajo se habría realizado, también a Angelo Velarde por su generosa ayuda y a Gustavo Kato por sus sabios consejos.

Al más especial de todos, a ti Señor porque hiciste realidad este sueño, por todo el amor con el que me rodeas y porque me tienes en tus manos.

ÍNDICE

RESUMEN.....	1
INTRODUCCIÓN.....	2
1 PLANTEAMIENTO DEL PROBLEMA.....	3
1.1 Introducción	3
1.2 Antecedentes.....	6
1.2.1 Estado del arte.....	6
1.2.2 Desarrollo de un sistema de adquisición de datos de temperatura para evaluación funcional de incubadoras.....	6
1.3 Justificación	6
1.4 Objetivos.....	6
1.4.1 Objetivo general.....	6
1.4.2 Objetivos específicos.....	6
2 MARCO CONCEPTUAL.....	7
2.1 Incubadoras neonatales.....	7
2.2 Neonatos críticos.....	8
2.3 Requerimientos para la seguridad en incubadoras neonatales (IEC 60601)....	9
3 DESARROLLO DEL SISTEMA.....	11
3.1 Diseño del sistema	11
3.2 Implementación del sistema	17
3.2.1 Hardware	17

3.2.2	Software.....	37
3.3	Pruebas del sistema	62
4	RESULTADOS	64
	CONCLUSIONES.....	68
	RECOMENDACIONES.....	69
	BIBLIOGRAFÍA.....	70



ÍNDICE DE FIGURAS

Figura 1.1 Diagrama de flujo del proceso de evaluación térmica cuando no se cuenta con un instrumento especializado.	5
Figura 2.1 Incubadora neonatal convencional. [6].	7
Figura 2.2 Distribución de los puntos de evaluación. Todos puntos se ubican a 10cm sobre la base de habitáculo.	9
Figura 2.3 Temperatura vs. Tiempo el error máximo en el estado estable debe ser de $\pm 0.5^{\circ}\text{C}$	10
Figura 3.1 Diagrama de bloques del sistema.	12
Figura 3.2 Diagrama de bloques principal del sistema destacando una unidad de medición de temperatura independiente.	13
Figura 3.3 Diagrama de bloques de una unidad de medición de temperatura independiente.	14
Figura 3.4 Módulo de adquisición de datos.	15
Figura 3.5 Módulo de procesamiento de datos y comandos	15
Figura 3.6 Módulo de gestión de tráfico de datos y comandos.	16
Figura 3.7 Módulo de interfaz de usuario.	16
Figura 3.8 Respuesta del sensor de temperatura YSI44018.	18
Figura 3.9 Izquierda: Diagrama eléctrico del sensor. Derecha: Forma física del sensor. [8]	18
Figura 3.10 Circuito esquemático sugerido por el fabricante. [8].	19
Figura 3.11 Error de medición producido por la regresión lineal.	20
Figura 3.12 Diagrama de bloques del conversor analógico digital ADS1110 [11].	22
Figura 3.13 Proceso de escalamiento	22

Figura 3.14 Diagrama esquemático con la configuración del puente de Wheatstone. .23

Figura 3.15 Respuesta del sensor YSI44018 usando la configuración de Wheatstone.
.....24

Figura 3.16 Resultado del proceso de escalamiento de la respuesta del sensor.24

Figura 3.17 Diagrama esquemático del circuito de adquisición de datos.25

Figura 3.18 Dibujo del circuito impreso del módulo de adquisición de datos.25

Figura 3.19 Fotografía de la tarjeta del módulo de adquisición terminada.26

Figura 3.20 Diagrama de bloques del módulo de procesamiento de datos y comandos.
.....27

Figura 3.21 Diagrama esquemático de la unidad de procesamiento de datos y
comandos.....28

Figura 3.22 Fotografía de la tarjeta del módulo procesamiento de datos y comandos
terminada.....29

Figura 3.23 Dibujo del circuito impreso del módulo de procesamiento de datos y
comandos.....29

Figura 3.24 Diagrama de bloques de la unidad de gestión de tráfico de datos y
comandos.....30

Figura 3.25 Diagrama esquemático del hardware del módulo de gestión de tráfico de
datos y comandos.31

Figura 3.26 Dibujo del circuito impreso del circuito de impreso del módulo adquisición
de datos y comandos.32

Figura 3.27 Fotografía del hardware del módulo de gestión de datos y comandos
terminada.....32

Figura 3.28 Diagrama esquemático del circuito de integración de hardware.34

Figura 3.29 Dibujo del circuito impreso de integración de módulos.....34

Figura 3.30 Fotografía de la tarjeta de integración de módulos terminada.....	35
Figura 3.31 Representación física de todo el hardware del sistema.	36
Figura 3.32 Modos de configuración del ADC. [11].	37
Figura 3.33 Diagrama de tiempos de la Comunicación con el ADC vía I2C [11]......	38
Figura 3.34 Diagrama esquemático con la configuración del puente de Wheatstone. .	40
Figura 3.35 Código de salida máximo, mínimo y nominal.	43
Figura 3.36 Límites del código de salida por efecto de la tolerancia.	44
Figura 3.37 Límites del código de salida por efecto de la temperatura de exposición..	45
Figura 3.38 Límites del código de salida por efecto de la tolerancia y la temperatura de exposición.....	45
Figura 3.39 Proceso de interpolación.	48
Figura 3.40 Diagrama de estados del sistema operativo de la unidad de procesamiento de datos y comandos.	51
Figura 3.41 Formato de los paquetes SPI.	52
Figura 3.42 Formato de los paquetes en el bus SPI.....	53
Figura 3.43 Formato de los paquetes en la interfaz RS232.....	54
Figura 3.44 Topología de la red de comunicaciones.	55
Figura 3.45 Capas de software aplicativo del módulo de interfaz de usuario.	57
Figura 3.46 Método para identificar la interfaz de comunicaciones específica.	58
Figura 3.47 Fragmentación de paquetes en la comunicación.	58
Figura 3.48 Desfragmentación de paquetes.....	59
Figura 3.49 Comprobación de la integridad de los paquetes recibidos.	60
Figura 3.50 Vista de la interfaz de usuario.	61

Figura 3.51 Representación Temperatura vs. Tiempo.	61
Figura 3.52 Datos exportados a MS EXCEL.	61
Figura 3.53 Primera prueba: error precisión de 0.18°C.	63
Figura 3.54 Segunda prueba: error de precisión máximo de 0.01°C.....	63
Figura 4.1 Tarjetas electrónicas completamente terminadas.	64
Figura 4.2 A la derecha la tarjeta de gestión de tráfico de datos y comandos, a la izquierda una de las tarjetas de procesamiento de datos y comandos.	65
Figura 4.3 Tarjeta de adquisición de datos y el sensor de temperatura ubicada dentro del habitáculo de una incubadora.....	65
Figura 4.4 Sensores dispuestos para su calibración.	66
Figura 4.5 Aplicación de software mostrando las variaciones de temperatura.....	66
Figura 4.6 Sistema completo evaluando el desempeño térmico de una incubadora....	67

RESUMEN

La mayoría de termómetros en el mercado presentan deficiencias para la medición de cambios de temperatura en incubadoras neonatales, como: lento tiempo de respuesta, dificultad para medir con precisión, difícil instalación, e inapropiados para registrar cambios de temperatura en varios puntos a la vez. Para superar estas limitaciones se desarrolló un sistema electrónico (hardware y software) especializado para incubadoras neonatales que permite realizar mediciones simultáneas de temperatura en cinco puntos, el rango de medición va desde los 0°C a $+100^{\circ}\text{C}$ con un error de medición de $\pm 0.015^{\circ}\text{C}$ en el rango de $+20^{\circ}\text{C}$ y $+55^{\circ}\text{C}$, una resolución de 0.01°C , con capacidad de visualizar las mediciones en tiempo real y registrar hasta 24 horas de evaluación en una computadora personal, que forma parte del sistema.

El sistema desarrollado está compuesto por cinco unidades de medición independientes, cada una de ellas se compone de módulos que cuentan con hardware y software dedicados, las mediciones se aprecian una computadora personal mediante el software aplicativo propio.

El presente trabajo ha logrado incrementar la precisión y confiabilidad de las evaluaciones de los prototipos de incubadoras neonatales que se desarrolla en nuestra universidad, y diagnosticar incubadoras comerciales en centros de salud públicos [1]. También ha sido utilizado en el desarrollo de trabajos de tesis dentro del GIMDES [2].

INTRODUCCIÓN

La carencia de un instrumento especializado que permita la evaluación térmica según la norma IEC 60601 [3] en incubadoras neonatales, y que brinde un resultado global concluyente, motivó el desarrollo del presente trabajo que consiste en un sistema electrónico de medición y evaluación de la distribución de temperaturas.

El sistema de medición y evaluación que se ha desarrollado cuenta con un soporte electrónico automático; dedicado a la adquisición de datos de temperatura de alta precisión, un software aplicativo de entorno visual intuitivo de fácil uso, capaz de exportar los resultados de la evaluación a otras aplicaciones para un análisis mucho más profundo. Con ello se logra: disminuir la carga laboral del personal; acortando el tiempo que demanda realizar esta actividad, además se ha logrado incrementar la precisión y confiabilidad en las evaluaciones de las incubadoras neonatales que se desarrollan en la Pontificia Universidad Católica del Perú (PUCP) y diagnosticar incubadoras en centros de salud públicos.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

1.1 Introducción

Uno de los principales problemas que existe dentro el desarrollo tecnológico en la línea de la ingeniería neonatal, es la carencia de un instrumento especializado que permita evaluar el desempeño térmico de incubadoras para recién nacidos, y además, sirva para contrastar los resultados obtenidos con los establecidos por normas internacionales.

Actualmente los termómetros con los que cuentan los diferentes laboratorios de la PUCP tienen varias limitaciones que dificultan el desarrollo de equipos como: la “Burbuja Neonatal” y el “Equipo Tele-médico de Soporte de Vida de Neonatos Críticos”, proyectos que actualmente están siendo desarrollos por el Grupo de Investigación y Desarrollo de Equipos Médicos y Sistemas (GIDEMS) dentro de la PUCP.

Las limitaciones más importantes de estos instrumentos son:

- a. Dificultad para registrar cambios de temperatura en varios puntos dentro de la incubadora neonatal de manera simultánea, puesto que los termómetros poseen solo un sensor.
- b. Lento tiempo de respuesta; pues, los termómetros son de propósito general.
- c. Dificultad para medir con precisión, debido a la interpretación visual del usuario del rango de medición y en algunos casos el número de dígitos es insuficiente.
- d. De difícil instalación; ya que, los sensores no están preparados para instalarse en incubadoras neonatales.

- e. Dificultad para procesar datos; porque se hace de forma semi manual y la probabilidad de cometer errores de cálculo es elevada, no existe un software especializado.
- f. Resultado general deficiente; no es posible apreciar de manera ordenada y global el diagnóstico del ensayo térmico al que se sometió la incubadora neonatal.

En conclusión; el problema es la falta de un instrumento especializado que permita medir y evaluar la distribución de temperaturas en incubadoras neonatales.

En la Figura 1.1 se presenta los procesos que se realizan para evaluar el desempeño térmico de una incubadora neonatal, en esta figura también se aprecian las limitaciones antes mencionadas.



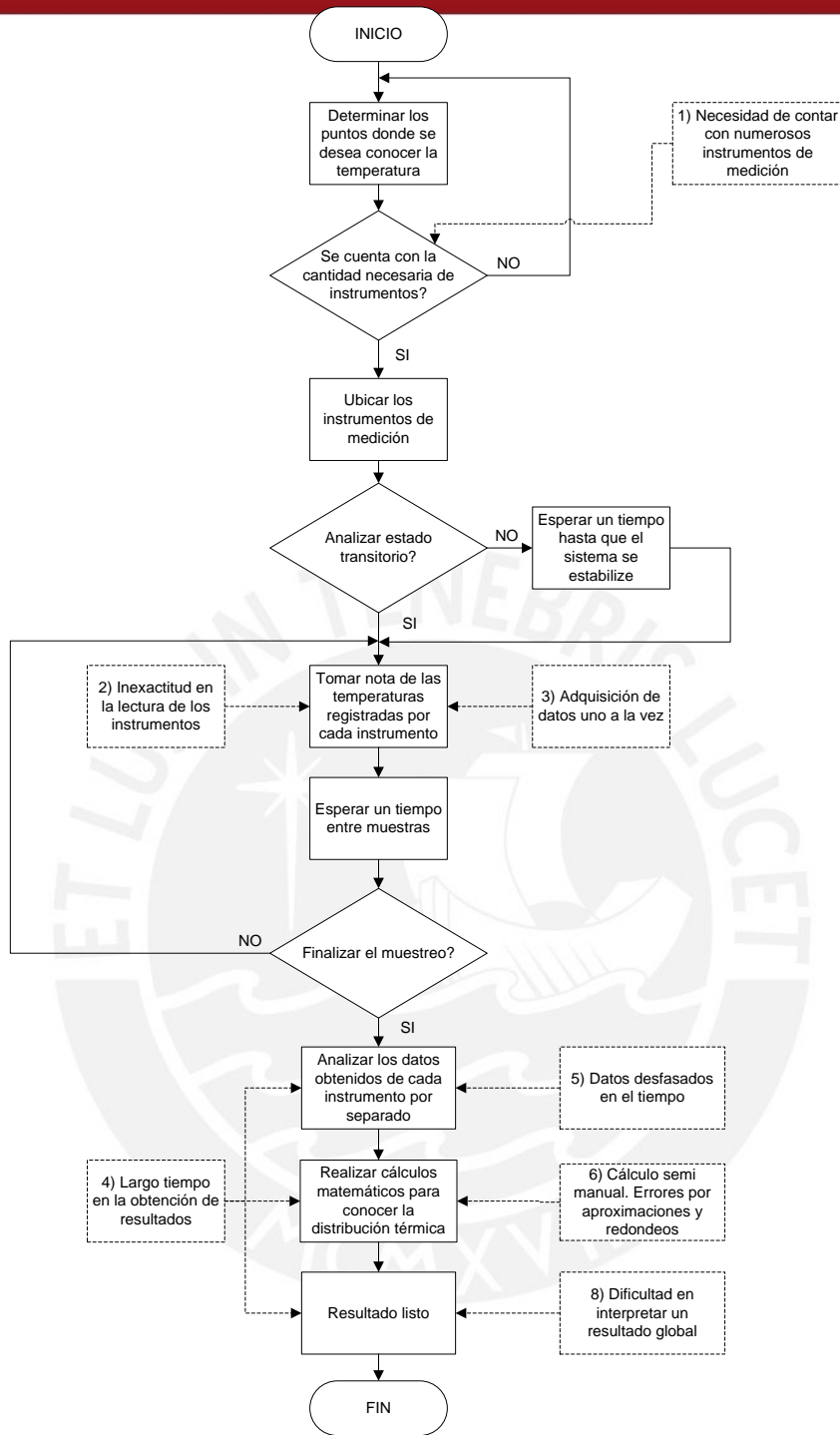


Figura 1.1 Diagrama de flujo del proceso de evaluación térmica cuando no se cuenta con un instrumento especializado.

1.2 Antecedentes

1.2.1 Estado del arte

El trabajo de L. Bychto y M. Malinski titulado: “*1D or 3D spatial temperature distribution-comparison (2006)*” [4]. Muestra la comparación de los resultados de cálculos de la distribución espacial de la temperatura y los espectros de las señales de un piezoeléctrico en una (1D) y tres (3D) dimensiones usando modelos térmicos. Estos resultados evidencian que un enfoque en una dimensión (1D) puede ser aplicado con éxito. [4].

1.2.2 Desarrollo de un sistema de adquisición de datos de temperatura para evaluación funcional de incubadoras.

Tesis desarrollada por el Ing. Miguel Obregón. [5].

1.3 Justificación

La temperatura es el parámetro de control más importante en una incubadora neonatal; por ello, el presente trabajo de tesis tiene como finalidad diseñar e implementar un instrumento electrónico especializado que ayude en la evaluación térmica de incubadoras neonatales; y que además, permita el desarrollo e investigación de dichos equipos en menos tiempo y a un menor costo.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar e implementar un sistema electrónico que permita medir y evaluar la distribución de temperaturas en incubadoras neonatales.

1.4.2 Objetivos específicos

- ❖ Diseñar e implementar un hardware que permita la adquisición, procesamiento y transporte de datos de temperatura.
- ❖ Diseñar e implementar un software que permita la adquisición, procesamiento, transporte y visualización de datos de temperatura.
- ❖ Integrar el sistema electrónico de software y hardware.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1 Incubadoras neonatales

Según el Centro Nacional de Excelencia Tecnológica en Salud, del gobierno mexicano se define una incubadora como:

La incubadora es un equipo médico que posee una cámara, dentro de la cual se coloca al neonato (Figura 2.1) con el fin de proporcionarle un medio ambiente controlado. Dependiendo del tipo de incubadora, estas pueden: controlar la temperatura, la humedad relativa y la concentración de oxígeno en el aire que rodea al paciente. Por lo general, las paredes de la cámara (capacete) se construyen con materiales transparentes, lo que permite aislar al paciente sin perder el contacto visual con él. [6].



Figura 2.1 Incubadora neonatal convencional. [6].

2.2 Neonatos críticos

Según el departamento de salud del gobierno Inglés:

La definición de neonato de estado crítico está vinculada a la definición de Neonatología de Cuidados Críticos de la Salud. Las definiciones contenidas en el informe del Departamento de Salud 2003 "Informe del Grupo de Revisión de Servicios Neonatal de Cuidados Intensivos", detallan la aplicación de las normas básicas para el cuidado del neonato crítico y son las siguientes:

A todos los pacientes (con exclusión de las madres) que están en una sala de Cuidados Intensivos para bebés a los que uno o más de las siguientes actividades de cuidados críticos se le aplican por un período mayor de 4 horas, se les pueden considerar neonatos de estado crítico. [7].

- 1 Asistencia respiratoria a través de un tubo traqueal.
- 2 Presión positiva continua en las vías nasales aéreas (CPAPN).
- 4 Intercambio de Transfusión.
- 5 Diálisis Peritoneal.
- 6 Infusión continua de inotrópico, vasodilatador pulmonar o prostaglandina.
- 7 Nutrición parental.
- 10 Terapia de oxígeno (Oxígeno terapia).
- 11 Síndrome de abstinencia neonatal.
- 12 Cuidados para un catéter intra-arterial o drenaje torácico.
- 13 Transfusión de intercambio diluido.
- 14 Cuidado para Traqueotomía por parte de personal de enfermería.
- 15 Cuidado para Traqueotomía por parte de personal externo.
- 16 Apnea recurrente.
- 17 Hemofiltración.
- 18 Monitoreo continuo.
- 19 Glucosa intravenosa y soluciones de electrolitos.
- 20 Realimentación tubos alimentados.
- 21 Atención de Barrera.
- 22 Fototerapia.
- 23 Monitoreo especial.
- 24 Observación a intervalos regulares.
- 25 Medicación vía intravenosa.

2.3 Requerimientos para la seguridad en incubadoras neonatales (IEC 60601)

La norma internacional IEC 60601 establece rangos de funcionamiento y exigencias para el desarrollo de equipos médicos. El apartado IEC 60601-2-19 está referido a la seguridad en incubadoras neonatales.

A continuación se hace referencia a los requerimientos térmicos de la norma. [3].

- ❖ Sección dos.- Requerimientos de seguridad: La temperatura ambiental donde operará la incubadora debe estar entre $+20^{\circ}\text{C}$ y $+30^{\circ}\text{C}$.
- ❖ Sección siete.- Protección contra temperaturas excesivas, fuego y otros riesgos, tales como errores humanos; La temperatura de las superficies internas metálicas que harán contacto con el bebe no excederán los $+40^{\circ}\text{C}$ y para otro tipo de materiales no excederán los $+43^{\circ}\text{C}$.
- ❖ Sección ocho.- Precisión en la operación de datos y protección contra salidas incorrectas: Durante la condición de estado estable la temperatura en el punto *A* no excederá por más de 0.5°C al promedio de temperaturas de los puntos *A, B, C, D* y *E* (Figura 2.2) se verificarán las pruebas de medición fijando el sistema de control de temperatura, en valores de 32°C a 36°C por periodos de una hora por lo menos.

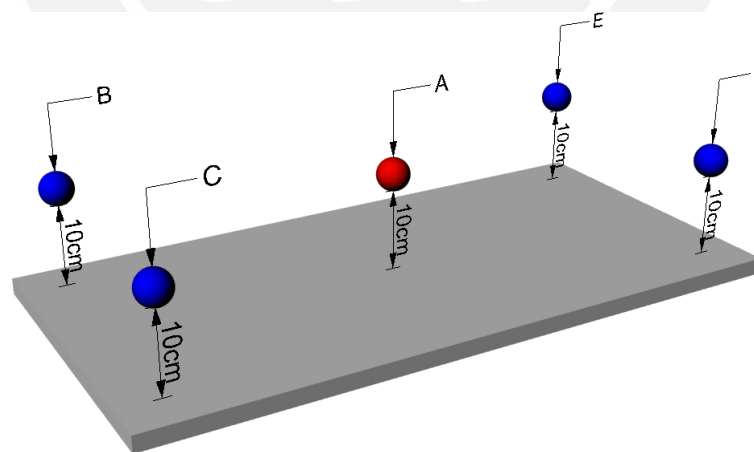


Figura 2.2 Distribución de los puntos de evaluación.

Todos puntos se ubican a 10cm sobre la base de habitáculo.

- ❖ Sección ocho.- El tiempo de calentamiento del equipo no excederá el 20% del tiempo de calentamiento especificado en las instrucciones. Se verificará mediante la siguiente prueba: con el control de temperatura fijado a 12°C por encima de la

temperatura ambiental; se encenderá empezando desde una condición fría y medirá el tiempo que demore en alcanzar 11°C por encima de la temperatura ambiental (Figura 2.3).

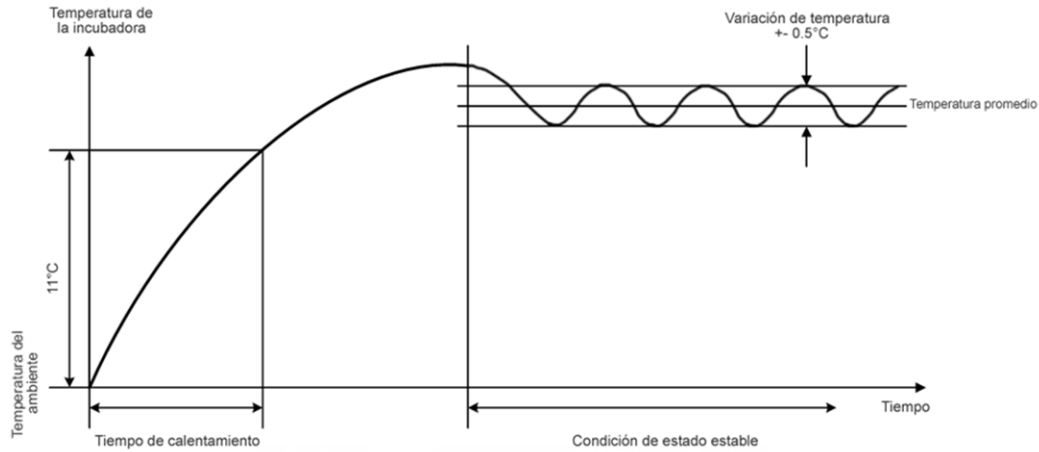


Figura 2.3 Temperatura vs. Tiempo
el error máximo en el estado estable debe ser de $\pm 0.5^{\circ}\text{C}$.

En conclusión, el instrumento de medición debe ser capaz de por lo menos:

- ❖ Registrar temperaturas con 0.5°C de resolución.
- ❖ Registrar temperaturas en el rango de $+20^{\circ}\text{C}$ hasta $+43^{\circ}\text{C}$
- ❖ Registrar temperaturas con una exactitud de $\pm 0.2^{\circ}\text{C}$
- ❖ Registrar temperaturas con una precisión de $\pm 0.5^{\circ}\text{C}$
- ❖ Contar al menos con siete sensores de temperatura, ampliables a nueve.
- ❖ Mostrar un resultado similar a la Figura 2.3 (temperatura vs. tiempo).

CAPÍTULO 3

DESARROLLO DEL SISTEMA

Introducción

El sistema se desarrolló para permitir al usuario hacer mediciones simultáneas de temperatura y registrar los cambios en nueve puntos en el interior del habitáculo de una incubadora neonatal.

3.1 Diseño del sistema

El sistema está representado en la Figura 3.1, y está compuesto por unidades de medición de temperatura independientes (Figura 3.2), cada unidad está conformada por módulos con funciones específicas (Figura 3.3).

Una unidad de medición es capaz de registrar temperaturas de forma independiente, haciendo uso de los módulos que la componen; sin embargo, existen módulos de uso compartido como el módulo de gestión de tráfico de datos y comandos y el módulo de interfaz de usuario.

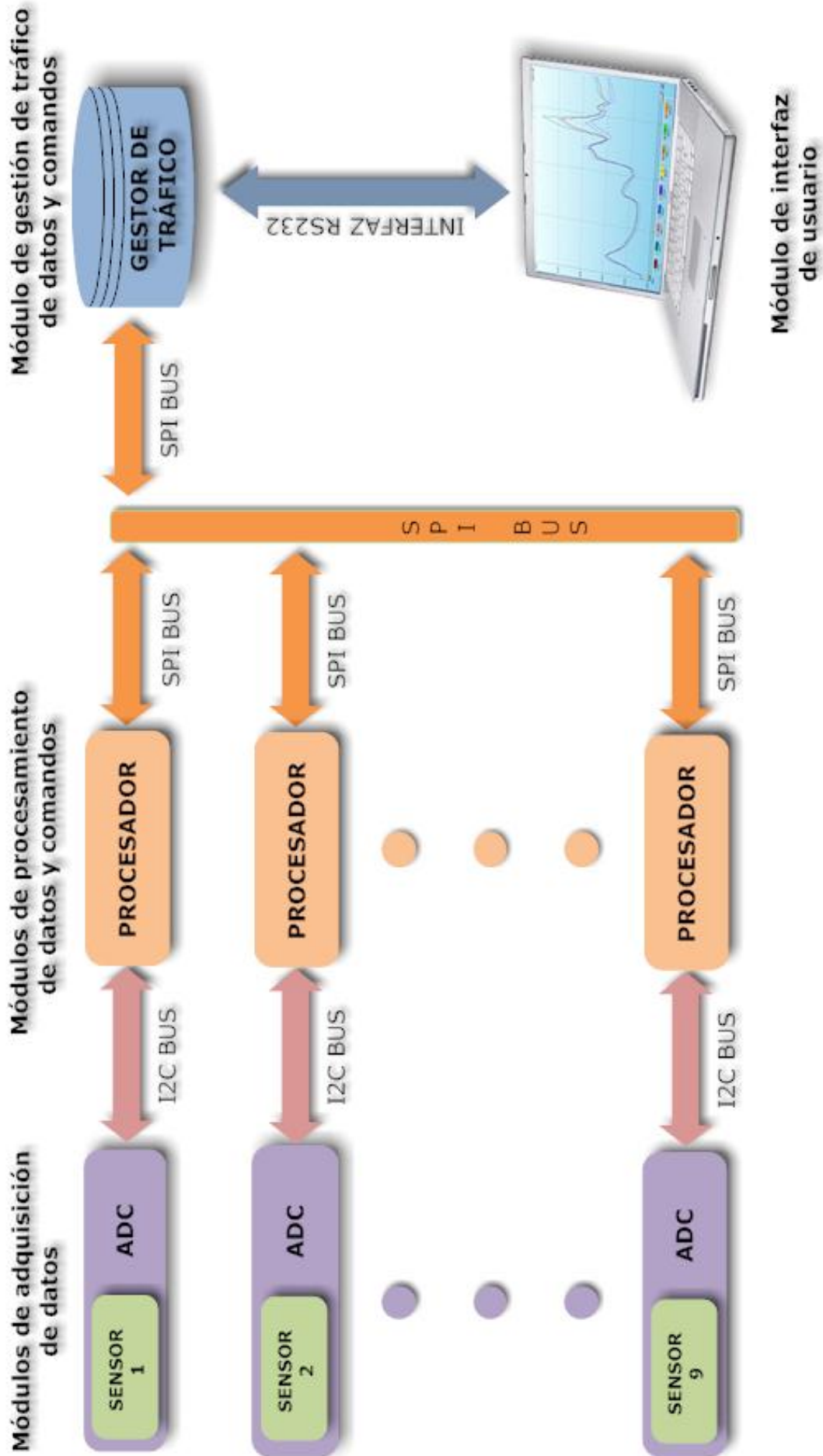


Figura 3.1 Diagrama de bloques del sistema.

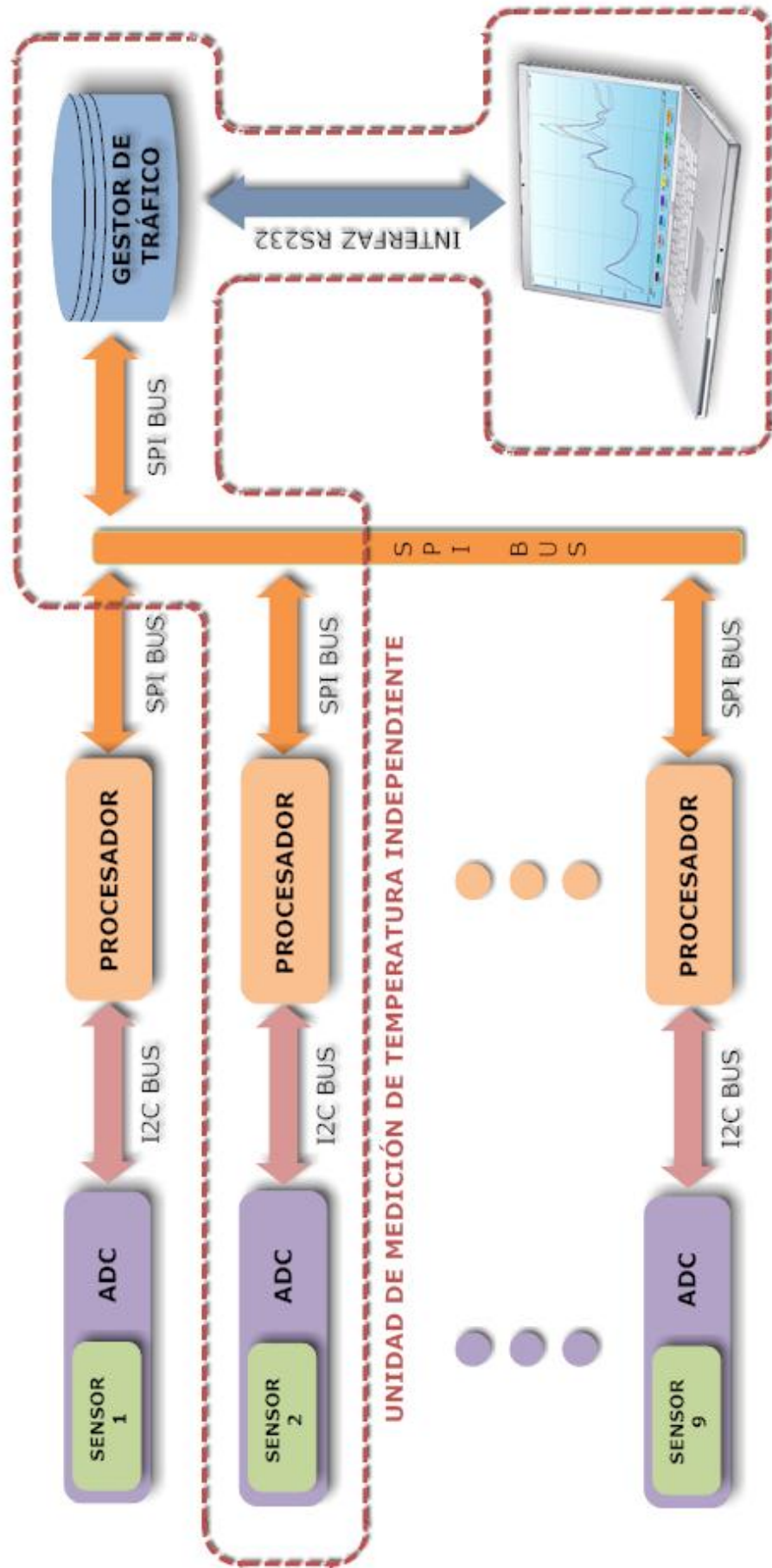


Figura 3.2 Diagrama de bloques principal del sistema destacando una unidad de medición de temperatura independiente.

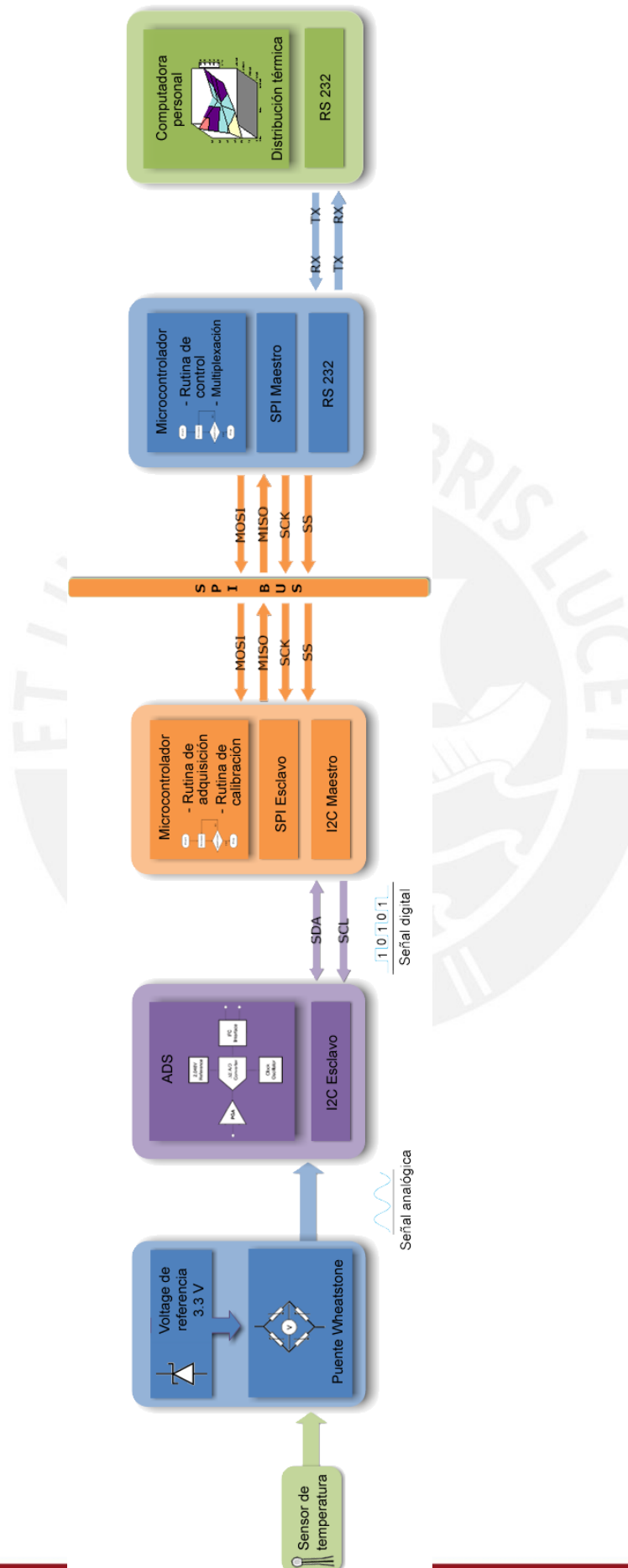


Figura 3.3 Diagrama de bloques de una unidad de medición de temperatura independiente.

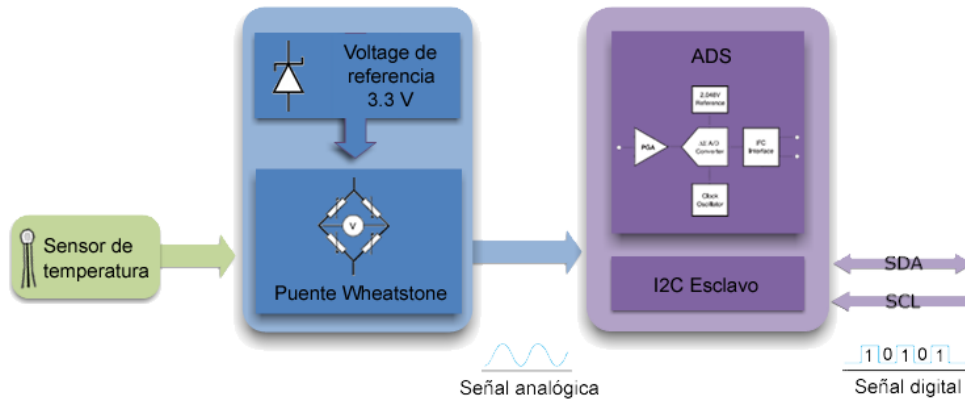


Figura 3.4 Módulo de adquisición de datos.

La Figura 3.4 representa al módulo de adquisición de datos, el cual está compuesto por el sensor de temperatura, un circuito eléctrico de acondicionamiento de señales y un convertor analógico - digital. Al cambiar la temperatura del sensor éste cambia sus propiedades eléctricas esto se refleja en una señal eléctrica, la cual, es inmediatamente acondicionada a niveles de voltaje adecuados para ser registrada por el convertor analógico – digital, que al concluir una conversión ubica el resultado en el bus de comunicaciones I2C.

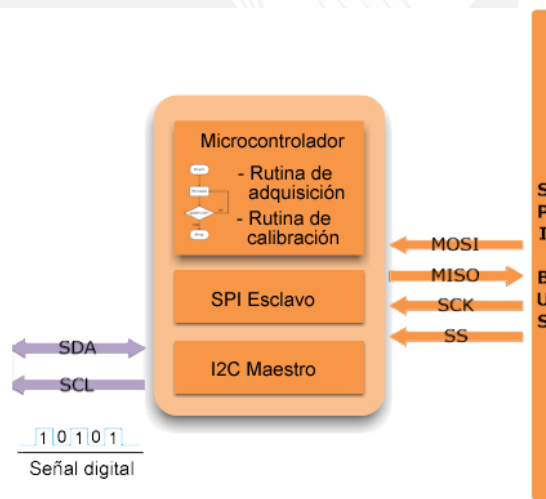


Figura 3.5 Módulo de procesamiento de datos y comandos

La Figura 3.6 representa al módulo de procesamiento de datos y comandos, el cual se encarga del procesamiento de datos y comandos, lee el dato del bus I2C que proporcionó el módulo anterior, luego lo convierte a una magnitud escalar en grados centígrados, termina colocando este dato de temperatura en un segundo bus de comunicaciones SPI. Este módulo ejecuta una serie de comandos que están

relacionados a la adquisición de datos de temperatura, ajustes en la conversión, comandos del usuario, manejo de errores y otros procedimientos que serán detallados en el capítulo de software,

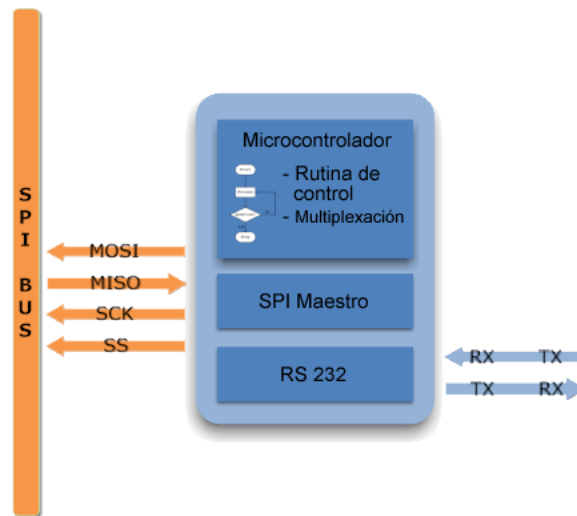


Figura 3.6 Módulo de gestión de tráfico de datos y comandos.

La Figura 3.6 representa al módulo de gestión de datos y comandos, el cual se encarga de gestionar el tráfico de datos y comandos entre el bus SPI y la interfaz de comunicaciones RS232. Recolecta, empaqueta y envía los datos de temperatura provenientes de los módulos de procesamiento de datos y comandos enviándolos al módulo de interfaz de usuario, así mismo, direcciona y envía los comandos provenientes del módulo de interfaz de usuario a los módulos de procesamiento de datos que corresponda.

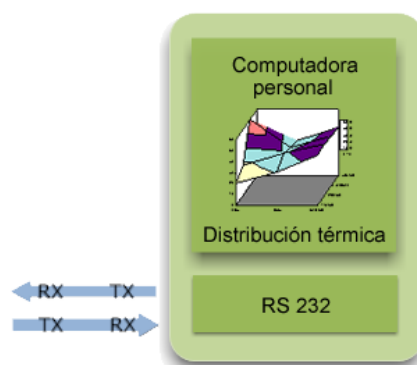


Figura 3.7 Módulo de interfaz de usuario.

La Figura 3.7 representa al módulo de la interfaz de usuario, el cual consiste de un aplicativo de software que recibe y procesa los paquetes de datos provenientes del

módulo de gestión de tráfico de datos y comandos; conforma una imagen dinámica que permite visualizar las mediciones de temperatura en tiempo real, modifica parámetros en los módulos de procesamiento de datos y comandos, identifica los puntos de medición; calibra los sensores, guarda y revisa mediciones realizadas, exporta datos a otros aplicativos como MS EXCEL y MATLAB, etc.

3.2 Implementación del sistema

La implementación se realizó de forma modular. Cada módulo que compone una unidad de medición independiente de temperatura, cuenta con un hardware y software dedicado.

3.2.1 Hardware

Comprende el soporte físico de los módulos de: adquisición de datos, procesamiento de datos y comandos; gestión de tráfico de datos y comandos e interfaz de usuario. Cada módulo cuenta con un hardware dedicado.



3.2.1.1 Módulo de adquisición de datos

En este módulo se usó el sensor de temperatura YSI44018, este sensor de uso médico se caracteriza por tener $\pm 0.2^{\circ}\text{C}$ de error por intercambiabilidad. Está compuesto de dos resistencias variables en función de la temperatura de exposición. [8].

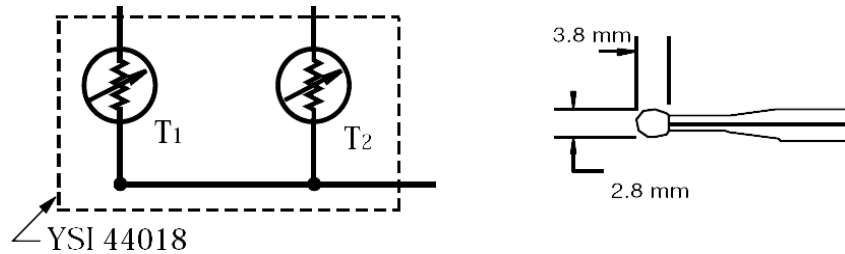


Figura 3.9 Izquierda: Diagrama eléctrico del sensor. Derecha: Forma física

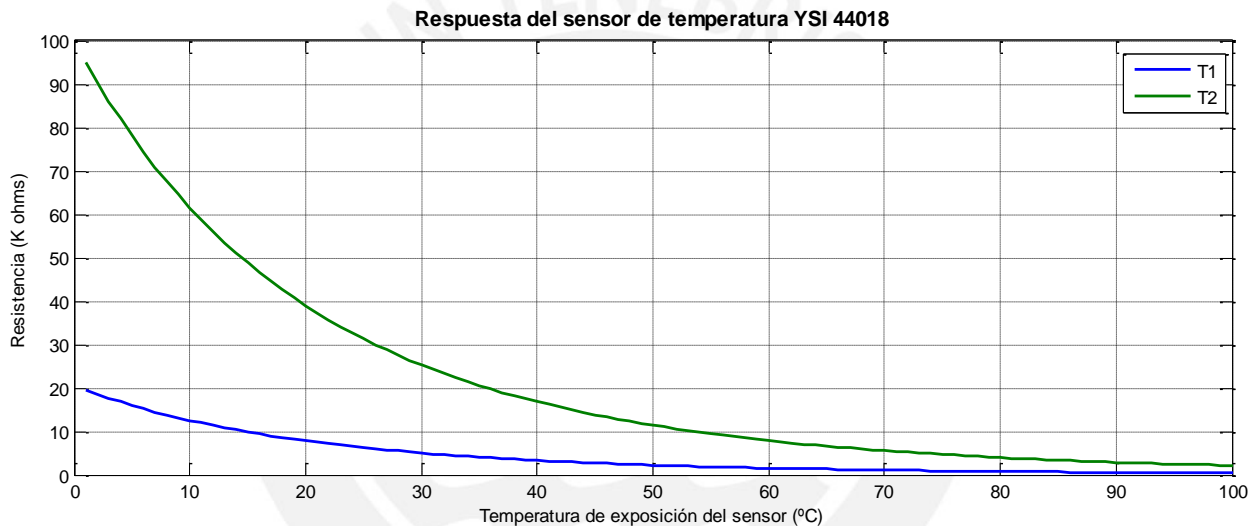


Figura 3.8 Respuesta del sensor de temperatura YSI44018.

La Figura 3.9 muestra la relación entre los valores de las resistencias internas $T1$ y $T2$ y la temperatura de exposición del sensor (valores proporcionados por el fabricante [8]). Como se aprecia claramente la respuesta del sensor no es lineal.

El fabricante recomienda usar el circuito mostrado en la Figura 3.10. [8] Las consideraciones para el cálculo de las resistencias $R1$ y $R2$ son el rango de trabajo del sensor y el valor del voltaje de referencia V_{ref} . El fabricante recomienda entre otras opciones: $R1 = 5700 \Omega$, y $R2 = 12000 \Omega$, ambas resistencias tienen una tolerancia de $\pm 1\%$, esto permite un rango de trabajo de -5°C a $+45^{\circ}\text{C}$ con una desviación máxima de $\pm 0.065^{\circ}\text{C}$ [8]. Para el propósito de este trabajo de tesis el rango mencionado no es suficiente; ya que, planea hacer mediciones desde 0°C hasta $+100^{\circ}\text{C}$.

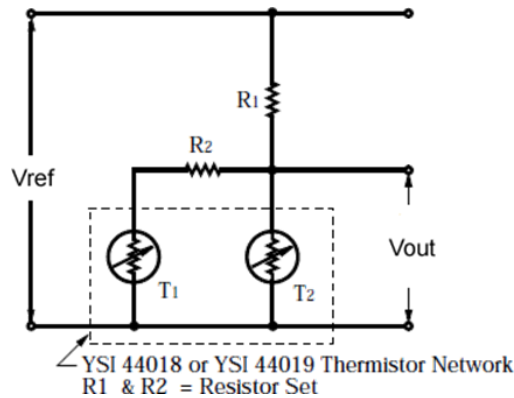


Figura 3.10 Circuito esquemático sugerido por el fabricante. [8]

Manteniendo la recomendación del fabricante y haciendo $V_{ref} = 3.3V$, se resolvió el circuito de la Figura 3.10. Ya que los valores de T_1 y T_2 son conocidos para cada grado centígrado, se obtuvieron los valores de V_{out} en función de la temperatura, este análisis se hizo implementando una rutina en la herramienta de software MATLAB [9].

Cabe señalar que los efectos en los componentes derivados por la tolerancia y la temperatura de exposición sobre valor de V_{out} serán analizados en el la sección 3.2.1.2 “Módulo de procesamiento de datos y comandos”.

Por el momento se determinaran las magnitudes nominales considerando las resistencias de capa metálica de alta precisión y gran estabilidad del fabricante VISHAY [10], de tolerancia $\pm 1\%$ y coeficiente térmico del orden de los $15ppm/^{\circ}C$, así como, el voltaje de referencia con una tolerancia de $\pm 0.04\%$ y coeficiente térmico de $3ppm/^{\circ}C$.

Luego de resolver el circuito de la Figura 3.10 se obtuvieron los resultados que se muestran en la Figura 3.11. El gráfico A muestra la curva de respuesta del circuito de la Figura 3.10 Circuito esquemático sugerido por el fabricante. , la cual aparenta ser lineal. El gráfico B muestra una recta y su ecuación, producto de la regresión lineal hecha a la curva del gráfico A. El gráfico C, muestra el error derivado de la regresión lineal, es decir, si se asume que la respuesta del sensor corresponde a la recta $Y = -0.019092 * X + 2.6655$; se aprecia en este gráfico que el error no es constante y está en función de la temperatura de exposición, llegando a su máximo valor de $-1.5^{\circ}C$ alrededor de los $+100^{\circ}C$.

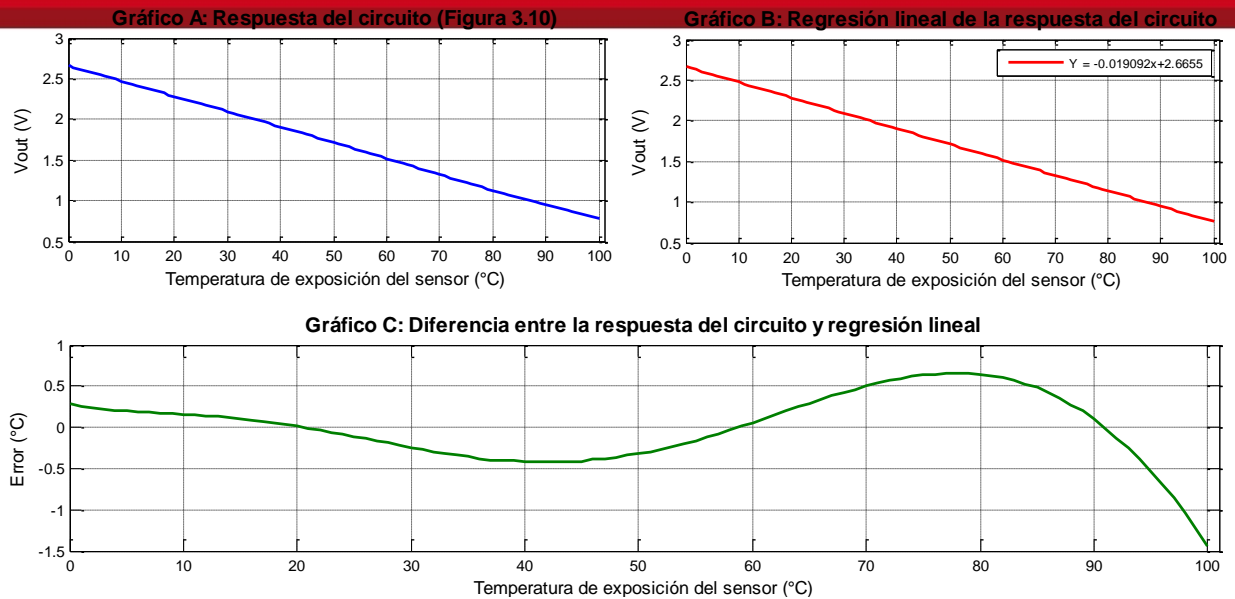


Figura 3.11 Error de medición producido por la regresión lineal.

Tomando como base el Gráfico A de la Figura 3.11 se deduce que para medir temperaturas desde 0 °C hasta 100 °C se tendrá que medir voltajes desde 0.7 V hasta 2.7 V aproximadamente. Debido a que existe un error variable y de magnitud considerable no se usará la recta del Gráfico B para representar el comportamiento del sensor.

Desarrollando el circuito de la Figura 3.10 para los casos en que la temperatura registrada es 0°C y +100°C, tenemos:

$$V_{out} = \frac{V_{ref} * (R2 + T1) * T2}{R1 * (R2 + T1 + T2) + T2 * (R2 + T1)}$$

Para T = 0°C

$$T1 = 19590 \Omega, T2 = 94980 \Omega, R1 = 5700 \Omega, R2 = 12000 \Omega \text{ y } V_{ref} = 3.3V$$

$$V_{out @ 0^\circ C} = 2.66 V$$

Para T = 100°C

$$T1 = 407.6 \Omega, T2 = 2069 \Omega, R1 = 5700 \Omega, R2 = 12000 \Omega \text{ y } V_{ref} = 3.3V$$

$$V_{out @ 100^\circ C} = 0.83 V$$

El rango de voltaje a cuantificarse es:

$$\Delta V_{0^{\circ}\text{C a } 100^{\circ}\text{C}}^{out} = 1.8217 \text{ V}$$

Ya que la resolución de la medición es de 0.01°C y el rango de medición es de $+100^{\circ}\text{C}$, se requieren 10000 niveles de discriminación en el conversor analógico – digital (ADC), por consiguiente la unidad mínima de cuantificación debe ser:

$$1 \text{ LSB}_{necesario} = 0.00018217 \text{ V/Bit}$$

Este requerimiento se cumple usando un ADC de por lo menos 14 BITS de resolución, con el cual se obtienen 16384 niveles de discriminación.

Para definir las características del conversor analógico digital se tomó en cuenta los siguientes parámetros: tiempo de respuesta del sensor de temperatura igual a 6.3 segundos [8], el voltaje de referencia de 3.3V , el rango de medición de temperatura; desde 0°C hasta $+100^{\circ}\text{C}$, resolución de medición de 0.01°C .

Para definir la tasa de muestreo, según el teorema de Nyquist, se tiene:

$$F_{muestreo\ mínima} = \frac{1}{6.3} * 2, \quad F_{muestreo\ mínima} = 0.3174 \text{ Hz}$$

Ya que el rango de medición es desde 0°C hasta $+100^{\circ}\text{C}$ la variación de voltaje correspondiente a este rango es:

$$\Delta V_{0^{\circ}\text{C a } 100^{\circ}\text{C}}^{out} = 1.8217 \text{ V}$$

El ADC, que se eligió es el ADS1110 [11] . por contar con las siguientes características: tener integrado un voltaje de referencia de 2.048V con un error de $\pm 0.05\%$, un amplificador de ganancia programable, entradas diferenciales, resolución de conversión de hasta 16 BITS, una frecuencia de muestreo de hasta 240 SPS e interfaz de comunicaciones I2C para el control y el retorno de datos. En la Figura 3.12 se aprecia el diagrama de bloques de este dispositivo.

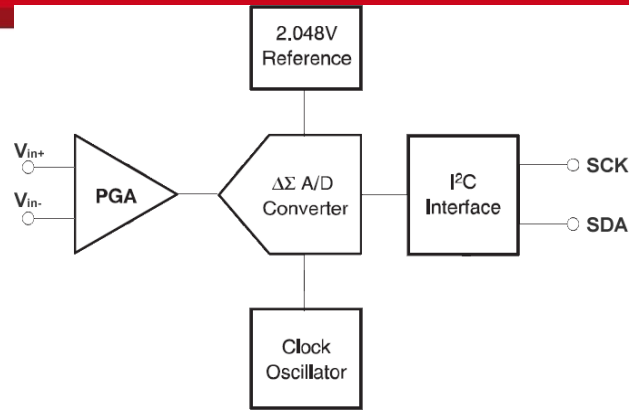


Figura 3.12 Diagrama de bloques del conversor analógico digital ADS1110 [11].

El circuito de la Figura 3.10 determina que para medir temperaturas desde 0°C hasta $+100^{\circ}\text{C}$ se tendrá que medir voltajes desde 0.7V hasta 2.7V , rango que no podrá cuantificar directamente el ADC seleccionado; porque, la referencia de voltaje interna es de solo 2.048V , esto impediría hacer mediciones por encima de este valor. La Figura 3.13 permite apreciar con más detalle este inconveniente. En la columna (A) se observa la escala completa de conversión del ADC; la columna (B) muestra el rango que sería necesario que el ADC pueda cuantificar (desde $+0.7\text{V}$ hasta $+2.7\text{V}$) y en la columna (C) se distingue el rango de conversión útil con el cual se podría ponderar temperaturas desde $+35^{\circ}\text{C}$ hasta $+100^{\circ}\text{C}$. En resumen, la columna (D) muestra cuatro segmentos: el segmento (i) representa el rango imposible de cuantificar, el segmento (ii) representa el rango útil de conversión y los segmentos (iii) y (iv) son rangos disponibles pero no usados.

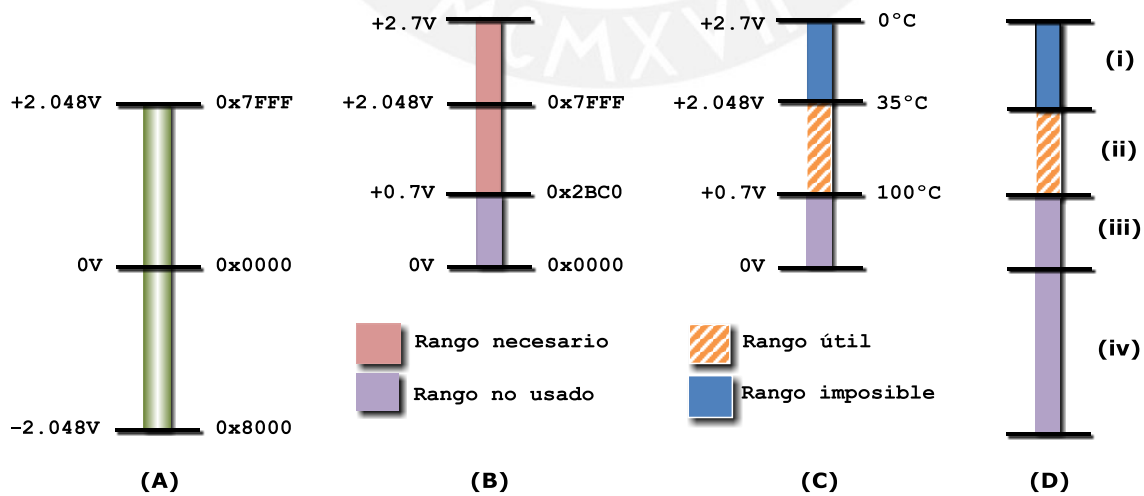


Figura 3.13 Proceso de escalamiento

Es posible mejorar la respuesta del circuito superando el inconveniente mencionado con el uso de un puente de Wheatstone como se muestra en la Figura 3.14.

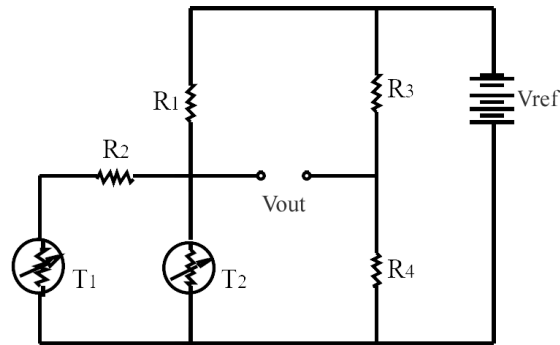


Figura 3.14 Diagrama esquemático con la configuración del puente de Wheatstone.

Se pueden desplazar los segmentos (i) y (ii) a niveles de voltajes adecuados, para lograr un rango de conversión suficiente.

Fijando: $V_{ref} = 3.3V$ $R3$ y $R4$ deben cumplir la condición:

$$2.7 = 3.3 * \frac{R4}{R3 + R4}, \quad \text{entonces } R4 = \frac{9}{2} * R3$$

Si $R3 = 1.3 K\Omega$ entonces $R4 = 5.85 K\Omega$

Resolviendo el circuito de la Figura 3.14, se obtuvieron los resultados que se muestran en la Figura 3.15. En el Gráfico A, se aprecia la respuesta del circuito sugerido por el fabricante; donde el rango de voltaje necesario va desde $+2.7V$ hasta $+0.7V$ aproximadamente. En el Gráfico B se muestra la respuesta del circuito con la configuración del puente de Wheatstone donde se aprecian dos cambios importantes: el primero refleja que el nuevo rango de voltaje va desde $0V$ hasta $2V$ aproximadamente y el segundo muestra que la relación voltaje – temperatura se ha invertido, como resultado de considerar $(-)Vout$ en vez de $(+)Vout$, aprovechando la entrada de tipo diferencial del ADC.

Gráfico A: Respuesta del circuito sugerido por el fabricante (Figura 3.10)

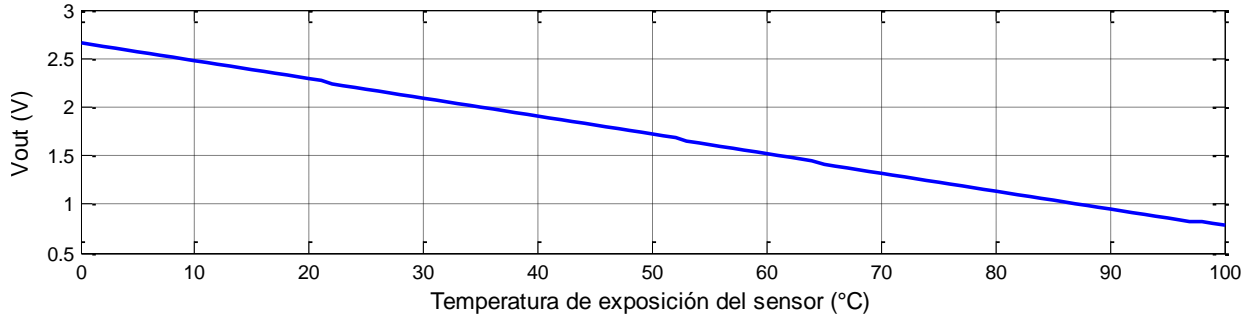


Gráfico B: Respuesta del circuito con la configuración Wheatstone (Figura 3.14)

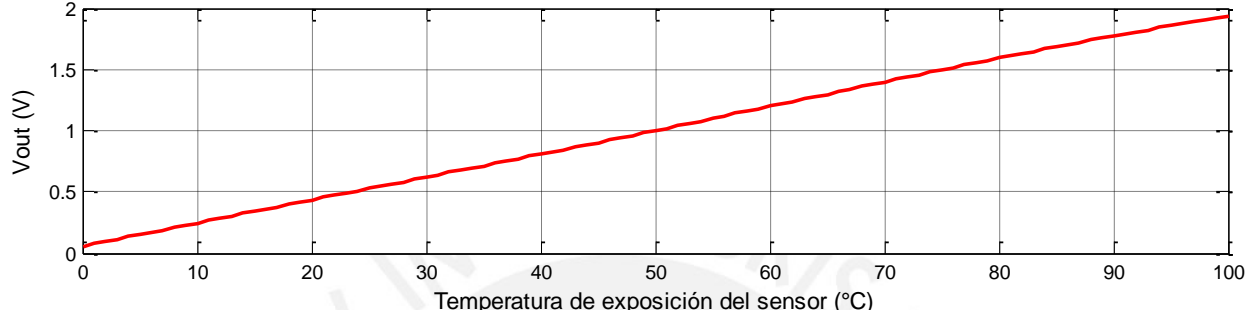


Figura 3.15 Respuesta del sensor YSI44018 usando la configuración de Wheatstone.

En conclusión, usando el puente de Wheatstone se logró acondicionar la señal del sensor tal como se muestra en la Figura 3.16, columna (C).

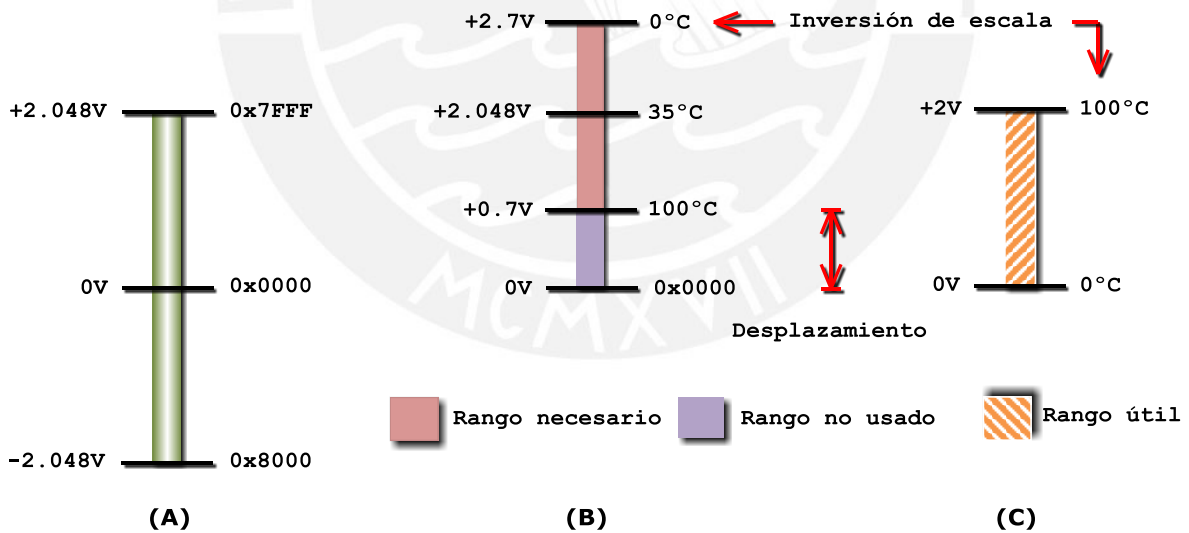


Figura 3.16 Resultado del proceso de escalamiento de la respuesta del sensor.

Definidos los componentes se completó el diagrama esquemático (Figura 3.17) del módulo de adquisición de datos.

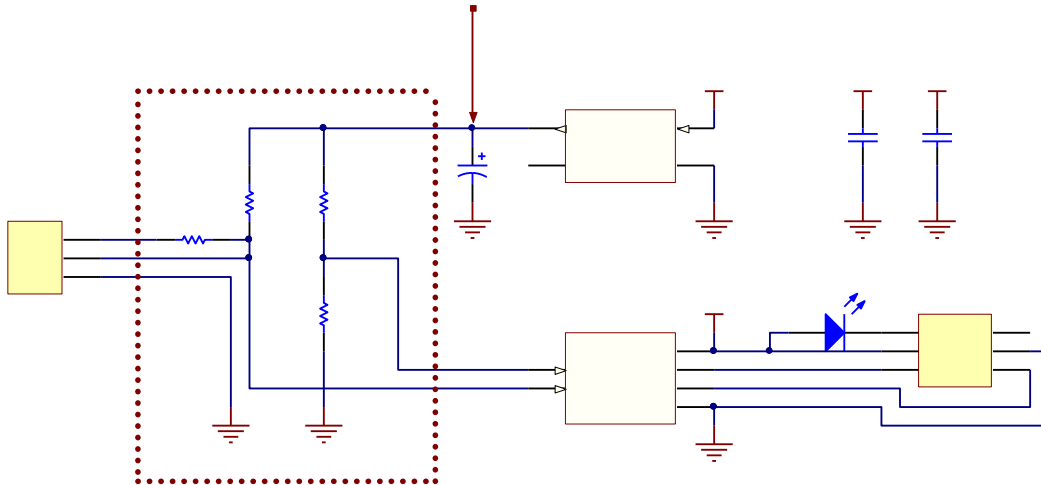


Figura 3.17 Diagrama esquemático del circuito de adquisición de datos.

La Figura 3.18 muestra el dibujo de las dos caras del circuito impreso del módulo de adquisición de datos.

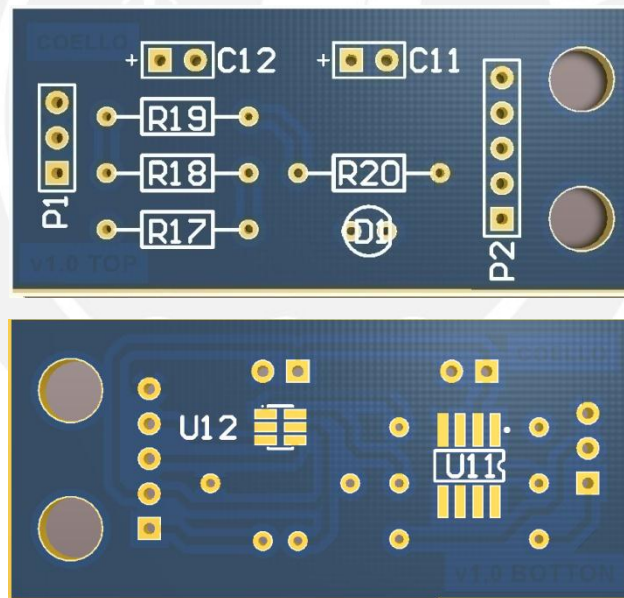


Figura 3.18 Dibujo del circuito impreso del módulo de adquisición de datos.

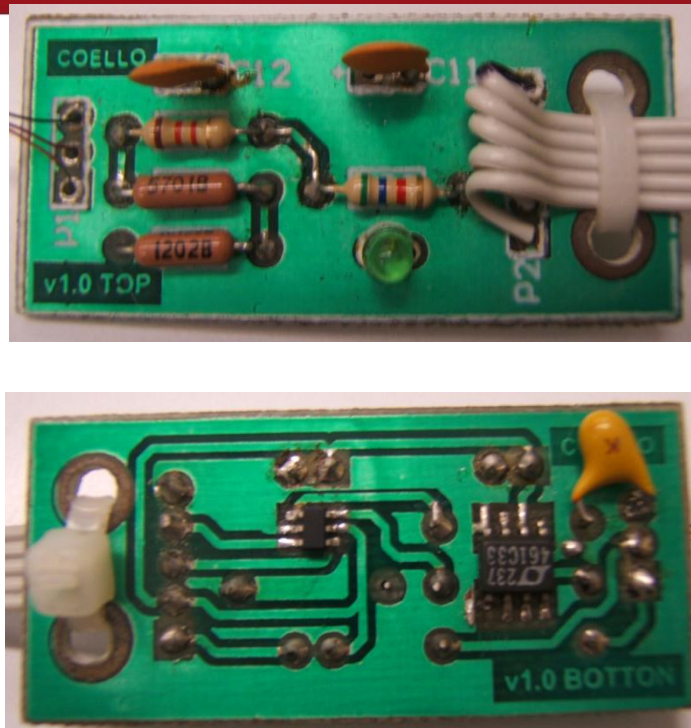


Figura 3.19 Fotografía de la tarjeta del módulo de adquisición terminada.

3.2.1.2 Módulo de procesamiento de datos y comandos

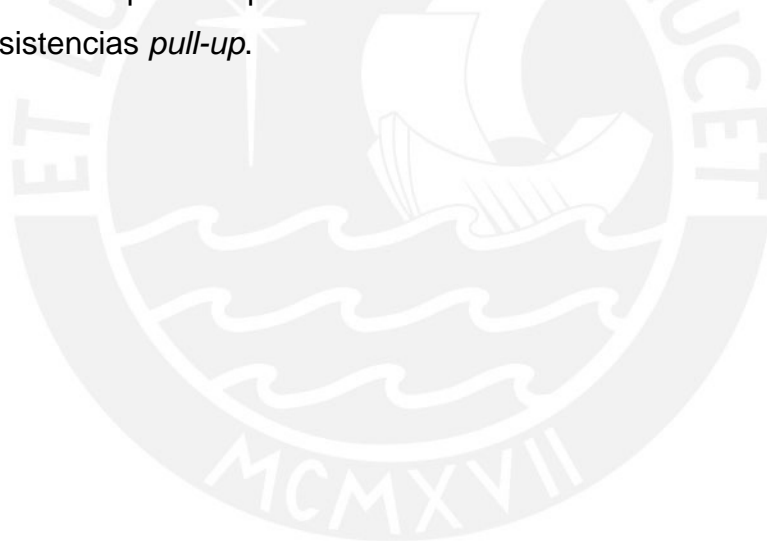
El hardware de este módulo está conectado con el hardware del módulo de adquisición de datos y el hardware del módulo de gestión de tráfico de datos y comandos.

De acuerdo al diseño principal, este módulo brinda el soporte físico a la unidad de procesamiento de datos y comandos, conformada básicamente por un microcontrolador ATMEGA8, el cual cuenta con una unidad de comunicaciones I2C y otra para comunicaciones SPI. En la Figura 3.20 se muestra el diagrama de bloques de esta unidad, la cual integra una fuente de voltaje regulada de 5V basada en el regulador LM7805, un puerto de programación ISP y un circuito integrado CD4053 del cual se usan 4 de los 6 *buffers* que dispone para la protección entre los puertos del microcontrolador y el bus de comunicaciones SPI.



Figura 3.20 Diagrama de bloques del módulo de procesamiento de datos y comandos.

La Figura 3.21, muestra el diagrama esquemático de la unidad de procesamiento de datos y comandos. Se aprecia claramente las partes antes mencionada, es conveniente señalar el uso de las resistencias de *pull-up* para las líneas SS y MOSI, y las de *pull-down* para las líneas MISO y SCK del bus de comunicaciones SPI; de igual modo las dos líneas que componen el bus de comunicaciones I2C (SDA y SCL) cuentan con resistencias *pull-up*.



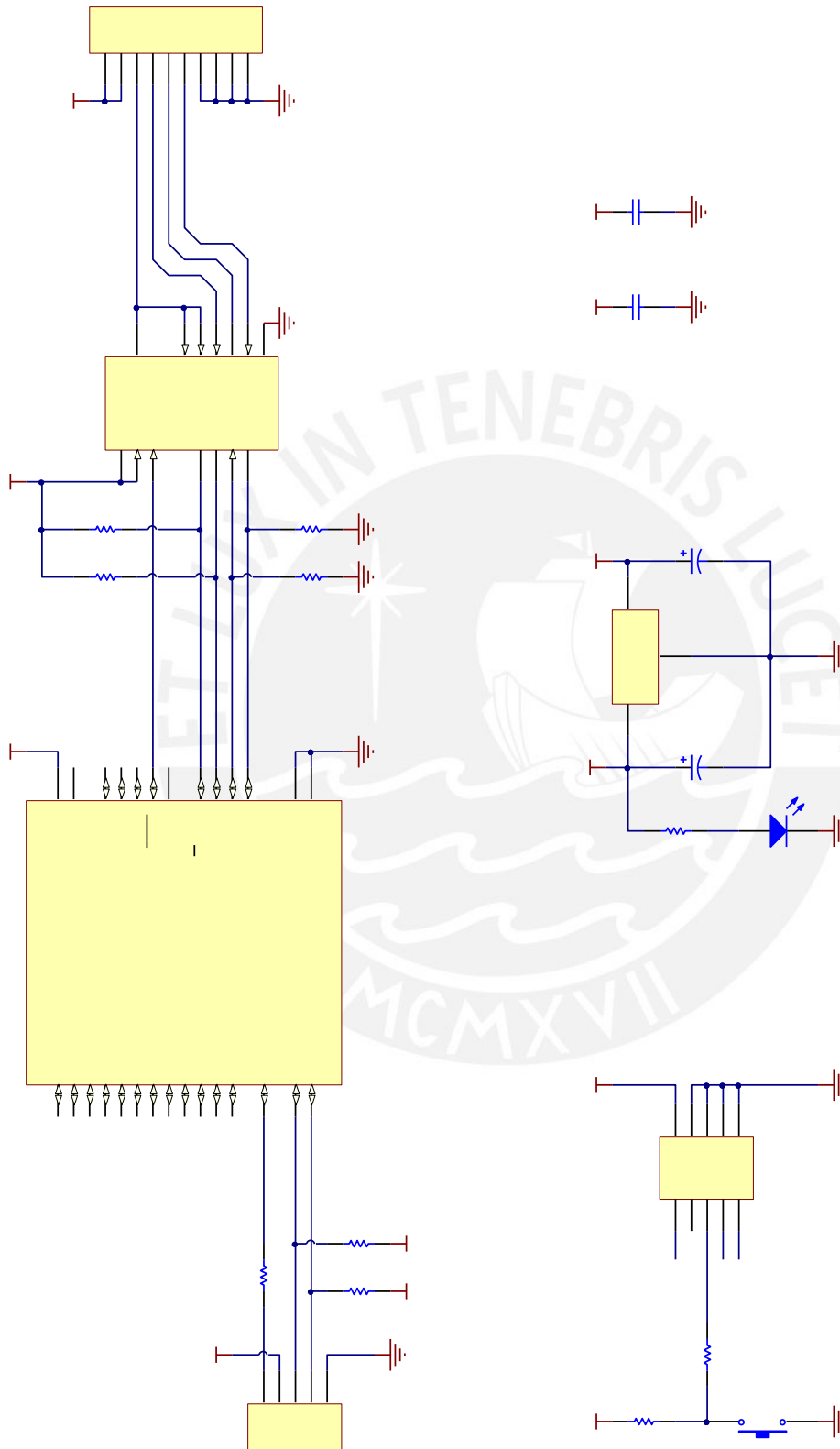


Figura 3.21 Diagrama esquemático de la unidad de procesamiento de datos y comandos.

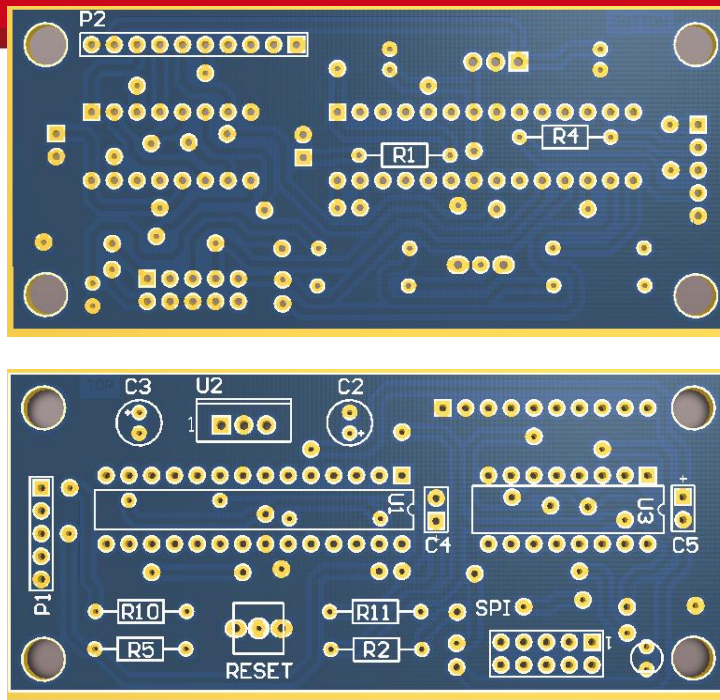


Figura 3.23 Dibujo del circuito impreso del módulo de procesamiento de datos y comandos.

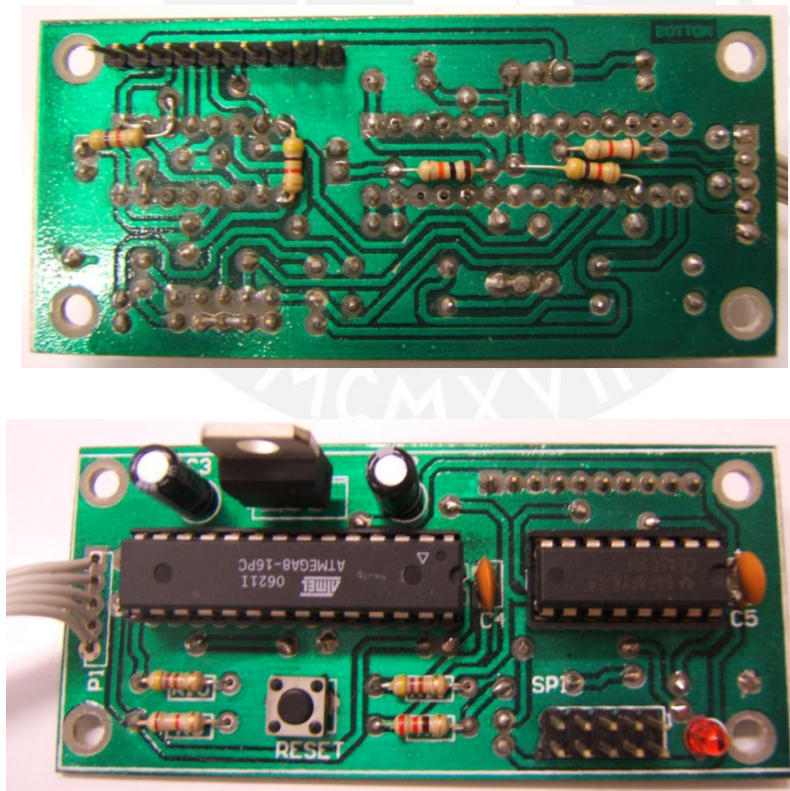


Figura 3.22 Fotografía de la tarjeta del módulo procesamiento de datos y comandos terminada.

3.2.1.3 Módulo de gestión de tráfico de datos y comandos

Este módulo se conecta con las unidades de procesamiento de datos y comandos, también con el módulo de interfaz de usuario.

Se encarga de recolectar los datos de temperatura de cada módulo de procesamiento usando comandos de interrogación a través del bus de comunicaciones SPI, luego de recolectar los datos disponibles los empaqueta y envía a una computadora personal usando la interfaz RS232, para ello, usa los periféricos de comunicaciones USART y SPI. Se usó el microcontrolador ATMEGA8, pues cumple los requerimientos necesarios; una fuente regulada de 5V, basada en el regulador LM7805; un puerto de programación ISP; una interfaz TTL/RS232, basada en el circuito MAX232, 4 de los 6 buffers del circuito integrado CD4053 para la protección de los puertos del microcontrolador que se conectan al bus de comunicaciones SPI; un codificador de 4 BITS, basado en el circuito integrado 74HC42, para direccionar a cada una de las tarjetas de los módulos de procesamiento de datos y comandos que se conectan al bus SPI. En la Figura 3.24 se puede apreciar el diagrama de bloques del hardware de este módulo.

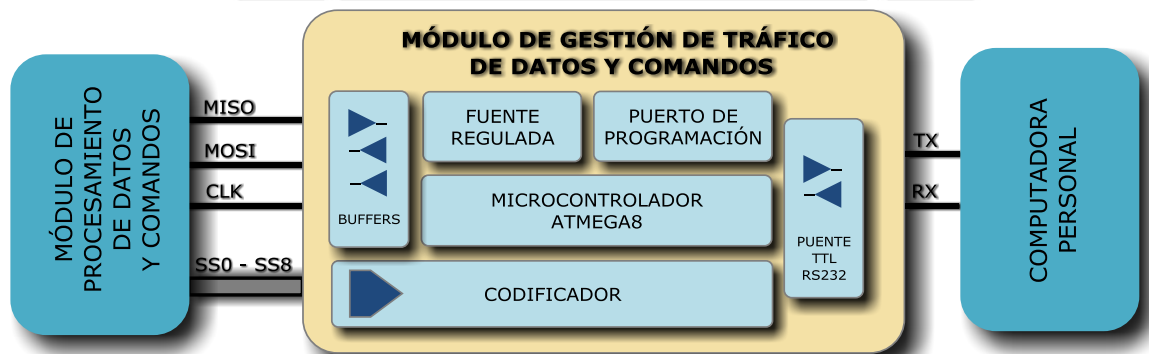


Figura 3.24 Diagrama de bloques de la unidad de gestión de tráfico de datos y comandos.

La Figura 3.25 muestra el diagrama esquemático con todos los componentes antes mencionados.

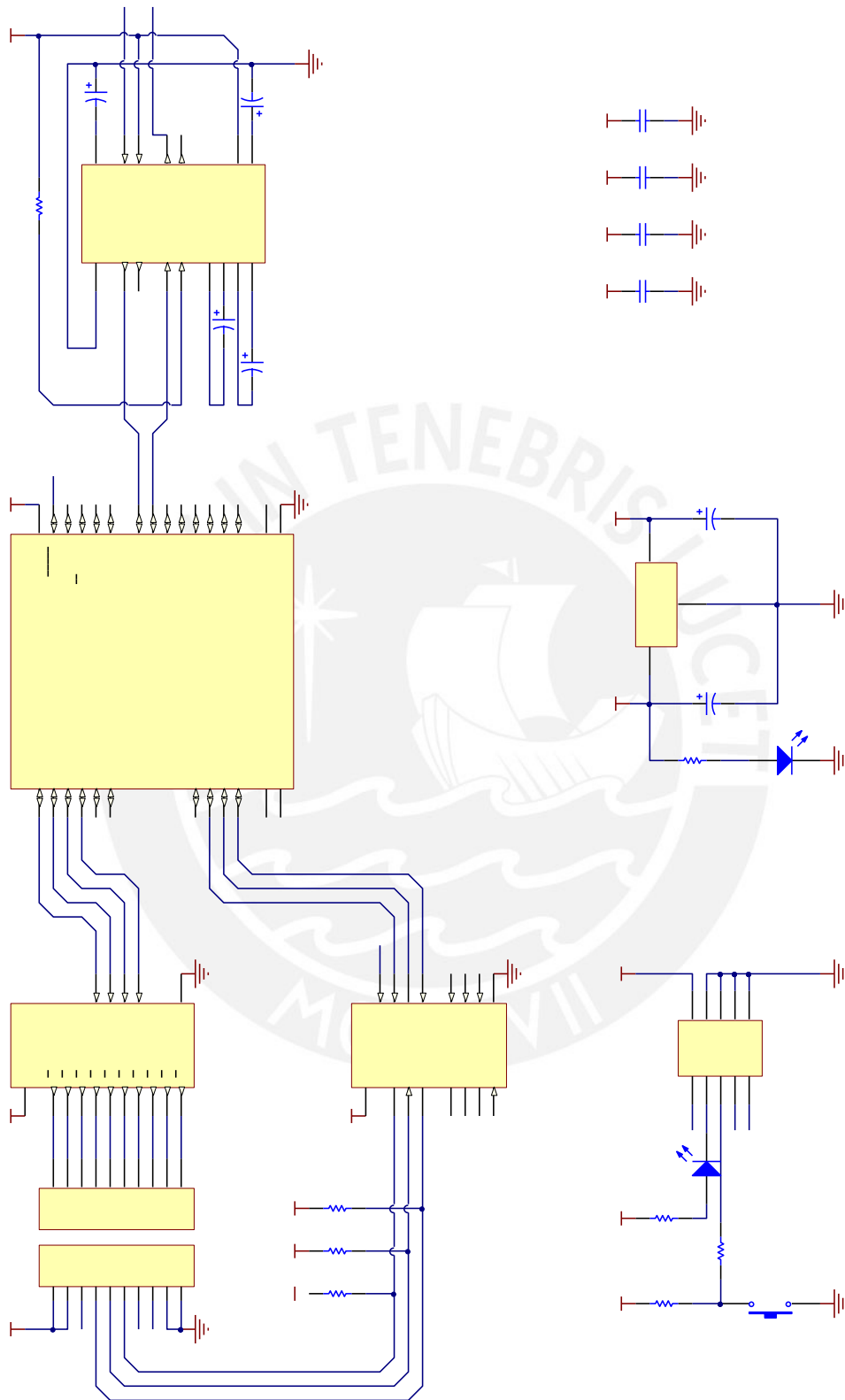


Figura 3.25 Diagrama esquemático del hardware del módulo de gestión de tráfico de datos y comandos.

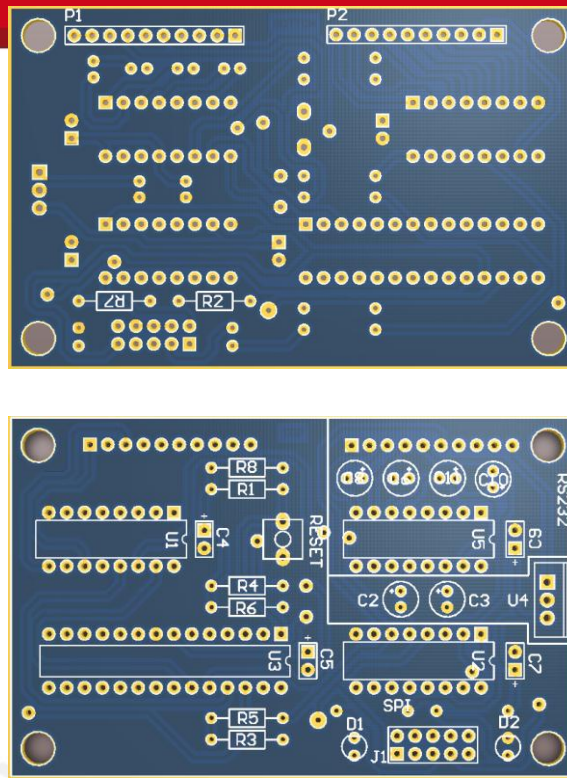


Figura 3.26 Dibujo del circuito impreso del circuito de impresión del módulo adquisición de datos y comandos.

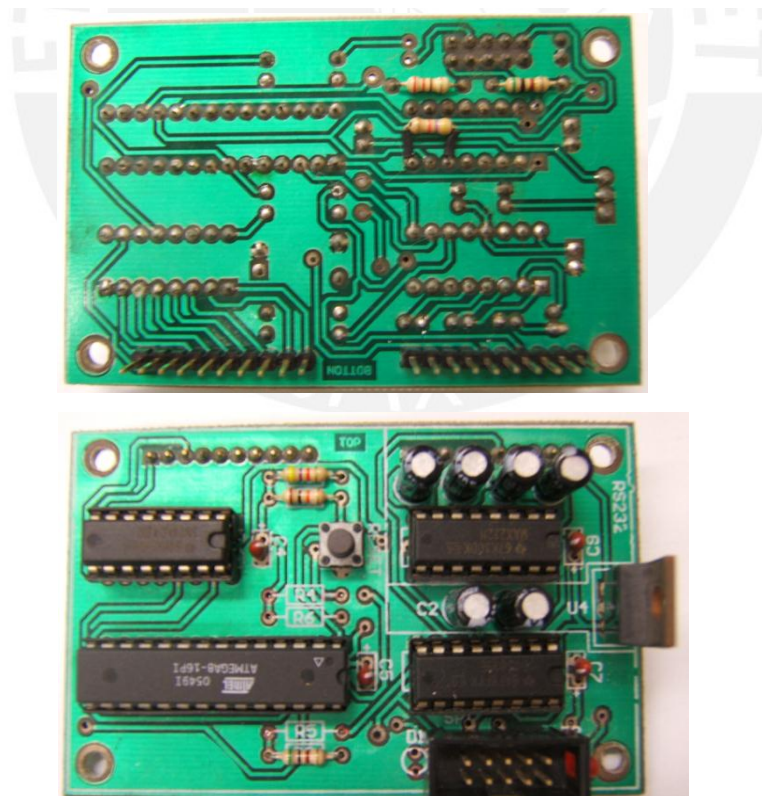


Figura 3.27 Fotografía del hardware del módulo de gestión de datos y comandos terminada.

3.2.1.4 Módulo de interfaz de usuario

Este módulo es una computadora personal que tiene por lo menos las siguientes características: Pentium 4, 512Mb de memoria RAM, 10Mb de espacio libre en el disco duro, controladora de video de con 64Mb, interfaz de comunicaciones RS232, monitor a color, teclado y mouse, sistema operativo Windows 98, MS Frameworks 2.0.

3.2.1.5 Integración del hardware

Ya que la filosofía del sistema es modular, es necesario un soporte de integración físico y eléctrico; así, cada uno de los módulos cuenta con un conector adecuado. Entre el hardware del *módulo adquisición de datos* y el hardware del *módulo de procesamiento de datos y comandos*; la conexión es mediante un cable *flat* de cinco hilos, en ambos casos el cable *flat* está soldado a las tarjetas. Entre la tarjeta del *módulo de procesamiento de datos y comandos* y la tarjeta del *módulo de gestión de tráfico de datos y comandos*, la conexión se hace mediante una tarjeta auxiliar denominada “hardware para la integración de módulos” que cuenta con conectores receptores para cada una de las nueve tarjetas de los *módulos de procesamiento de datos y comandos*; así, como para la tarjeta del *módulo de gestión de tráfico de datos y comandos*. Finalmente, la conexión física entre el hardware para la integración de los módulos y la computadora personal, se hace a través de un cable blindado de nueve hilos con conectores DB9 complementarios en cada extremo.

La Figura 3.28 muestra el diagrama esquemático del hardware para la integración de los módulos, este provee energía eléctrica a cada módulo y el soporte físico para el bus de comunicaciones SPI y la interfaz RS232; también, cuenta con un arreglo de diodos en la entrada principal de energía para proteger el sistema de posibles negligencias en la polaridad del conector externo.

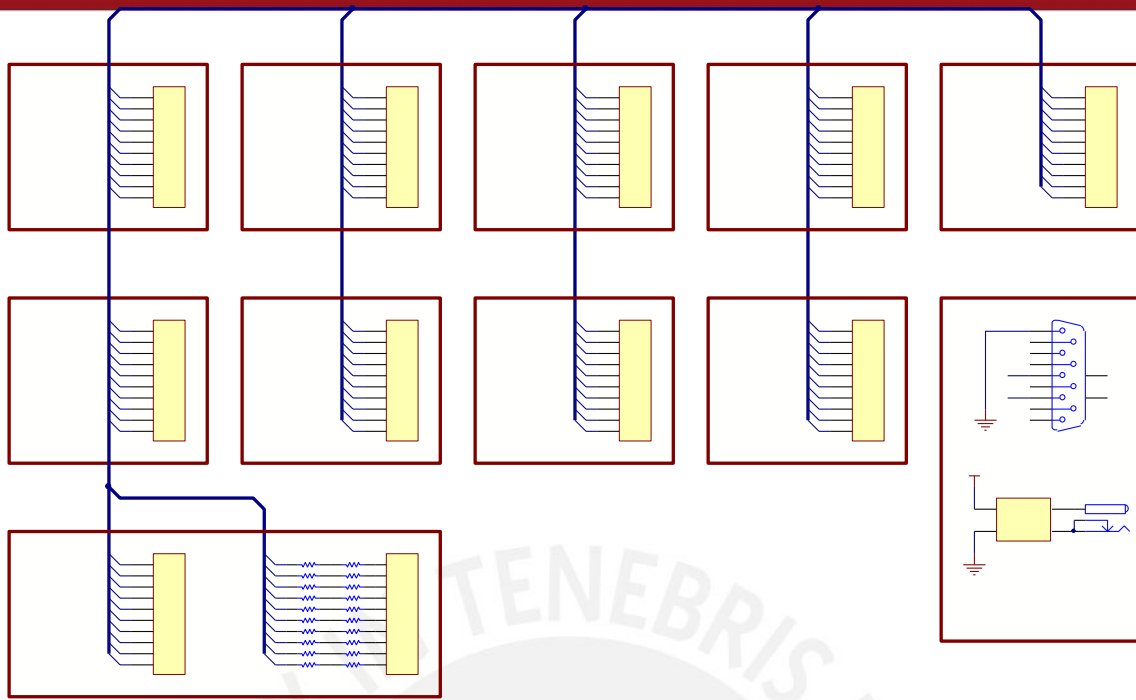


Figura 3.28 Diagrama esquemático del circuito de integración de hardware.



Figura 3.29 Dibujo del circuito impreso de integración de módulos.

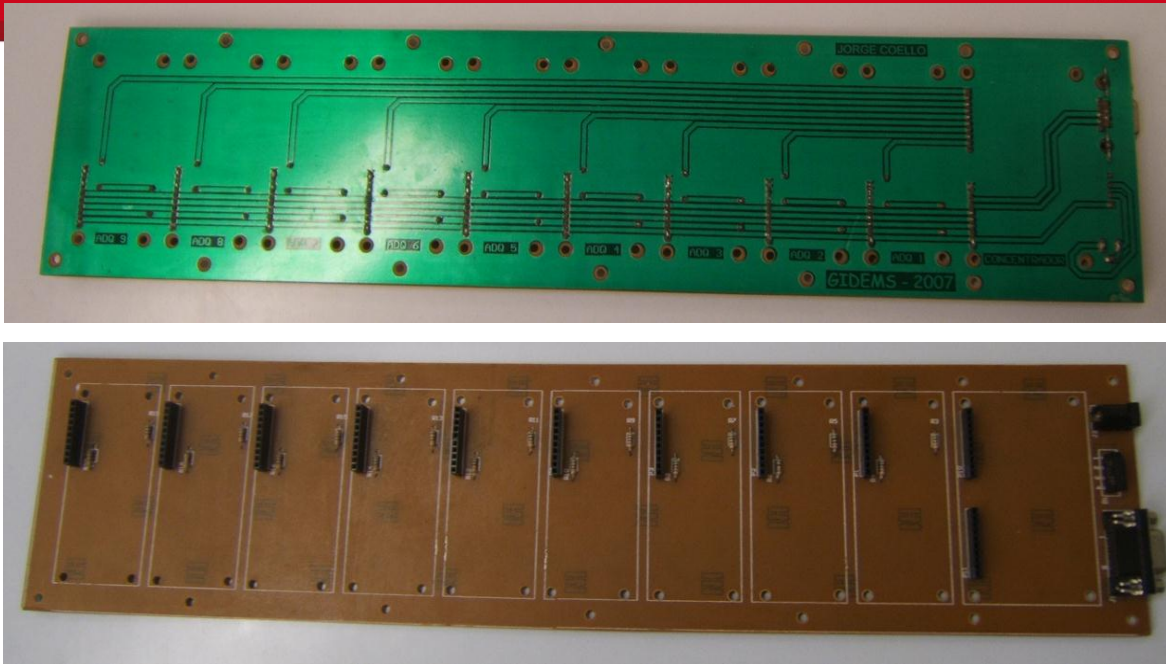
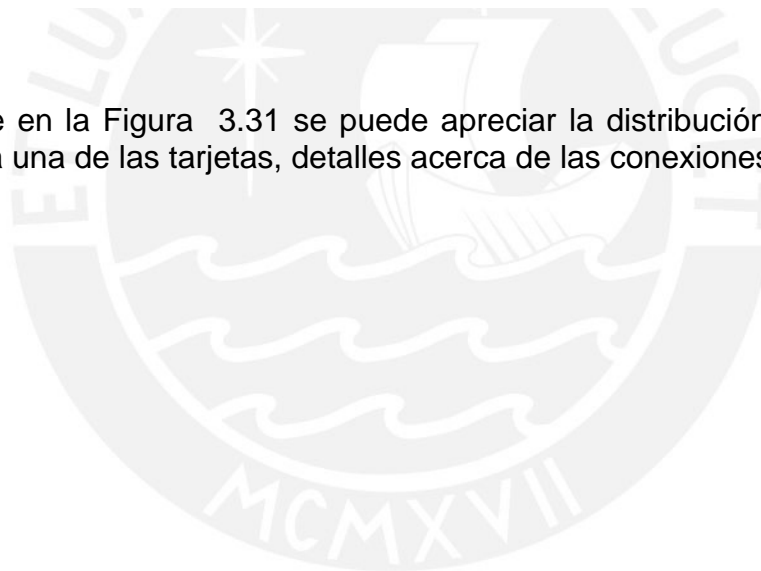


Figura 3.30 Fotografía de la tarjeta de integración de módulos terminada.

Adicionalmente en la Figura 3.31 se puede apreciar la distribución física del sistema completo, cada una de las tarjetas, detalles acerca de las conexiones, puertos, etc.



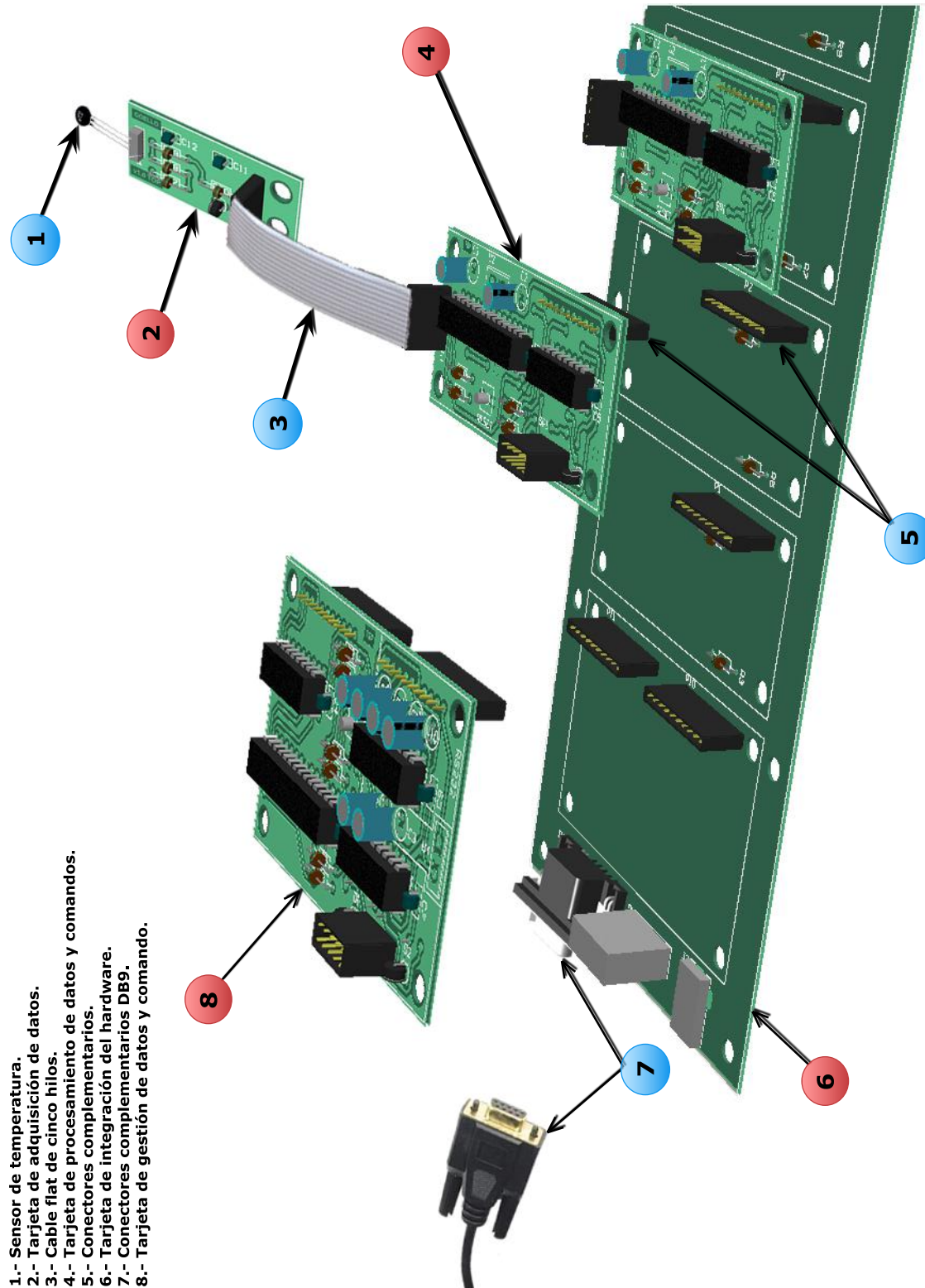


Figura 3.31 Representación física de todo el hardware del sistema.

3.2.2 Software

El software del sistema brinda el soporte informático a cada uno de los módulos que componen una unidad de medición independiente de temperatura. El programa aplicativo se desarrolló usando la herramienta VISUAL BASIC EXPRESS 2008, y los programas de los microcontroladores se desarrollaron con la herramienta CODEVISION v2.03.09.

3.2.2.1 Módulo de adquisición de datos

El software de este módulo establece comunicación con del ADC mediante el protocolo de comunicaciones I2C. Primero, configura los parámetros de operación del ADC tales como la frecuencia de muestreo, factor de ganancia y resolución de la conversión; enseguida, inicia rutinariamente la adquisición de los datos del ADC. Como se determinó en el capítulo de diseño del hardware del módulo de adquisición de datos el rango de conversión es desde 0V hasta 2V aproximadamente, así que, no será necesario amplificar la señal; por lo tanto, el parámetro de ganancia será igual a uno. En la Figura 3.32 [11] se puede observar la relación entre la frecuencia de muestreo y la resolución de la conversión.

DATA RATE	NUMBER OF BITS	MINIMUM CODE	MAXIMUM CODE
15SPS	16	-32,768	32,767
30SPS	15	-16,384	16,383
60SPS	14	-8192	8191
240SPS	12	-2048	2047

Figura 3.32 Modos de configuración del ADC. [11].

Se aprecia que para obtener una resolución de 16 BITS, es necesaria una frecuencia de muestreo no mayor a 15 SPS. Ya que, las variaciones de temperatura no son repentinas se consideró sacrificar velocidad de muestreo a cambio de una mayor resolución. Del análisis hecho, los parámetros de configuración para el ADC son:

- Frecuencia de muestreo: 15Hz
- Factor de ganancia: 1
- Resolución de conversión: 16BITS
- Modo de conversión continuo

El fabricante proporciona la descripción de la trama I2C para la comunicación con el ADC. [11], esta se puede observar en la Figura 3.33, donde se aprecia lo siguiente: el

primer octeto corresponde a la dirección física del periférico (ADC), el segundo y tercer octeto representan el valor numérico del resultado de la conversión, finalmente el cuarto octeto contiene los parámetros de configuración.

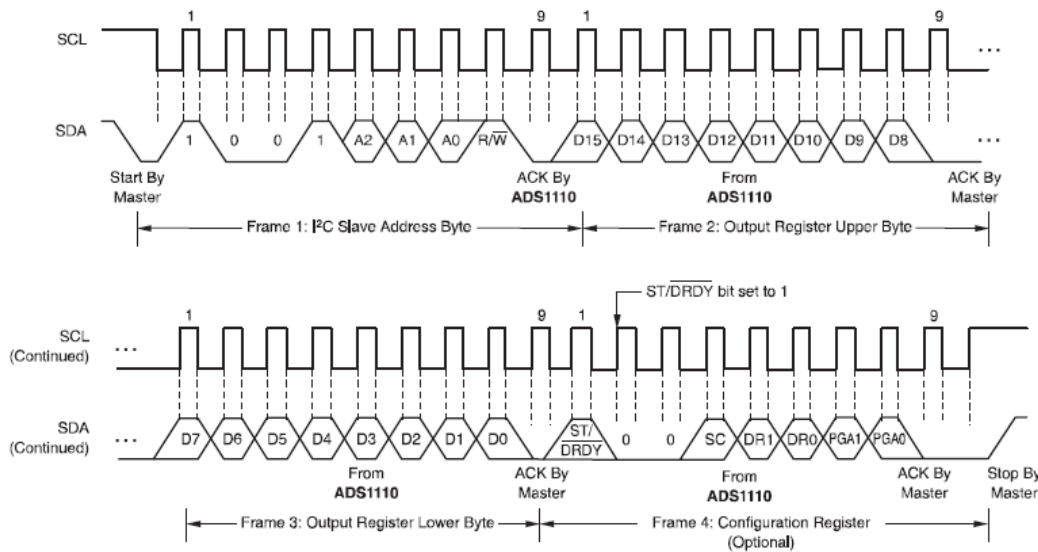


Figura 3.33 Diagrama de tiempos de la Comunicación con el ADC vía I2C [11].

Con respecto al octeto de configuración, este tendrá el valor **0x8C**, el cual, corresponde a las características de configuración ya mencionadas.

Para establecer comunicación con el ADC, se desarrolló un manejador (*Driver*) en lenguaje C, denominado `ads1110.h`; el cual se detalla a continuación:

Variables globales:

`unsigned int temperature`

2 bytes que contienen el resultado de la conversión del ADC

`unsigned char config`

1 bytes que contiene la información de configuración para el ADC

Procedimientos:

`void ADS1110_init(void)`

Este procedimiento inicializa la configuración de los pines PC5 (SCL) y PC4 (SDA) del puerto C.

`void ADS1110_reset()`

Este procedimiento reinicia el ADC escribiendo el valor 0x00 en el bus I2C.

void ads1110_start(void)

Este procedimiento envía la señal conocida como “start”, la cual es parte del protocolo I2C y tiene como objetivo indicar el inicio de una comunicación.

void ads1110_stop(void)

Este procedimiento envía la señal conocida como “stop”; la cual, es parte del protocolo I2C y tiene como objetivo indicar el fin de una comunicación.

void one_pulse_clock(void)

Este procedimiento genera un pulso en la línea SDA que consiste de un flanco de subida y otro de bajada, se usa en todos los procedimientos y funciones que leen o escriban datos en el bus I2C.

void send_ack(void)

Este procedimiento envía la señal conocida como “acknolage”; la cual es parte del protocolo I2C y tiene como objetivo indicar que la trama enviada o recibida ha sido reconocida.

Funciones:

unsigned char ADS1110_read_data(unsigned char address)

Esta función establece comunicación con el ADC, tiene como parámetro de entrada la dirección física de este periférico. Invoca a los procedimientos `ads1110_start`, `ads1110_stop`, `one_pulse_clock`, `send_ack`. Así como a la función `ads1110_write_byte`, su objetivo es leer el resultado de la conversión analógica digital y guardarlo en la variable global `temperature` finalmente devuelve el valor `0x00` si es que la operación se realizó con éxito, de lo contrario devuelve un `0x01`.

**unsigned char ADS1110_write_config(unsigned char address,
unsigned char options)**

Esta función establece comunicación con el ADC, tiene como parámetros de entrada la dirección física de este periférico y número que representa la configuración del ADC. Invoca a los procedimientos `ads1110_start`, `ads1110_stop`, `one_pulse_clock`, `send_ack`. Así, como a la función `ads1110_write_byte`, su objetivo es escribir el byte de configuración de parámetros del ADC; después, devuelve el valor `0x00` si es que la operación se realizó con éxito, de lo contrario devuelve un `0x01`.

`unsigned ads1110_write_byte(unsigned char byte)`

Esta función escribe un byte en el bus I2C. Invoca a los procedimientos `ads1110_start`, `ads1110_stop`, `one_pulse_clock`, `send_ack`; finalmente, devuelve el valor `0x00` si es que la operación se realizó con éxito, de lo contrario devuelve un `0x01`.

3.2.2.2 Módulo de procesamiento de datos y comandos

El software de este módulo se encarga de procesar los datos recibidos por el módulo de adquisición de datos, también procesar comandos enviados desde el módulo de interfaz de usuario. Antes de explicar el proceso global es conveniente explicar las diversas actividades que lo conforman.

A) En cuanto al procesamiento de datos

Cada vez que el microcontrolador y el ADC se comunican, el microcontrolador lee tres bytes del ADC; los primeros dos bytes contienen el valor numérico de la última conversión realizada y el tercer byte; concretamente, el “*bit*” más significativo indica si el dato leído es nuevo o ya ha sido leído anteriormente.

Los datos obtenidos del ADC, corresponden a valores de voltaje entre los terminales de entrada. Este valor, al que se denominará Código de Salida (CS), está en función de los parámetros T_1 , T_2 , R_1 , R_2 , R_3 , R_4 y V_{ref} . Que conforman el circuito esquemático que se muestra en la Figura 3.34

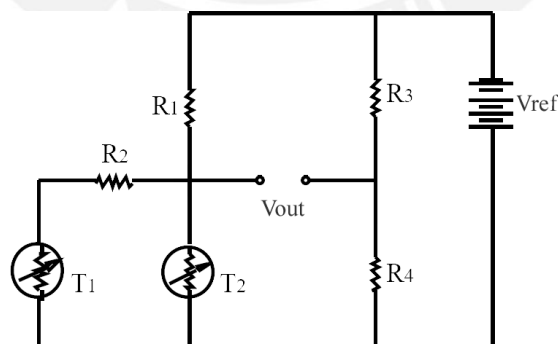


Figura 3.34 Diagrama esquemático con la configuración del puente de Wheatstone.

En esta sección se analizará el efecto que tienen la tolerancia y el coeficiente térmico sobre el código de salida y cuanto error introducen en la medición de temperatura.

Se deduce de la Figura 3.34 que:

$$V_{out} = V_{ref} * \left(\frac{(R2 + T1) * T2}{R1 * (R2 + T1 + T2) + T2 * (R2 + T1)} - \frac{R4}{R4 + R3} \right)$$

Según el fabricante del ADC [11], el código de salida es:

$$CS = -1 * -32768 * 1 * \frac{(V_{in+} - V_{in-})}{2.048V}$$

Como $(V_{in+} - V_{in-}) = -V_{out}$ se obtiene:

$$CS = -16000 * V_{ref} * \left(\frac{(R2 + T1) * T2}{R1 * (R2 + T1 + T2) + T2 * (R2 + T1)} - \frac{R4}{R4 + R3} \right)$$

Las resistencias seleccionadas tienen una tolerancia de fabricación de $\pm 1\%$ y un coeficiente térmico resistivo (TRC) igual a $\pm 15 \text{ppm}/^\circ\text{C}$ en el rango de -20°C a $+85^\circ\text{C}$ [10], ya que el rango de medición propuesto es mayor a $+85^\circ\text{C}$, el fabricante recomienda adicionar $0.05 \text{ppm}/^\circ\text{C}$ por cada grado centígrado que escape al rango regular, de este modo el coeficiente térmico para el rango de -20°C a $+100^\circ\text{C}$ es de $\pm 15.75 \text{ppm}/^\circ\text{C}$, en caso del voltaje de referencia este tiene una tolerancia de $\pm 0.04\%$ y un coeficiente térmico de $\pm 3 \text{ppm}/^\circ\text{C}$ en el rango de -40°C a $+125^\circ\text{C}$. [12].

Cada resistencia que compone el circuito así como el voltaje de referencia variarán su magnitud en función de la temperatura a la que están expuestos; del mismo modo el valor nominal cambiará a causa de la tolerancia de fabricación, estas alteraciones tendrán repercusión en la medición de temperatura.

Las magnitudes de las resistencias estarán en función a su valor nominal, su tolerancia y la temperatura de exposición, esto es:

$$R_n(R_{Rn}, R_{tol}, T) = R_{Rn} \pm R_{tol}_{Rn} \pm R(T)_{Rn}, \text{ y:}$$

$$R(T)_{Rn} = R_{Rn @ 25^\circ\text{C}} * (1 + TRC_{Rn} * (T - 25^\circ\text{C})), \text{ donde:}$$

TCR_{Rn} : Coeficiente térmico

R_{Rn} : Resistencia nominal

R_{tol}_{Rn} : Resistencia debido a la tolerancia

$R(T)_{Rn}$: Resistencia en función de la temperatura

De igual modo, la magnitud del voltaje de referencia estará en función a su valor nominal, su tolerancia y la temperatura de exposición, esto es:

$$Vn(V_{Vn}Vtol, T) = V_{Vn} \pm Vtol_{Vn} \pm V(T)_{Vn}, y:$$

$$V(T)_{Vn} = V_{Vn@25^{\circ}C} * (1 + TVC_{Vn} * (T - 25^{\circ}C)), \text{ donde:}$$

TVC_{Vn} : Coeficiente térmico

V_{Vn} : Voltaje nominal

$Vtol_{Vn}$: Voltaje debido a la tolerancia

$V(T)_{Vn}$: Voltaje en función de la temperatura

Se conoce que los valores nominales, tolerancias y coeficientes térmicos son:

Valor nominal	Tolerancia	Coeficiente Térmico (-20°C a 100°C)
$Vref = 3.3 V$	$Vtol_{Vref} = \pm 0.04\%$	$TVC_{Vref} = \pm 3 ppm/^{\circ}C$
$R1 = 5700 \Omega$	$Rtol_{R1} = \pm 1\%$	$TRC_{R1} = \pm 15.75 ppm/^{\circ}C$
$R2 = 12000 \Omega$	$Rtol_{R2} = \pm 1\%$	$TRC_{R2} = \pm 15.75 ppm/^{\circ}C$
$R3 = 1300 \Omega$	$Rtol_{R3} = \pm 1\%$	$TRC_{R3} = \pm 15.75 ppm/^{\circ}C$
$R4 = 5900 \Omega$	$Rtol_{R4} = \pm 1\%$	$TRC_{R4} = \pm 15.75 ppm/^{\circ}C$

Tabla 3.1 Valores nominales, tolerancias y coeficientes térmicos

Partiendo de la ecuación que describe al código de salida:

$$CS = -16000 * Vref * \left(\frac{(R2 + T1) * T2}{R1 * (R2 + T1 + T2) + T2 * (R2 + T1)} - \frac{R4}{R4 + R3} \right)$$

Si: $CS = A * (B - C)$ donde:

$$A = -16000 * V_{Vref} \pm Vtol_{Vref} \pm V(T)_{Vref}$$

$$B = \frac{(R_{R2} \pm Rtol_{R2} \pm R(T)_{R2} + T1) * T2}{R_{R1} \pm Rtol_{R1} \pm R(T)_{R1} * (R_{R2} \pm Rtol_{R2} \pm R(T)_{R2} + T1 + T2) + T2 * (R_{R2} \pm Rtol_{R2} \pm R(T)_{R2} + T1)}$$

$$C = \frac{R_{R4} \pm Rtol_{R4} \pm R(T)_{R4}}{R_{R4} \pm Rtol_{R4} \pm R(T)_{R4} + R_{R3} \pm Rtol_{R3} \pm R(T)_{R3}}$$

Entonces, se puede afirmar que: por efecto de la tolerancia habrá un rango de valores alrededor del valor nominal donde se encuentra la magnitud real del componente; por efecto de la temperatura de exposición, existirá un rango de valores dependientes del coeficiente térmico y de la temperatura de exposición. Para cuantificar los efectos sobre el código de salida se implementó una rutina en la herramienta de software MATLAB [13], dicha rutina resuelve la ecuación del código de salida en función de la tolerancia, coeficiente térmico y temperatura de exposición. Parte calculando el rango de valores por efecto de la tolerancia de cada componente, luego, calcula el rango de posibles valores en función de la temperatura y coeficiente térmico que inicia en 0°C, seguidamente resuelve la ecuación para cada posible combinación, obteniendo un código de salida en cada caso, el máximo y mínimo son registrados, se prosigue a incrementar un grado centígrado, y se calculan los rangos nuevamente, el proceso se repite hasta llegar a los +100°C. El resultado de este cuantioso cálculo se aprecia en la Figura 3.35.

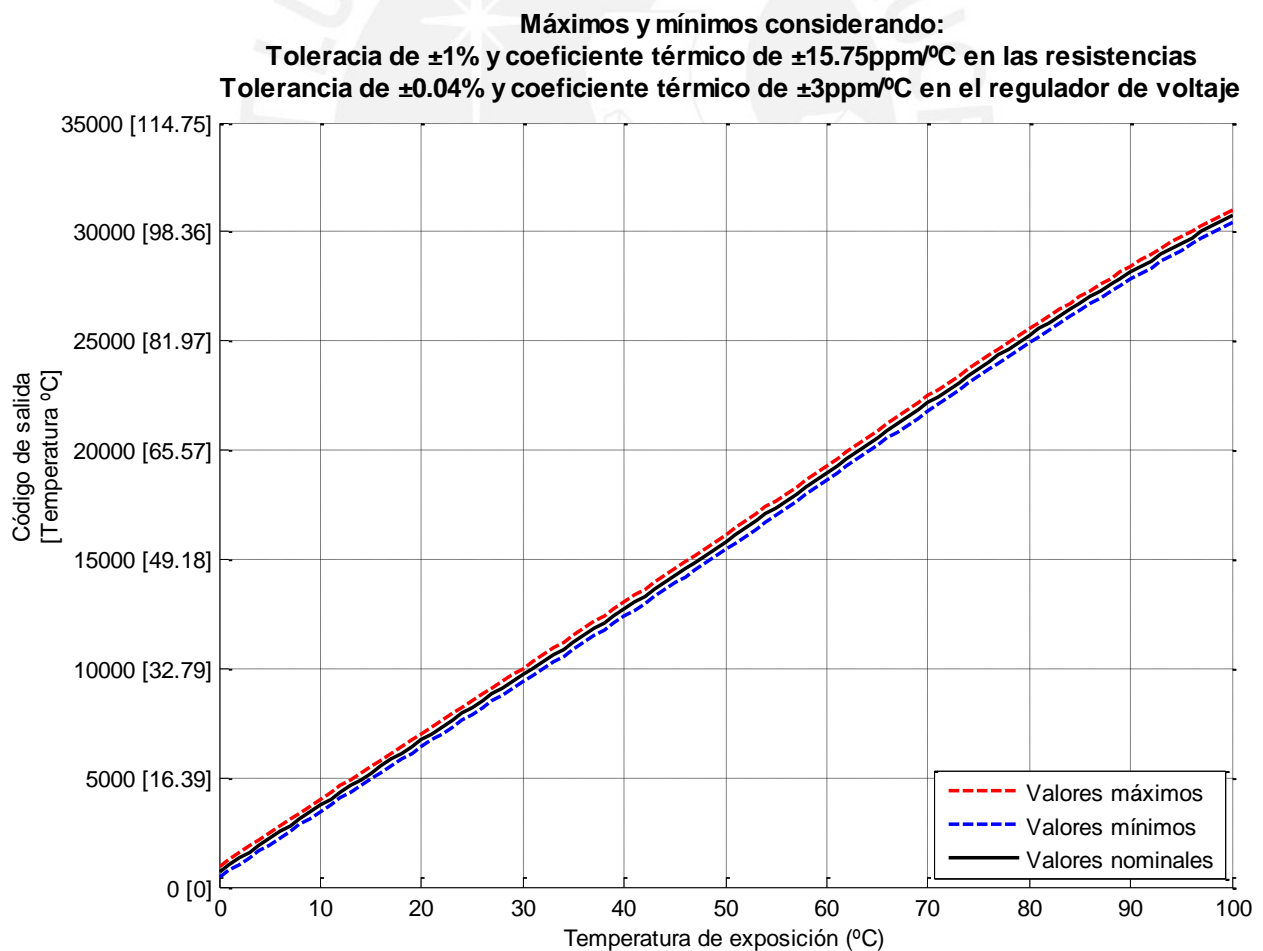


Figura 3.35 Código de salida máximo, mínimo y nominal.

La Figura 3.36 muestra únicamente el efecto de la tolerancia, tomando como referencia la respuesta del circuito a los valores nominales, se aprecia que el error introducido varía de acuerdo a la temperatura de exposición; y puede tomar algún valor entre los límites superior e inferior. Por ejemplo, se aprecia que al rededor de los $+45^{\circ}\text{C}$ de exposición se podría presentar el error de medición máximo que es de $\pm 1.08^{\circ}\text{C}$ aproximadamente.

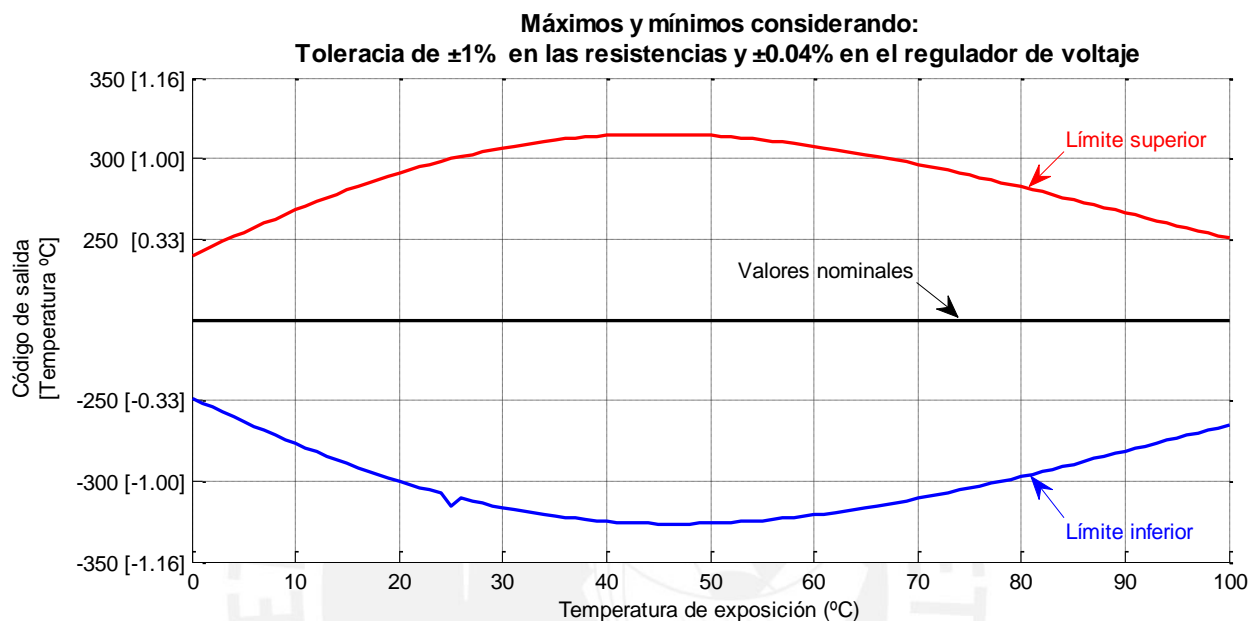


Figura 3.36 Límites del código de salida por efecto de la tolerancia.

En la Figura 3.37 se aprecia el efecto de la temperatura sobre la magnitud nominal de los componentes, que por consecuencia afectará el valor del código de salida. Se observa que alrededor de los $+100^{\circ}\text{C}$ se espera el nivel más alto del error de medición, pues a esa temperatura el código de salida podría tomar algún valor entre 30 y -35 , lo que representa un error de medición de $\pm 0.12^{\circ}\text{C}$ aproximadamente; también se observa que a los $+25^{\circ}\text{C}$ el error por efecto de temperatura es casi despreciable, por descripción de los fabricantes [10] [12].

Máximos y mínimos considerando:
Coeficiente térmico de $\pm 15.75 \text{ ppm}/^\circ\text{C}$ en las resistencias $\pm 3 \text{ ppm}/^\circ\text{C}$ en el regulador de voltaje

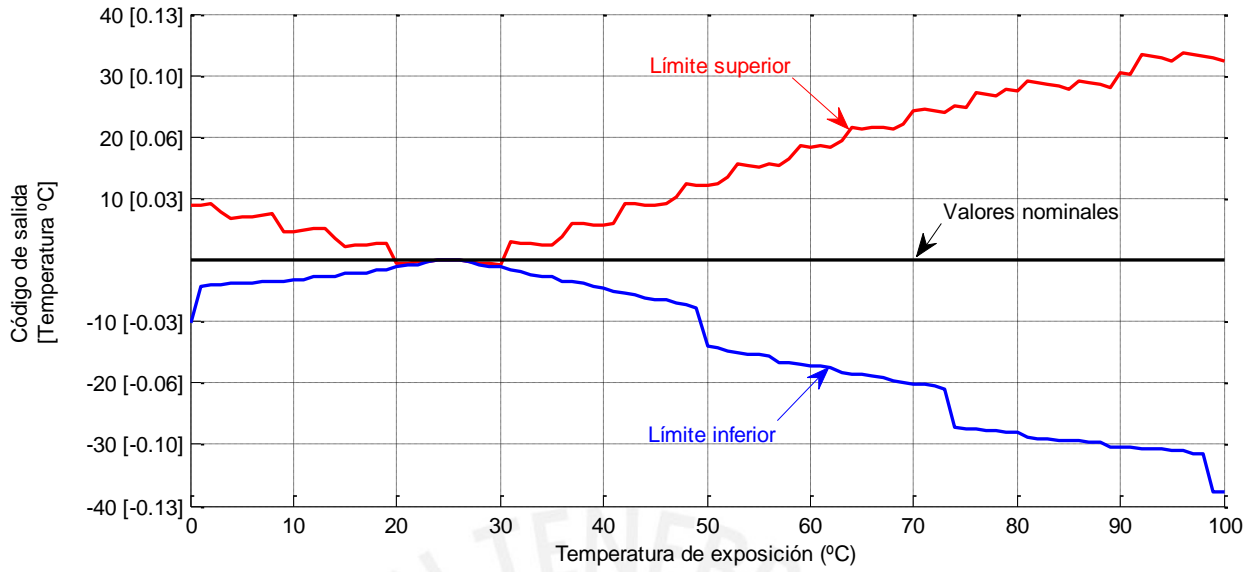


Figura 3.37 Límites del código de salida por efecto de la temperatura de exposición

Finalmente, la Figura 3.38, muestra el efecto combinado de la tolerancia y la temperatura de exposición de los componentes sobre el código de salida, se aprecia que alrededor de los $+60^\circ\text{C}$ se tendría el valor máximo del error de medición, que puede llegar a alcanzar los $\pm 1.16^\circ\text{C}$ aproximadamente.

Máximos y mínimos considerando:
Tolerancia de $\pm 1\%$ y coeficiente térmico de $\pm 15.75 \text{ ppm}/^\circ\text{C}$ en las resistencias
Tolerancia de $\pm 0.04\%$ y coeficiente térmico de $\pm 3 \text{ ppm}/^\circ\text{C}$ en el regulador de voltaje

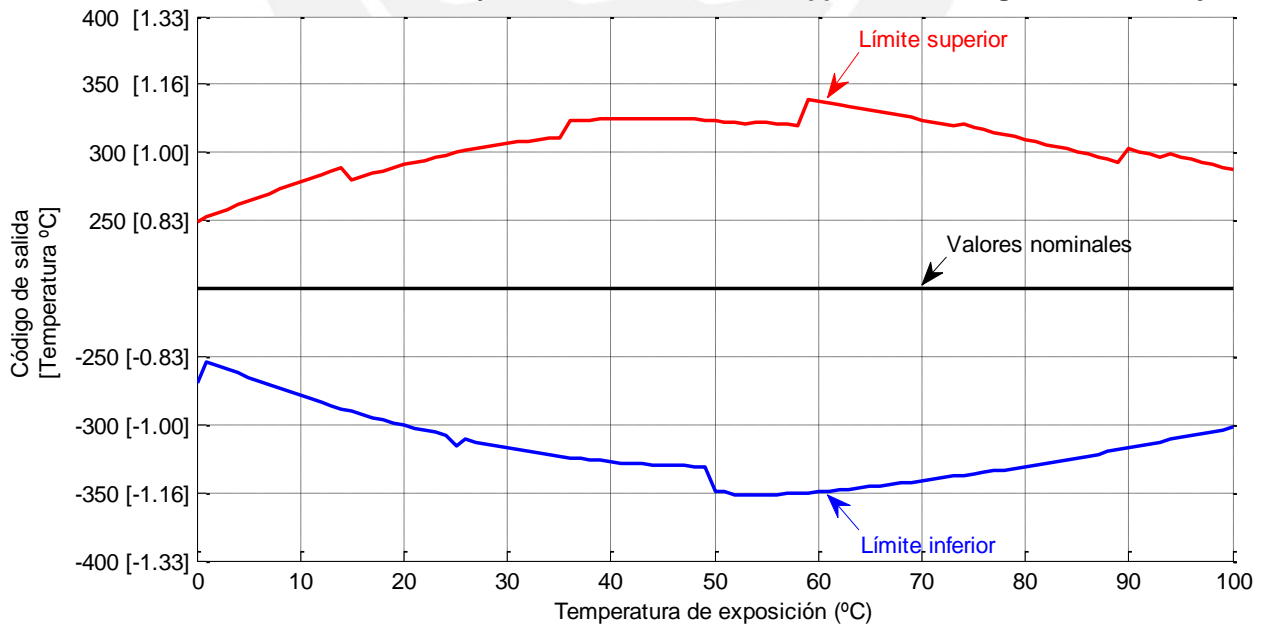


Figura 3.38 Límites del código de salida por efecto de la tolerancia y la temperatura de exposición

En conclusión, existirá un error de medición de hasta $\pm 1.16^{\circ}\text{C}$ por efecto de la tolerancia y la temperatura de exposición en los componentes, el 93% de ese error, aproximadamente, se debe a la tolerancia de los componentes.

Para solucionar el inconveniente producto de la tolerancia se obtuvieron los valores reales de los componentes. Para mitigar el error producto de la temperatura de exposición los sensores fueron calibrados con un patrón. De este modo se relacionó el código de salida con la temperatura de exposición, este procedimiento se detallará en la sección 3.3 “*Pruebas del sistema*”.

Ya que el código de salida depende de la temperatura de exposición y las magnitudes reales de los componentes, el procedimiento para obtener las magnitud verdaderas de $R1, R2, R3, R4$ y V_{ref} fue mediante la medición directa una vez soldados los componentes. Se usó el multímetro de la marca FLUKE modelo 179 con un error de medición de resistencias de 0.9%. [12]. Con ello, se logró conocer la magnitud de cada componente, se contempló una herramienta de cálculo en el software aplicativo del módulo de interfaz de usuario, de este modo, es posible obtener una tabla con los valores de los códigos de salida para cada grado centígrado ingresando los valores de $R1, R2, R3$ y $R4$ y V_{ref} . Dicha tabla es calculada en función a los parámetros propios y únicos de cada hardware de adquisición de datos y guardada en la memoria EEPROM, del hardware de procesamiento de datos y comandos del módulo correspondiente.

Como ya se mencionó anteriormente los valores de $T1$ y $T2$ sirven para calcular la temperatura del sensor, pero el fabricante brinda información únicamente en unidades de grados centígrados, es decir 1°C , 2°C , etc. Para determinar temperaturas intermedias, fue necesario usar el método de aproximación por interpolación, asumiendo que el comportamiento de la curva entre cada grado centígrado es lineal. Se implementó una librería en lenguaje C, denominada `index.h` con las siguientes características:

Variables globales:

```
eeprom int ADS_code_table[111]
```

Es un arreglo de 111 *palabras de 16 bits*, las primeras 101 *palabras* son los valores de los códigos de salida calculados para cada grado centígrado desde 0°C hasta +100°C, las 10 *palabras* restantes son usadas para albergar el número de serie y constantes para la calibración del sensor.

```
unsigned int ADS_code.Out
```

Esta variable albergará el código de salida devuelto por el ADC.

```
unsigned int ADS_code.Index_botton
```

Esta variable albergará el índice superior de la tabla `ADS_code_table`, con respecto al valor `ADS_code.Out`.

```
unsigned int ADS_code.Index_Top
```

Esta variable albergará el índice inferior de la tabla `ADS_code_table`, con respecto al valor `ADS_code.Out`.

Procedimientos:

```
void Index_Manager(void)
```

Este procedimiento modifica las variable globales `ADS_code.Index_Top` y `ADS_code.Index_botton` en función al valor de `ADS_code.Out` y los valores de la tabla `ADS_code_table[]`.

```
float Interpolate(void)
```

Este procedimiento calcula el valor intermedio de temperatura correspondiente a la interpolación hecha en base a los valores superior e inferior y el valor del código de salida (`ADS_code.Index_botton`, `ADS_code.Index_Top`, `ADS_code.Out`).

La Figura 3.39 ayudará a ilustrar el proceso que se realizará. En este el ejemplo luego de una comunicación con el ADC se ha leído el valor 9115, `ADS_code.Out = 9115`, este valor será buscado en la tabla de la columna (A), cabe señalar que esta tabla ha sido calculada usando parámetros propios de cada hardware de adquisición de datos, esta búsqueda se llevará a cabo usando el procedimiento `Index_Manager`; una vez ubicado el rango en el que se encuentra el valor 9115, el cual está entre 8943 y 9240 (27°C y 28°C) se asignan los valores `ADS_code.Index_Top = 9240` y `ADS_code.Index_botton = 8943`, como se aprecia en la columna (B), debido a

que no se cuenta con el valor de temperatura correspondiente al valor 9115, se realizará una interpolación usando el procedimiento **Interpolate** como lo muestra la columna (C). Finalmente se obtiene el valor de temperatura para el código 9115 el cual es igual a $+27.57^{\circ}\text{C}$

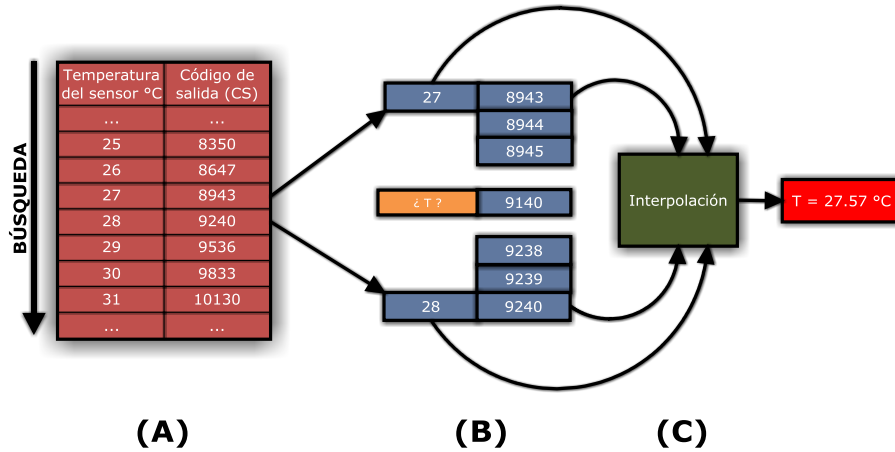


Figura 3.39 Proceso de interpolación.

B) En cuanto al procesamiento de comandos

Con respecto al procesamiento de comandos es necesario señalar los comandos disponibles:

```
#define order_START
#define order_RESET
#define order_LPM
#define order_EPM
#define order_RESOLUTION
```

order_START:

Este comando es inicia el proceso de adquisición y procesamiento de datos, antes de que este comando sea ejecutado el microcontrolador configurará todos los periféricos e inicializará todas las variables.

order_RESET:

Este comando reinicia el microcontrolador.

order_LPM:

Este comando tiene como fin modificar el estado del LED indicador pudiendo tomar cualquiera de los tres siguientes estados: Encendido, Apagado, Intermitente.

order_EPM:

Este comando tiene como fin leer y escribir los valores que contiene el arreglo `ADS_code_table[]` ubicado en la memoria EEPROM, estos valores son necesarios para procesar los datos de temperatura.

order_RESOLUTION:

Este comando tiene como fin cambiar la resolución de la temperatura calculada, pudiendo ser de dos modos: centésimas y milésimas.

Todos estos comandos son enviados en algún momento de forma automática o por acción del usuario mediante el software aplicativo del módulo de interfaz de usuario que se ejecuta en la computadora personal.

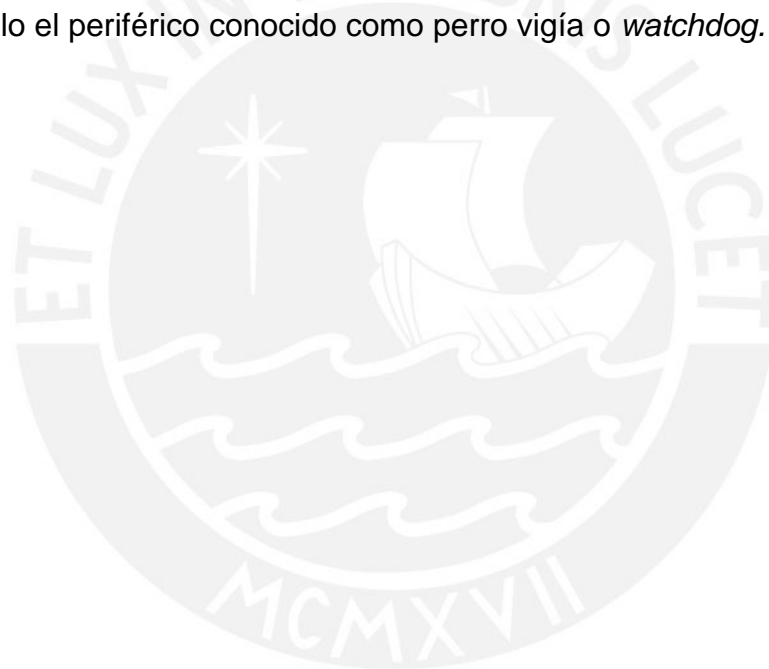
El funcionamiento el sistema operativo se encarga de la adquisición y procesamiento de datos. El proceso iniciará configurando un temporizador, el cual producirá una interrupción cada 61ms, este es un tiempo menor a los 66.6 ms (15Hz) que se demora el ADC en realizar una conversión, y se debe al hecho que tanto el ADC como el microcontrolador tienen su propia fuente de reloj, y no están sincronizados.

Una vez que transcurrió el tiempo programado se inicia una ráfaga de comprobaciones, así, el microcontrolador lee continuamente los datos del ADC y comprueba si éstos son producto de una conversión reciente o de una anterior. Si se determina que un dato es nuevo, se reiniciará el temporizador de modo tal que a partir de ese momento transcurran 55ms hasta la próxima interrupción. Una vez obtenido el código de salida (CS) es procesado mediante las operaciones antes descritas. Finalmente, se obtiene un valor numérico de temperatura correspondiente a la medición realizada. Debido a que este valor es almacenado en una variable del tipo *float*, necesaria para almacenar datos con punto decimal y con el fin de aminorar el volumen de tráfico necesario (4 bytes por una variable *float* de cada dato de temperatura). Las temperaturas serán representadas desplazando las décimas y centenas; por ejemplo, si la temperatura registrada es $+27.57^{\circ}\text{C}$; entonces el sistema operativo considerará que esta magnitud corresponde al número 2757, posteriormente en la capa del software aplicativo se hará

la conversión necesaria para representar adecuadamente dicha magnitud. Con este ajuste se logrará disminuir en 50% la cantidad de *bytes* necesarios para representar un dato de temperatura ya que únicamente se necesitarán dos *bytes*.

Los dos bytes que contienen la temperatura registrada se colocarán en un buffer relacionado al periférico de comunicaciones SPI, a la espera de que el software del módulo de gestión de tráfico de datos y comandos se encargue de leerlos.

En la Figura 3.40 se muestra el diagrama de estados del sistema operativo del módulo de procesamiento de datos y comandos. Se puede apreciar el lazo principal compuesto por W2, M1, M2, M3, M4, M5 que se encarga de procesar los datos de temperatura, también se observan los lazos que involucran al procesamiento de comandos. Un aspecto resaltante es el manejo de errores en cada uno de los procesos y estados; usando para ello el periférico conocido como perro vigía o *watchdog*.



Estado	Evento	Descripción
1	W1	Mientras no llegue la orden "Start"
	M0	Cuando llegue la orden "Start"
	W2	Mientras no llegue alguna orden
2	P4	Cuando llegue la orden "EEPROM Program mode"
	P2	Cuando llegue la orden "LED Program mode"
	R1	Cuando llegue la orden "Reset"
	M1	Cuando el timer indique que transcurrieron 62ms
3	M4	Cuando el timer indique que transcurrieron 72ms
	S1	Cuando el watchdog se active (transcurran 2s)
	E2	Mientras se intenta obtener un dato nuevo del ADS
	M2	Cuando se haya obtenido un dato nuevo del ADS
	M3	Cuando el timer indique que transcurrieron 72ms
4	M4'	Cuando el timer indique que transcurrieron 2s
	E3	Cuando el watchdog se active (transcurran 2s)
	M5	Cuando se haya terminado exitosamente el procesamiento de un dato
	E4	Cuando el watchdog se active (transcurran 2s)
5	P1	Cuando se haya terminado exitosamente la programación del LED
	E5	Cuando el watchdog se active (transcurran 2s)
	W3	Mientras se escribe la nueva frecuencia de parpadeo del LED
6	E6	Cuando se haya terminado exitosamente la programación de la memoria EEPROM
	W4	Mientras se escriben los datos en la memoria EEPROM
7	S2	Cuando se haya completado con éxito el envío del estado y los registros del sistema
	E7	Cuando el watchdog se active (transcurran 2s)
8	W5	Mientras se envían los registros de estado
	R2	Cuando se invoque al reinicio de todo el sistema

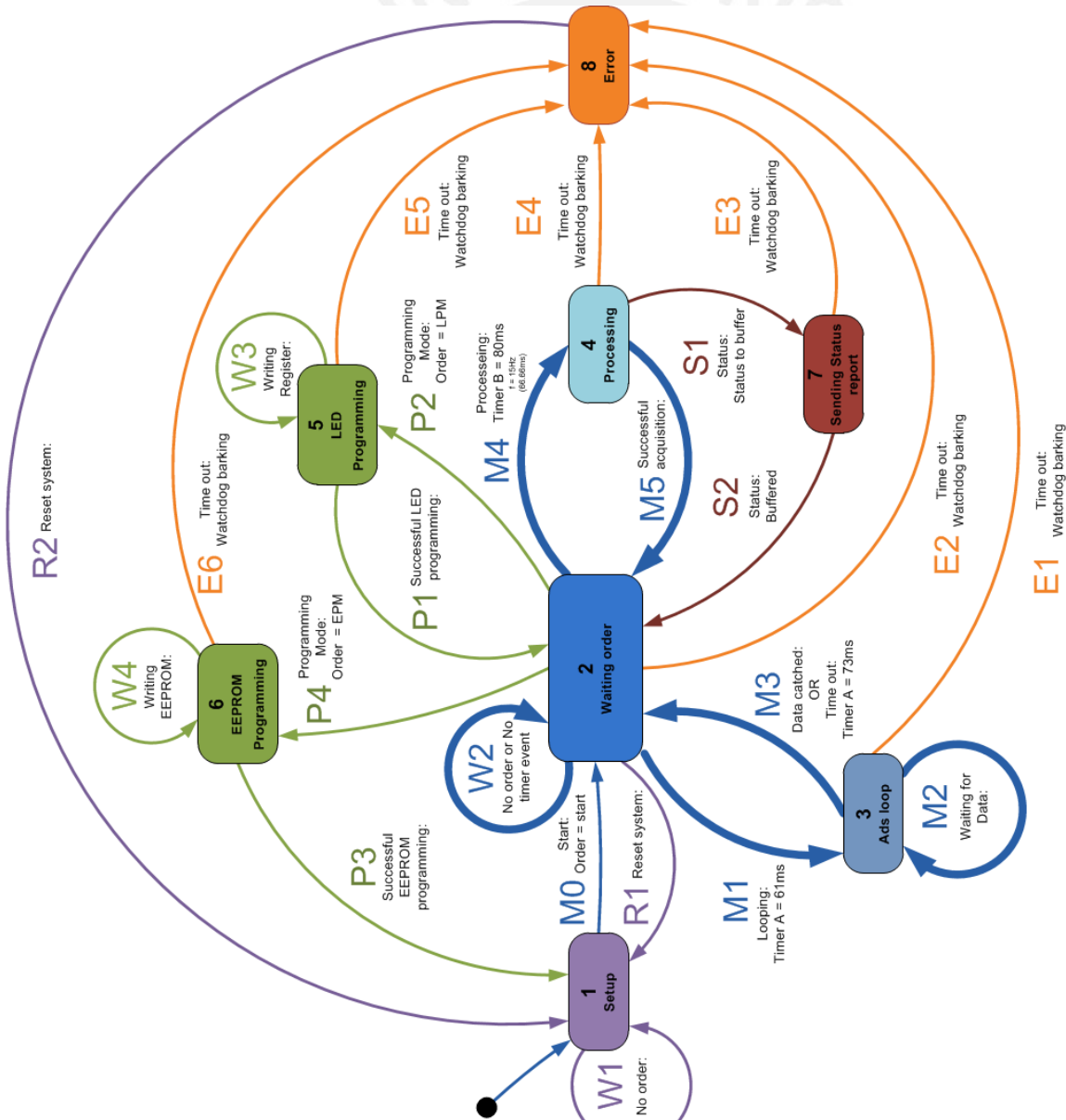


Figura 3.40 Diagrama de estados del sistema operativo de la unidad de procesamiento de datos y comandos.

3.2.2.3 Módulo gestión de tráfico de datos y comandos

El software de este módulo está encargado de manejar el tráfico de datos y comandos a través del bus SPI y la interfaz RS232.

Es conveniente detallar el tipo de paquetes que manejará el software de este módulo, tanto para la comunicación con la computadora personal, a través de la interfaz RS232 como para la comunicación con los módulos de procesamiento de datos y comandos mediante el bus SPI.

Formato de un paquete en el bus SPI:

Está compuesto por 4 bytes, como se aprecia en la Figura 3.41, el primer y último byte son valores constantes que representan las etiquetas de inicio y fin de un paquete de datos. Los bytes MSB y LSB corresponden al dato contenido en el paquete.

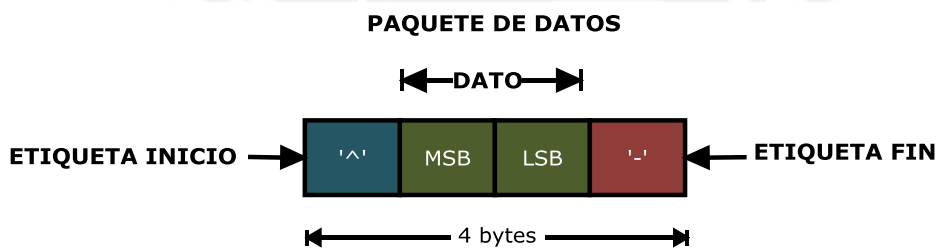


Figura 3.41 Formato de los paquetes SPI.

Los datos en un paquete pueden ser de dos tipos: datos de temperatura o comandos de control, que están relacionados a la dirección del flujo de datos, como se aprecia en la Figura 3.42, en donde la dirección (A) representa el flujo de datos desde el módulo de gestión de tráfico de datos y comandos, hasta alguno de los módulos de procesamiento de datos y comandos, como se observa el dato contenido en el paquete está conformado por un comando (MSB) y un argumento (SLB). Por otro lado, la dirección (B) representa el flujo de datos desde algún de los módulos de procesamiento de datos y comandos hasta el módulo de gestión de tráfico de datos y comandos, este paquete contiene un dato de temperatura que ocupa el byte MSB y el LSB.

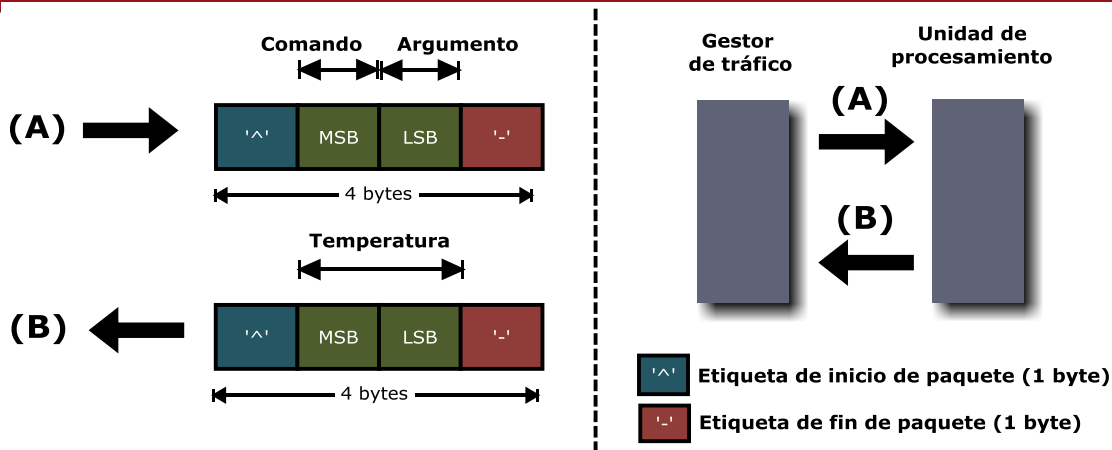


Figura 3.42 Formato de los paquetes en el bus SPI.

Formato de los paquetes en la interfaz RS232:

La comunicación entre el software del módulo de interfaz de usuario (software aplicativo) y el módulo de gestión de tráfico de datos y comandos es en ambos sentidos, por una parte, el módulo de gestión de tráfico de datos y comandos envía los datos a través de la interfaz RS232 hacia el módulo de interfaz de usuario, estos son datos de temperatura y de estado, los últimos indican: el estado de los *led* indicadores de los módulos de adquisición de datos y el estado de los módulos de procesamiento de datos y comandos. Por otro lado, el módulo de interfaz de usuario envía comandos hacia el módulo de gestión de tráfico de datos y comandos para modificar parámetros de configuración o solicitar información acerca del estado de los componentes mencionados. Los formatos de los paquetes de datos son los que se aprecian en la Figura 3.43, como se observa en (A), el sentido de la comunicación es del módulo de interfaz de usuario hacia el módulo de gestión de datos y comandos, este paquete de datos lo conforman 10 *bytes*; el primero, es el comando pudiendo indicar el inicio del proceso de adquisición u otra actividad, los siguientes 9 *bytes* corresponden al argumento del comando, y la posición de cada uno de estos *bytes* corresponden a una unidad de procesamiento específica. El flujo de datos en la dirección (B1) va desde el módulo de gestión de tráfico de datos y comandos hacia el módulo de interfaz de usuario, al igual que el flujo de datos (B2), ambos están conformados por cinco *bytes*. El byte más significativo FF corresponde a una etiqueta que indica el inicio de un paquete de datos, el segundo *byte* indica el tipo de paquete, pudiendo ser un paquete de datos (B1) o de estado (B2), el tercer y cuarto *byte* corresponden al dato transmitido, si es un paquete de datos, el dato corresponderá a una valor de temperatura, si el

paquete es de estado, contendrá el estado de alguno de los módulos de adquisición de datos o procesamiento de datos y comandos, el quinto byte indica la dirección del módulo del cual proviene el paquete.

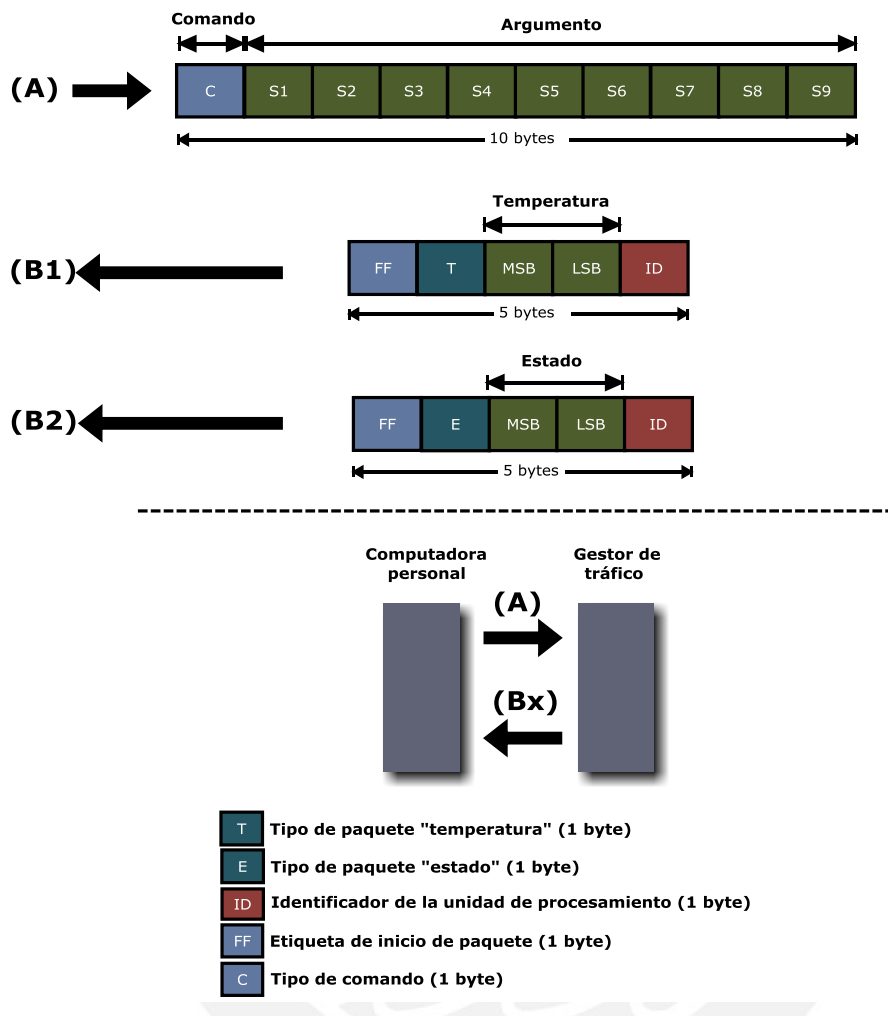


Figura 3.43 Formato de los paquetes en la interfaz RS232.

A) En cuanto al tráfico de datos

El tráfico de datos de temperatura comienza recolectando los datos que cada módulo de procesamiento de datos y comandos haya terminado de procesar, como el tiempo que se demanda en tener un dato disponible es de aproximadamente 66ms, esta unidad cuenta con un temporizador dedicado para cada uno de los nueve módulos de procesamiento de datos y comandos, de modo que se produzca una interrupción por módulo cada 62ms. Una vez producida una interrupción la unidad Maestro (módulo de gestión de tráfico de datos y comandos) lee los datos que tenga disponible una unidad Esclavo (módulo de procesamiento de datos y comandos); luego, estos datos serán almacenados en un *buffer*, transcurrirán 62ms, antes de volver a leer los datos de ese

módulo. Si por algún motivo no se obtiene respuesta, al hacer la petición de datos se abandona el procedimiento, y espera que se produzca otra interrupción. Una vez obtenidos los datos de los módulos de procesamiento de datos y comandos estos son empaquetados de acuerdo al formato descrito anteriormente, el cual consta de cinco bytes, y son enviados al módulo de interfaz de usuario de forma constantemente. La topología de la red de comunicaciones se puede apreciar en la Figura 3.44.

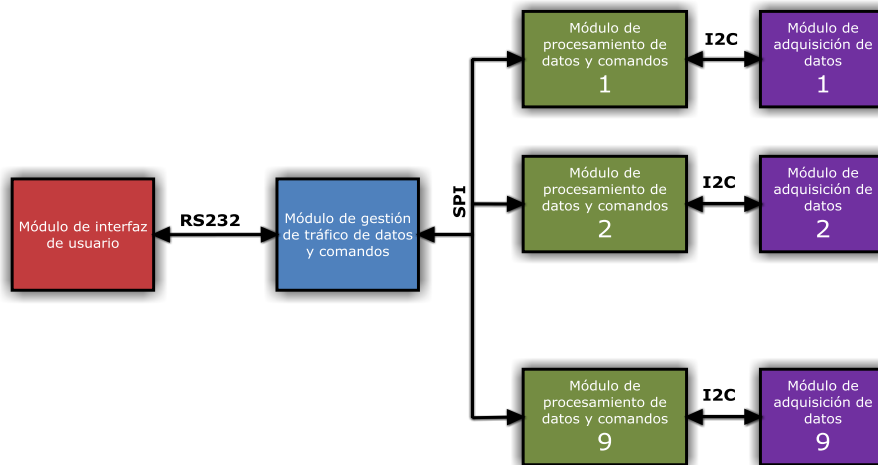


Figura 3.44 Topología de la red de comunicaciones.

B) En cuanto al tráfico de comandos

Respecto al tráfico de comandos, estos son enviados por el módulo de interfaz de usuario usando el paquete de datos descrito anteriormente (Figura 3.43), dichos comandos pueden ser:

‘S’-INICIO

Este comando inicia el proceso de adquisición y procesamiento de datos a cada unidad cuyo argumento Sx sea igual a “1”. Ejemplo:

```
S-S1-S2-S3-S4-S5-S6-S7-S8 → S 0-0-0-0-0-1-1-0-0-0
```

Se envía el comando de inicio a las unidades 5 y 6, solamente.

‘R’-REINICIO:

Este comando reinicia el proceso de adquisición y procesamiento de datos a cada unidad cuyo argumento Sx sea igual a “1”. Ejemplo:

```
R-S1-S2-S3-S4-S5-S6-S7-S8 → R 0-0-0-0-0-0-0-0-1-1
```

Se envía el comando de reinicio a las unidades 8 y 9, solamente.

'L'-LED:

Este comando cambia el estado del led indicador de un módulo de adquisición de datos. Los argumentos pueden ser "e" encendido "a" apagado "p" parpadeo.

Ejemplo:

L-S1-S2-S3-S4-S5-S6-S7-S8 → L-e-e-e-a-a-a-p-p-p

Los led en los módulos 1, 2 y 3 se encenderán, los de los módulos 4, 5,6 se apagarán y los de los módulos 7, 8 y 9, parpadearan.

'E'-EEPROM:

Este comando tiene como fin leer y escribir los valores que contiene el arreglo `ADS_code_table[]` ubicado en la memoria EEPROM en cada módulo de procesamiento de datos y comandos, los argumentos pueden ser 'l' lectura, 'e' escritura y '0' reinicio, Ejemplo:

E-S1-S2-S3-S4-S5-S6-S7-S8 → E 0-0-0-0-0-0-0-0-e

Se inicia el proceso de actualización de la tabla de valores `ADS_code_table[]`. para el módulo de procesamiento de datos y comandos número 9, esto hace a que todas las demás unidades se detengan. No es posible enviar la orden de lectura o escritura a dos unidades al mismo tiempo, con el fin de resguardar la integridad de los datos enviados, el software realiza una comprobación de *checksum* cada 10 bytes enviados.

'T'- RESOLUTION:

Este comando cambia la resolución de la temperatura calculada. Los argumentos pueden ser 'H' milésimas y 'L' centésimas. Ejemplo:

T-S1-S2-S3-S4-S5-S6-S7-S8 → T-L-L-L-L-L-L-L-L-L

Todas las unidades se configuran para devolver valores con centésimas.

'M'- MUESTREO:

Este comando cambia la frecuencia de muestreo de la temperatura. Los argumentos son: 1, 10, 20, 50, 100 y 200. Ejemplo:

T-S1-S2-S3-S4-S5-S6-S7-S8 → T-20-0-0-0-0-0-0-0-0

El período entre muestras será igual a $100\text{ms} * 20 = 2$ segundos, el valor de 100ms siempre es constante, y se multiplicará por el argumento recibido. Este cambio afecta a todos los módulos de adquisición de datos.

3.2.2.4 Módulo de interfaz de usuario

El software del módulo comprende un aplicativo y tiene como fin reflejar las mediciones realizadas en gráficos; los cuales puedan ser visualizados y personalizados por el usuario, esto permite mostrar la evolución del ensayo, al cual se someterá el equipo médico, brindando un resultado global, dicho resultado puede ser exportado para su análisis a otras aplicaciones más avanzadas con el software MATLAB, y el software MS EXCEL, por ser de uso muy cotidiano actualmente, los resultados pueden ser guardados en un formato propio de modo que se pueda contar con esta información en el futuro.

Por otro lado, este aplicativo permite al usuario reconfigurar el sistema, actualizar las tablas de cálculo de cada módulo de procesamiento de datos y enviar comandos para iniciar, reiniciar, detener, cambiar el estado de los *led* indicadores, etc. En la Figura 3.45 se muestra las capas que conforman al software aplicativo



Figura 3.45 Capas de software aplicativo del módulo de interfaz de usuario.

Protocolo de conexión: Esta capa se encarga de establecer la conexión entre el módulo de interfaz de usuario y el módulo de gestión de tráfico de datos y comandos mediante el intercambio de mensajes. Debido que muchos ordenadores poseen más de una interfaz de comunicaciones RS232, el protocolo considera cada una de las interfaces registradas por el sistema operativo como una candidata por la cual intentará establecer conexión. De este modo, envía un mensaje por interfaz esperando recibir una respuesta, cuando se recibe una, el nombre de la interfaz se grabará en un registro, de esta manera la próxima vez que se establezca conexión no será necesario

que se haga nuevamente una búsqueda. La Figura 3.46 ilustra el proceso mencionado, donde la interfaz a la cual está conectado el módulo de gestión de tráfico de datos y comandos es la COM3.

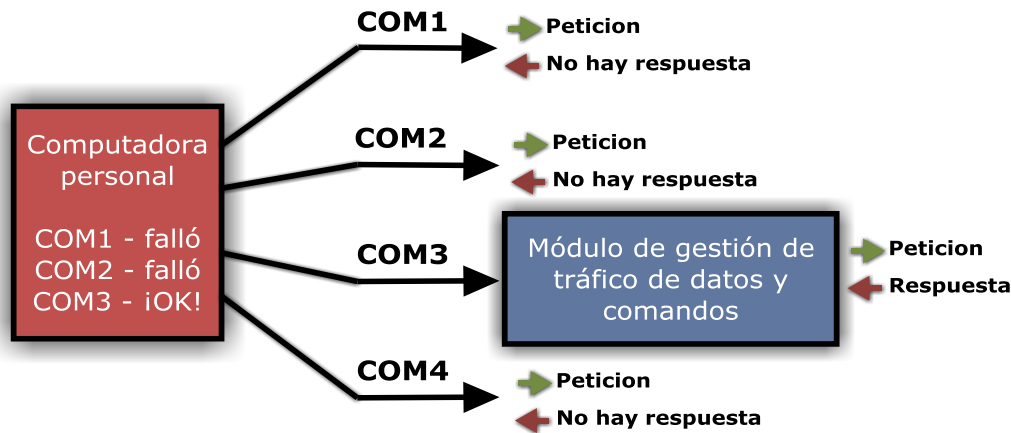


Figura 3.46 Método para identificar la interfaz de comunicaciones específica.

Protocolo de comunicación: Esta capa se encarga de que las capas superiores cuenten con un flujo permanente de datos de temperatura, tiene como función revisar la integridad de los paquetes que serán enviados por el módulo de gestión de datos y comandos, descartando las tramas que presenten errores, en algunos casos reconstruyendo tramas partidas, además se hace la corrección de escala de los datos recibidos (Ejemplo. convierte el dato 2717 a 27.17°C. En la Figura 3.47 se muestra el proceso de verificación de la integridad de los paquetes recibidos.

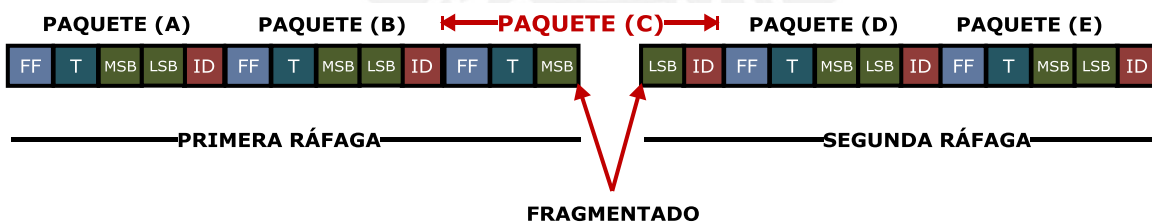


Figura 3.47 Fragmentación de paquetes en la comunicación.

Debido a que los datos son recibidos por ráfagas, implica que la densidad de bytes por unidad de tiempo no es constante, en algunos casos ocurre una fragmentación de paquete, como lo muestra la figura 3.43, se puede observar que en la primera ráfaga se recibieron los paquetes A y B, en cambio el paquete C se fragmentó, y no fue recibido en su totalidad, y que en la segunda ráfaga se reciben los bytes restantes del paquete

C, también los paquete D y E. Este problema se soluciona contando los bytes de cada paquete si un paquete está incompleto los bytes recibidos son guardados hasta que llegan los bytes restantes y el paquete es desfragmentado, como lo muestra la Figura 3.48, donde se aprecia que luego de llegar la segunda ráfaga el paquete C es reconstruido.

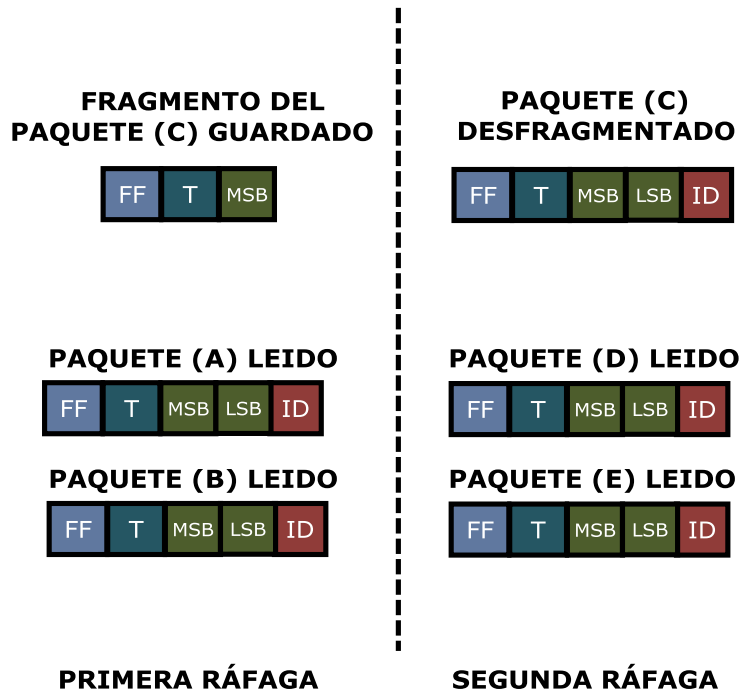


Figura 3.48 Desfragmentación de paquetes.

Un segundo problema es la pérdida de bytes, en la Figura 3.49 se muestra una trama que contiene un error en el paquete B, este error podría producir que los contadores antes mencionados malinterpreten los límites de un paquete de datos. Esto se soluciona verificando los marcadores 'FF' 'T' y 'ID', de este modo cada segmento de una trama será revisada buscando el inicio de un paquete; luego, verificando que contenga los valores de los marcadores mencionados. Si hay pérdida de uno o varios bytes, el paquete es descartado, como lo muestra la Figura 3.49 en la fila (iv).

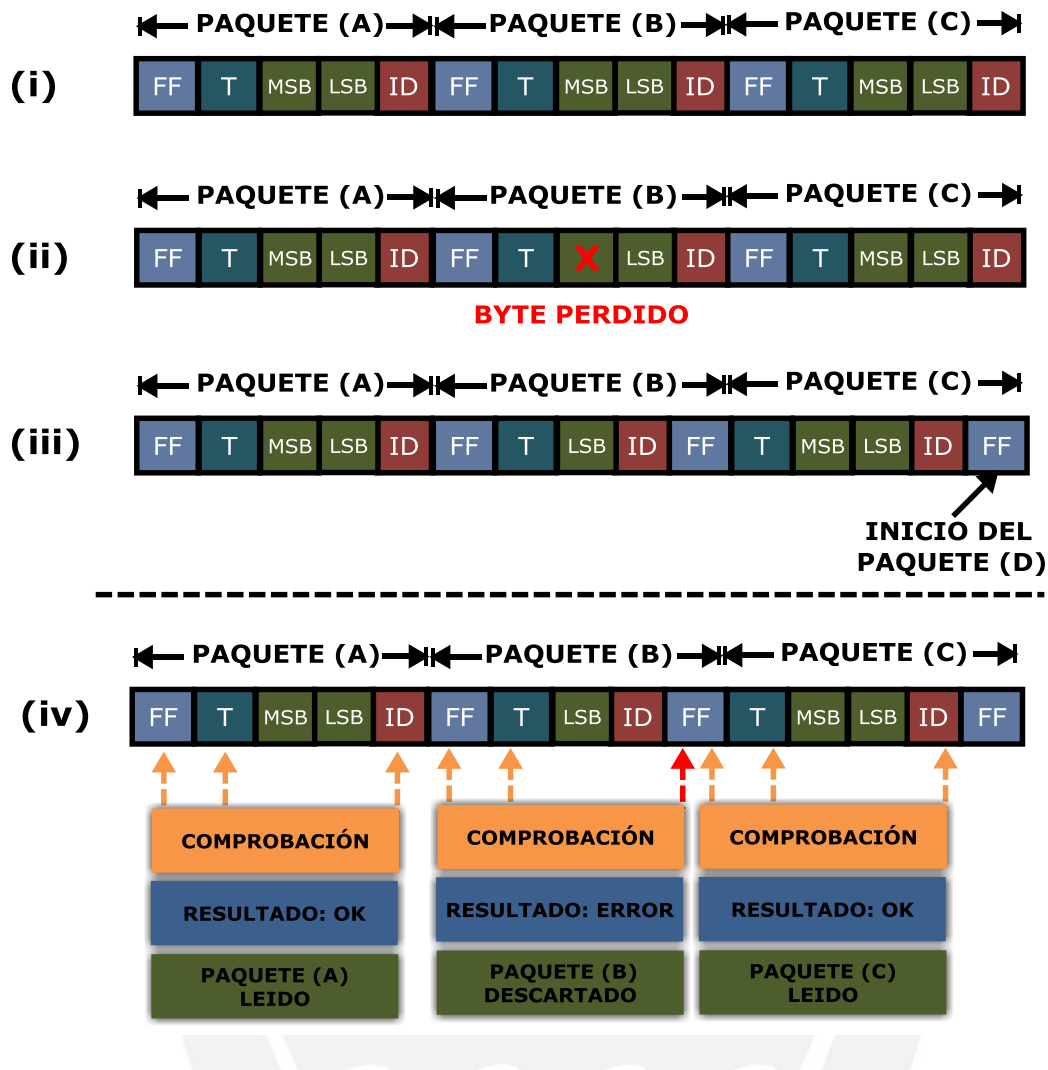


Figura 3.49 Comprobación de la integridad de los paquetes recibidos.

En (i) se muestra una trama íntegra y correcta, en (ii) se muestra que el tercer byte del paquete (B) se ha perdido, en (iii) se aprecia cómo se produce un desplazamiento de bytes confundiendo los límites de cada paquete, en (iv) el protocolo de comunicación comprueba la integridad de cada paquete, determinando su validez, los paquetes defectuoso se descartan.

Conformador gráfico: Esta capa tiene la finalidad de dibujar las curvas correspondientes a los datos que proporciona la capa de comunicación, dichas curvas representarán el comportamiento de la temperatura del al ambiente donde se hayan ubicado los sensores de temperatura

El aplicativo permite observar dos tipos de gráficos uno de ellos muestra la temperatura versus la ubicación del sensor, el segundo modo muestra la temperatura versus

tiempo, como lo muestra la Figura 3.50. Además, es posible exportar los datos en formato de hoja de cálculo de MS EXCEL, como se muestra en la Figura 3.52.

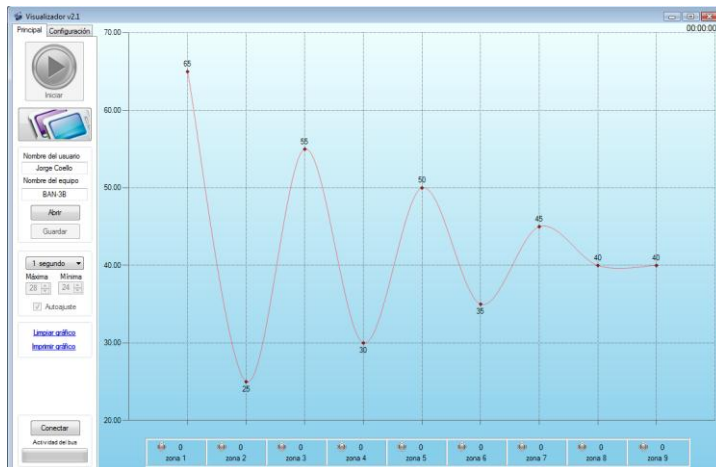


Figura 3.50 Vista de la interfaz de usuario.

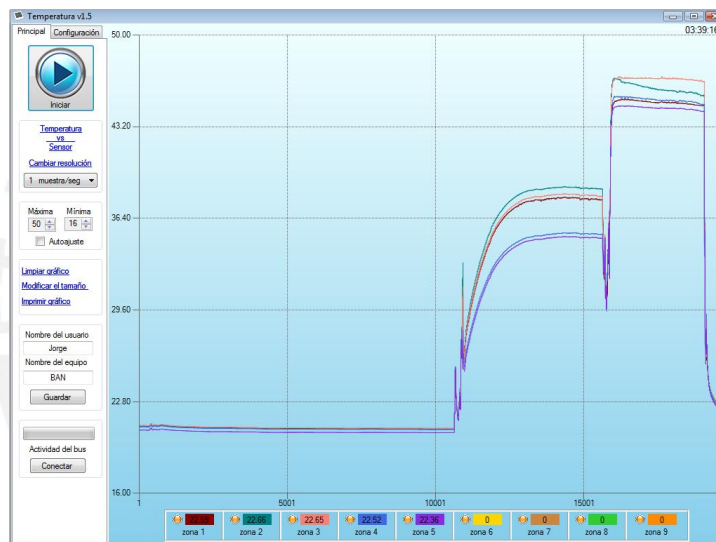


Figura 3.51 Representación Temperatura vs. Tiempo.

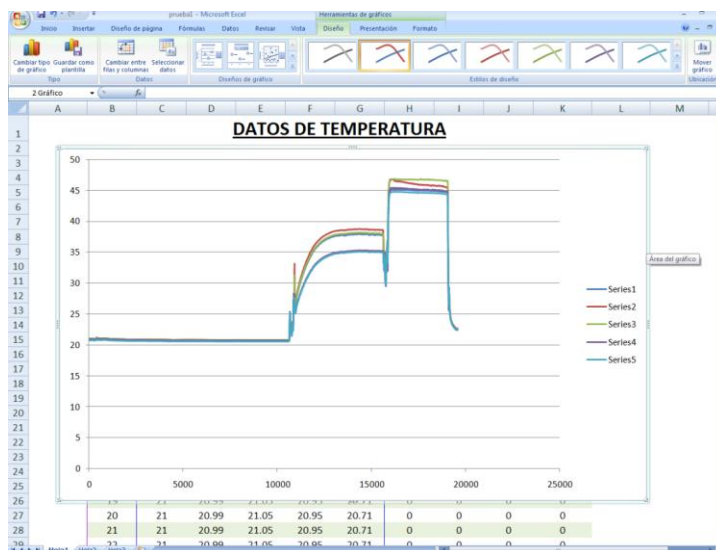


Figura 3.52 Datos exportados a MS EXCEL.

Usuario: En esta capa se ubican todas las funciones y ventanas a las que el usuario tiene acceso, brinda información acerca de los parámetros internos de cada módulo de adquisición de datos y procesamientos de datos y comandos, permite guardar, recuperar y exportar las mediciones realizadas; cambiar el estado de los *led* indicadores, cambiar características de visualización como establecer las cotas máximas y mínimas; hacer *zoom*, desplazarse en el eje de tiempo de la gráfica, actualizar tablas de cálculo de los módulos de procesamiento de datos y comandos, etc.

3.2.2.5 Integración de software

La integración de software fue primeramente entre la etapa el software del módulo de adquisición de datos y el módulo de procesamiento de datos y comandos, una vez comprobada la comunicación se integró el software del módulo de procesamiento de datos y comando con el módulo de gestión de datos y comandos; finalmente, se procedió a integrar el módulo de gestión de tráfico de datos y comandos y el módulo de interfaz de usuario. Cabe señalar que el proceso de integración entre todos los módulo se comprobó al establecer la comunicación entre todos ellos. Una vez se logró comunicar cada uno de los módulos, se implementaron los protocolos definidos por el autor, estos básicamente consisten en comunicaciones mediante paquetes de datos usando señalizaciones de inicio y fin así como comprobaciones para garantizar a la integridad de los datos.

3.3 Pruebas del sistema

Al encender el sistema por primera vez los sensores registraron temperaturas muy similares, sin embargo existía una dispersión de la precisión de hasta 0.18°C con respecto al patrón que registraba 24.11°C , como se aprecia en la Figura 3.53, luego de calibrar los sensores se logró incrementar la precisión a $\pm 0.01^{\circ}\text{C}$ como se observa en la Figura 3.54.

El patrón usado fue el termómetro Ertco-Eutechnics. 4400 con resolución de 0.01°C y una exactitud de $\pm 0.015^{\circ}\text{C}$, entre -5°C y 110°C . [13]

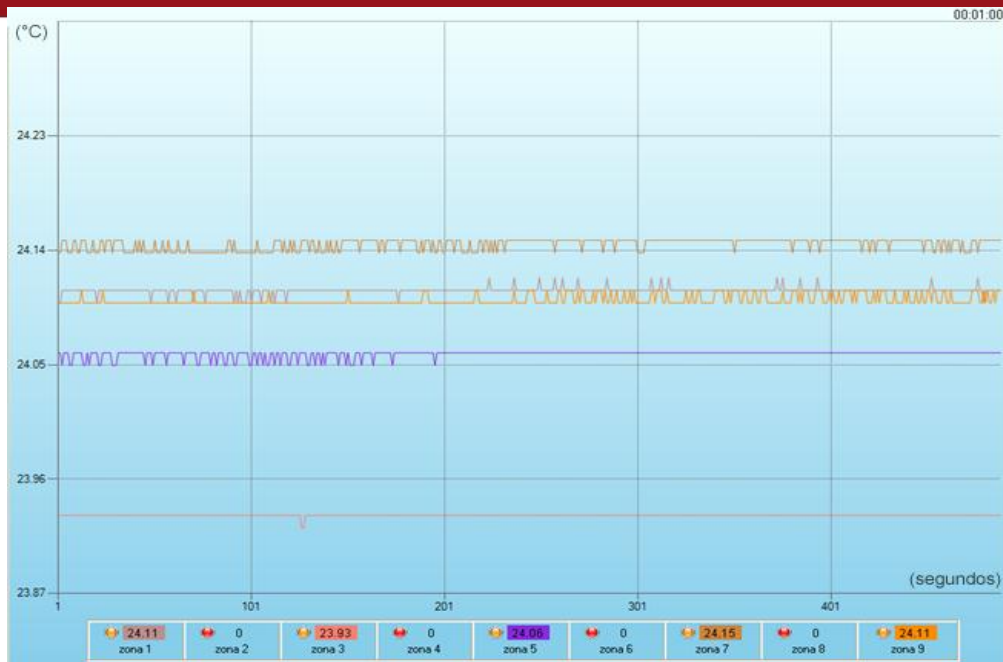


Figura 3.53 Primera prueba: error precisión de 0.18°C.

Tiempo	zona 1	zona 2	zona 3	zona 4	zona 5	zona 6	zona 7	zona 8	zona 9
1	24.1	0	23.93	0	24.06	0	24.14	0	24.1
2	24.11	0	23.93	0	24.05	0	24.15	0	24.1
3	24.11	0	23.93	0	24.06	0	24.15	0	24.1

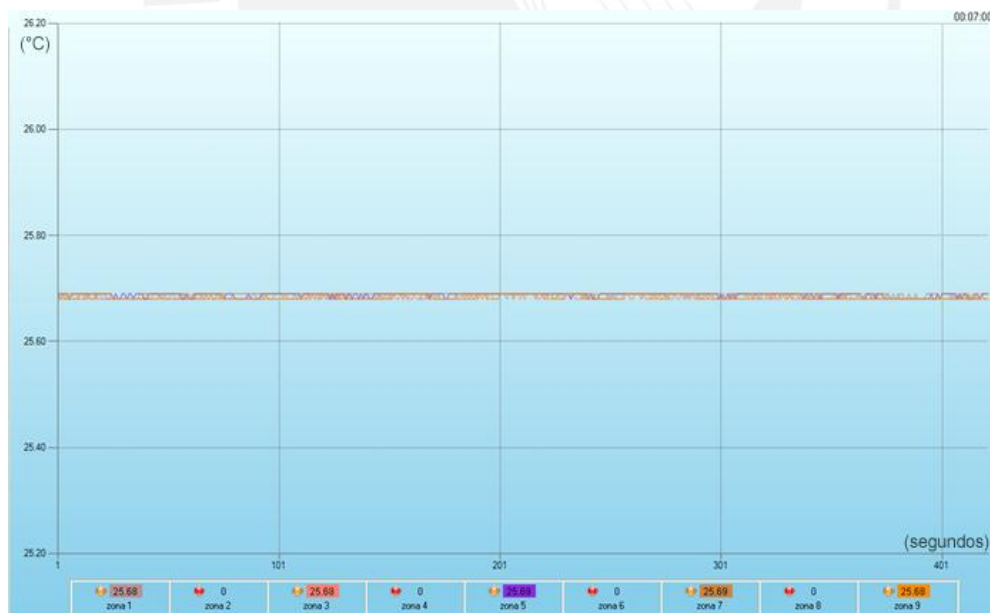


Figura 3.54 Segunda prueba: error de precisión máximo de 0.01°C

Tiempo	zona 1	zona 2	zona 3	zona 4	zona 5	zona 6	zona 7	zona 8	zona 9
1	25.68	0	25.68	0	25.69	0	25.68	0	25.68
2	25.69	0	25.68	0	25.69	0	25.69	0	25.69
3	25.68	0	25.68	0	25.69	0	25.69	0	25.68

CAPÍTULO 4

RESULTADOS

El resultado de este trabajo de tesis que es el sistema electrónico para la de medición y evaluación de la distribución de temperaturas en incubadoras ha sido diseñado, implementado y probado. El sistema actualmente posee cinco unidades independientes de medición de temperatura, ampliables a nueve, las cuales pueden a medir temperaturas entre 0°C hasta 100°C; sin embargo, solo se calibró el instrumento con el patrón [13], entre +20°C y +55°C.

En la Figura 4.1 se aprecia el sistema con las cinco unidades de medición de temperatura.

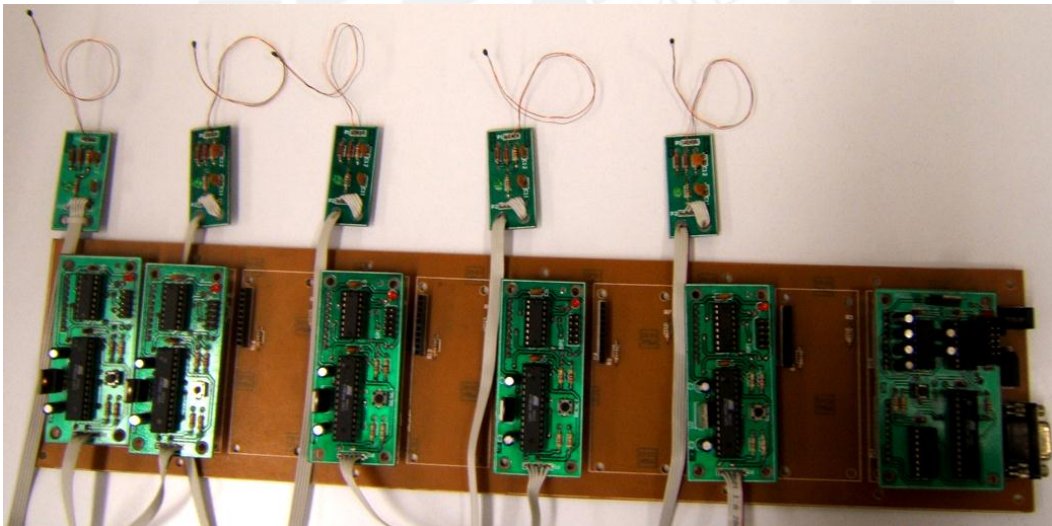


Figura 4.1 Tarjetas electrónicas completamente terminadas.

Se puede observar en la Figura 4.2 dos tarjetas electrónicas, la de la izquierda corresponde a un módulo de procesamiento de datos y comandos y la de la derecha al módulo de gestión de tráfico de datos y comandos.

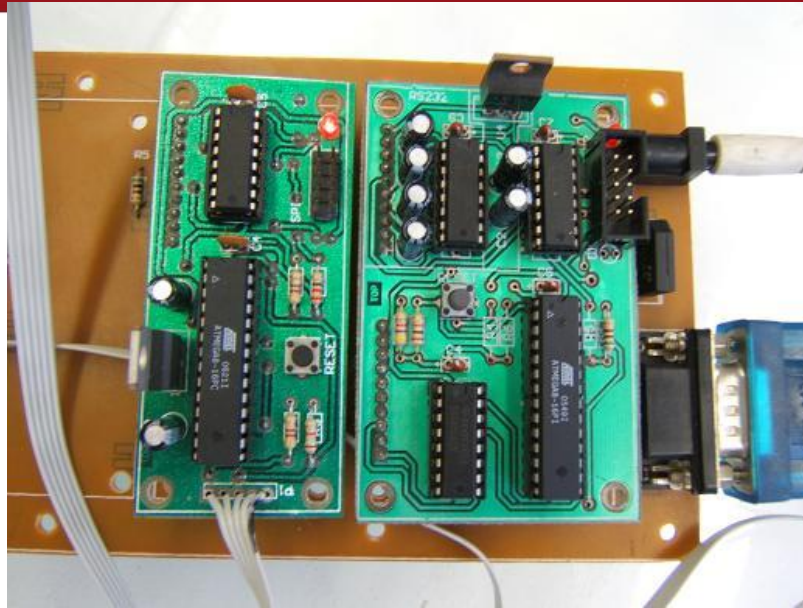


Figura 4.2 A la derecha la tarjeta de gestión de tráfico de datos y comandos, a la izquierda una de las tarjetas de procesamiento de datos y comandos.

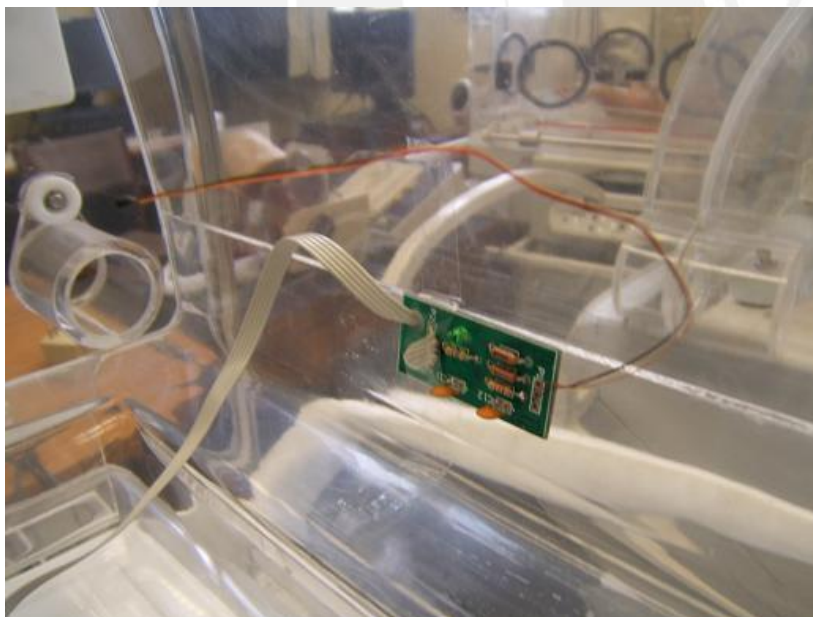


Figura 4.3 Tarjeta de adquisición de datos y el sensor de temperatura ubicada dentro del habitáculo de una incubadora.

En la Figura 4.3, se puede apreciar un módulo de adquisición de datos con su respectivo sensor de temperatura.

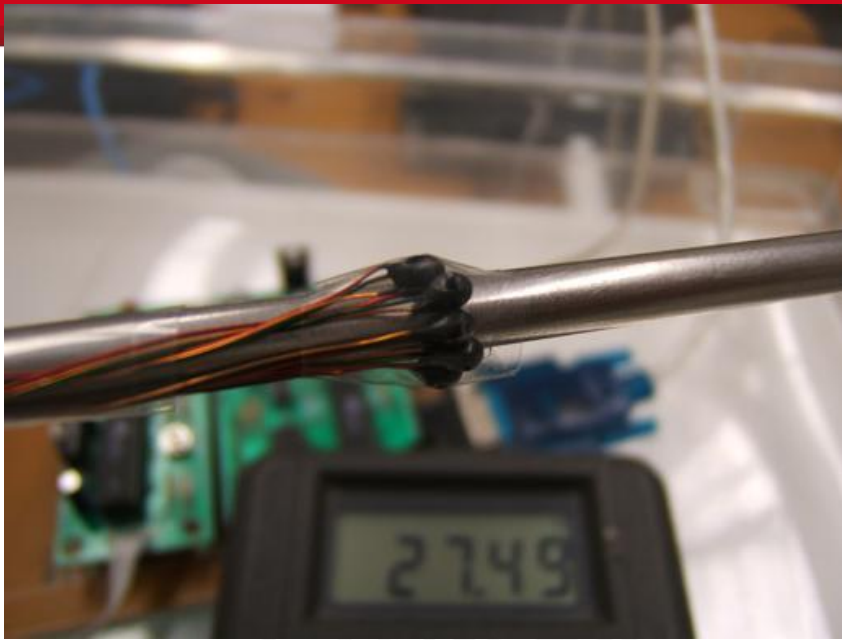


Figura 4.4 Sensores dispuestos para su calibración.

La Figura 4.4 muestra cinco sensores acoplados a un sensor mucho más grande (barra metálica), con el propósito de ser calibrarlos mediante software.

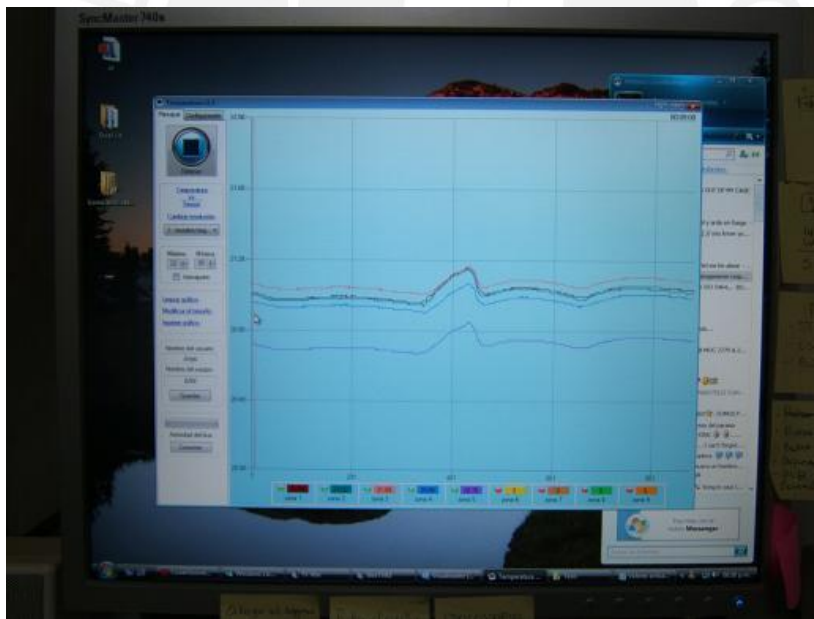


Figura 4.5 Aplicación de software mostrando las variaciones de temperatura.

La Figura 4.5 muestra la aplicación de software terminada, en la cual se observa la variación de temperaturas con respecto al tiempo

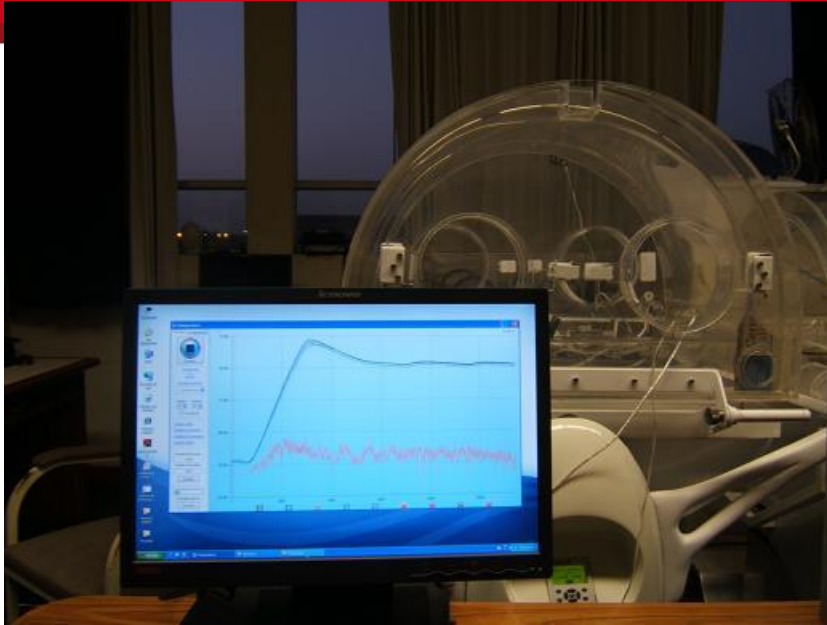


Figura 4.6 Sistema completo evaluando el desempeño térmico de una incubadora.

La eficiencia del sistema se comprobó evaluando el desempeño térmico de un prototipo de incubadora denominado “Burbuja Neonatal Andina”, como se aprecia en la Figura 4.6. Como resultado de esta evaluación se determinó que en esta incubadora la temperatura en distintos puntos al interior del habitáculo neonatal tiene una variación máxima de 0.6°C , un tiempo de establecimiento de aproximadamente 2.4 horas, y un sobre impulso de 3.4°C .

CONCLUSIONES

Del desarrollo del presente trabajo de tesis, se desprenden las siguientes conclusiones relevantes:

Los objetivos iniciales han sido alcanzados en su totalidad. Se diseñó e implementó con éxito el hardware necesario para la adquisición, procesamiento y transporte de datos de temperatura. Se culminó exitosamente el diseño e implementación del software necesario para la adquisición, procesamiento, transporte y visualización de datos de temperatura. Se logró integrar satisfactoriamente todos y cada uno de los módulos en un único sistema, conformando un instrumento dedicado a la evaluación del desempeño térmico en incubadoras neonatales.

Este instrumento, aminorará el tiempo necesario para la evaluación de incubadoras neonatales, contribuirá mejorando las capacidades de funcionamiento de dichos equipos, esto a su vez, incrementará la calidad de atención a neonatos de alto riesgo en los centros de salud.

RECOMENDACIONES

En cuanto al hardware; es recomendable usar componentes de montura superficial para reducir el tamaño y en algunos casos evitar el uso de disipadores, sería conveniente que las próximas implementaciones consideren el uso de una etapa única para regular y distribuir el voltaje, esto disminuiría costos de fabricación. En cuanto al método de conexión entre las tarjetas, sería recomendable el uso de conectores del tipo SIMM, ello ahorraría espacio y se traduciría en un menor costo en cuanto a circuito impreso.

El bus de comunicaciones que conecta al módulo de adquisición de datos y el módulo de procesamiento de datos y comandos, podría ser implementado usando tecnología inalámbrica, esto incrementaría la facilidad de uso del instrumento. Actualmente, la interfaz RS232 no es muy usada en dispositivos portátiles, sería conveniente adicionar un puente de comunicaciones USB-RS232 o usar un puente de comunicaciones *Bluetooth*, el cual, permitiría mantener el sistema intacto y al mismo tiempo ampliar su conectividad y la facilidad de uso, en el caso de usar una interfaz USB esta podría, suministrar energía eléctrica al sistema (hasta 400mA), de modo tal, que se descarte la fuente externa que actualmente se usa. Es recomendable, no implementar los puertos de programación para cada módulo de procesamiento de datos y comandos, pues aumentan los costos y no tienen mayor uso salvo en el momento de la implementación del sistema ya que es posible programar cada uno de los micros controladores usando el bus de comunicaciones SPI que todos comparten adicionando una unidad de control alternativa para direccionar a cada microcontrolador y habilitarlo en modo de programación. En cuanto al recinto que albergará a todas las tarjetas electrónicas, es prudente que cuente ranuras que permitan una ventilación adecuada.

Finalmente; en cuanto al *firmware* del sistema este puede ser mejorado principalmente en el aspecto de aminorar el tráfico de datos; sin embargo, el software en su totalidad debería ser optimizado. Con respecto al aplicativo sería recomendable implementar una sección de ayuda para que el usuario tenga referencias de cómo usar el sistema.

BIBLIOGRAFÍA

- [1] Pizarro, Alejandro. *IMPLEMENTACION DE UN SISTEMA DE EVALUACION TERMICA DE INCUBADORAS NEONATALES EN EL HOSPITAL NACIONAL DOS DE MAYO*. Lima: Tesis de pregrado PUCP, 2010.
- [2] Castellón, Bruno. *Grupo de Investigación de Equipos Médicos y Sistemas*. Lima: <http://www.pucp.edu.pe/grupo/gidems/>, 1994.
- [3] Iec. *INTERNATIONAL ELECTROTECHNICAL COMMISSION*. <http://www.iec.ch>, 1996.
- [4] Malinski, L. Bychto and M. *1D or 3D spatial temperature distribution- comparison*. Koszalin: http://jp4.journaldephysique.org/index.php?option=com_article&access=doi&doi=10.1051/jp4:2004117004&Itemid=129, 2004.
- [5] Obregón Núñez, José Miguel. *DESARROLLO DE UN SISTEMA DE ADQUISICIÓN DE DATOS DE TEMPERATURA PARA EVALUACIÓN FUNCIONAL DE INCUBADORAS*. Editado por Tesis de pregrado PUCP. Lima, 2007.
- [6] Gobierno, México. *CENTRO NACIONAL DE EXCELENCIA TECNOLÓGICA EN SALUD, DEL GOBIERNO MEXICANO*. www.cenetec.salud.gob.mx/descargas/equipo_guias/guias_tec/4gt_incubadora.pdf, 2004.
- [7] Gobierno, Ingles. *DEPARTAMENTO DE SALUD DEL GOBIERNO INGLES*. <http://www.dh.gov.uk/en/index.htm>, 2003.
- [8] —. *YSI PRECISION THERMISTORS & PROBES*. "\Anexos\Hardware\Hojas Técnicas\DataSheet Termilineal.pdf", 2007.
- [9] Coello, Jorge. *Linealización*. Lima: "\Anexos\Software\Análisis para la linealización\Linealización (código matlab para el análisis).pdf", 2007.
- [10] Vishay. *Metal Film Resistors, High Precision, High Stability*.

<http://www.vishay.com/>, 2008.

- [11] Instrumens, Texas. *16-Bit ANALOG-TO-DIGITAL CONVERTER*. USA, 2003.
- [12] Linear, Technology. *LINEAR TECHNOLOGY CORPORATION*.
<http://www.ortodoxism.ro/datasheets/lineartechnology/1461f.pdf>, 1999.
- [13] Coello, Jorge. *Cálculo del error por tolerancia y coeficiente térmico*. Lima:
"Anexos\Software\Análisis para la linealización\Cálculo del error por tolerancia y
coeficiente térmico (código matlab para el análisis) (código matlab para el análisis).pdf",
2007.
- [14] Instruments, Fluke. *SERIE 170 DE MULTÍMETROS DIGITALES FLUKE*.
http://www.fluke.nl/comx/show_product.aspx?pid=29983&product=HMASUB&type=3&locale=PEes, 2008 .
- [15] Instruments, Ertco. *ERTCO ® THERMOMETERS*.
http://www.coleparmer.com/catalog/product_index.asp?cls=16899, 2008.
- [16] Wolf, W.H. Dept. of Electr. Eng., Princeton Univ., NJ. *HARDWARE-SOFTWARE CO-DESIGN OF EMBEDDED SYSTEMS*.
http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=293155, 1994.
- [17] Webster, Jhon G. *THE MEASUREMENT, INSTRUMENTATION AND SENSORS HANDBOOK (ELECTRICAL ENGINEERING HANDBOOK)*. 1999.
- [18] Universidad de Cádiz, Grupo de electrónica. *Conversión Digital/Analógica*.
http://www2.uca.es/grup-nvest/instrument_electro/ppjjgdr/, 2005.
- [19] Universidad Autónoma de Guadalajara, Curso. *MATERIAL DE ENSEÑANZA CURSO REDES DE COMPUTADORAS*.
http://genesis.uag.mx/edmedia/material/comuelectro/uni1_2_7.cfm, 2005.
- [20] *MEASUREMENT AND INSTRUMENTATION PRINCIPLES*. 2001.
- [21] Silverman, William A. *DIAGNOSIS AND TREATMENT: USE AND MISUSE OF*

TEMPERATURE AND HUMIDITY IN CARE OF THE NEWBORN INFANT.

<http://pediatrics.aappublications.org/cgi/reprint/33/2/276>, 1964.

- [22] Schlichthärle, Dietrich. *DIGITAL FILTERS: BASICS AND DESIGN* . 2000.
- [23] Ramón Pallás-Areny, John G. Webster. *SENSORS AND SIGNAL CONDITIONING, 2ND EDITION*. 2001.
- [24] Rae. *DICCIONARIO EN LINEA DE LA REAL ACADEMIA DE LA LENGUA ESPAÑOLA*. <http://www.rae.es>, 2007.
- [25] Plassche, Rudy J. van de. *CMOS INTEGRATED ANALOG-TO-DIGITAL AND DIGITAL-TO-ANALOG CONVERTER*. 2003.
- [26] PALLAS, Ramón. Editorial Marcombo. *SENSORES Y ACONDICIONADORES DE SEÑAL*. 1998.
- [27] Norberto Santos et al., Asociación Argentina de Perinatología. *ACTUALIZACIÓN DE TEMAS NEONATALES TOMO I*. 2005.
- [28] Maniktala, Sanjaya. *SWITCHING POWER SUPPLIES A TO Z*. 2006.
- [29] Lenk, Ron. *PRACTICAL DESIGN OF POWER SUPPLIES* . 1998.
- [30] Kuhnel, Claus. *AVR RISC MICROCONTROLLER HANDBOOK*. 1998.
- [31] Khandpur, R. S. *BIOMEDICAL INSTRUMENTATION: TECHNOLOGY AND APPLICATIONS*. 2005.
- [32] J. G. Webster, Ed. John Wiley & Sons Inc., USA. *MEDICAL INSTRUMENTATION. APPLICATION AND DESIGN*. 1998.
- [33] J. G. Webster, Ed. John Wiley & Sons Inc., USA. *ENCYCLOPEDIA OF MEDICAL DEVICES AND INSTRUMENTATION*. 1998.
- [34] Heath, Steve. *MICROPROCESSOR ARCHITECTURES RISC, CISC AND DSP*. 2005.

- [35] Government, FEDERAL STANDARD U.S. *INTERCHANGEABILITY DEFINITION*.
http://www.its.bldrdoc.gov/fs-1037/dir-019/_2778.htm, 1994.
- [36] Githler, Charles E. *THERMOMETER*.
<http://www.google.com/patents?id=RMx4AAAAEBAJ&dq=THERMOMETER+1999173,1993>.
- [37] Corporation, Atmel. *Hoja de datos microcontrolador ATmega8*.
http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf, 2007 .
- [38] Chen, Wai-Kai. *THE CIRCUITS AND FILTERS HANDBOOK*. 2002.
- [39] Bychto, L., y M. Maliński. *1D OR 3D SPATIAL TEMPERATURE DISTRIBUTION-COMPARISON*. <http://adsabs.harvard.edu/abs/2004JPhy4.117...17B>, 2004.
- [40] Burley. *WHEATSTONE BRIDGE CIRCUIT FOR THE MEASUREMENT OF A PHYSICAL VARIABLE REMOTE*.
<http://www.google.com/patents?id=zRpKAAAAEBAJ&dq=Wheatstone+bridge+circuit,1961>.
- [41] Bruno Castellón, Eduardo Ajito. *GIDEMS NEONTAL ENGINEERING*. 2007.
- [42] al, David M. Masarik et. *SIGNAL PROCESSOR FOR INFRARED CAMERA*.
<http://www.google.com/patents?id=91lcAAAAEBAJ&dq=infrared+camera,1994>.
- [43] Kaiser, Cletus J. *The Resistor Handbook*. Second Edition. USA, 1998.