

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Electrónica**



**Desarrollo de un sistema de monitoreo de temperatura  
para la sala de equipos de telecomunicaciones del ICE**

**Informe de Proyecto de Graduación para optar por el título de Ingeniero en  
Electrónica con el grado académico de Licenciatura**

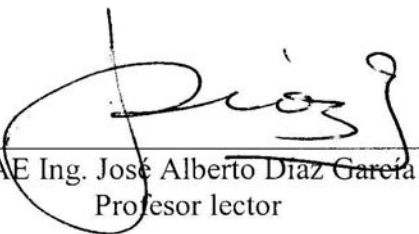
**Enrique Con Hong**

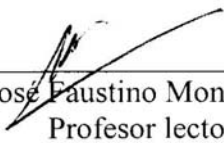
**Cartago, Junio del 2006**


**INSTITUTO TECNOLOGICO DE COSTA RICA**  
**ESCUELA DE INGENIERIA ELECTRONICA**  
**PROYECTO DE GRADUACIÓN**  
**TRIBUNAL EVALUADOR**

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

  
MAE Ing. José Alberto Díaz García  
Profesor lector

  
Ing. José Faustino Montes de Oca  
Profesor lector

  
Msc. Ing. Julio Stradi Granados  
Profesor asesor



Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 28 de junio, 2006

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 2006



Enrique Manuel Con Hong

Céd: 1-1090-0996

## Resumen

El Proceso de Transmisión del ICE de San Pedro cuenta con una sala de equipos de telecomunicaciones, cuyo sistema de enfriamiento (aire acondicionado y ventilas) no logra mantener una temperatura baja uniforme, lo que da como resultado un sobrecalentamiento de los equipos con los efectos negativos que esto conlleva (disminución del ciclo de vida y averías).

Este problema representa a la empresa un mayor costo económico, debido que requieren de un personal técnico para la reparación y mantenimiento, tener un inventario de componentes de repuestos y el tiempo invertido en la búsqueda de averías y en su debida reparación.

Por estas razones es necesario realizar un sistema de monitoreo de la temperatura de la sala para determinar las zonas que pueden perjudicar los equipos y analizar posibles soluciones.

Para implementar el sistema de monitoreo se colocaron módulos con sensores de temperatura en los bastidores, en donde se encuentra ubicados los equipos de telecomunicación. Estos módulos se comunican entre sí por medio de enlaces inalámbricos infrarrojos.

Una vez recolectados los datos, éstos son enviados a una computadora que se encarga de visualizar por medio de gráficos el comportamiento térmico de la sala en estudio y además guardar la información en una base de datos. También se desarrolló el software para automatizar proceso de cambiar la temperatura de acuerdo a los valores recolectados.

## **Abstract**

The ICE Transmission Department of San Pedro has a telecommunications equipment room, in which they have many kinds of equipment, for example, routers, RF transceivers, and fiber optic equipment. This variety of equipment produces a non uniform temperature across the room, which present a high temperature zones that results in an overheating problem in some equipments, giving as a result life cycle degradation and potential equipment damage.

This problem represents a significant company investment in: technical personnel for repair and maintenance, spare parts stock and troubleshooting time.

For these reasons a temperature monitor system is needed, so the high risk areas can be identify and also analyzed to find the proper solution

To implement the monitor system, modules with temperature sensors were distributed across the equipment racks. The wireless communication between the modules is via infrared links.

The collected data is sent to a personal computer which includes software with the ability to display the thermal behavior of the room in 3D, and store the data in a data base. Also an automatization program was developed in order to facilitate the process of adjust the optimal temperature.

*A mi familia y amigos, por toda la ayuda brindada  
durante toda mi carrera universitaria*

*Muchas gracias*

### *Agradecimientos*

Agradezco al profesor Ing. Julio Stradi por todo el tiempo y ayuda que me brindó para llevar adelante el presente proyecto.

Agradezco también a todos mis compañeros que de una u otra manera me ayudaron en el desarrollo de este trabajo, con su tiempo y sus consejos.

Muchas gracias

## INDICE GENERAL

Resumen .....	III
Abstract .....	IV
Capítulo 1: Introducción.....	1
1.1 Problema existente e importancia de su solución .....	1
1.2 Solución seleccionada .....	3
Capítulo 2: Meta y objetivos .....	4
2.1 Meta.....	4
2.2 Objetivo General.....	4
2.3 Objetivos específicos .....	4
a. Objetivos de hardware .....	4
b. Objetivos de software.....	5
c. Objetivos de documentación .....	5
d. Objetivos de implementación .....	6
Capítulo 3: Marco Teórico .....	7
3.1 Sensores de Temperatura .....	7
3.1.1 Termopares .....	7
3.1.2 Resistivos.....	8
3.1.2.1 Detector de resistencia por temperatura (RTD).....	8
3.1.2.2 Termistores.....	9
3.1.3 Semiconductores.....	9
3.1.4 Sensor de temperatura LM35 .....	10
3.2 Redes inalámbricas .....	11
3.3 Comunicación Infrarroja.....	13
3.4 Clasificación de redes por transmisión de datos.....	15
3.5 Variables Referenciadas (Casting o transformaciones de tipos).....	15
3.5.1 Implícito.....	15
3.5.2 Explícito.....	16
3.6 Universal Asynchronous Receiver/Transmitter (UART).....	16
3.6.1 Transmisión serial sincrónica .....	16
3.6.2 Transmisión serial asincrónica .....	17



3.7	Estándar RS232.....	18
3.8	Amplificador operacional en configuración no inversora.....	19
3.9	Análisis de la repuesta en el tiempo .....	20
3.9.1	Respuesta en el tiempo de sistemas en tiempo continuo .....	20
3.9.2	Señal de prueba .....	21
3.9.3	Parámetros de la respuesta ante un escalón .....	22
Capítulo 4: Procedimiento metodológico.....		24
4.1	Reconocimiento y definición del problema.....	24
4.2	Obtención y análisis de la información.....	25
4.3	Evaluación de alternativas y síntesis de una solución .....	26
4.3.1	Investigación de experiencias anteriores sobre proyectos realizados	26
4.3.2	Recomendación de expertos en la materia .....	26
4.3.2	Evaluación de alternativas .....	27
4.4	Metodología .....	29
4.4.1	Investigación y desarrollo del comportamiento de la temperatura en la sala de estudio .....	29
4.4.2	Investigación de los componentes necesarios para el desarrollo del proyecto .....	29
4.4.3	Prueba de los componentes adquiridos .....	30
4.4.4	Desarrollo del software encargado de la adquisición de datos .....	30
4.4.5	Desarrollo del software encargado del despliegue de datos en la PC	30
4.4.6	Medios de difusión para dar a conocer los resultados .....	31
4.5	Reevaluación y rediseño.....	31
Capítulo 5: Descripción detallada de la solución (Explicación del diseño) .....		32
5.1	Descripción del hardware.....	32
5.1.1	Diagrama de primer nivel .....	32
5.1.2	Diagrama de segundo nivel.....	33
5.1.2.1	Módulo Principal .....	34
5.1.2.2	Módulos Intermedios .....	35
5.1.2.3	Módulos extremos .....	37
5.1.3	Diagrama de tercer nivel .....	37
5.1.3.1	Módulo principal.....	37
5.1.3.1.1	Bloque de alimentación.....	37

5.1.3.1.2	Bloque de procesamiento de datos .....	37
5.1.3.1.3	Bloque de comunicación.....	38
5.1.3.2	Módulo intermedio .....	39
5.1.3.2.1	Bloque de recolección de datos de temperatura.....	39
5.1.3.2.2	Bloque de procesamiento de datos .....	39
5.1.3.2.3	Bloque de comunicación.....	40
5.1.3.3	Módulo extremo .....	43
5.1.3.3.1	Bloque de comunicación.....	43
5.1.4	Diagrama de cuarto nivel .....	45
5.1.4.1	Bloque de alimentación .....	45
5.1.4.2	Bloque de procesamiento de datos .....	46
5.1.4.3	Bloque de recolección de datos.....	50
5.1.4.4	Bloque de comunicación infrarroja (IR).....	54
5.1.4.5	Bloque Interfaz RS232.....	59
5.1.4.6	Bloque Tri-State.....	60
5.1.5	Diagrama de quinto nivel.....	63
5.2	Descripción del software.....	66
5.2.1	Criterios de diseño .....	66
5.2.2	Descripción del programa principal .....	68
5.2.2.1	Arreglo de datos .....	68
5.2.2.2	Esquema del protocolo implementado.....	70
5.2.2.2.1	Petición de datos .....	71
5.2.2.2.2	Recolección de datos .....	71
5.2.2.2.3	Generación de error.....	72
5.2.3	Descripción de la interfaz de usuario .....	74
5.2.4	Diagramas de Flujo .....	75
5.2.4.1	Diagrama de flujo del Modulo Principal .....	75
5.2.4.2	Diagrama de flujo del Modulo Intermedio .....	77
5.2.4.3	Diagrama de flujo del modulo final.....	79
5.2.4.4	Diagrama de flujo del programa de Delphi .....	80
5.2.4.5	Diagrama de flujo del programa de control de temperatura.....	82
5.2.5	Métodos utilizados.....	84
5.2.5.1	Rutinas de conversión de tipos.....	84
5.2.5.2	Factor de conversión .....	85
Capítulo 6:	Análisis de Resultados.....	86
6.1	Prueba de los componentes principales .....	86
6.1.1	Sensores de temperatura.....	86
6.1.2	Transceivers infrarrojos.....	88

6.1.3	Microcontrolador Pic16F873.....	92
6.2	Comunicación entre microcontrolador y PC.....	93
6.3	Cuantificación de la temperatura en los equipos .....	94
6.4	Modelado matemático.....	97
6.5	Interfaz con el usuario.....	101
Capítulo 7: Conclusiones y recomendaciones.....		108
7.1	Conclusiones .....	108
7.2	Recomendaciones .....	109
Bibliografía .....		110
Apéndices.....		112
A.1	Glosario, abreviaturas y simbología .....	113
A.2.	Información sobre la empresa/institución .....	115
A.2.1	Descripción de la empresa .....	115
A.2.2	Descripción del área donde se realizará el proyecto .....	116
A.2.3	Antecedentes prácticos.....	117
Apéndice A.3 Programas .....		118
Apéndice A.3.1 Programa genérico de los nodos del sistema .....		118
Apéndice A.3.2 Programa de control de temperatura .....		126
Apéndice A.4 Circuitos impresos .....		128
Apéndice A.4.1 Nodo principal .....		128
ANEXOS .....		131

## INDICE DE FIGURAS

Figura 1.1	Esquema general de la sala de equipos de telecomunicaciones .....	1
Figura 3.1	Esquema de un amplificador no inversor .....	19
Figura 3.2	Respuesta ante un escalón .....	22
Figura 5.1	Diagrama de primer nivel .....	32
Figura 5.2	Esquema de ubicación física de los dispositivos .....	33
Figura 5.3	Esquema general del módulo principal .....	34
Figura 5.4	Esquema general del módulo intermedio .....	36
Figura 5.5	Diagrama de tercer nivel del módulo principal .....	38
Figura 5.6	Diagrama de tercer nivel del módulo intermedio .....	42
Figura 5.7	Diagrama de tercer nivel del módulo extremo .....	44
Figura 5.8	Diagrama del regulador LM340 .....	45
Figura 5.9	Diagrama de pines del pic16F873 .....	49
Figura 5.10	Esquema del sensor de temperatura LM35 .....	50
Figura 5.11	Esquema del amplificador de señal .....	53
Figura 5.12	Diagrama de pines del MCP2120 .....	56
Figura 5.13	Diagrama de pines del Transceiver HSDL3612 .....	57
Figura 5.14	Esquema del Transceiver HSDL3612 .....	58
Figura 5.15	Diagrama de conexión del bloque de comunicación IR .....	59
Figura 5.16	Esquema de conexión de componentes externos del MAX232 .....	60
Figura 5.17	Esquema interno del dispositivo Tri-State DM74LS125A .....	62
Figura 5.18	Circuito del módulo principal .....	63
Figura 5.19	Circuito del módulo intermedio .....	64
Figura 5.20	Circuito del módulo extremo .....	65
Figura 5.21	Diagrama del arreglo de datos .....	68
Figura 5.22	Diagrama del flujo de datos del protocolo de comunicación implementado .....	73
Figura 5.23	Diagrama de flujo del módulo principal .....	76
Figura 5.24	Diagrama de flujo del módulo intermedio .....	78
Figura 5.25	Diagrama de flujo del módulo final .....	79
Figura 5.26	Diagrama de flujo del programa Delphi .....	81
Figura 5.27	Diagrama de flujo del sistema de control de temperatura .....	83
Figura 5.28	Relación entre Temperatura vs Resolución .....	85
Figura 6.1	Esquema del circuito de prueba del sensor de temperatura .....	86
Figura 6.2	Programa de prueba para visualizar los datos de temperatura .....	87
Figura 6.3	Circuito impreso del sensor HSDL3612 .....	88
Figura 6.4	Circuitos de prueba del módulo extremo (izquierda) y del módulo principal (derecha) .....	89
Figura 6.5	Esquema del circuito para prueba del HSDL3612 .....	89
Figura 6.6	Recepción de datos utilizando la herramienta Hyper Terminal .....	90

Figura 6.4	Esquema del ángulo máximo entre transceivers IR .....	91
Figura 6.5	Interfaz de prueba de comunicación entre PC y microcontrolador ....	94
Figura 6.6	Gráfica térmica del equipo NEC DMR-200 A .....	94
Figura 6.7	Gráfica térmica del equipo NEC DMR-200 B .....	95
Figura 6.8	Gráfica térmica del equipo NEC DMR-200 C .....	95
Figura 6.9	Gráfica térmica del equipo NEC DMR-200 D .....	96
Figura 6.10	Temperatura vs Tiempo .....	99
Figura 6.11	Respuesta al escalón .....	100
Figura 6.12	Menú de opciones de la interfaz con el usuario .....	102
Figura 6.13	Tabla con los valores de temperatura capturados por el sistema .	102
Figura 6.14	Gráfica 3D de los valores capturados por el sistema .....	103
Figura 6.15	Ayuda del sistema (manual de usuario) .....	104
Figura 6.16	Comando de ayuda Obtener Datos.....	105
Figura 6.17	Comando de ayuda Visualizar datos.....	105
Figura 6.18	Comando de ayuda Graficar .....	106
Figura 6.19	Comando de ayuda Archivo Excel .....	106
Figura 6.20	Comando de ayuda Créditos.....	107
Figura A.4.1	Impreso del nodo principal .....	128
Figura A.4.2	Impreso nodo intermedio .....	129
Figura A.4.3	Impreso nodo extremo final.....	130

## INDICE DE TABLAS

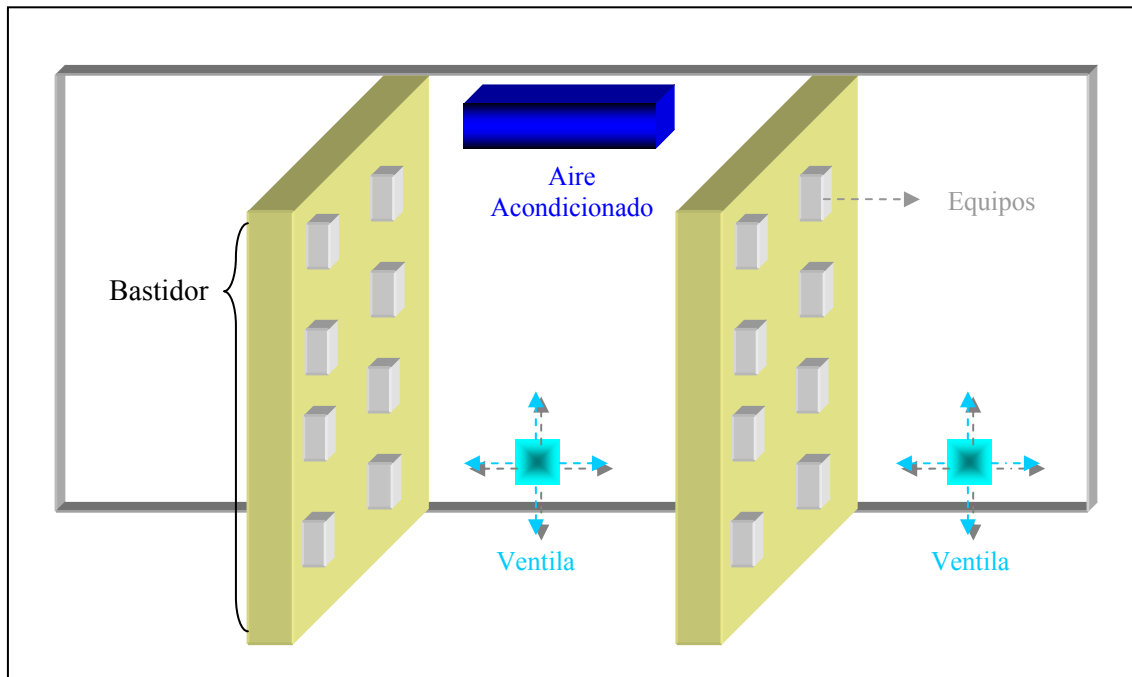
Tabla 5.1	Lista Comparativa entre Microcontroladores .....	47
Tabla 5.2	Lista comparativa entre Sensores de Temperatura.....	51
Tabla 5.3	Tabla de configuración de frecuencia del MCP2120 .....	55
Tabla 5.4	Configuración de las funciones de transmisión y recepción .....	57
Tabla 6.1	Comparación de valores del milímetros vs LM35 .....	88
Tabla 6.2	Valores de distancia de prueba para la recepción de datos entre módulos.....	90
Tabla 6.3	Valores de ángulos de prueba para establecer el alcance .....	91
Tabla 6.4	Valores experimentales del tiempo y temperatura del sistema de enfriamiento.....	98

## Capítulo 1: Introducción

### 1.1 Problema existente e importancia de su solución

El Proceso de Transmisión del ICE cuenta con una sala de equipos de telecomunicaciones, la cual cuenta con equipos de transmisión (radio, fibra óptica y telefonía inalámbrica), enrutadores y otros equipos de diversa naturaleza. Esta variedad de equipos provoca que la temperatura no sea uniforme en todos los puntos de la sala, por lo que existen zonas que presentan altas temperaturas que dan como resultado un sobrecalentamiento de los equipos.

En la Figura 1.1 se muestra un esquema general de cómo se encuentran distribuidos los equipos en la sala y el sistema de enfriamiento (la cantidad dispositivos del esquema solo es para propósitos de visualización).



**Figura 1.1** Esquema general de la sala de equipos de telecomunicaciones

El sistema actual de enfriamiento es insuficiente, debido al aumento de equipos y a la falta de un sistema de control que indique las zonas donde es necesaria una compensación de temperatura. Dicha insuficiencia del sistema está demostrada en un estudio termográfico que se realizó en la sala de telecomunicaciones mediante una cámara térmica, donde se muestran las altas temperaturas en los equipos, que en su mayoría sobrepasan los datos especificados por el fabricante.

Este sistema de enfriamiento actual está conformado por un sistema de aire acondicionado que se encuentra instalado dentro del cuarto y cuenta con ventilas para lograr distribuir de una forma más uniforme el aire para el enfriamiento.

En general los sistemas electrónicos deben de trabajar bajo ciertos parámetros controlados, como por ejemplo la humedad y la temperatura, para que tengan un desempeño óptimo.

Como consecuencias del aumento del procesamiento de los equipos se pueden mencionar el daño de los dispositivos internos y la reducción del ciclo de vida de tales equipos debido a una degradación más acelerada. Esto le representa a la empresa un costo económico tanto en reparaciones, repuestos y cambio de equipos.

Es por ello que es necesario un estudio del comportamiento de la temperatura en dicha sala, para analizar el caso y tomar las medidas necesarias para solucionar el problema, ya sea implementando un sistema de control en el sistema actual de enfriamiento, agregando una mayor cantidad de aires acondicionados en las zonas críticas o reemplazar totalmente el sistema actual por uno que cumpla con los requisitos dados por el estudio que se pretende realizar.

## 1.2 Solución seleccionada

Entre los criterios que se tomaron en cuenta a la hora de seleccionar la solución se pueden mencionar las siguientes:

- Se decidió utilizar un sistema inalámbrico debido a que por experiencias pasadas no se obtuvieron los resultados esperados con sistemas cableados, entre los problemas presentados se pueden mencionar la distribución de los cables, falta de seguridad por alimentación remota, e interfase entre equipo/software.

- Debido a la naturaleza de los equipos de telecomunicaciones, las señales de radiofrecuencia producen interferencia, por lo que el sistema se implementó con una comunicación infrarroja.

- Un factor importante es la escalabilidad del sistema, debido a que el proyecto no abarca toda la sala de telecomunicaciones. Además, se debe tomó en cuenta la posibilidad de que la empresa adquiera nuevos equipos, aumentando de esta manera la cantidad de sectores que se deben monitorear.

- Además se consideró que los dispositivos con los que se implementaron el sistema final se puedan adquirir dentro del país, debido a que el ICE tiene una política estricta respecto a compras en el exterior.

Como solución final se decidió implementar varios módulos distribuidos en los bastidores donde se encuentran los equipos para monitorear el comportamiento térmico de la sala en general. Una vez recolectados los valores de temperatura, éstos son enviados a una computadora que contendrá el software desarrollado para efectos de visualizar y almacenar la información en una base de datos.



## **Capítulo 2: Meta y objetivos**

### **2.1 Meta**

Implementar un sistema capaz de determinar las zonas que puedan presentar un efecto negativo en los equipos que se encuentran en la sala de telecomunicaciones del Proceso de Transmisión para obtener un adecuado funcionamiento de dichos equipos.

### **2.2 Objetivo General**

Desarrollar un sistema de monitoreo de temperatura para la sala de equipos de telecomunicaciones del Proceso de Transmisión, capaz de representar gráficamente el estado de cada zona analizada y cuya interacción con el usuario sea amigable.

### **2.3 Objetivos específicos**

#### **a. Objetivos de hardware**

- Definir el equipo que mejor se adecue al sistema que se desea implementar.
- Realizar pruebas con los sensores de temperatura para definir su correcto funcionamiento.
- Realizar pruebas con los transceivers adquiridos para definir su correcto funcionamiento.
- Determinar mediante pruebas el correcto funcionamiento de los circuitos una vez montados en los circuitos impresos.

### **b. Objetivos de software**

- Diseñar e implementar un protocolo de comunicación entre los diferentes módulos.
- Escribir un programa en ambiente Windows, que cuente con una interfaz gráfica amigable con el usuario, que permita describir mediante gráficos el comportamiento térmico en las zonas donde se distribuirán los sensores.
- Desarrollar un programa capaz de automatizar el proceso de ajustar la temperatura respecto a los valores recolectados por el sistema de monitoreo.

### **c. Objetivos de documentación**

- Realizar un manual de usuario para el uso del software que se encarga de la visualización de la temperatura (este manual se encuentra como una opción de ayuda en la interfaz del usuario).
- Documentar el código fuente que se utilizará tanto para el software de la PC como el del sistema de automatización del proceso de ajuste de temperatura.
- Elaborar un informe final que documente el estudio teórico práctico, los resultados, conclusiones y recomendaciones del proyecto

#### **d. Objetivos de implementación**

- Implementar el prototipo del sistema en protoboards para verificar el funcionamiento correcto del mismo.
- Realizar los circuitos impresos de los diferentes módulos.
- Crear una interfaz amigable de usuario para el software elaborado
- Presentar el proyecto al Proceso de Transmisión y realizar la defensa del mismo a la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

## **Capítulo 3: Marco Teórico**

### **3.1 Sensores de Temperatura<sup>1</sup>**

En muchas aplicaciones electrónicas es común la medición de la temperatura para evaluar sus efectos en ciertos procesos. Debido a ello se ha desarrollado una gran cantidad de dispositivos sensores y transductores, los cuales van desde la simple unión bimetálica de los termostatos, hasta dispositivos semiconductores más complejos.

Las principales clases de sensores de temperatura se basan en:

- a) la expansión térmica
- b) el cambio de resistencia
- c) propiedades termoeléctricas de diversas sustancias en función de la temperatura.

Los termómetros de mercurio y alcohol son ejemplos comunes de sensores de expansión térmica. Sin embargo, su valor es limitado en redes de monitoreo in situ o remotas debido a que no tienen la capacidad de registrar datos automatizados.

Se pueden mencionar los siguientes tipos de sensores de temperatura:

#### **3.1.1 Termopares**

Estos dispositivos utilizan la tensión generada en la unión de dos metales en contacto térmico, debido a sus distintos comportamientos eléctricos.

Entre sus ventajas está el bajo costo y su robustez, además de que son estables a lo largo del tiempo. Gracias a su reducido tamaño, tienen una respuesta a los cambios temperatura bastante rápida.

Como se mencionó antes, está constituido de dos metales, debido a que el número de electrones libres en un metal depende de la temperatura y de la

---

<sup>1</sup> Tomado de [8], [9] y [10].

composición del metal, dos metales de desigual isoterma, dan una diferencia de potencial que es una función repetible de la temperatura. Es decir, básicamente lo que tiene es un dispositivo de medida diferencial, donde la temperatura de una de las uniones funciona como referencia y la así la otra temperatura se obtiene de la diferencia de voltaje.

El material con que están fabricados estos dispositivos es especialmente seleccionado, el cual se caracteriza por su relación de voltaje/temperatura, que normalmente es la del punto de agua/hielo de 0°C.

Los termopares trabajan con bajos niveles de tensión y además con baja impedancia, por lo que el acondicionamiento de estas señales es imprescindible. Además que no todos los termopares tienen un comportamiento totalmente lineal, pero el hecho que sea predecible y repetible (el comportamiento), permite que la señal pueda ser compensada analógicamente y/o digitalmente.

La instalación de estos sensores, exige de ciertos cuidados para evitar corrientes de inducción de fuentes cercanas de corriente alterna que podrían ocasionar errores en la medición. Los termopares también son susceptibles al voltaje espurio causado por la humedad.

### **3.1.2 Resistivos**

#### **3.1.2.1 Detector de resistencia por temperatura (RTD)**

También conocido como PT100, las cuales se basan en la relación de la resistividad de un conductor con la temperatura.

Los RTD comúnmente están compuestos de una resistencia de platino (se les denomina PRTD). Por lo general su resistencia se encuentra entre los 20Ω y 20KΩ. Su mayor ventaja es que tienen un comportamiento lineal dentro del rango de temperatura entre -200°C y 850°C.

### 3.1.2.2 Termistores

Mucho más económicos que las RTD, los cuales aunque no tienen un comportamiento lineal, pueden tener un coeficiente de temperatura positivo (PTC) o negativo (NTC).

Estos dispositivos semiconductores son muy utilizados debido a su bajo costo y sensibilidad. Sin embargo la dependencia con la temperatura no es simple, la cual se puede expresar mediante la siguiente ecuación matemática:

$$R(T) = R(T_0) \cdot \exp\left(\beta \cdot \left(\frac{1}{T} - \frac{1}{T_0}\right)\right) = R_0 \cdot \exp\left(\frac{E_g}{k} \cdot \left(\frac{1}{T} - \frac{1}{T_0}\right)\right) \quad (3.1)$$

Donde T y T<sub>0</sub>: son las temperaturas absolutas

R(T<sub>0</sub>) = R<sub>0</sub>: la resistencia a T<sub>0</sub>

β y E<sub>g</sub>: constantes a determinar

k: constante de Boltzmann

### 3.1.3 Semiconductores

Quizás el efecto más conocido es la variación de la diferencia de potencial en una juntura con la temperatura. Para un semiconductor de silicio, una juntura sencilla polarizada directamente exhibe una diferencia de potencial que ronda los 0.6 a 0.7 V. Este valor cambia con la temperatura a razón de unos 8mV por grado centígrado, lo que permite realizar un fácil muestreo de la temperatura.

Los semiconductores normalmente no se utilizan en el control y medición de temperaturas de procesos industriales debido a sus rangos limitados y/o limitaciones de amplitud de alcance, su falta de comportamiento lineal, características de deriva e inexactitudes.

### 3.1.4 Sensor de temperatura LM35 <sup>2</sup>

La salida de este sensor de precisión tiene un voltaje que es proporcional a la temperatura en la escala Celsius ( $10\text{mV}/^{\circ}\text{C}$ ). Una de las ventajas del LM35 sobre los sensores de temperaturas lineales calibrados en grados Kelvin, es que no ocupa utilizar un voltaje alto para poder obtener una escala en centígrados conveniente. Además no requiere de calibración externa para obtener una precisión de  $\pm 0.25^{\circ}\text{C}$  a temperatura ambiente y  $\pm 0.75^{\circ}\text{C}$  a un rango de  $-55$  hasta  $+150^{\circ}\text{C}$ .

Este sensor puede ser utilizado con una sola fuente de alimentación o con una bipolar. Solo le demanda a la fuente unos  $60\mu\text{A}$  por lo que tiene un calentamiento del mismo muy bajo, menos de  $0.1^{\circ}\text{C}$  al aire libre. Además tiene un rango de operación entre los 4 y los 30 voltios.

Debido a que este dispositivo cuenta con una baja impedancia ( $0.1$  ohmios por carga de  $1\text{mA}$ ), salida lineal y calibración precisa inherente, hace que la interfaz para la lectura o el control del mismo sea muy sencilla.

---

<sup>2</sup> Tomado de hojas de datos del sensor LM35 marca National Semiconductor

### **3.2 Redes inalámbricas<sup>3</sup>**

Se conoce como red inalámbrica de datos aquel sistema cuyos equipos de comunicación se encuentran interconectados sin el uso de algún tipo de cable como medio de conexión.

Las redes inalámbricas le ofrecen al diseñador una gran cantidad de ventajas sobre las redes de datos cableadas, por lo que su uso es más común cada día. Entre dichas ventajas se pueden mencionar las siguientes:

#### **- Escalabilidad**

Los sistemas inalámbricos pueden ser configurados en una variedad de topologías para satisfacer las necesidades de las instalaciones y aplicaciones específicas. Las configuraciones son fáciles de cambiar, al igual que la incorporación de nuevos dispositivos a la red.

#### **- Movilidad**

Debido a su independencia de una conexión física, las redes inalámbricas permiten que los dispositivos sean trasladados de un lugar a otro sin perder conectividad siempre y cuando se mantengan en el rango de cobertura.

#### **- Reducción de costos**

La reducción de costos se debe a la innecesidad de realizar cambios en la infraestructura, a una menor cantidad de personal para la instalación y además no es necesario un mantenimiento regular (carencia de conexiones físicas).

#### **- Flexibilidad**

La instalación de este tipo de redes es rápida, sencilla y elimina la necesidad de tirar cables a través de paredes y techos. La tecnología inalámbrica permite comunicar dispositivos en lugares donde una red cableada no lo lograría.

---

<sup>3</sup> Tomado de [7].



La transmisión inalámbrica no utiliza medios guiados, su principal medio es el aire. Su funcionamiento se basa en la irradiación de energía electromagnética por medio de una antena, para luego ser recibida por otra antena.

Existen dos configuraciones para la transmisión/recepción de la energía electromagnética:

**- Direccional**

La energía es transmitida en una sola dirección, por lo que el dispositivo transmisor debe de estar alineado con el receptor. A mayor frecuencia de transmisión, mayor será la direccionalidad de la señal.

**- Omnidireccional**

Esta configuración consiste en enviar la energía en múltiples direcciones, por lo que la señal será captada por aquellas antenas que se encuentren dentro del área de recepción.

Las señales microondas se caracterizan por su alta frecuencia, por lo que generalmente son utilizadas para enlaces punto a punto. Mientras que las ondas de radio (bajas frecuencias) son utilizadas para configuraciones omnidireccionales. Las señales infrarrojas son utilizadas para cortas distancias, como por ejemplo, un cuarto.

### 3.3 Comunicación Infrarroja<sup>4</sup>

Las comunicaciones infrarrojas, como su nombre indica, están basadas en el principio de la luz infrarroja. Esta luz infrarroja es una radiación electromagnética cuya frecuencia permite que sea visible al ojo humano, La luz visible viaja en ases de luz que van desde los 400 ángstrom (violeta oscuro), a 700 ángstrom (rojo oscuro). Las frecuencias del infrarrojo son de 700 a 1,000 ángstrom. Conforme a los estándares del IrDA (Infrared Developers Association) la mayoría de las computadoras personales y equipo de comunicaciones se mantienen entre los 850 y 900 ángstrom.

Este tipo de comunicación también utiliza pulsos para transmitir datos. Estos pulsos varían con respecto a los digitales en que mientras los anteriores son constantes durante un ciclo de reloj, los pulsos IrDA duran sólo una fracción del ciclo básico de reloj (celda estándar de bit). Estos pulsos son distribuidos ampliamente entre ellos, lo que los hace fáciles de recibir y distinguir en el receptor IrDA

Cuando un puerto IrDA funciona a 115,000 bits por segundo cada pulso emitido debe de ser de al menos 1.41 microsegundos de duración. Típicamente uno de estos pulsos dura únicamente 3/16 de la longitud de una celda estándar de bit. Conforme se aumenta la velocidad de transmisión los pulsos se acortan en tiempo y longitud. Este rango de pulsos va de los 295.2 nanosegundos para los 576 Kbits por segundo a 115 nanosegundos para una tasa de transferencia de 4 megabits por segundo.

En la comunicación infrarroja también existe la modulación. Dependiendo de la velocidad de conexión IrDA se utiliza dos formas de modulación.

---

<sup>4</sup> Tomado de [6]

### **- RZI (Return-to-Zero Invert)**

Esta modulación es utilizada para velocidades de transmisión de datos menores a 4.0 Mbps. Con este tipo de modulación el emisor envía un pulso para indicar un cero binario, y cuando no envía un pulso, indica un uno binario.

### **- PPM (Pulse Position Modulation)**

Esta modulación es utilizada para velocidad de datos mayores a 4.0 Mbps. También se le conoce como 4PPM, debido a las cuatro posiciones de su pulso. La longitud y posición de un pulso es determinado por un período de reloj. Estos períodos son determinados por la duración del símbolo y son divididos en 4 segmentos iguales. Estos segmentos son referidos como Chips (Pedazos).

Los datos transmitidos por un dispositivo IrDA son transmitidos en un formato de 8 bits, conforme al estándar de la IRDA, 8 bits de datos, bit de paridad, y bit de paro para un total de 10 bits por carácter.

Los dispositivos de transmisión y recepción infrarrojos deben estar alineados o ubicarse en línea, para así evitar posibles problemas de comunicación por reflexión de rayo en las superficies. Este tipo de transmisión está exento de problemas de seguridad debido a que los rayos infrarrojos no pueden atravesar objetos. Además no existe una regulación para su utilización, a diferencia de las microondas y la radiofrecuencia.

### **3.4 Clasificación de redes por transmisión de datos**

#### **- Redes simplex (transmisión simple)**

Son aquellas redes en que los datos solo pueden viajar en un solo sentido.

#### **- Redes Half-Duplex**

Son aquellas en las que los datos viajan en ambos sentidos pero no simultáneamente, es decir, en el canal sólo puede haber transferencia de datos en un sentido a la vez.

#### **- Redes Full-Duplex**

Son aquellas en que los datos pueden viajar en ambos sentidos a la vez.

### **3.5 Variables Referenciadas (Casting o transformaciones de tipos)<sup>5</sup>**

El casting es un procedimiento para transformar una variable primitiva de un tipo a otro, o transformar un objeto de una clase a otra clase siempre y cuando haya una relación de herencia entre ambas.

Dentro del casting de variables primitivas se distinguen dos clases:

#### **3.5.1 Implícito**

No se necesita escribir código para que se lleve a cabo. Ocurre cuando se realiza una conversión ancha (widening casting), es decir, cuando se coloca un valor pequeño en un contenedor grande.

---

<sup>5</sup> Tomado de [1].

### **3.5.2 Explícito**

Sí es necesario escribir código. Ocurre cuando se realiza una conversión estrecha (narrowing casting), es decir, cuando se coloca un valor grande en un contenedor pequeño. Son susceptibles de pérdida de datos.

## **3.6 Universal Asynchronous Receiver/Transmitter (UART)<sup>6</sup>**

El controlador UART es el componente clave para lograr establecer la comunicación serial entre una computadora y otro sistema. Este controlador toma los bytes de los datos y los envía como bits individuales de un modo serial. En el sistema destino, un segundo controlador UART recolecta los bits y los vuelve a ensamblar en bytes completos.

Existen dos tipos principales de transmisión serial: Asíncrona (UART) y Síncrona (USART).

### **3.6.1 Transmisión serial síncrona**

Para este tipo de transmisión es necesario que tanto el dispositivo que envía como el que recibe los datos, compartan una misma señal de reloj. También se puede establecer este tipo de transmisión si el dispositivo fuente envía una señal indicándole al dispositivo destino cuando puede leer el siguiente bit de datos. En casi todas las formas de la transmisión síncrona, cuando no hay datos disponibles en algún momento determinado, algún dato debe ser enviado de manera que siempre se esté enviando información. Este tipo de comunicación es más eficiente debido a que los datos solo son transmitidos entre transmisor y receptor, pero por lo general tienen un mayor costo económico debido a que

---

<sup>6</sup> Tomado de [11].

ocupan mayor cableado y dispositivos para que ambos compartan la señal de reloj.

Las computadoras no cuentan con este tipo de transmisión.

### **3.6.2 Transmisión serial asincrónica**

La transmisión asincrónica puede establecerse sin necesidad que el dispositivo que envía los datos mande un señal de reloj. En lugar de ello, el transmisor y el receptor se ponen de acuerdo de antemano sobre los parámetros de tiempo y además bits especiales son añadidos a cada palabra, los cuales se encargan de la sincronización de las unidades enviadas y recibidas.

Cuando una palabra es dada a la UART para entablar una transmisión asincrónica, un bit llamado "Start Bit" es añadido al principio de cada palabra a transmitir. Este Start Bit es usado para alertar al receptor que la palabra está a punto de enviarse, lo cual fuerza al receptor a sincronizar la señal de reloj con la del transmisor. Estos dos relojes deben de tener tal precisión, de manera que la frecuencia no se desvíe en más de un 10% durante la transmisión de los bits restantes de la palabra.

Después del Start Bit, los bits de la palabra son enviados, siendo el bit menos significativo el primero en enviarse. Cada bit es transmitido con exactamente la misma cantidad de tiempo entre uno y otro. El receptor leerá el dato cuando se encuentre aproximadamente a la mitad del período asignado a cada bit para determinar si es un 1 o un 0 lógico.

Por último se enviará al menos un bit llamado "Stop Bit" encargado de informarle al receptor que la palabra ha terminado de transmitirse.

### 3.7 Estándar RS232<sup>7</sup>

El puerto serie también es conocido por el estándar que lo gobierna, el RS232, el cual fue creado con el propósito de ser utilizado como interfase entre los equipos terminales de datos (Data Terminal Equipment, DTE), y el equipo de comunicación de datos (Data Communications Equipment, DCE) empleando intercambio serial de datos binarios.

El estándar RS232 es un protocolo para la conexión entre un DET y un DCE que define:

- Tipo de conector a emplear
- Características eléctricas
- Niveles de tensión
- Longitudes máximas a distintas velocidades

Sus niveles lógicos están definidos por una señalización eléctrica bipolar:

- 0 lógico: de +3 hasta +15V
- 1 lógico: de -3 hasta -15V

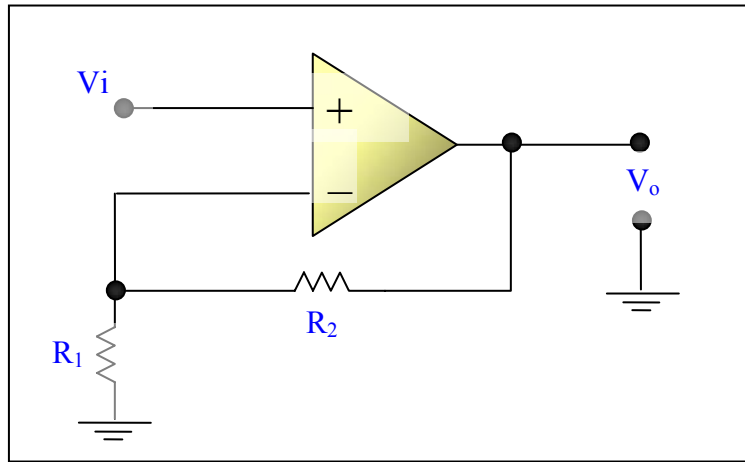
Las velocidades de transmisión que puede soportar este estándar van desde los 0bps hasta los 20Kbps. Con respecto a las distancias máximas se recomienda que no sean superiores a 15 metros.

---

<sup>7</sup> Tomado de [5].

### 3.8 Amplificador operacional en configuración no inversora<sup>8</sup>

La configuración no inversora del amplificador operacional se muestra en la Figura 3.1.



**Figura 3.1** Esquema de un amplificador no inversor

En este circuito, la tensión  $V_i$  se aplica a la entrada no inversora (+), y una fracción de la señal de salida,  $V_o$ , se aplica a la entrada inversora (-) a través del divisor de tensión  $R_1 - R_2$ . Puesto que, no fluye corriente de entrada en ninguna terminal de entrada, y la diferencia de voltaje entre la entrada inversora con la no inversora es igual a 0 ( $V_d = 0$ ), la tensión en  $R_1$  será igual a  $V_i$ .

Por lo que tendremos que:

$$V_i = I \cdot R_1 \quad (3.2)$$

y como:

$$V_o = I \cdot (R_1 + R_2) \quad (3.3)$$

---

<sup>8</sup> Tomado de [2].



Sustituyendo la ecuación 3.2 en 3.3:

$$V_o = \frac{V_i}{R_1} \cdot (R_1 + R_2) \quad (3.4)$$

La ganancia la podemos expresar de la siguiente manera:

$$\frac{V_o}{V_i} = \frac{R_1 + R_2}{R_1} \Rightarrow G = \frac{R_2}{R_1} + 1 \quad (3.5)$$

que corresponde a la ecuación característica de ganancia para el amplificador no inversor ideal.

También se pueden deducir propiedades adicionales para esta configuración. El límite inferior de ganancia se produce cuando  $R_2 = 0$ , lo que da lugar a una ganancia unidad.

### 3.9 Análisis de la repuesta en el tiempo<sup>9</sup>

#### 3.9.1 Respuesta en el tiempo de sistemas en tiempo continuo

Debido a que la mayoría de los sistemas de control utilizan el tiempo como variable independiente, es usual obtener la respuesta en el tiempo del mismo. El análisis se realiza aplicando una señal de entrada de referencia al sistema, y se evalúa el desempeño de este último con el comportamiento de la respuesta en el dominio del tiempo.

La respuesta en el tiempo se separa, por lo general, en dos secciones: la respuesta transitoria y la respuesta en estado estable. Describiendo la respuesta por medio de una ecuación, obtenemos:

$$y(t) = y_i(t) + y_{ss}(t) \quad (3.6)$$

siendo  $y_i(t)$  la respuesta transitoria y  $y_{ss}(t)$  la respuesta en estado estable.

---

<sup>9</sup> Tomado de [3] y [4]

En sistemas de control. La respuesta transitoria se conoce como la parte de la respuesta en el tiempo que tiende a cero cuando el tiempo se hace muy grande. Mientras que la respuesta en estado estable es la parte de la respuesta final que permanece después de que la transitoria ha desaparecido.

### 3.9.2 Señal de prueba

Para propósitos de análisis y diseño, por lo general se utilizan diferentes tipos básicos de entradas de prueba para evaluar el desempeño del sistema. Para el sistema en estudio se utilizará de entrada la función escalón, la cual representa un cambio instantáneo en la entrada de referencia. La representación matemática se describe a continuación:

$$r(t) = \begin{cases} R & t \geq 0 \\ 0 & t < 0 \end{cases} \quad (3.7)$$

donde R es una constante real.

Esta tipo de función es muy útil, debido a que su brinco instantáneo revela el comportamiento del sistema (que tan rápido o que tan lento responde) ante entradas con cambios abruptos.

### 3.9.3 Parámetros de la respuesta ante un escalón

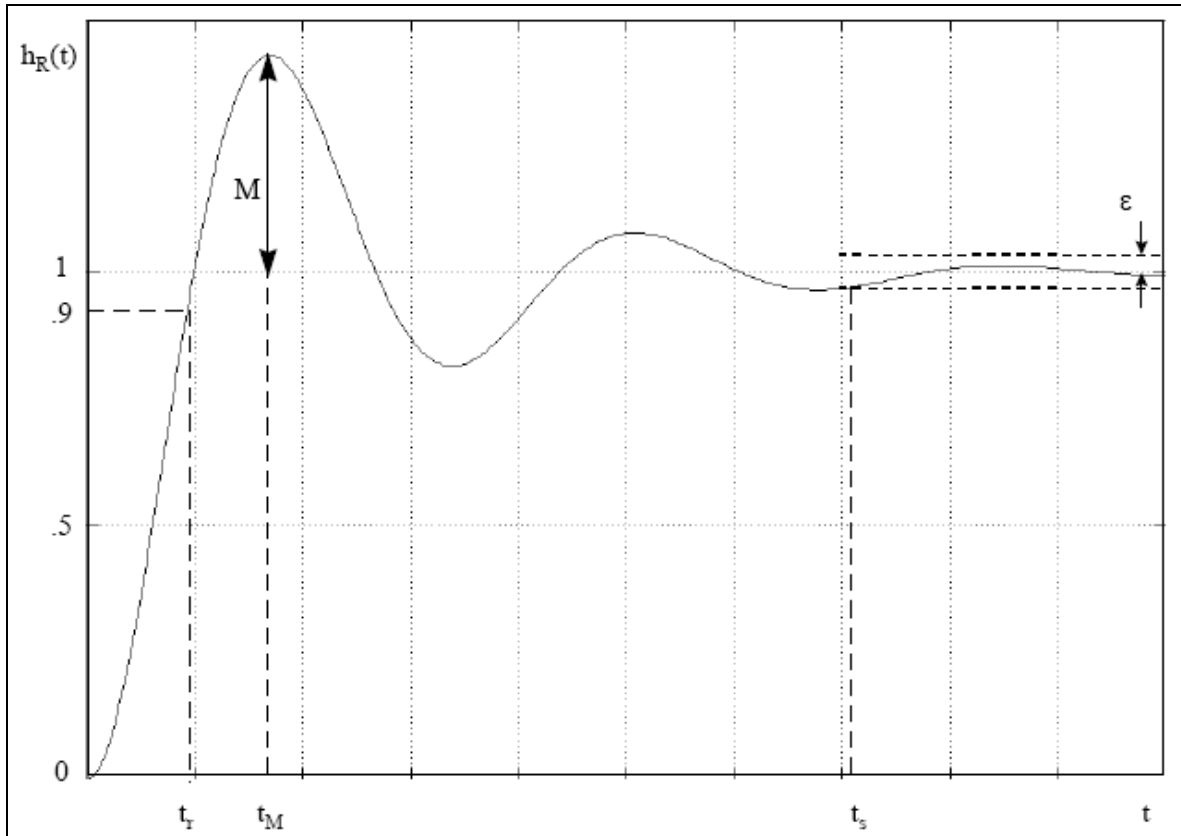


Figura 3.2 Respuesta ante un escalón<sup>10</sup>

<sup>10</sup> Tomado de [3]

### - Error de estado estacionario ( $\epsilon$ )

El resultado final que se desea de los sistemas de control, es que la respuesta de salida del sistema siga una señal de referencia específica en forma exacta en el estado estable. La diferencia entre la señal de salida y la de referencia en el estado estable se conoce como error de estado estable o estacionario. En los sistemas reales, debido a gran cantidad de factores, es muy difícil que ambas señales concuerden exactamente, por lo que el error de estado estacionario es casi inevitable. Por ello se busca minimizarlo al máximo.

### - Sobrepasso máximo (M)

Asúmase que  $y(t)$  es la respuesta al escalón unitario. También que  $y_{\max}$  enuncia el valor máximo de  $y(t)$ , y  $y_{ss}$  es el valor en estado estable de  $y(t)$ . Además  $y_{\max} \geq y_{ss}$ . Por lo tanto el sobrepasso máximo se define como:

$$M = y_{\max} - y_{ss} \quad (3.8)$$

### - Tiempo de levantamiento ( $t_r$ )

Se define como el tiempo requerido para que la respuesta al escalón se eleve del 10 al 90% de su valor final.

### - Tiempo de asentamiento ( $t_s$ )

Se define como el tiempo requerido para que la respuesta al escalón disminuya y permanezca dentro del porcentaje específico de su valor final. Para efectos del proyecto se utilizó un valor del 2%.

## **Capítulo 4: Procedimiento metodológico**

### **4.1 Reconocimiento y definición del problema**

La propuesta de proyecto fue presentada por el Ing. Julio Stradi en las instalaciones del departamento de Transmisión del ICE en San Pedro, donde se expuso la problemática que presenta la sala de equipos de telecomunicaciones. Además se discutieron las restricciones del proyecto, las cuales fueron mencionadas en apartado 2.1 del presente documento.

Conforme se fue aumentando la cantidad de equipos en la sala de estudio, también se identificó un aumento no uniforme en la temperatura de dicha sala (ver sección 6.3 sobre el estudio termográfico), dando como resultado el problema de las averías en los equipos de telecomunicaciones. Esto demostró que el sistema de enfriamiento actual no cuenta con el sistema necesario para identificar las zonas con temperaturas excesivas.

## **4.2 Obtención y análisis de la información**

La información preliminar se obtuvo mediante entrevistas y por medio de consultas realizadas con el Ing. Julio Stradi, quien explicó los detalles del proyecto considerando también las experiencias pasadas de proyectos similares que no lograron cumplir las expectativas.

Sobre la información obtenida se analizó los factores por los cuales las soluciones que se intentaron implementar no lograron el desempeño esperado, y además la factibilidad de posibles soluciones.

Además se definieron los siguientes criterios a tomar en cuenta a la hora de diseñar la solución:

- Considerar la opción más económica.
- Debido a la política de compras de la empresa, todos los componentes deben de adquirirse dentro del territorio nacional.
- La solución no debe de afectar el funcionamiento de los equipos que se encuentran en la sala de telecomunicaciones.

### **4.3 Evaluación de alternativas y síntesis de una solución**

#### **4.3.1 Investigación de experiencias anteriores sobre proyectos realizados**

En el año 2000 se realizó un proyecto semejante elaborado como proyecto de graduación por el Ing. Alonso Piedra Gutiérrez, el cual se denominó “Diseño e implementación de un sistema de monitoreo de temperaturas para la sala de transmisión del ICE San Pedro”. El Ing. Piedra fue asesorado tanto de parte del Instituto Tecnológico de Costa Rica como del ICE por el Ing. Julio Stradi.

Dicho proyecto no logró alcanzar las expectativas debido a las siguientes razones:

- Los módulos que se diseñaron contaban con una conexión alamburada, lo que presenta un problema a la hora de colocarlos debido a la cantidad de cables que se encuentran actualmente y la carencia de fuentes de poder para la alimentación de los módulos
- El sistema no era escalable debido a la estructura física del prototipo realizado.
- El sistema presentaba problemas de interfase con el software.

#### **4.3.2 Recomendación de expertos en la materia**

Entre las recomendaciones que presentó el Ing. Julio Stradi se pueden mencionar las siguientes:

- El sistema a implementar debe ser inalámbrico para evitar el problema de saturación de cables en la sala.
- Utilizar baterías como fuente de energía de los módulos debido a la carencia de fuentes de alimentación en la sala.
- Utilizar mecanismo de detección de errores para facilitar el arreglo de los mismos.

- El protocolo de comunicación puede ser implementado de manera que se presente una señal inicial indicando que se realizó una petición de datos y una vez que dicha señal llega hasta el último módulo, se comenzarán a recolectar todos los datos de temperatura hasta llegar al módulo principal.

Para realizar el modelado matemático de la temperatura de la sala en estudio, se recurrió a la ayuda del Ing. Eduardo Interiano, el cual recomendó realizar un modelado experimental que consistiera en introducirle un cambio de temperatura (escalón) al sistema actual de enfriamiento y así recolectar los datos de temperatura con su respectivo tiempo para luego ser procesados por la herramienta Matlab y de esta manera obtener un modelado que contemplara los diversos factores que afectan la sala en estudio.

#### **4.3.2 Evaluación de alternativas**

Entre las posibilidades se consideró sustituir el sistema de enfriamiento actual para lograr la temperatura adecuada. Dicha posibilidad presentaba otro problema, el cual es que no se sabía con exactitud qué tipo de sistema de enfriamiento era necesario para compensar la disipación de calor de los equipos y exactamente en cuales zonas de la sala, por lo que habría que sobredimensionar el equipo de enfriamiento que debería de adquirirse, lo que le representaría a la empresa un mayor costo económico. Además no se consideraba la posibilidad de una mayor cantidad de equipos en la sala, lo cual podría presentar el mismo problema en un futuro.

Se consideró como una alternativa más viable colocar módulos con sensores de temperatura en los bastidores que contienen los equipos de la sala para obtener el comportamiento de la disipación térmica. Los valores de temperatura recolectados en toda la sala son transmitidos a través de los módulos hasta la PC con el propósito de lograr la visualización de la temperatura en los



equipos de telecomunicaciones por medio de gráficos para el posterior análisis de los mismos.

La propuesta anterior puede realizarse de diversas maneras, específicamente en la forma de implementar los medios de comunicación entre los sensores. Entre ellos se puede mencionar la comunicación por radiofrecuencia (RF), infrarroja (IR) o por una solución cableada.

Como se mencionó en el apartado 1.2 la comunicación por radiofrecuencia no es considerada como opción, aunque es cierto que se puede escoger alguna frecuencia que no sea utilizada por los equipos que se encuentran en la sala, existe la posibilidad de que en algún momento se adquieran equipos que sí sean sensibles a la señal escogida, además de que los transceivers que se consiguen en el mercado trabajan a frecuencias específicas, eliminando la posibilidad de escoger una frecuencia que se adecue al sistema que se desea diseñar.

Por otro lado una comunicación cableada entre sensores no es la mejor solución debido a que por experiencias pasadas no se obtuvieron los resultados esperados, entre los problemas presentados se pueden mencionar la distribución de los cables, falta de seguridad por alimentación remota, e interfase entre equipo/software.

Se consideró como mejor opción trabajar la comunicación entre sensores por un medio infrarrojo, debido a que no afecta los equipos que se encuentran en la sala, y la distancia entre sensores no es tan amplia para que se produzcan problemas en la transmisión y recepción de datos.

Sobre el software que se encarga de la automatización del sistema de enfriamiento, se realizó un análisis con los datos obtenidos del sistema de monitoreo para poder plantear una solución viable.

## **4.4 Metodología**

### **4.4.1 Investigación y desarrollo del comportamiento de la temperatura en la sala de estudio**

Utilizando como base un estudio termográfico que había sido realizado en la sala en estudio se obtuvieron los datos sobre la temperatura de operación actual de cada uno de los equipos que se encuentran en la sala de telecomunicaciones (ver sección 6.3)

Además se desarrolló el modelado matemático del sistema en estudio con el cual se analizó el comportamiento de la temperatura en el mismo.

### **4.4.2 Investigación de los componentes necesarios para el desarrollo del proyecto**

Mediante investigación se obtuvo la información necesaria para la escogencia de los componentes más relevantes del proyecto (ver sección 5.1.4). Entre ellos están los sensores de temperatura los cuales cuentan con una lectura adecuada, los transceivers (transmisor/receptor) infrarrojos (IR) que lograron un buen desempeño bajo las condiciones de trabajo y el microcontrolador que puede soportar la cantidad de sensores que se pretenden instalar.

#### **4.4.3 Prueba de los componentes adquiridos**

Se llevaron a cabo pruebas de laboratorios para confirmar el funcionamiento correcto tanto de los sensores de temperatura, como de los transceivers infrarrojos (IR) adquiridos (ver sección 6.1 sobre prueba de componentes).

#### **4.4.4 Desarrollo del software encargado de la adquisición de datos**

Se realizó un protocolo de comunicación bidireccional entre módulos (a través de los transceivers) para lograr recolectar y procesar toda la información necesaria, para que posteriormente esta llegue a la PC para efectos de visualización y almacenaje.

Para la implementación de cada una de las rutinas fue necesario una serie de iteraciones y pruebas de laboratorio, para verificar la validez de los datos obtenidos por el microcontrolador. Además se consideraron todos los posibles casos de error mediante la inducción manual de los mismos en el código enviado y así observar la respuesta del sistema ante tales errores.

#### **4.4.5 Desarrollo del software encargado del despliegue de datos en la PC**

Se realizó un software en ambiente windows con una interfaz amigable con el usuario, capaz de visualizar por medio de gráficos la temperatura a través del tiempo y el comportamiento térmico en las diversas secciones donde se encuentren los equipos de telecomunicaciones. Además este programa almacena los valores de temperatura de todos los módulos para efectos de análisis de comportamiento.

#### **4.4.6 Medios de difusión para dar a conocer los resultados**

Mediante la opción de Ayuda de la herramienta desarrollada para la interfaz del usuario, este último puede buscar la información necesaria para poder manejar adecuadamente dicha herramienta (manual de usuario).

Además del presente documento donde se presentan todos los resultados, análisis, conclusiones y recomendaciones del proyecto.

#### **4.5 Reevaluación y rediseño**

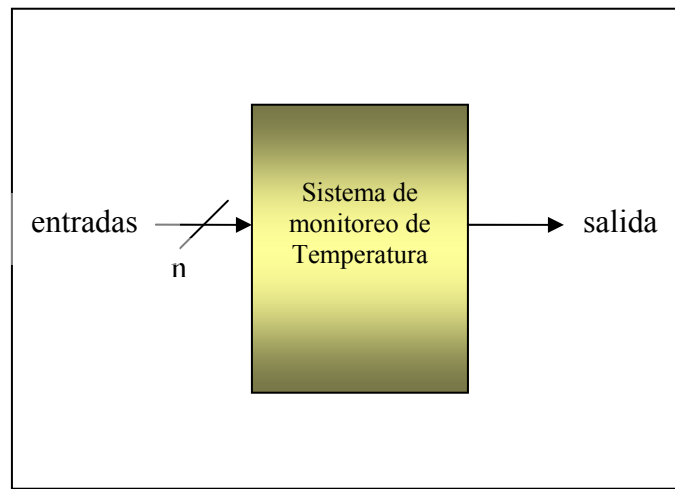
Mediante los resultados obtenidos por el monitoreo del comportamiento térmico se podrá analizar la posibilidad de cambiar el sistema de enfriamiento actual (con el criterio necesario) o se podrá implementar un sistema de control a partir del trabajo realizado.

El sistema final tendrá la posibilidad de aumentar la cantidad de sensores cuando sea necesario (sistema modular escalable), lo que evitaría tener que volver a realizar un proyecto semejante si se aumentara la cantidad de equipos en la sala de telecomunicaciones.

## Capítulo 5: Descripción detallada de la solución (Explicación del diseño)

### 5.1 Descripción del hardware

#### 5.1.1 Diagrama de primer nivel



**Figura 5.1.** Diagrama de primer nivel

Como se mencionó en el apartado 1.1 (Problema existente e importancia de su solución) la problemática de la degradación y fallos que se presentan en los equipos de telecomunicaciones se debe a una distribución no uniforme de la temperatura en la sala en estudio, por lo tanto se diseñó un sistema capaz de monitorear el comportamiento térmico de dicha sala.

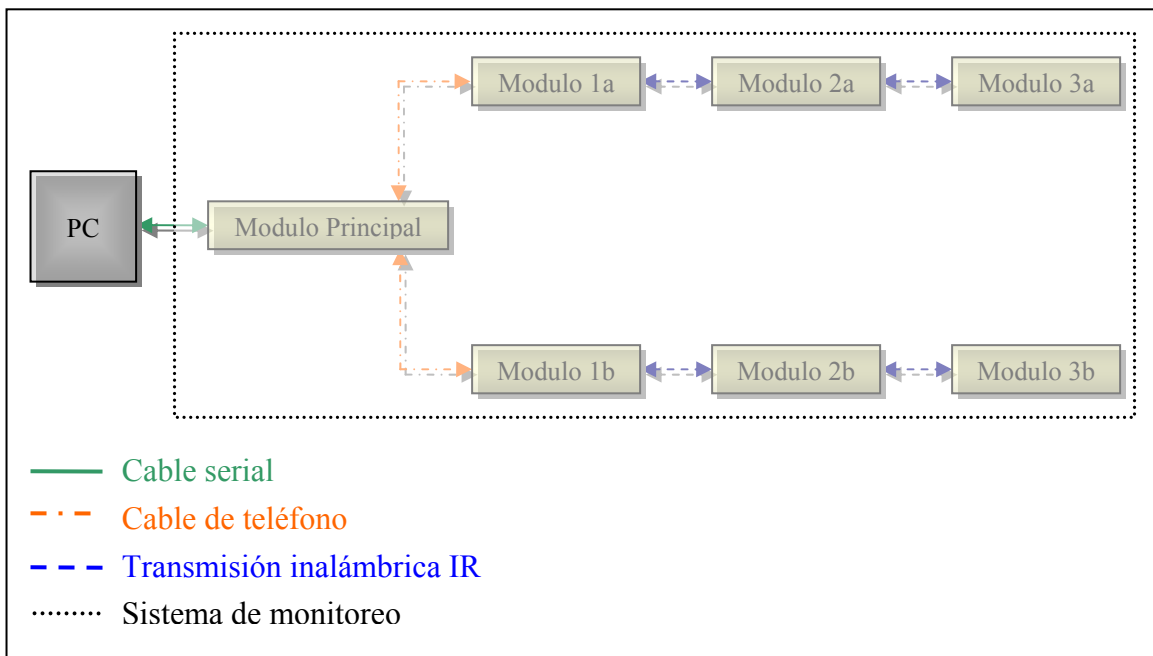
En la Figura 5.1 se observa un esquema general del sistema que se implementó. Las entradas del sistema corresponden a los valores de temperatura que se pretenden monitorear, en este caso dichos valores se presentan con una cantidad indefinida  $n$  debido a la escalabilidad del sistema (no existe un límite de módulos que se pueden implementar para la recolección de datos). Por otro lado, la salida corresponde a la recolección de todos los valores de temperatura que son enviados a una PC que contiene un software capaz de visualizar y almacenar dichos datos.

### 5.1.2 Diagrama de segundo nivel

Para lograr un muestreo más preciso de la temperatura, se decidió por diseñar un sistema que tuviera ubicados los módulos de recolección de temperatura a lo largo del bastidor y a dos diferentes alturas del mismo, una inferior y otra superior.

En la Figura 5.2 se observa un esquema general de la ubicación física de los diferentes módulos que conforman el sistema de monitoreo. El módulo principal encargado de establecer la comunicación con los diferentes módulos se encuentra conectado a través del puerto serie a la PC y mediante cable telefónico al primer módulo, tanto del nivel inferior (b) como del nivel superior (a) del bastidor. Los demás módulos se comunican mediante enlaces inalámbricos.

La cantidad de módulos en la fila superior (a) y la fila inferior (b) depende de la cantidad de equipos que se encuentren en el bastidor.



**Figura 5.2** Esquema de ubicación física de los dispositivos

### 5.1.2.1 Módulo Principal

Encargado de la comunicación entre el sistema y la computadora y la distribución los de datos, está compuesto básicamente por los siguientes bloques (ver Figura 5.3):

- **Bloque de alimentación:** encargado de administrar la energía necesaria a los dispositivos electrónicos.
- **Bloque de interfaz RS232:** tiene como función establecer la comunicación con la PC utilizando el protocolo RS232.
- **Bloque de procesamiento de datos:** este bloque se encarga de identificar las funciones que se deben de realizar, acomodar los datos según el protocolo diseñado y descartar la información que no le corresponde.
- **Bloque de comunicación:** el bloque de comunicación es el medio físico utilizado para conectar el módulo principal con los módulos intermedios, tanto del nivel inferior como del superior.

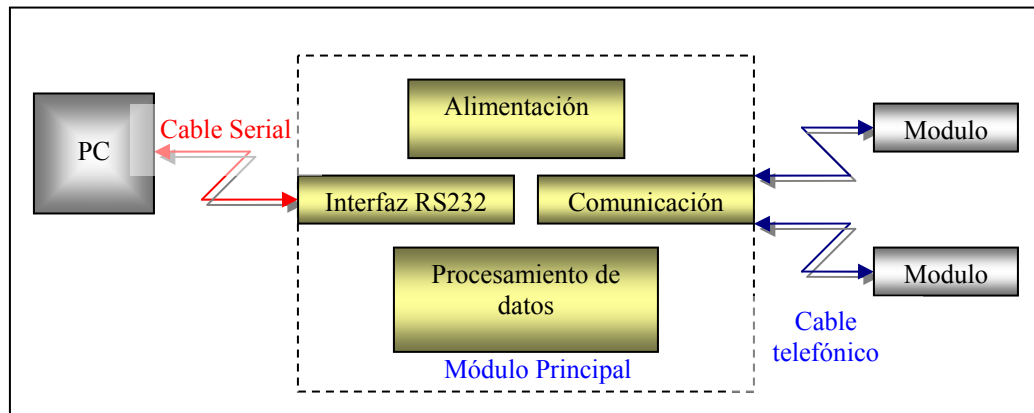


Figura 5.3. Esquema general del módulo principal

Este módulo tiene como entradas/salidas los enlaces de comunicación con la PC y los módulos intermedios. Las entradas corresponden a la información con la solicitud de datos proveniente de la PC y a los datos de temperatura por parte de los módulos intermedios. Respecto a la salida hacia la PC se tiene la cadena de información que se le envía con todos los valores de temperatura y como salida hacia los módulos intermedios se tiene la solicitud de recolección de datos.

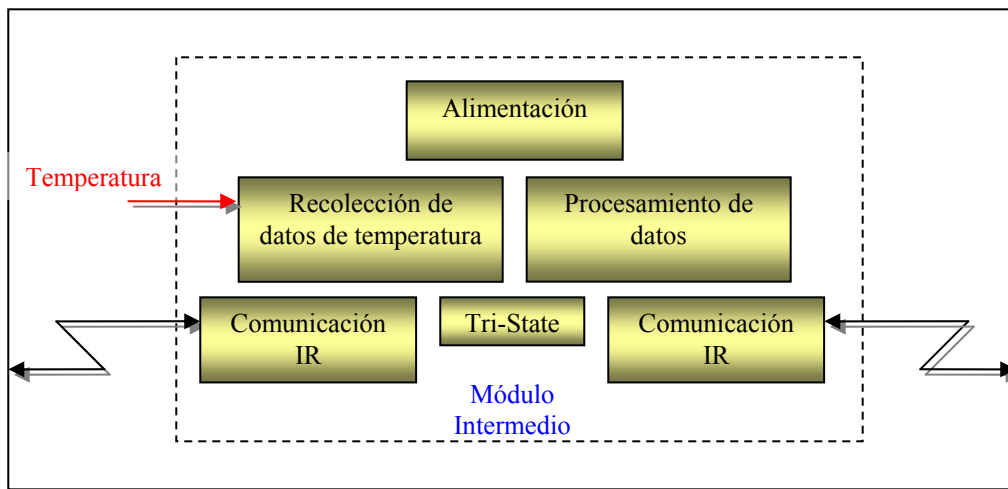
### 5.1.2.2 Módulos Intermedios

Estos módulos se encargan de la recolección de los datos de temperatura y cuentan con dos interfaces de comunicación infrarroja, lo que le permite transmitir y recibir datos tanto del módulo anterior, como del posterior. Está compuesto por los siguientes bloques (ver Figura 5.4):

- **Bloque de alimentación:** encargado de administrar la energía necesaria a los dispositivos electrónicos.
  
- **Bloque de recolección de datos de temperatura:** este bloque se encarga de la recolección del valor de temperatura y acondicionar dicha señal de manera que el bloque de procesamiento de datos pueda reconocer el valor capturado.
  
- **Bloque de procesamiento de datos:** este bloque se encarga de identificar las funciones que se deben de realizar, acomodar los datos según el protocolo diseñado y descartar la información que no le corresponde.
  
- **Bloque comunicación IR:** este bloque es el encargado de codificar y decodificar las señales provenientes de la UART del módulo intermedio para lograr establecer los enlaces de comunicación infrarrojos con los demás módulos.



- **Bloque Tri-State:** este bloque tiene una doble función, la primera es como medio de seguridad al desactivar los sensores infrarrojos que no se están utilizando, y así evitar que reciban información que no les corresponde; y su segunda funcionalidad es la de disminuir la cantidad de codificadores/decodificador IR, lo que permite un ahorro de espacio y costo económico.



**Figura 5.4** Esquema general del módulo intermedio

Al igual que el módulo principal descrito en el apartado 5.1.2.1, este sistema cuenta con dos entradas/salidas que corresponden a los enlaces de comunicación con el módulo anterior y posterior (solicitud y envío de datos de temperatura). Además cuenta con una tercera entrada, que corresponde al valor de temperatura que se pretende monitorear.

### **5.1.2.3 Módulos extremos**

Estos módulos tienen las mismas características que el caso anterior, con la diferencia de que solo cuentan con una interfaz infrarroja. En el caso del primer módulo, este está conectado mediante cable al módulo principal, y en el caso del último módulo solo establecerá comunicación con el módulo anterior.

## **5.1.3 Diagrama de tercer nivel**

### **5.1.3.1 Módulo principal**

En la Figura 5.5 se presenta un diagrama de bloques mas detallado correspondiente al módulo principal. Como se puede observar, este presenta tres etapas principales: bloque de alimentación, bloque de procesamiento de datos y el bloque de comunicación. A continuación se describen cada uno de estos bloques:

#### **5.1.3.1.1 Bloque de alimentación**

Esta etapa tiene la función de suministrar la energía a todos los dispositivos que conforman el módulo principal. Está conformado por una batería de 9V y el regulador de voltaje LM340. Como entrada tiene el voltaje suministrado por la batería y como salida se tiene el voltaje regulado a 5V.

#### **5.1.3.1.2 Bloque de procesamiento de datos**

Esta etapa está conformada por el microcontrolador pic16F873, el cual se encarga del procesamiento de datos provenientes de la PC, el primer módulo del nivel inferior y del primer módulo del nivel superior. Sus tres enlaces de comunicación mencionados anteriormente son bidireccionales, por lo tanto se consideran como de entrada y salida (I/O).

### 5.1.3.1.3 Bloque de comunicación

Esta etapa está conformada por los dispositivos necesarios para que el bloque de procesamiento de datos logre establecer la comunicación con los demás módulos. Para el caso de la comunicación con la PC es necesario utilizar el dispositivo Max232, el cual se encarga de acoplar los niveles de voltaje de las señales de entrada/salida. Por otro lado, para lograr establecer la comunicación con los módulos de nivel inferior y superior se utilizó cables telefónicos como medio físico para la transmisión y recepción de datos.

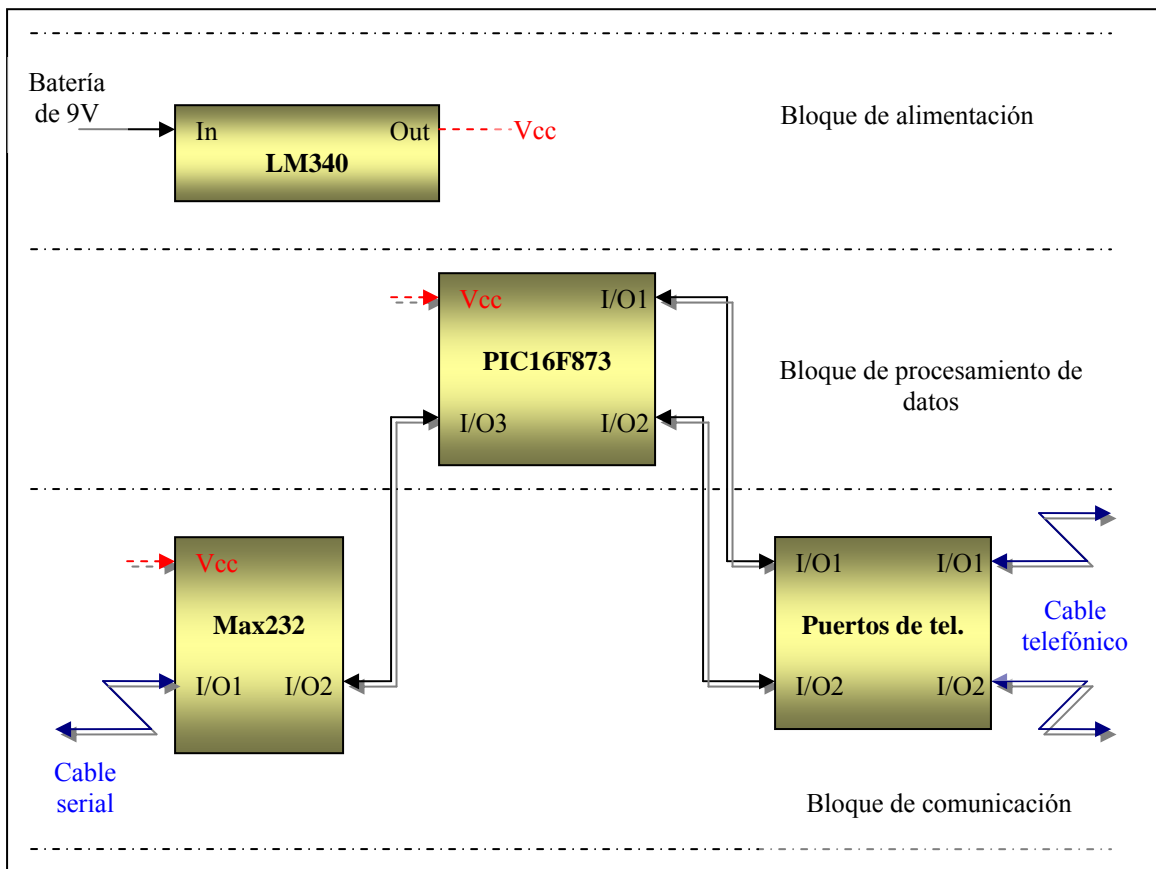


Figura 5.5 Diagrama de tercer nivel del módulo principal

### **5.1.3.2 Módulo intermedio**

En la Figura 5.6 se presenta un diagrama de bloques mas detallado correspondiente al módulo intermedio. Como se puede observar, este presenta cuatro etapas principales: bloque de alimentación, bloque de recolección de datos de temperatura, bloque de procesamiento de datos y el bloque de comunicación. El bloque de alimentación es exactamente igual al descrito en la sección anterior, por lo tanto referirse a la sección 5.1.3.1 para más información. A continuación se describen los demás bloques:

#### **5.1.3.2.1 Bloque de recolección de datos de temperatura**

Conformada por los dispositivos LM35 y LM158. El primero es un sensor de temperatura cuya entrada es la temperatura ambiente y su salida corresponde al valor eléctrico de la misma. El segundo dispositivo es un amplificador operacional, cuya entrada es el valor eléctrico de temperatura proveniente del LM35 y cuya salida corresponde al valor de temperatura acondicionado de manera que el bloque de procesamiento de datos pueda codificar dicho valor.

#### **5.1.3.2.2 Bloque de procesamiento de datos**

De igual manera que el módulo principal, este bloque está conformado por el microcontrolador pic16F873, el cual se encarga de administrar las funciones del módulo. A continuación se describen sus entradas, salidas y líneas de control:

- Entrada (In): se tiene el valor de temperatura proveniente del bloque de recolección de recolección de datos, el cual es utilizado para enviarlo a los demás módulos a través del protocolo de comunicación implementado.

- Señales de control (Ctrl): en este módulo hay dos dispositivos que deben de ser controlados o configurados por medio del microcontrolador. El primero es el dispositivo tri-state del bloque de comunicación encargado de definir la dirección del flujo de datos. El segundo es el codificador/decodificador MCP2120 el cual cuenta con algunas señales de control (enable, reset y mode) que son definidas por el pic16F873.
- Señales de entrada/salida (I/O): son las encargadas de establecer la comunicación entre los demás módulos, por lo que conectan al dispositivo MCP2120 para codificar/decodificar las señales infrarrojas.

#### **5.1.3.2.3 Bloque de comunicación**

Este bloque está conformado por tres tipos de dispositivos, el decodificador/codificador MCP2120, el transceiver infrarrojo HSDL3612 y el tri-state DM74LS125A, los cuales se encargan de establecer la comunicación del módulo con los demás periféricos.

A continuación se describen las entradas, salidas y líneas de control de cada uno de los dispositivos:

MCP2120:

- Señales de control (ctrl): las líneas de control provienen del microcontrolador, las cuales se encargan de configurar el dispositivo de manera que se comporte según las especificaciones del sistema.
- Entradas/salidas (I/O): las entradas/salidas provienen tanto del microcontrolador (I/O1) como del tri-state (I/O2), que son utilizadas para la transmisión y recepción de datos entre los dos dispositivos mencionados anteriormente.

HSDL3612:

- Entradas/salidas (I/O): estas líneas provienen tanto del tri-state (I/O1) como del enlace infrarrojo (I/O2). Las señales del enlace infrarrojo corresponden a los datos de transmisión y recepción que provienen de un módulo exterior (módulo extremo o intermedio).

DM74LS125A:

- Señales de control (ctrl): las líneas de control provenientes del microcontrolador son las encargadas de definir cuál de los dos transceivers HSDL3612 que se encuentran en el módulo estará habilitado, de manera que pueda establecer una comunicación con un módulo exterior (módulo extremo o intermedio).

- Entradas/salidas (I/O): las señales de transmisión/recepción provienen de tres fuentes: el MCP2120 (I/O1), o cualquiera de los dos HSDL3612 (I/O2). El establecimiento de la comunicación con alguno de los dos HSDL3612 está definido por las señales de control.

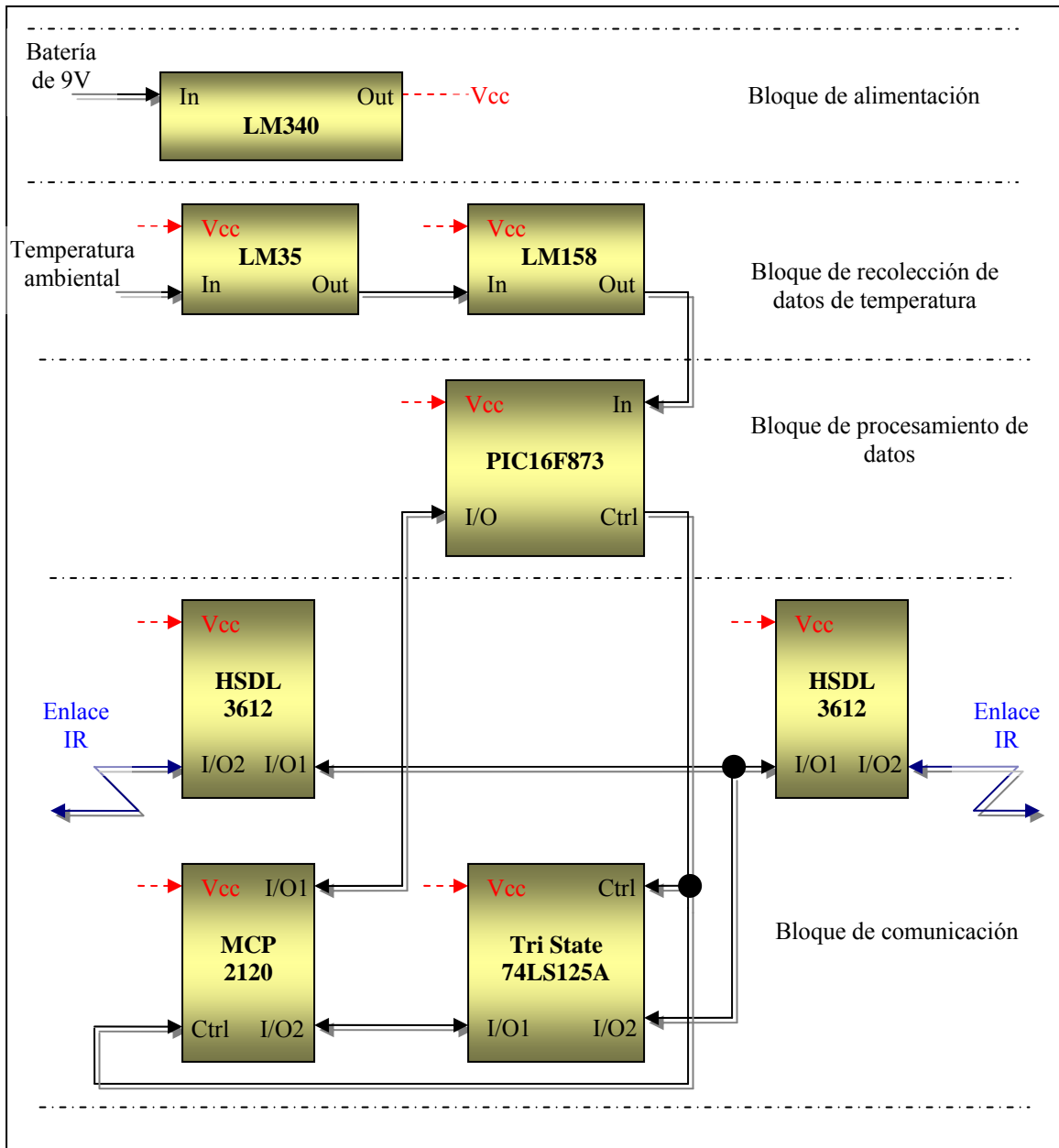


Figura 5.6 Diagrama de tercer nivel del módulo intermedio

### **5.1.3.3 Módulo extremo**

En la Figura 5.7 se presenta un diagrama de bloques mas detallado correspondiente al módulo extremo. Como se puede observar, este presenta cuatro etapas principales: bloque de alimentación, bloque de recolección de datos de temperatura, bloque de procesamiento de datos y el bloque de comunicación. El bloque de alimentación, recolección de datos y procesamiento de datos son exactamente iguales al descrito en la sección anterior, por lo tanto referirse a la sección 5.1.3.2 para más información. Respecto al bloque de comunicación existe una pequeña diferencia en relación con el módulo intermedio, éste solo cuenta con un transceiver infrarrojo, por lo que se descarta la necesidad del dispositivo tri-state (encargado de escoger cuál de los dos transceiver habilitar en el módulo intermedio). A continuación se describe este bloque:

#### **5.1.3.3.1 Bloque de comunicación**

Este bloque está conformado por dos tipos de dispositivos, el decodificador/codificador MCP2120 y el transceiver infrarrojo HSDL3612, los cuales se encargan de establecer la comunicación del módulo el módulo intermedio.

A continuación se describen las entradas, salidas y líneas de control de cada uno de los dispositivos:

MCP2120:

- Señales de control (ctrl): las líneas de control provienen del microcontrolador, las cuales se encargan de configurar el dispositivo de manera que se comporte según las especificaciones del sistema.



- Entradas/salidas (I/O): las entradas/salidas provienen tanto del microcontrolador (I/O1) como del HSDL3612 (I/O2), las cuales son utilizadas para establecer la comunicación entre los dos dispositivos mencionados anteriormente.

HSDL3612:

- Entradas/salidas (I/O): estas líneas provienen tanto del MCP2120 (I/O1) como del enlace infrarrojo (I/O2). Las señales del enlace infrarrojo corresponden a los datos de transmisión y recepción provenientes del módulo intermedio.

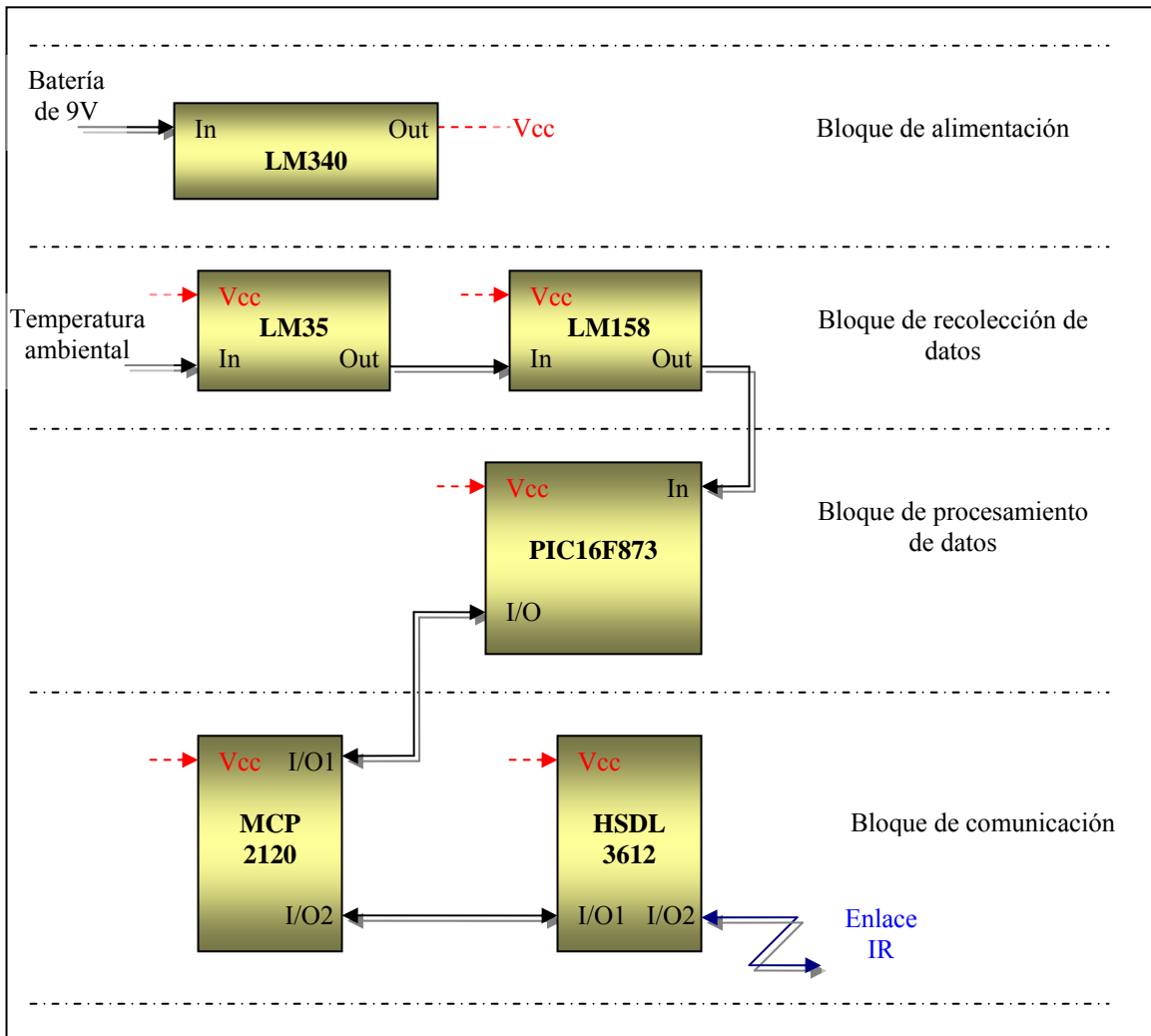


Figura 5.7 Diagrama de tercer nivel del módulo extremo

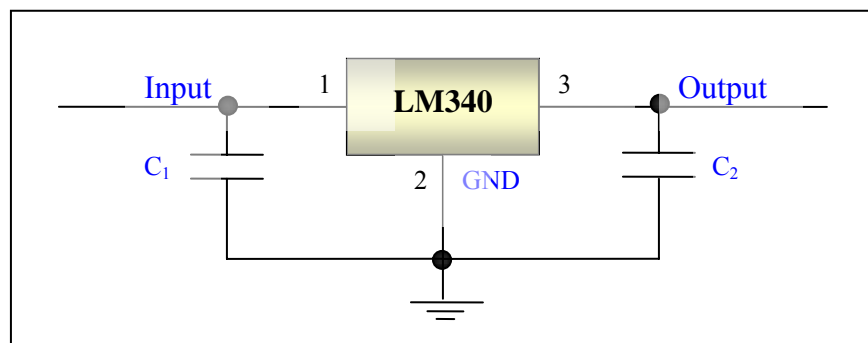
## 5.1.4 Diagrama de cuarto nivel

### 5.1.4.1 Bloque de alimentación

Debido a que todos los dispositivos seleccionados trabajan a 5V, se seleccionó el dispositivo LM340 de National Semiconductor, el cual brinda un voltaje de salida de 5V y requiere un voltaje mínimo de entrada de 7.5V. Este dispositivo cuenta con protección interna ante corto circuitos y sobrecarga, lo cual brinda mayor seguridad al sistema. Para su funcionamiento, solo ocupa dos capacitores externos, uno en su entrada y otro en su salida.

Según la Figura 5.8 correspondiente al esquema del dispositivo LM340 se pueden apreciar las siguientes conexiones:

- Puerto 1: entrada del sistema que va conectada a una batería de 9V encargada de suministrar la energía requerida del sistema.
- Puerto 2: conexión a tierra.
- Puerto 3: salida del dispositivo que brinda un voltaje de 5V a todos los demás componentes que integran el sistema de monitoreo.



**Figura 5.8.** Diagrama del regulador LM340

#### **5.1.4.2 Bloque de procesamiento de datos**

Este bloque está constituido por un microcontrolador pic16F873, el cual se encarga de diferentes tareas según la función del módulo. Dichas tareas se describen a continuación:

- a) Convertidor analógico-digital (ADC): mediante este puerto se recibe los datos del bloque de recolección de datos de temperatura, cuyos valores analógicos son convertidos a digitales de manera que el sistema pueda procesarlos.
- b) Múltiples puertos con protocolo RS232: este microcontrolador tiene la característica de que sus puertos pueden ser configurados vía software para que trabajen con el protocolo RS232, lo cual es una gran ventaja debido a la necesidad del sistema de comunicarse con varios dispositivos. Esto facilita el trabajo ya que se logra trabajar con un protocolo conocido, fiable y universal. En el caso del módulo principal se utilizan tres puertos con este protocolo, uno para la comunicación con la computadora y dos para la comunicación con el primer módulo de cada nivel (inferior y superior).  
Además permite una escalabilidad al sistema al darle la oportunidad de conectar una mayor cantidad de módulos.
- c) Contiene el protocolo de comunicación: el microcontrolador se encuentra programado con el protocolo creado para establecer la comunicación entre los diversos módulos (para más detalles del protocolo ver sección 5.2.2.2)
- d) Procesamiento de datos: En todos los módulos (principal, intermedios y extremos) se realiza un cálculo matemático con todos los valores de temperatura (cada vez que recibe un dato) el cual tiene como función la detección de errores. En el caso del módulo principal, además del cálculo matemático, se procesan los datos que recibe de todos los módulos,

acomodándolos y eliminando la información innecesaria para enviársela a la PC.

- e) Cantidad de puertos: en los dispositivos intermedios es necesario gobernar el dispositivo tri-state para determinar cuál sensor infrarrojo debe estar activo, por lo cual es necesario 4 líneas de control para dicha tarea.

Entre los criterios que se tomaron para escoger este dispositivo se pueden mencionar los siguientes:

- a) Cuenta con las características necesarias que cada módulo ocupa (ADC y múltiples puertos RS232).
- b) De los microcontroladores de la marca Microchip es el componente más pequeño (aproximadamente 34x7mm) y económico que cuenta con las características mencionadas en el punto anterior.
- c) Familiaridad con el dispositivo (en proyectos pasados ya se había trabajado con dicho componente) y se tenía a disposición el programador del mismo (Olimex, pic-pg2c).

En la tabla 5.1 se presenta un resumen comparativo de los diversos microcontroladores que se analizaron para determinar el que mejor se adecuaba al sistema.

**Tabla 5.1** Lista Comparativa entre Microcontroladores

<b>Modelo</b>	<b>Características</b>	<b>Precio</b>
PIC10F204	No soporta UART ni ADC	\$2.00
8259 - PIC	No soporta UART ni ADC	\$4.00
PIC16F630	No soporta UART	\$4.47
PIC16F676	No soporta UART	\$5.00
PIC16F628 20MHz	No soporta ADC	\$6.00
PIC16F84A 4MHz	No soporta UART ni ADC	\$7.00
PIC16F873A 20MHz	Cuenta con UART y ADC	\$8.50
PIC16F876A 20MHz	Cuenta con UART y ADC	\$9.50
PIC16F870	Cuenta con UART y ADC	\$10.0

En la Figura 5.9 se muestra el diagrama de pines del dispositivo PIC16F873, a continuación se describen los puertos utilizados y la función que desempeñan:

- Puerto 1 (MCLR/Vpp): este pin se encarga de reiniciar el sistema cuando su entrada esta en bajo, por lo tanto se tiene conectado a Vcc para que el dispositivo se encuentre en su funcionamiento normal.
- Puerto 2 (RA0/AN0): este puerto es utilizado como entrada analógica en los módulos que cuentan con un sensor de temperatura, por lo tanto se encuentra conectado físicamente a la salida del amplificador operacional del bloque de recolección de datos.
- Puerto 8, 19 y 20 ( $V_{SS}$ ,  $V_{DD}$ ): los puertos 8 y 19 son utilizados como referencia de tierra, mientras que el 20 es el encargado de suplir la energía del dispositivo.
- Puerto 9 y 10 (OSC1/CLK1, OSC2/CLK0): entre estos dos puertos se conecta el oscilador que se encarga de suplir la señal de reloj para el funcionamiento del dispositivo.
- Puerto 17 y 18 (RC6/Tx, RC7/Rx): utiliza la USART del dispositivo para la transmisión y recepción asincrónica respectivamente. Estos puertos están conectados físicamente a los puertos de transmisión (pin 7) y recepción (pin 11) del dispositivo MCP2120.
- Puertos 15, 16 (RC4, RC5): en el caso de los módulos intermedios y extremos, estos pines son utilizados para controlar el decodificador/codificador MCP2120. El primero (puerto 15) se encarga de habilitar/deshabilitar el MCP2120 y el segundo

(puerto 16) se escoge el modo de operación del mismo (para más información del modo de operación referirse a la sección 5.4.4).

- Puertos 21, 22, 23, 24, 25, 26 (RB0, RB1, RB2, RB3, RB4 Y RB5): en el caso de los módulos intermedios y extremos, los cuatro primeros pines (21, 22, 23 y 24) son utilizados para controlar el dispositivo tri-state 74LS125. Estas cuatro líneas de control habilitan y deshabilitan las señales de transmisión/recepción provenientes del MCP2120, lo que permite que escoger si se desea enviar los datos al módulo anterior o al posterior; el pin 26 se encarga de reiniciar el dispositivo MCP2120 en caso de ser necesario; el pin 25 en todos los módulos va a conectado a un led, el cual tiene como propósito informarle al usuario si el dispositivo se encuentra apagado o encendido.

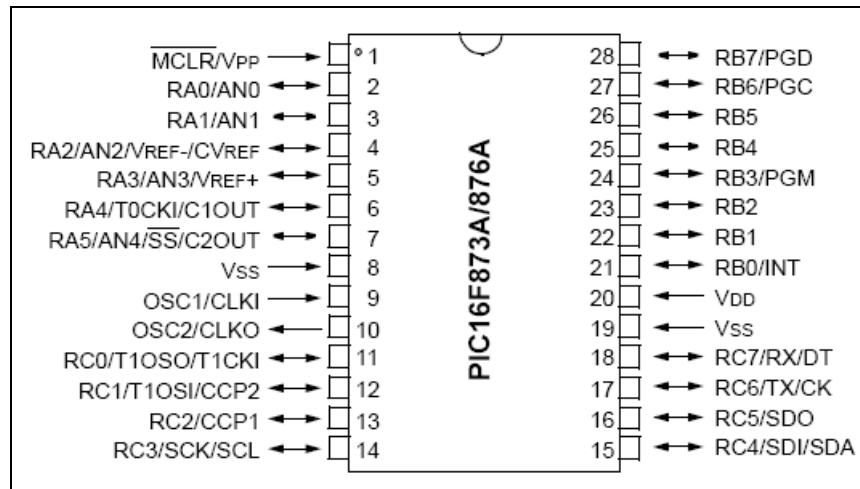


Figura 5.9 Diagrama de pines del pic16F873<sup>11</sup>

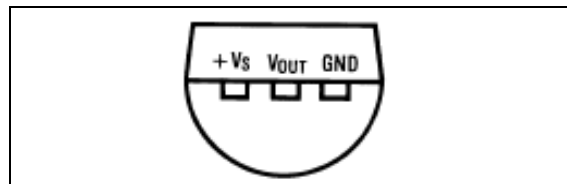
<sup>11</sup> Tomado de las hojas de datos del pic16F873.

### 5.1.4.3 Bloque de recolección de datos

Este bloque está dividido en dos partes: el sensor de temperatura y el amplificador de señal.

El dispositivo utilizado para recolectar los valores de la temperatura es el LM35 de National Semiconductor. Este componente se puede adquirir en diferentes tipos de encapsulado, se optó por el TO-92 debido a que es el ocupa menor espacio físico y es el más económico. Entre sus características principales están:

- Salida de voltaje lineal respecto a la temperatura en grados Celsius (ventaja que se aprovecha en la simplicidad al programar).
- No ocupa calibración ni componentes externos.
- Bajo costo y consumo de potencia.
- Puede funcionar con una sola fuente.



**Figura 5.10** Esquema del sensor de temperatura LM35<sup>12</sup>

<sup>12</sup> Tomado de las hojas de datos del LM35.

A continuación se resumen las principales ventajas y desventajas de los sensores considerados:

**Tabla 5.2** Lista comparativa entre Sensores de Temperatura

<b>Sensor de Temperatura</b>	<b>Ventajas</b>	<b>Desventajas</b>
Termopar	<ul style="list-style-type: none"> <li>• Bajo costo</li> <li>• Robustez</li> <li>• Estables a largo tiempo</li> <li>• Tamaño reducido</li> <li>• Respuesta rápida</li> <li>• Variedad de tipos</li> </ul>	<ul style="list-style-type: none"> <li>• Acondicionamiento de señal imprescindible.</li> <li>• No todos tienen comportamiento lineal.</li> <li>• Cuidado especial a la hora de instalar</li> <li>• Susceptibles al voltaje espurio (por humedad)</li> </ul>
RTD	<ul style="list-style-type: none"> <li>• Comportamiento lineal</li> <li>• Sensibles (precisos)</li> </ul>	<ul style="list-style-type: none"> <li>• Respuesta lenta</li> <li>• Alto costo</li> <li>• Rango limitado de temperatura</li> </ul>
Termistor	<ul style="list-style-type: none"> <li>• Económicos</li> <li>• Coeficiente de temperatura positivo o negativo</li> <li>• Sensibles</li> </ul>	<ul style="list-style-type: none"> <li>• Comportamiento no lineal</li> <li>• Alta disipación interna</li> <li>• Alto manejos de corrientes</li> </ul>
Semiconductores	<ul style="list-style-type: none"> <li>• Fácil sensado de temperatura</li> </ul>	<ul style="list-style-type: none"> <li>• Rangos limitados</li> <li>• Comportamiento no lineal</li> <li>• Características de deriva e inexactitudes</li> </ul>
LM35	<ul style="list-style-type: none"> <li>• Salida de voltaje proporcional a escala Celsius</li> <li>• Amplio rango de voltaje (4-30V)</li> <li>• No ocupa calibración externa</li> <li>• Bajo consumo de potencia</li> <li>• Comportamiento lineal</li> <li>• Económico</li> <li>• Fácil conexión a sistema de toma de datos</li> </ul>	<ul style="list-style-type: none"> <li>• Ocupa compensación de capacitancia.</li> </ul>



Según la Figura 5.10, este dispositivo cuenta con tres puertos descritos a continuación:

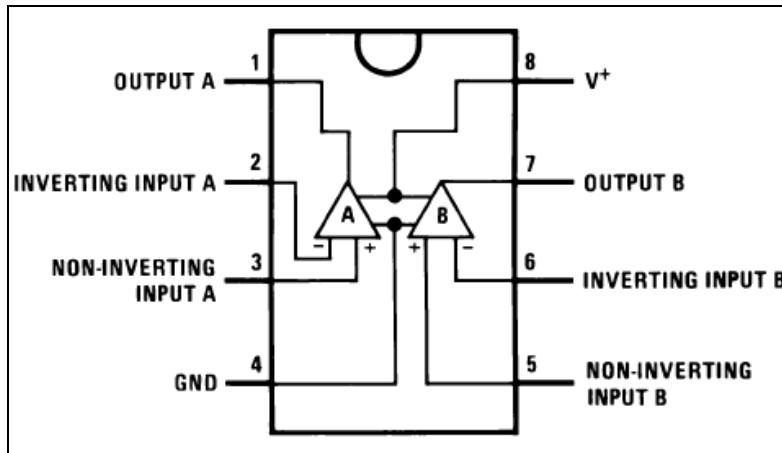
- Puerto de alimentación ( $V_{s+}$ ): este pin va conectado directamente a  $V_{cc}$  (5V).
- Puerto de salida ( $V_{out}$ ): este pin va conectado al puerto de entrada del dispositivo de amplificación (pin 2 del LM158).
- Referencia a tierra (Gnd): como su nombre lo describe, este pin va conectado a la referencia del sistema.

El propósito del amplificador de señal es el de acondicionar la señal de tal manera que el microcontrolador puede captar el cambio de temperatura de una manera precisa. El ADC del microcontrolador tiene una resolución de 256 bits y es alimentado con 5V, por lo tanto cada paso está definido en un valor de 0.01953V. Debido a que la razón de cambio del sensor es de 10mV por cada grado centígrado, la amplificación de la señal de salida del mismo, debe ser de 1.95.

Para llevar a cabo la tarea antes mencionada se optó por el amplificador operacional LM158 de National Semiconductor, debido a las características que se describen a continuación:

- Trabaja con una sola fuente de alimentación (principal razón de escogencia).
- No abarca mucho espacio físico (aproximadamente 9.5x6.5mm).
- Amplio rango de voltaje de alimentación unitaria: 3V – 32V.
- Muy baja corriente de drenaje de alimentación (500 $\mu$ A)

Se utilizó una configuración de amplificación no inversora como se muestra en la Figura 3.1 en la sección 3.8 del Marco Teórico



**Figura 5.11** Esquema del amplificador de señal<sup>13</sup>

A continuación se describen los pines utilizados del dispositivo LM158 (ver Figura 5.11 como referencia):

- Puerto 1 (output 1): este pin corresponde a la salida del amplificador operacional, el cual transmite la señal de temperatura con el voltaje adecuado de manera que el microcontrolador pueda procesarla. Dicha salida se encuentra conectada físicamente a la entrada analógica del microcontrolador (pin 2, AN0).
- Puerto 2 (inverting input A): este pin corresponde a la entrada inversora del amplificador operacional, el cual se encuentra conectado según la configuración de amplificación no inversora (ver Figura 3.1).
- Puerto 3 (non-inverting input A): este pin corresponde a la entrada no inversora del amplificador operacional, el cual tiene la función de recibir la señal analógica proveniente del sensor de temperatura. Físicamente se encuentra conectado a la salida Vout del dispositivo LM35.

<sup>13</sup> Tomado de las hojas de datos del dispositivo LM158 de la marca National Semiconductor

- Puerto 4 y 8 (GND y V<sup>+</sup>): estos puertos corresponden a la referencia a tierra y a la alimentación del dispositivo respectivamente.

#### **5.1.4.4 Bloque de comunicación infrarroja (IR)**

Este bloque está conformado por dos dispositivos principales, el codificador/decodificador (MCP2120) y el sensor Infrarrojo (HSDL3612).

El MCP2120 es el encargado de convertir la señal eléctrica en pulsos infrarrojos y viceversa. Entre las mayores ventajas de este dispositivo se pueden mencionar las siguientes:

- Utiliza el protocolo RS232 para comunicarse con el dispositivo de control (en este caso el microcontrolador Pic16F873), el cual como se mencionó antes, es un protocolo robusto y confiable.
- Totalmente inmune a interferencia electromagnética y ruido provocado por las fuentes de alimentación.
- No requiere de dispositivos externos.

A continuación se describen los pines utilizados del dispositivo MCP2120 (ver Figura 5.12):

- Puertos 1 y 14 (V<sub>DD</sub> y V<sub>SS</sub>): corresponden a la fuente de alimentación positiva y la referencia a tierra del dispositivo.
- Puertos 2 y 3 (OSC1 y OSC2): entre estas dos terminales se conecta un oscilador el cual determina junto con los pines de control (Baud2, Baud1, Baud0) la tasa de transferencia en baudios.

- Puerto 4 (Reset): se encarga de reiniciar el dispositivo. Esta entrada está conectada físicamente al pin 26 (RB5) del microcontrolador.
- Puertos 5 y 6 (RxIR Y TxIR): corresponden a las señales de recepción y transmisión asincrónicas provenientes del transceiver infrarrojo. Estas dos terminales se encuentran conectadas físicamente a los pines 8 (Rx) y 9 (Tx) del dispositivo HSDL3612.
- Puerto 7 (Mode): esta entrada se encarga de determinar si la tasa de transferencia de datos será configurada por medio de software o de hardware. Para el caso del sistema implementado se definió dicha tasa por medio hardware.
- Puertos 8, 9 y 10 (Baud2, Baud1, Baud0): estas entradas definen la velocidad de la tasa de transferencia de datos en baudios. Para el caso del sistema implementado se definió una tasa de 9600 con un cristal de 7.3728MHz, lo cual se logra conectando las tres terminales (Baud2, Baud1, Baud0) a tierra (ver tabla 5.3).

**Tabla 5.3** Tabla de configuración de frecuencia del MCP2120

BAUD2:BAUD0	Frecuencia (MHz)							Bit Rate
	0.6144	2.000	3.6864	4.9152	7.3728	14.7456	20.000	
000	800	2604	4800	6400	9600	19200	26042	Fosc/768
001	1600	5208	9600	12800	19200	38400	52083	Fosc/384
010	3200	10417	19200	25600	38400	78600	104167	Fosc/192
011	4800	15625	28800	38400	57600	115200	156250	Fosc/128
100	9600	31250	57600	78600	115200	230400	312500	Fosc/64

- Puertos 11 y 12 (Rx y Tx): estos pines corresponden a los puertos de recepción y transmisión asincrónicos provenientes del controlador UART del microcontrolador. Están conectados físicamente a los pines 18 (Rx) y 17 (Tx) del pic16F873.

- Puerto 13 (En): este puerto se encarga de habilitar el dispositivo. Este comando es controlado por medio del puerto 15 (RC4) del microcontrolador.

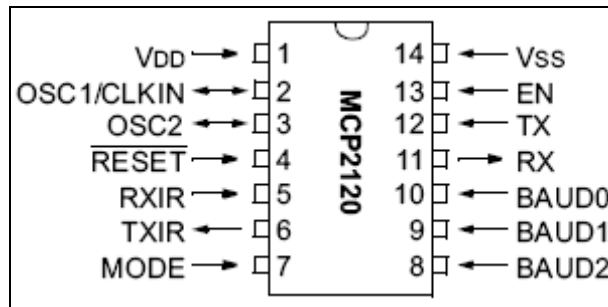


Figura 5.12 Diagrama de pines del MCP2120<sup>14</sup>

El HSDL3612 es un transceiver infrarrojo capaz de establecer una comunicación half duplex y serial a través del aire (ver esquema en Figura 5.14). Entre sus características se pueden mencionar las siguientes:

- Alcance de aproximadamente 1.5m
- Bajo consumo de potencia (entre 2.7 y 5.5V)
- Dimensiones: 4x12.2x5.1mm
- Inmunidad al ruido
- Sólo ocupa dos componentes externos

A continuación se describen los pines utilizados del dispositivo HSDL3612 (ver Figura 5.13 como referencia):

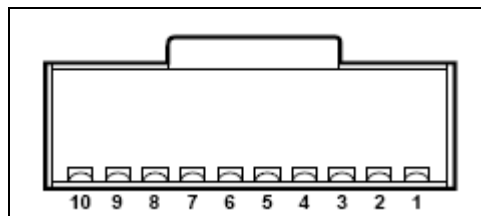
- Puerto 1 ( $V_{CC}$ ): entrada de alimentación del sistema (5V).
- Puerto 2, 3 y 7 (Gnd): referencias a tierra.

<sup>14</sup> Tomado de las hojas de datos del MCP2120 de la marca Microchip

- Puerto 4 y 5 (Mode, MD0 y MD1): estos dos pines de control permiten configurar la potencia de la señal de transmisión para definir el alcance de la señal infrarroja. En la tabla 5.4 se describen las diferentes configuraciones que se pueden utilizar. Para el sistema implementado se definió utilizar la máxima potencia de transmisión por lo que se conectaron ambos pines a tierra.

- Puerto 8 y 9 (RxD y TxD): estos puertos corresponden al receptor y transmisor del dispositivo, por lo que se conectan directamente al receptor (pin 5) y transmisor (pin 6) del MCP2120 respectivamente.

- Puerto 10 (Led Anode): esta conexión es utilizada para alimentar el led interno del componente por lo que se conecta a  $V_{CC}$  a través de una resistencia de  $15\Omega$  según especificaciones del fabricante (ver hoja de datos en el apéndice).

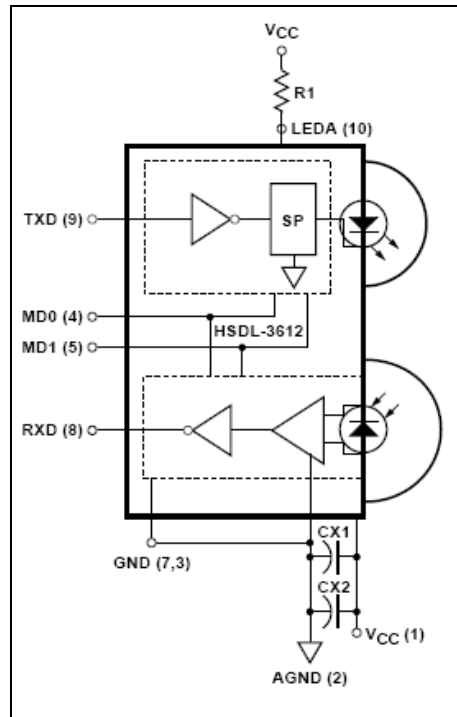


**Figura 5.13** Diagrama de pines del Transceiver HSDL3612<sup>15</sup>

**Tabla 5.4** Configuración de las funciones de transmisión y recepción

Mode 0	Mode 1	Función Rx	Función Tx
1	0	Apagado	Apagado
0	0	Encendido	Poder de distancia máxima
0	1	Encendido	Poder de 2/3 de distancia
1	1	Encendido	Poder de 1/3 de distancia

<sup>15</sup> Tomado de las hojas de datos del HSDL3612 de la marca Agilent



**Figura 5.14** Esquema del Transceiver HSDL3612<sup>16</sup>

En la Figura 5.15 se muestra el esquema de conexión del MCP2120, el HSDL3621 y el microcontrolador. Las señales enviadas por la UART del pic16F873 son moduladas y enviadas como pulsos eléctricos hacia el Transceiver (HSDL1612). De igual manera el Transceiver IR recibe información y la envía como pulsos eléctricos para que el MCP2120 la decodifique (demodulación) y envíe estas señales eléctricas por medio de la UART del mismo al microcontrolador. La modulación y demodulación de las señales cumplen con las especificaciones del estándar IrDA. Esta configuración de módulos se encuentra especificada en la hoja de datos del MCP2120 (ver hojas de datos en el apéndice).

<sup>16</sup> Tomado de las hojas de datos del HSDL3612 de la marca Agilent

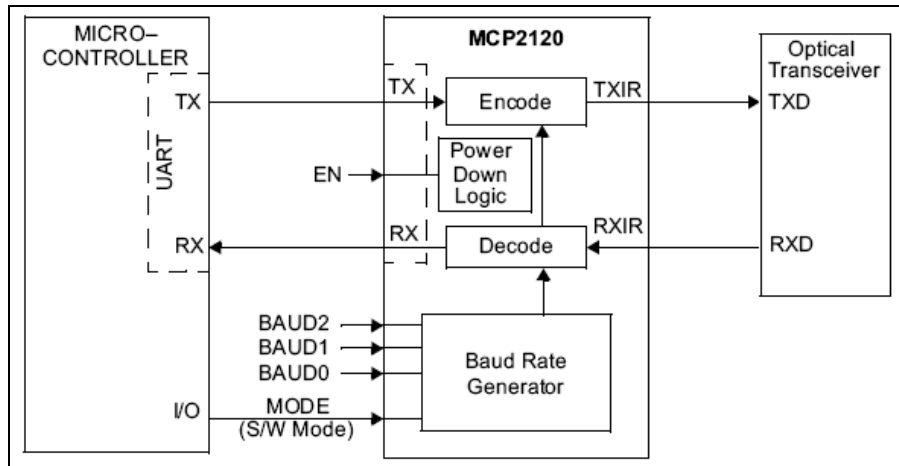


Figura 5.15 Diagrama de conexión del bloque de comunicación IR<sup>17</sup>

#### 5.1.4.5 Bloque Interfaz RS232

El puerto serie de la computadora trabaja a 12V, por lo cual es necesario de un dispositivo que acondicione la señal del módulo principal (5V) para que se logre establecer la comunicación. El MAX232 es un driver/receiver que contiene un generador de voltaje capacitivo, el cual cumple con las especificaciones de los niveles voltaje del TIA/EIA-232-F a partir de una sola fuente de 5V. El esquema de la conexión se puede observar en la Figura 5.16.

<sup>17</sup> Tomado de las hojas de datos del MCP2120 de la marca Microchip



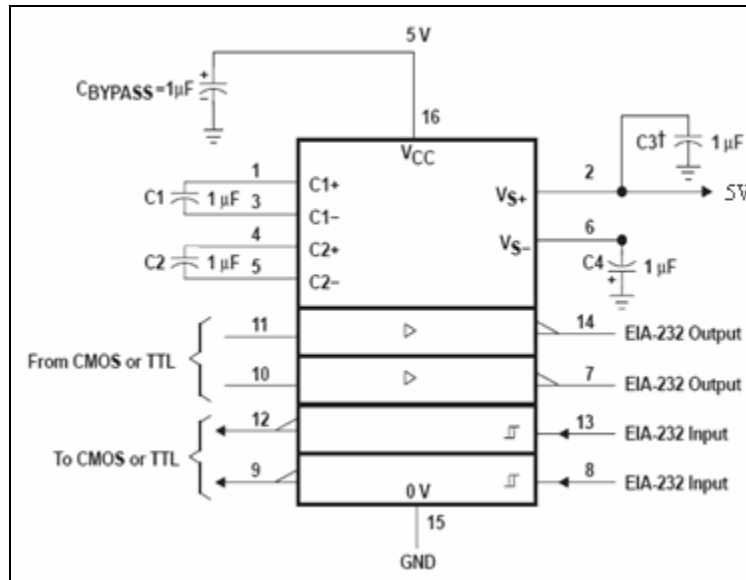


Figura 5.16 Esquema de conexión de componentes externos del MAX232<sup>18</sup>

De este dispositivo solo se utilizaron un transmisor un receptor (de los dos que posee) para poder establecer la comunicación entre el sistema de monitoreo y la PC. A continuación se describen las conexiones:

- Puerto 13 y 14 (R1In y T1Out): estos puertos se encuentran conectados al receptor y transmisor respectivamente del cable serial que se encuentra conectado a la computadora.
- Puerto 11 y 12 (T1In y R1Out): estos puertos se encuentran conectados a los puertos de transmisión y recepción de la UART del microcontrolador, que corresponden a los pines 17(Tx) y 18(Rx) respectivamente.

#### 5.1.4.6 Bloque Tri-State

Este bloque se encarga de seleccionar en que dirección están fluyendo los datos, es decir, si los datos provienen del lado izquierdo o derecho del módulo en

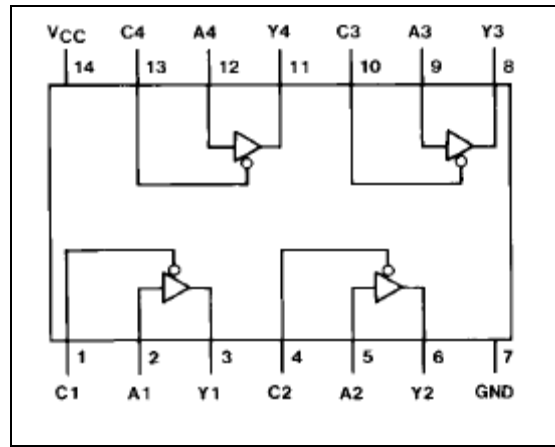
<sup>18</sup> Tomado de las hojas de datos del MAX232

cuestión. Esto permite utilizar un solo MCP2120 por módulo, por lo que el costo y el tamaño del mismo se ven disminuidos de igual manera. Además contribuye a disminuir la cantidad de errores debido a que evita que un módulo reciba datos que no le corresponden.

El dispositivo utilizado para este fin es el DM74LS125A de la marca Fairchild Semiconductor. Este componente cuenta con cuatro compuertas independientes que proveen un funcionamiento de almacenaje no inversor.

Utilizando como referencia la Figura 5.17, a continuación se describen los pines utilizados:

- Puertos 1, 4, 10 y 13 (C1, C2, C3 y C4): estos puertos corresponden a las líneas de control, encargadas de activar o deshabilitar las compuertas tri-state. Como se mencionó en la sección 5.1.4.3 (sobre las conexiones del microcontrolador), estas líneas se encuentran gobernadas por los puertos RB0, RB1, RB2 y RB3.
- Puertos 2, 5, 9 y 12 (A1, A2, A3 y A4): las entradas A1 y A4 se conectan al mismo punto, el cual corresponde a la salida TxIR (pin 6) del MCP2120 con el fin de que este pueda transmitir por cualquiera de los dos transceivers (HSDL3612). Por otro lado, las entradas A2 y A3 se encuentran conectadas al puerto RxD (pin 8) de cada uno de los dos transceivers con el fin de recibir información de cualquiera de los dos.
- Puertos 3, 6, 8 y 11 (Y1, Y2, Y3 y Y4): de igual manera que en los puertos de entrada, Y2 y Y3 se encuentran conectados al mismo punto, el cual corresponde a la entrada RxIR (pin 5) del MCP2120 con el fin de que este pueda recibir por cualquiera de los dos transceivers. Por otro lado, las salidas Y1 y Y4 van conectadas al puerto TxD (pin 9) del HSDL3612 de manera que cualquiera de los dos transceivers pueda transmitir la información necesaria.



**Figura 5.17** Esquema interno del dispositivo Tri-State DM74LS125A<sup>19</sup>

<sup>19</sup> Tomado de las hojas de datos de DM74LS125A de la marca Fairchild

### 5.1.5 Diagrama de quinto nivel

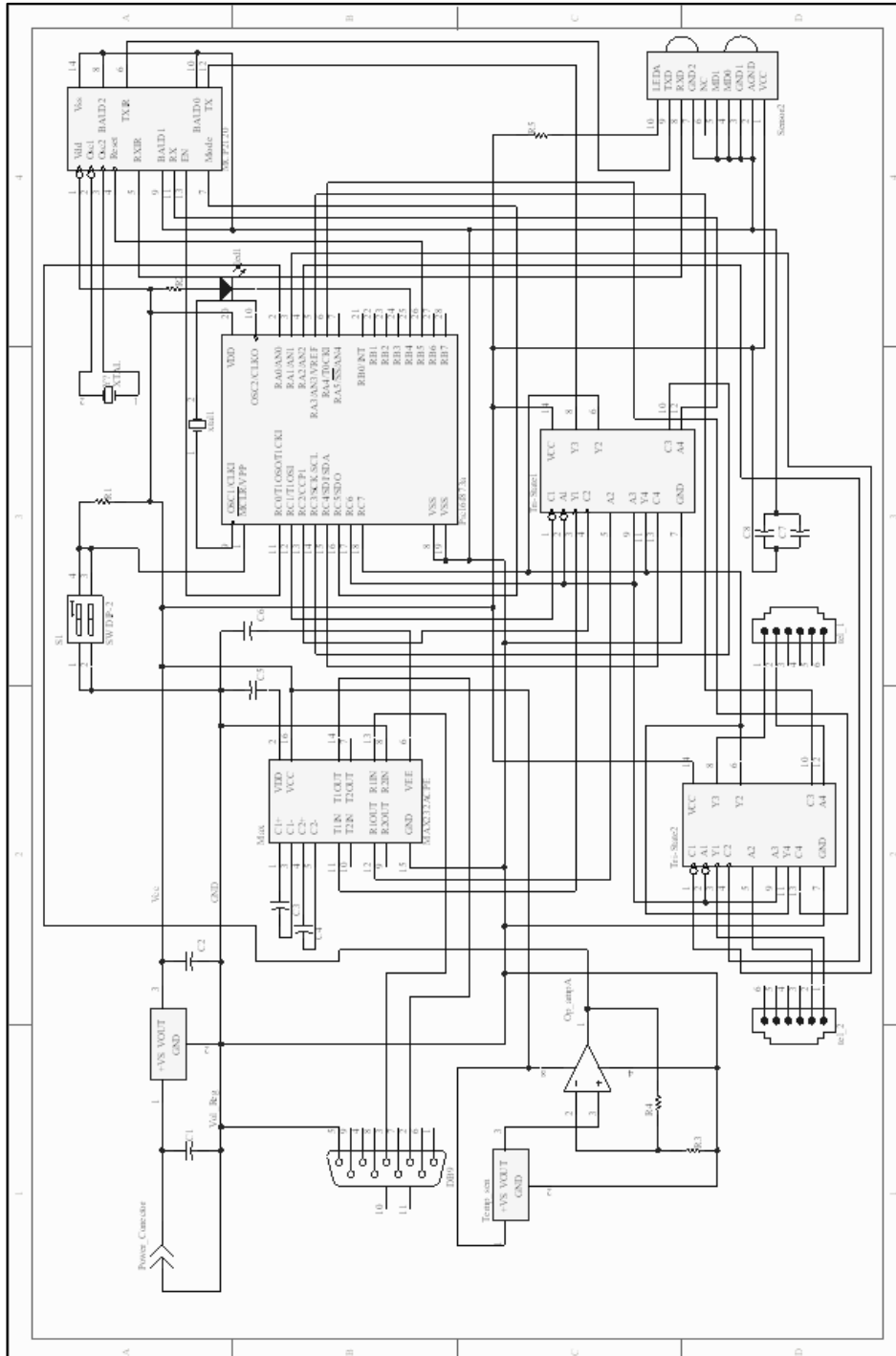


Figura 5.18 Circuito del módulo principal

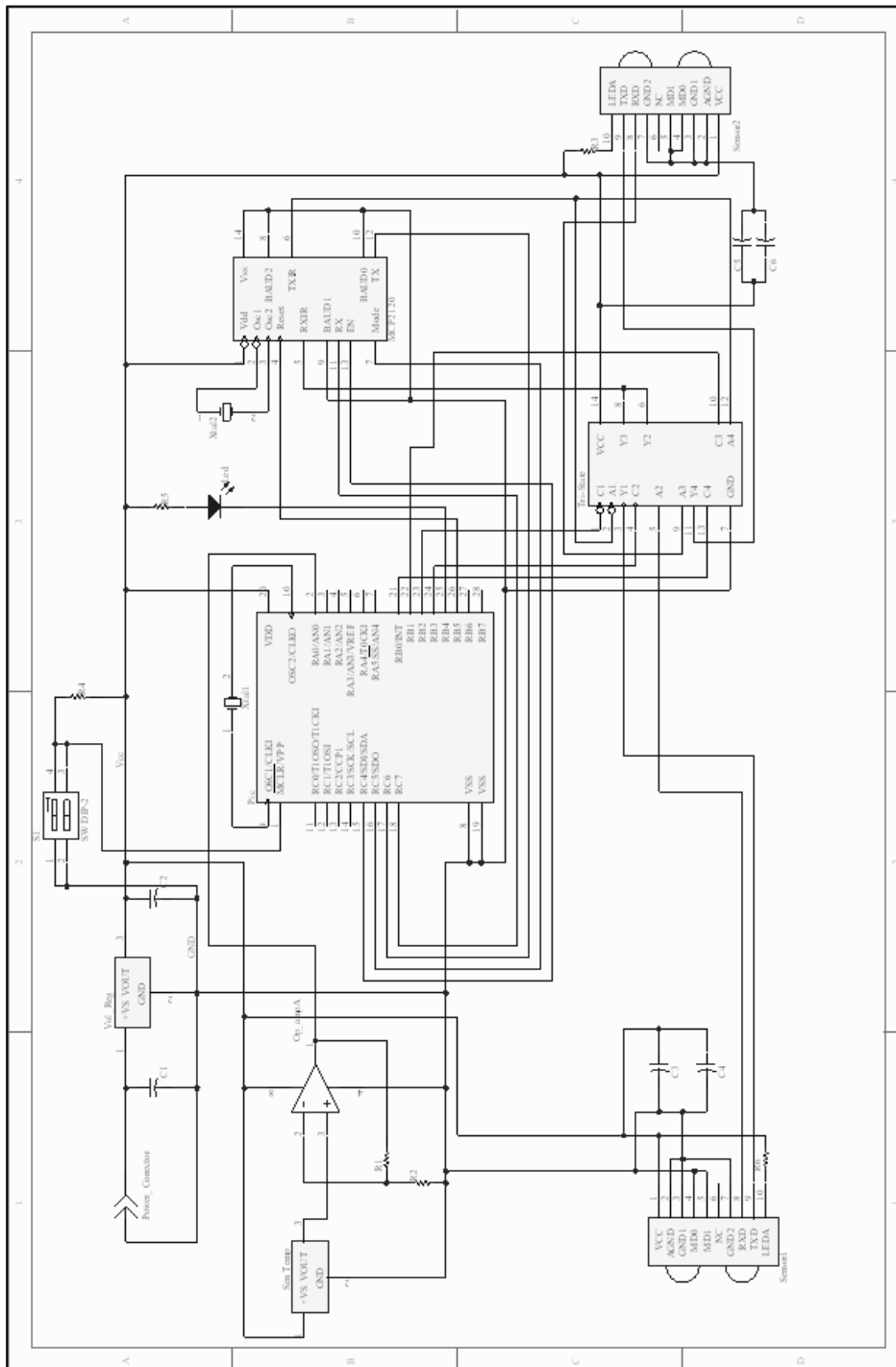


Figura 5.19 Circuito del módulo intermedio

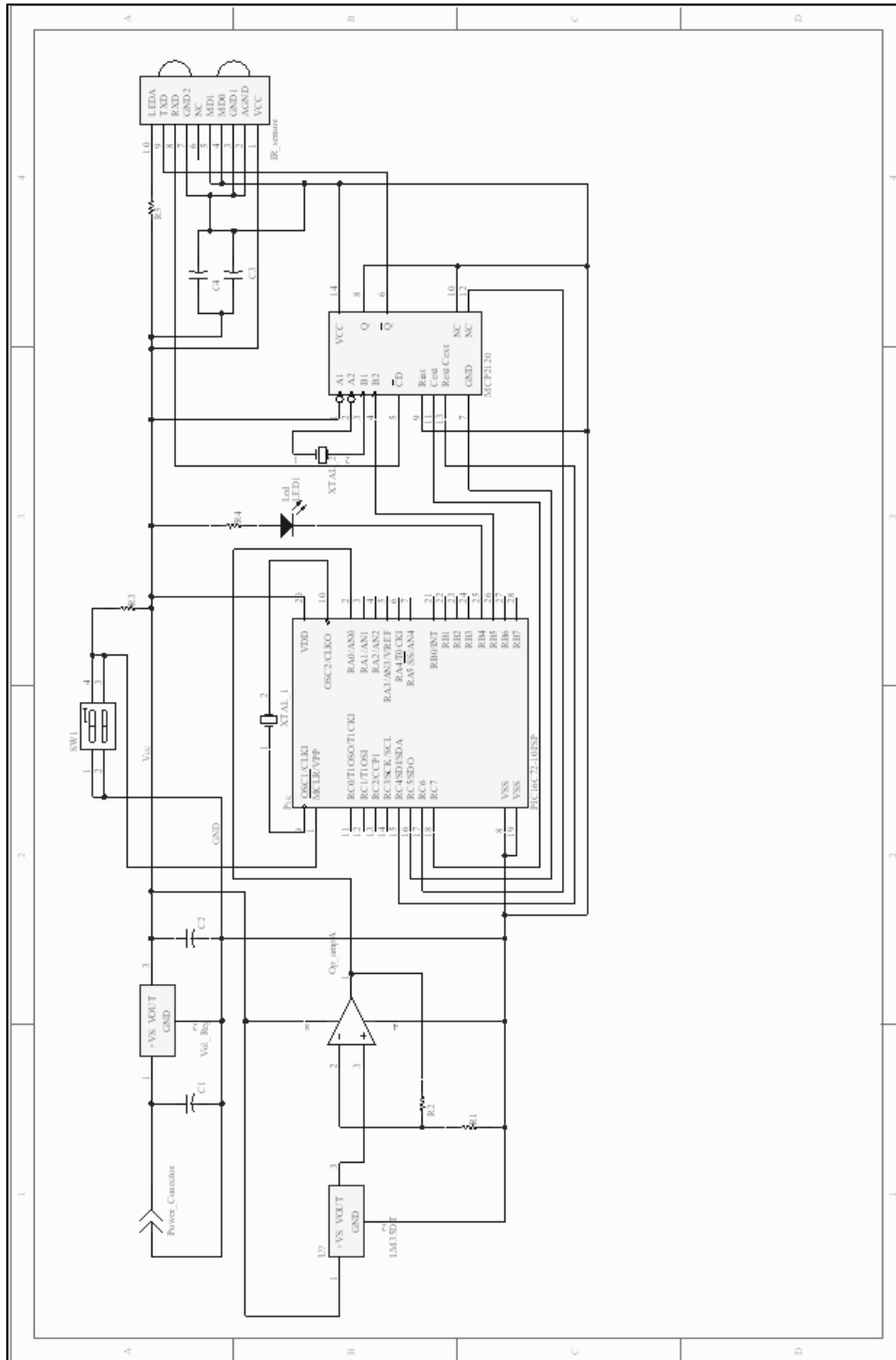


Figura 5.20 Circuito del módulo extremo

## **5.2 Descripción del software**

A lo largo del proyecto se desarrollaron los siguientes tres programas: el programa de recolección de datos de temperatura y establecimiento de comunicación entre los diferentes módulos (al cuál se le referirá como “programa principal”), la interfaz con el usuario y por último el programa que describe cómo podría automatizarse el sistema de enfriamiento del cuarto en estudio.

### **5.2.1 Criterios de diseño**

Programa principal:

Para que un sistema de comunicación sea confiable debe de contar con diversos mecanismos que logren evitar cualquier fuente de error. Entre las consideraciones que se tomaron en cuenta a la hora de escribir el protocolo de comunicación se pueden mencionar las siguientes:

- El protocolo debe ser capaz de detectar si se presentó una falla física en la comunicación e indicarle al usuario en qué lugar se encuentra el problema.
- El programa debe reconocer cuando los datos de temperatura presentan algún tipo de error y solicitar que se le envíen los valores correctos.
- Debido a que se está trabajando con un sistema inalámbrico infrarrojo, existe la posibilidad de que un módulo reciba los datos que no le corresponden, por lo que debe existir un mecanismo que elimine esta posibilidad.

#### Interfaz con el usuario:

La interfaz con el usuario se realizó de manera que fuera lo más amigable con el usuario, por lo que se decidió desarrollarlo con las siguientes características:

- La interfaz debe tener la menor cantidad de opciones para evitar cualquier tipo de confusión.
- El programa debe contar con un modo automático de recolección de datos para evitar cualquier intervención del usuario en caso que así se desee.
- El programa debe ofrecerle al usuario un sistema de visualización gráfico de los valores de temperatura para facilitarle al mismo la interpretación de los datos.

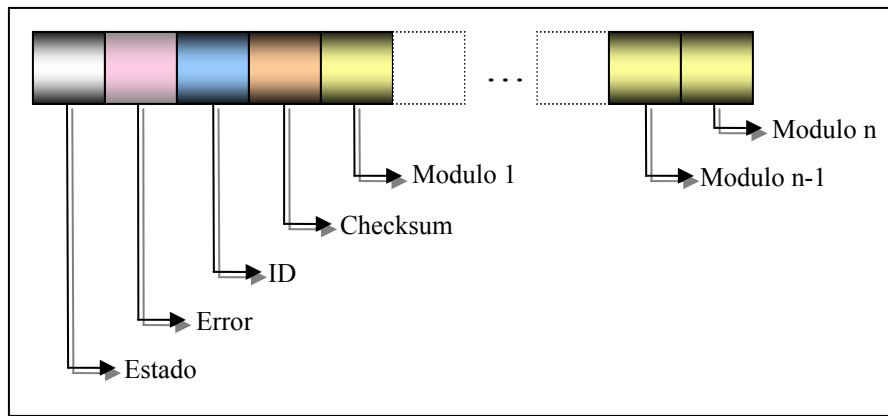
#### Automatización del sistema del sistema de enfriamiento:

- Se deben considerar todos los valores de temperatura de los diferentes módulos para abarcar todas las zonas en estudio.
- Se debe considerar las limitaciones del sistema actual de enfriamiento.
- Se debe utilizar el peor de los casos como referencia de cualquier tipo de cambio en la temperatura que deba realizarse.



## 5.2.2 Descripción del programa principal

### 5.2.2.1 Arreglo de datos



**Figura 5.21** Diagrama del arreglo de datos

El arreglo de datos mostrado en la Figura 5.21, muestra el formato que se decidió implementar para enviar los datos entre módulos. A continuación se describen los espacios que conforman este arreglo:

Estado:

Este campo le indica al microcontrolador que tipo de cadena está recibiendo. Existen tres estados posibles:

- Token: le indica al módulo que existe una solicitud de datos. En este caso el arreglo solamente está conformado por tres espacios (de un 1 byte cada uno): estado, error y ID. Los tokens sólo se envían cuando la dirección del flujo de datos es de izquierda a derecha. Este estado está representado con un valor hexadecimal de 0x30.

- Error: indica que no se logró establecer comunicación entre dos módulos (debido a que no se recibió respuesta después de tres intentos). Cuando se tiene este estado el arreglo está conformado por tres espacios. Los errores solo se envía con una dirección de derecha a izquierda. El estado está representado con un valor hexadecimal de 0x32.
- Cadena: Le indica al módulo que se está enviando un arreglo con valores de temperatura. En este caso el arreglo está conformado mínimo por 5 espacios (de un byte cada uno), el máximo está determinado por la cantidad de módulos. La dirección del flujo de datos para este tipo de estado es de derecha a izquierda. El estado está representado con un valor hexadecimal de 0x31.

#### Error:

En este espacio se guarda el valor del módulo donde se produjo el error, de manera que el programa de la interfaz con el usuario pueda desplegarle la ubicación donde se presentó la falla en la comunicación. En el caso en que se no exista falla, este espacio se encontrará con un valor hexadecimal de 0x00.

#### Identificación (ID):

Este espacio indica de cuál fue el módulo que envió los datos, es decir, la fuente de los datos. Esto le permite al microcontrolador determinar si la información le corresponde o si la debe de ignorar, dándole un mayor nivel de seguridad al sistema.

Cheksum:

En este espacio se guarda el valor de la ecuación matemática para que el módulo que recibe la cadena pueda realizar el mismo cálculo y así poder comparar valores. Esto permite eliminar errores en los valores de temperatura.

La ecuación matemática corresponde a una resta de 256 menos el módulo de la sumatoria de los valores de temperatura entre 256. Para efectos de visualización, le ecuación se presenta a continuación:

$$Checksum = 256 - \text{mod}\left(\frac{\sum \text{valores}_{\text{temperatura}}}{256}\right) \quad (5.1)$$

Módulo n a n-1 (valores de temperatura):

En estos espacios se almacenan los valores de temperatura de cada módulo, por lo tanto el tamaño del arreglo dependerá de la cantidad de módulos que existan.

### 5.2.2.2 Esquema del protocolo implementado

En la Figura 5.22 se observa el esquema general del protocolo de comunicación implementado para establecer la comunicación entre los diversos módulos, utilizando como elemento principal el arreglo de datos mencionado anteriormente (ver sección 5.2.2.1). Utilizando como base el esquema de la Figura 5.22, se puede dividir el flujo en tres bloques:

#### 5.2.2.2.1 Petición de datos

Este bloque está demarcado con las flechas punteadas - - - ->. Básicamente este flujo tiene la función de informarle a todos los módulos que se preparen debido a que se generó una petición de datos. Como se puede observar de la figura, este flujo comienza en el momento en que la interfaz de usuario (PC) solicita la recolección de datos de temperatura al módulo principal. A partir de este punto los arreglos de datos están conformados solamente por tres espacios: el primero que indica que el estado está en modo Token (T), el segundo que se encuentra vacío (con un valor hexadecimal de 0x00) debido a que no se ha presentado ningún error y por último la identificación (ID), el cual cambio de módulo a módulo como es de esperarse.

#### 5.2.2.2.2 Recolección de datos

Una vez que la petición de datos llega hasta el último módulo, el flujo cambia de dirección, por lo que empieza el proceso de recolección de datos, denotado en la figura por las flechas - · - · >. Durante este proceso el estado cambia a Cadena (cadena) y se le agregan los espacios de Checksum (CH) y Temperatura.

El espacio de Cheksum irá cambiando de módulo en módulo según cambien los valores de temperatura y cada módulo agregará un espacio en el arreglo de datos con el valor de temperatura que haya recolectado. De esta manera conforme se acerque el arreglo a la interfaz del usuario (PC), dicha cadena contendrá más espacios.

Cabe destacar que una vez que los datos hayan llegado al módulo principal, los espacios de Error e Identificación son eliminados, ya que el sistema reconoce que no se produjeron errores en ningún módulo y no existe equivocarse a la hora de identificar el módulo principal debido a que la conexión con la PC es cableada.

### 5.2.2.2.3 Generación de error

Esta etapa se encuentra denotada con las flechas  $\text{---}\dots\rightarrow$ . Este proceso solamente se lleva a cabo en el caso de que se produzca un error a la hora de establecer una comunicación entre los módulos. En el diagrama se presenta un ejemplo de una falla de comunicación entre el módulo 2 (M2) y el módulo 3 (M3), en este caso después de haber realizado 3 intentos de entablar comunicación, el módulo 2 devuelve el arreglo de datos al módulo 1 indicando en el espacio de Error que se produjo una falla en 2 (es decir, el último punto donde se alcanzó la comunicación). Una vez que el módulo 1(M1) recibe este arreglo, lo envía al módulo principal y éste último lo envía a la PC.

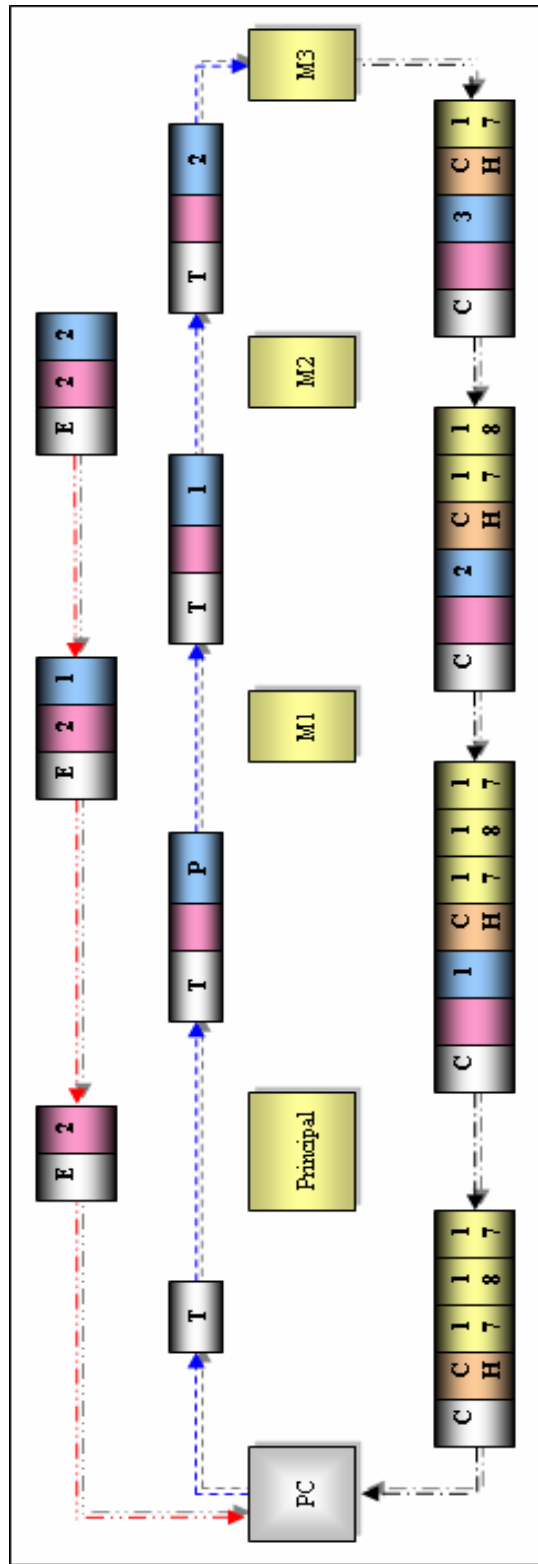


Figura 5.22 Diagrama del flujo de datos del protocolo de comunicación implementado

### **5.2.3 Descripción de la interfaz de usuario**

El programa de interfaz de usuario fue desarrollado en Delphi, el cual es un lenguaje de alto nivel con un enfoque a la orientación de objetos, con un compilador sumamente rápido, tiene gran soporte para creación de bases de datos y cuenta con un gran desarrollo de componentes.

Entre las ventajas que se consideraron para utilizar este lenguaje de programación, se pueden mencionar las siguientes: cuenta con una gran cantidad de componentes que facilitan la creación de base de datos, permite crear interfaces amigables para el usuario de una manera sencilla y permiten un fácil manejo de datos a través del puerto serie.

El programa desarrollado cuenta con las siguientes aplicaciones:

#### Petición de datos (automático/manual)

En el momento en que se enciende el programa, éste automáticamente solicitará datos de temperatura a los diferentes módulos cada 15 minutos. Mediante el botón Obtener Datos, el usuario tiene la potestad de solicitar la información de la temperatura en el momento que desee.

#### Base de datos

En el momento en que se recibe el primer valor de temperatura, se genera automáticamente un archivo Excel que almacena dichos datos clasificados por módulo y por nivel (inferior y superior). Además indica la hora (según el reloj de la computadora) y el día en que fueron tomados cada dato, lo que facilita identificar si algún evento en específico está afectando el funcionamiento normal de los equipos de telecomunicaciones.

### Visualización de datos en 3D

El programa le da la opción al usuario de observar los datos de temperatura de cada módulo por nivel en 3D. La gráfica despliega como máximo tres eventos respecto al tiempo, es decir, se pueden comparar visualmente los valores de temperatura entre un tiempo  $T_1$ ,  $T_2$  y  $T_3$ . Se tomó esta cantidad de eventos como máximo para facilitar la visualización de los datos, ya que se puede tornar confuso el tener muchos valores en una sola gráfica.

#### **5.2.4 Diagramas de Flujo**

Como convención, para las siguientes descripciones, se tomará la dirección del flujo de datos como guía, es decir, si los datos están siendo enviados de izquierda a derecha, el módulo anterior será el que se encuentre a la izquierda y el posterior será el que se encuentra a la derecha del módulo en cuestión y viceversa.

##### **5.2.4.1 Diagrama de flujo del Modulo Principal**

En el momento en que recibe una solicitud de parte de la PC, el microcontrolador se comunica con los primeros dos módulos de cada nivel (en cada bastidor hay dos filas de módulos, uno superior y otro inferior) y les solicita que realicen las transacciones necesarias para obtener todos los valores de temperatura.

Si no se recibe respuesta de la solicitud, se le vuelven a enviar la petición; si al tercer intento no se recibe respuesta, se le avisa a la PC que existe un problema de comunicación.

En el caso de que se reciba respuesta, el microcontrolador se quedará esperando a que se recolecten todos los datos de temperatura y en el momento en que los recibe, verifica si los datos son correctos (ver sección 5.2.2.1 en el apartado de checksum). Si los datos no son los esperados no enviará respuesta,



por lo que el anterior (según convención) interpretará como que los datos no fueron recibidos y volverá a enviar la cadena; por otro lado, si los datos son los esperados enviará respuesta a ambos módulos anteriores (superior e inferior) y además enviará la cadena actualizada de los datos de temperatura.

El diagrama de flujo se puede observar en la Figura 5.23.

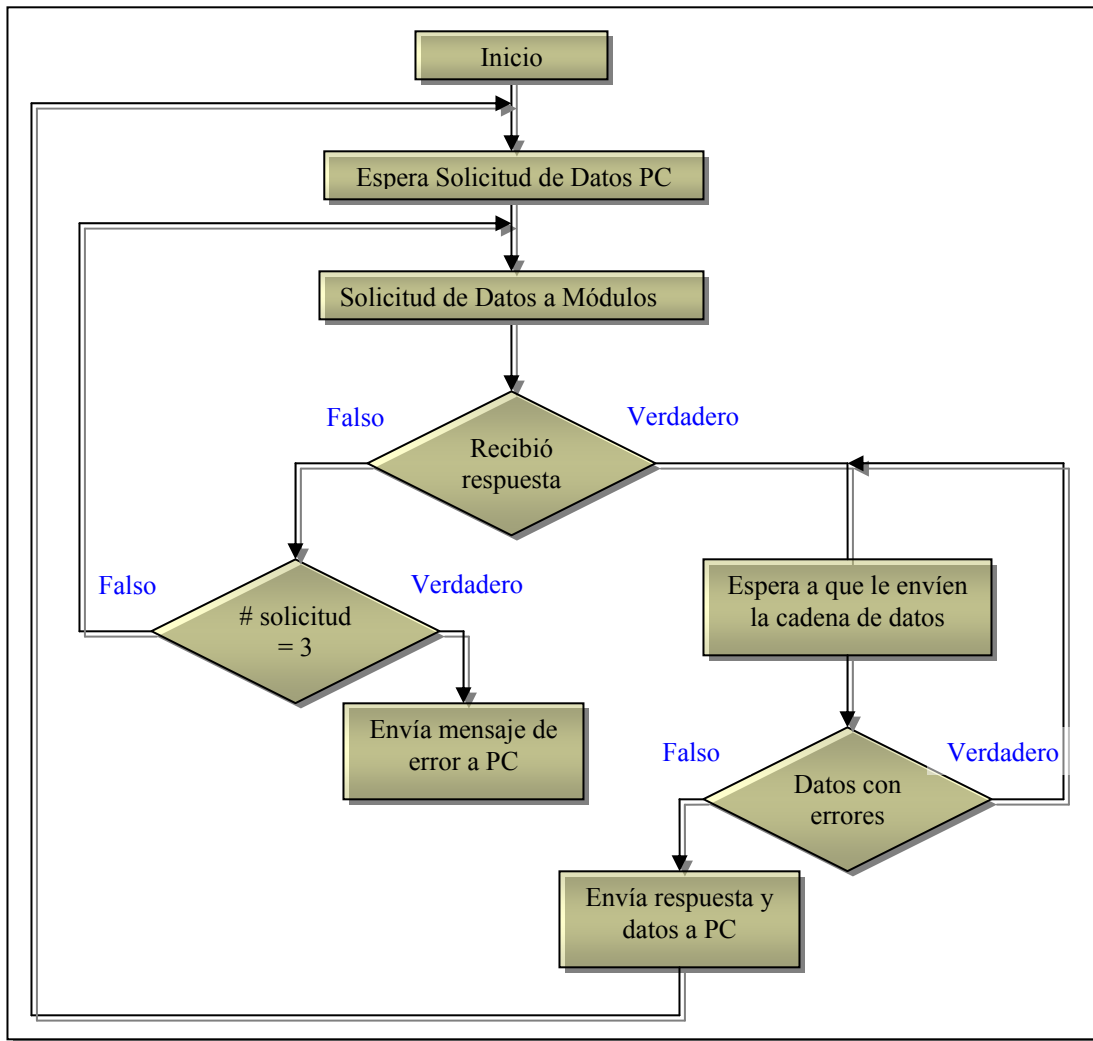


Figura 5.23 Diagrama de flujo del módulo principal

#### **5.2.4.2 Diagrama de flujo del Modulo Intermedio**

Este módulo se encuentra inactivo hasta el momento en que recibe una solicitud válida de datos (token) del módulo anterior y le envía una respuesta informándole que recibió dicha solicitud. Una vez realizada esta acción el microcontrolador desactiva por medio de los elementos tri-state los transceivers que establecen la comunicación con el módulo anterior y habilita los encargados de establecer la comunicación con el módulo posterior.

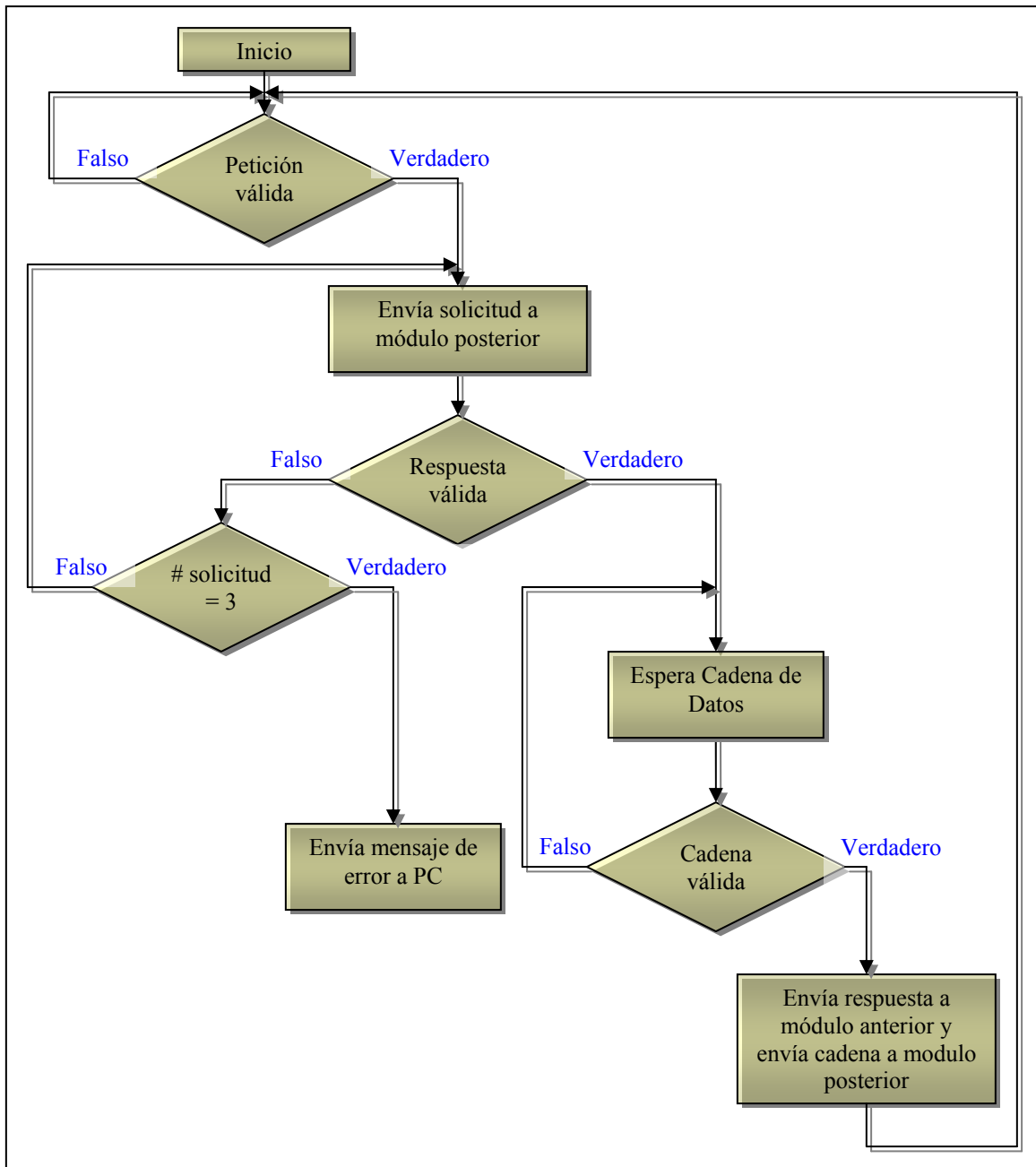
El siguiente paso es actualizar el espacio de identificación (ID) y enviársela al siguiente módulo, del cuál espera respuesta. En caso de que no recibir respuesta, intentará establecer comunicación dos veces más, y de no lograrla, enviará al módulo anterior la información de donde se produjo el error.

Si recibe respuesta, se mantendrá a la espera de la cadena de datos que contiene los valores de temperatura de todos los módulos anteriores (el flujo ahora va de derecha a izquierda, según convención antes descrita) y verificará si estos son correctos mediante la comparación del resultado de una ecuación matemática con el valor del espacio Cheksum de la cadena. Si la comparación es cierta, enviará respuesta al módulo anterior; en el caso de no ser correcta no se enviará respuesta, por lo que el módulo anterior lo interpretará como que la cadena no llegó a su destino y volverá a enviar los datos.

Una vez enviada respuesta, se le agregará a la cadena de datos el valor de temperatura del módulo en cuestión, se le actualizarán los espacios de identificación (ID), Checksum (ecuación matemática) y de estado (token, error o cadena).

Antes de enviar los datos al módulo siguiente, el microcontrolador activará y desactivará los transceivers necesarios (por medio de elementos tri-state) para lograr establecer la comunicación necesaria y enviará la nueva cadena de datos.

El diagrama de flujo se puede observar en la Figura 5.24.



**Figura 5.24** Diagrama de flujo del módulo intermedio

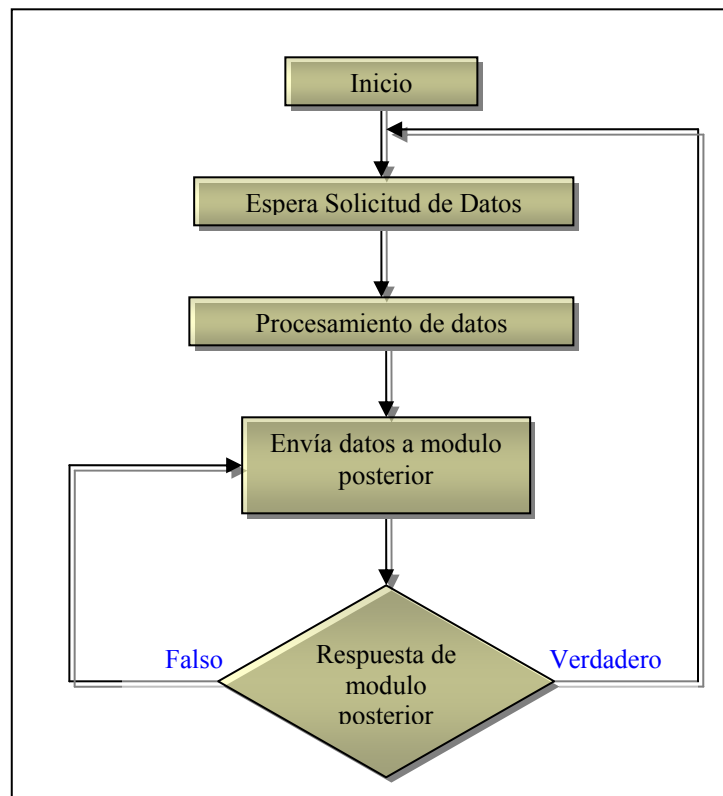
### 5.2.4.3 Diagrama de flujo del modulo final

En el momento en que este módulo recibe la solicitud, la dirección del flujo de datos cambio de sentido, es decir, la dirección de datos será de derecha a izquierda.

Lo primero que realiza es la captura del valor de temperatura, luego utiliza la ecuación matemática para rellenar el campo de Cheksum y finalmente actualiza los espacios de estado (cambia de token a cadena), identificación (ID), y valor de temperatura.

Una vez que tiene la cadena con todos los datos necesarios, esta es enviada al módulo posterior y espera la respuesta de que los datos hayan sido recibidos, en caso contrario, se volverá a enviar la cadena de datos.

El diagrama de flujo se puede observar en la Figura 5.25.



**Figura 5.25** Diagrama de flujo del módulo final

#### **5.2.4.4 Diagrama de flujo del programa de Delphi**

El programa realizado en Delphi le solicita datos al modulo principal de dos formas:

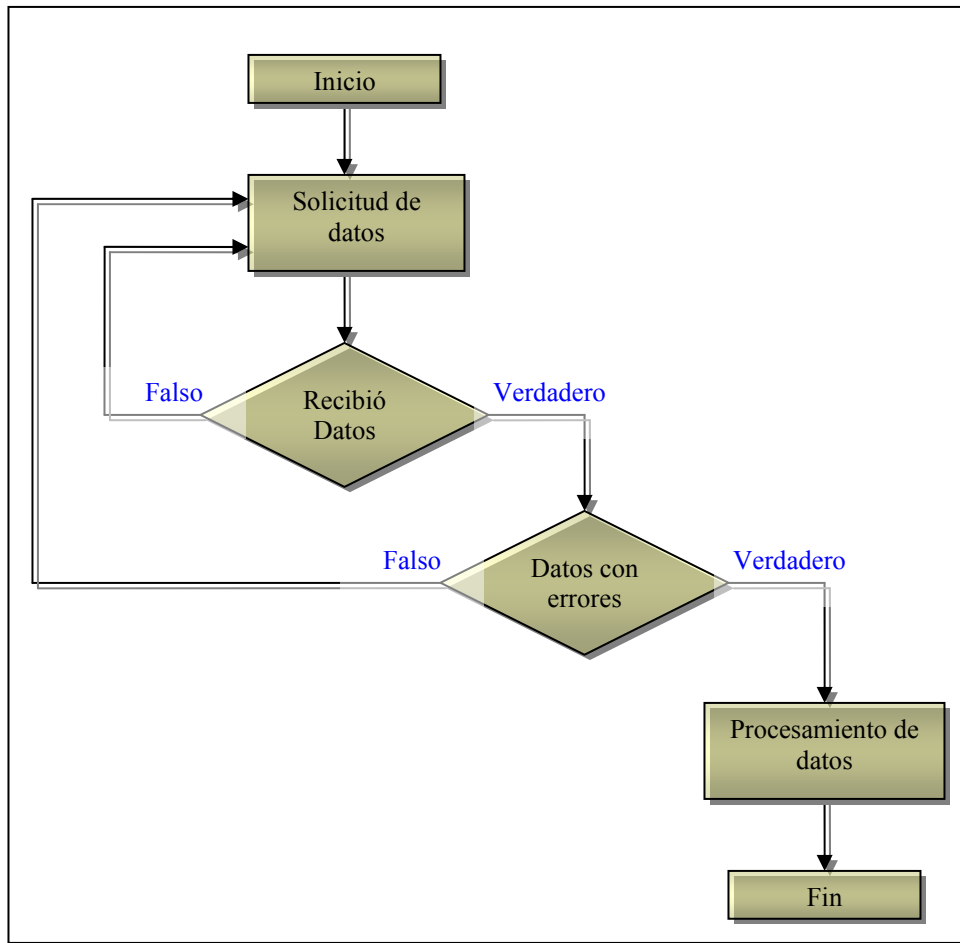
- Si el usuario lo solicita
- Automáticamente cada 15 minutos

La solicitud se realiza mediante el envío de un byte específico el cual será reconocido por el módulo principal.

Una vez enviado, el programa entra en un ciclo de espera para que el sistema recolecte todos los datos necesarios. En caso de que el tiempo designado expire, el programa realizará el mismo procedimiento dos veces más. Si no se recibe respuesta en dichos intentos, se desplegará en el monitor un mensaje de error en la comunicación entre la PC y el módulo principal.

Al recibir respuesta del módulo principal, se verifica si los datos son correctos mediante una operación matemática con los valores de temperatura de cada módulo y la comparación de dicho resultado con el valor guardado en un espacio de la cadena recibida (checksum). Si los valores comparados son diferentes, se vuelven a solicitar los datos; en el caso de que sean iguales, el sistema procesa los datos.

El diagrama de flujo se puede observar en la Figura 5.26.



**Figura 5.26** Diagrama de flujo del programa Delphi

#### **5.2.4.5 Diagrama de flujo del programa de control de temperatura**

Como se observa en el diagrama de flujo de la Figura 5.27 (ver apéndice A.3.2 para el código fuente) el protocolo inicia cuando recibe un valor de temperatura (se evalúan todos los valores de los diferentes módulos), el cual se compara para determinar si este es mayor que el valor actual de temperatura que presenta el sistema de enfriamiento. Si esta comparación es afirmativa se procederá a realizar los siguientes cálculos:

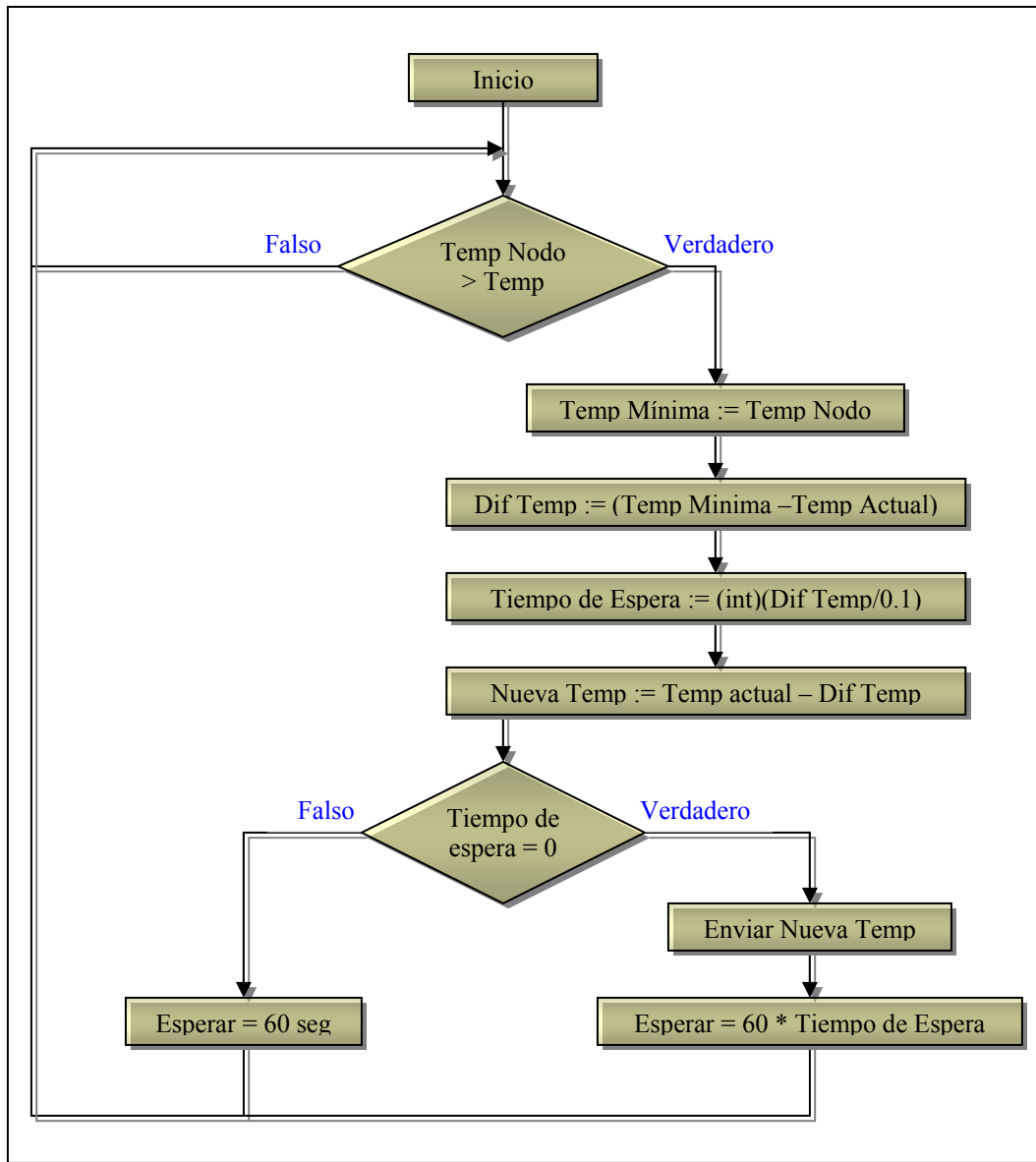
##### **- Diferencia de temperatura:**

Se resta el valor de temperatura del módulo con el valor actual para determinar cuántos grados el sistema de enfriamiento debe de disminuir.

##### **- Tiempo de espera:**

De acuerdo al diseño implementado el tiempo necesario para que el sistema de enfriamiento pueda disminuir un grado centígrado es de 10 minutos (ver sección 6.4), por lo tanto, existe una relación de cambio de 0.1 grados por minuto. Conociendo esta tasa de cambio, se toma la diferencia de temperatura calculada en el punto anterior, y se divide entre 0.1 (se utiliza conversión por casting para obtener valores enteros), para así determinar la cantidad de minutos que el sistema debe de esperar para llegar a la temperatura deseada.

Una vez que se tienen estos valores, se envía al sistema de enfriamiento el valor de temperatura al cual se desea llegar y se espera el tiempo necesario para que logre alcanzarla. Una vez realizada esta secuencia se vuelve a iniciar el ciclo cuando se reciba un nuevo valor de temperatura.



**Figura 5.27** Diagrama de flujo del sistema de control de temperatura



## 5.2.5 Métodos utilizados

### 5.2.5.1 Rutinas de conversión de tipos

Rutinas de redondeo Conversión explícita (Casting):

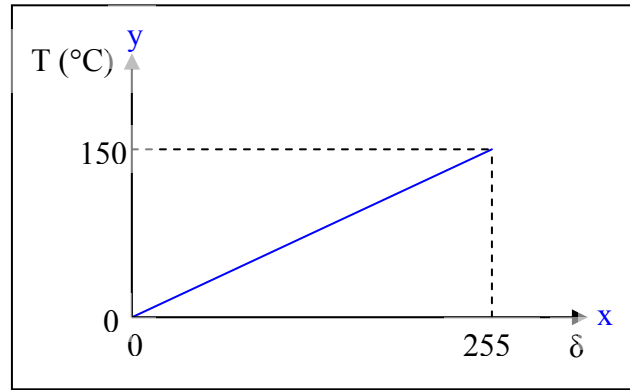
Durante el diseño del algoritmo se presentó la necesidad de procesar una misma variable pero con diferentes tipos de datos. La razón de ello se encuentra en el uso del protocolo RS232, en el cual cada dato es de tipo de string (la computadora reconoce el dato como un valor ASCII), lo que dificulta la manipulación del dato mediante operaciones matemáticas (necesarias para efecto de decodificar el valor real de la temperatura).

Como solución se decidió utilizar el método de conversión por casting, el cual en una sola línea de código logra la conversión del tipo de dato. El formato se muestra a continuación:

```
Var_integer := Integer(Var_string [1]);
```

Dicho formato muestra básicamente que se está “forzando” a la variable de tipo string (de un byte) a que sea almacenada en una variable tipo integer. Por ejemplo, si la variable Var\_string corresponde al carácter ASCII #, el valor almacenado en la variable Var\_integer correspondería a un valor decimal de 35.

### 5.2.5.2 Factor de conversión



**Figura 5.28** Relación entre Temperatura vs Resolución

Mediante la ecuación característica de una recta se obtuvo el factor de conversión necesario para obtener el valor real de temperatura (ver Figura 5.28), donde el eje y (°C) corresponde al rango de temperatura que posee el sensor de temperatura, y el eje x ( $\delta$ ) corresponde al rango de resolución del ADC del microcontrolador. A continuación se describen los cálculos para obtener el factor de conversión:

$$y = mx + b \quad (5.2)$$

$$T = m\delta + b \quad (5.3)$$

$$\text{donde } m = \frac{150 - 0}{255 - 0} = 0.59 \quad \text{y} \quad b = 0 \quad (5.4)$$

$$\Rightarrow T = 0.59 * \delta \quad (5.5)$$

Como ejemplo se podría utilizar el caso en que el valor proveniente de un módulo corresponde al carácter ASCII '\$', cuyo valor entero es 36 (ver sección 5.4.6), utilizando la ecuación anterior se obtendría el siguiente resultado:

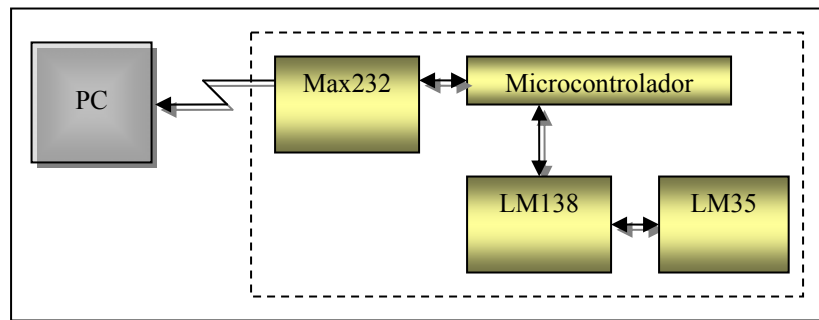
$$T = 0.59 * 36 = 21.24^{\circ}\text{C} \quad (5.6)$$

## Capítulo 6: Análisis de Resultados

### 6.1 Prueba de los componentes principales

#### 6.1.1 Sensores de temperatura

Para probar el correcto funcionamiento del sensor de temperatura LM35 se implementó un circuito de recolección de datos de temperatura conformado por el sensor LM35, el LM138 (etapa de acondicionamiento de señal), el microcontrolador para convertir la señal analógica en digital y enviarla a la computadora y el max232 como interfaz entre el microcontrolador y la computadora. El esquema del circuito se muestra en la Figura 6.1.

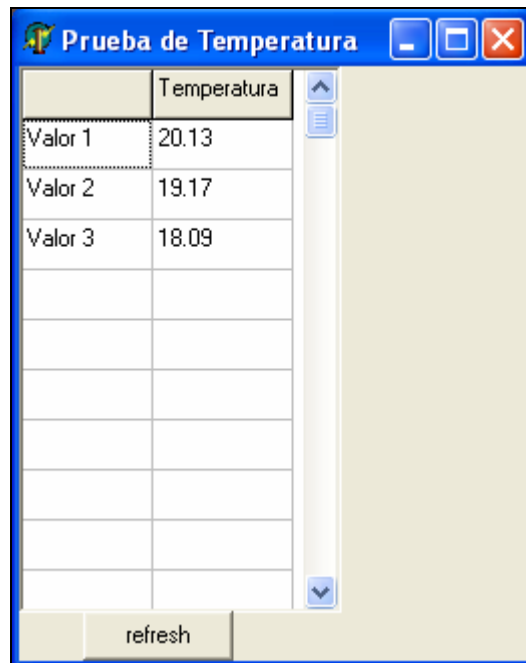


**Figura 6.1** Esquema del circuito de prueba del sensor de temperatura

En la sección 5.1.4.3 se mencionó la necesidad de una etapa de acondicionamiento de señal para acoplar la salida del sensor LM35 con el microcontrolador. En dicha sección se explicaba que el amplificador operacional debía tener un factor de amplificación de 1.95 ya que la resolución del microcontrolador es de 256 bits y es alimentado con 5V. Para lograr tal amplificación se escogió un valor de  $R_1$  de 100K y  $R_2$  de 95K, cuyos valores al ser sustituidos en la ecuación 3.3 descrita en la sección 3.8 del marco teórico, se obtiene como resultado una ganancia de 1.95 (ver ecuación 6.1).

$$G = \frac{R_2}{R_1} + 1 = \frac{95}{100} + 1 = 1.95 \quad (6.1)$$

Respecto al software, se programó el microcontrolador para que capturara el valor proveniente del circuito de recolección de datos por medio de su entrada analógica (A0) y enviara el mismo a la PC a través del max232. Del lado de la PC, se hizo un programa con el lenguaje de programación Delphi, el cual simplemente recibía el dato, utilizaba el factor de conversión (ver apartado 5.2.5.2) para luego desplegar los datos de temperatura en una tabla (ver Figura 6.2).



	Temperatura	
Valor 1	20.13	
Valor 2	19.17	
Valor 3	18.09	

**Figura 6.2** Programa de prueba para visualizar los datos de temperatura

Los valores obtenidos con el circuito de prueba fueron comparados con los valores obtenidos con un multímetro que contaba con un termopar. Este multímetro desplegaba valores de temperatura sin decimales. Los resultados

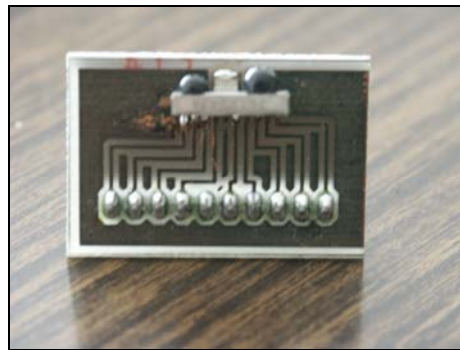
experimentales de la comparación entre los valores medidos con el multímetro y los medidos con el sensor LM35 se encuentran en la tabla 6.1

**Tabla 6.1** Comparación de valores del milímetros vs LM35

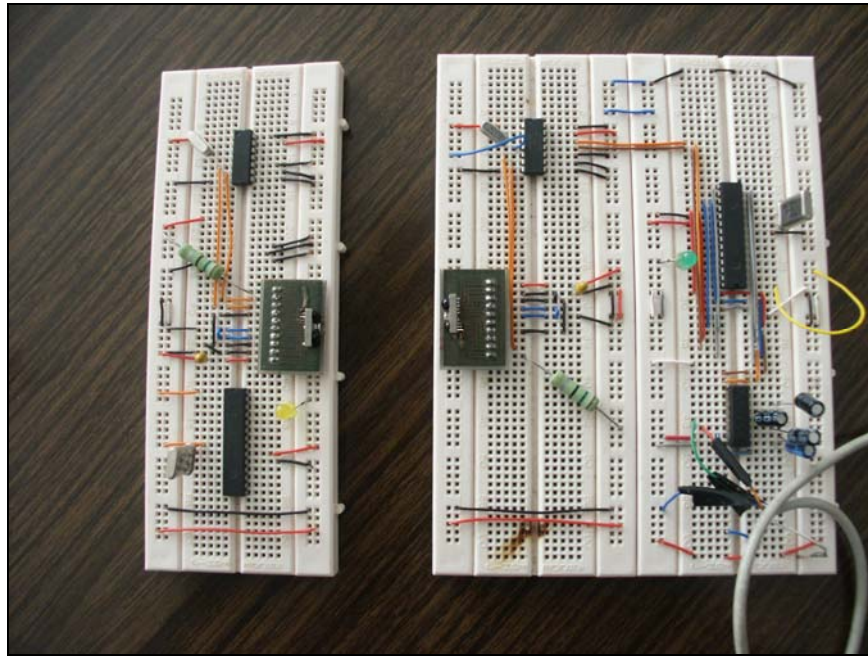
Temperatura medida con el termopar del multímetro (°C)	Temperatura medida con el LM35 (°C)
20	20.13
19	19.17
18	18.09

### 6.1.2 Transceivers infrarrojos

Para comprobar el correcto funcionamiento de los transceivers infrarrojos (montaje de superficie, modelo HSDL3612), se tuvo que implementar un impreso que solo contuviera estos dispositivos (ver figura 6.3) y así realizar pruebas conectándolos a una protoboard con los demás componentes necesarios (ver Figura 6.4 ).

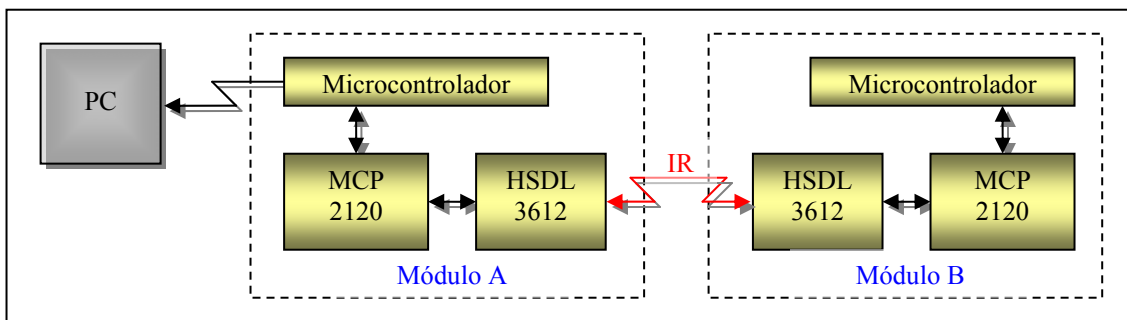


**Figura 6.3** Circuito impreso del sensor HSDL3612



**Figura 6.4** Circuitos de prueba del módulo extremo (izquierda) y del módulo principal (derecha)

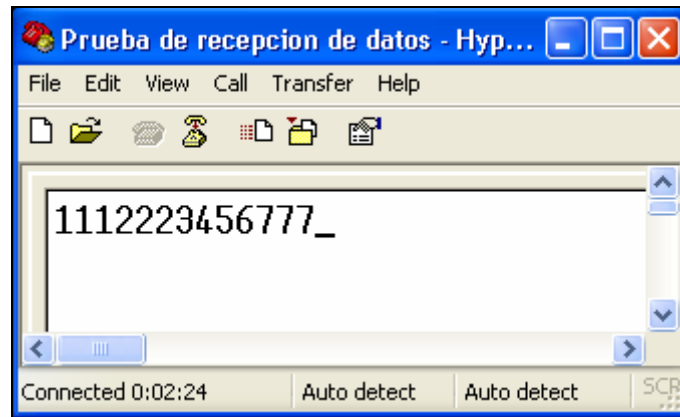
La Figura 6.5 muestra un esquema del circuito de prueba implementado, cuyo objetivo principal es lograr definir si se está logrando establecer la comunicación infrarroja (IR) entre el módulo A y el módulo B.



**Figura 6.5** Esquema del circuito para prueba del HSDL3612

Una vez montado el circuito de prueba se procedió a enviar cadenas de datos (string cuyos valores correspondían a 1112223456777) del módulo B hacia

el módulo A, para que éste último las enviara a la PC y así observar que llegaran los datos correctos (se utilizó la herramienta de comunicación Hyper Terminal de la PC, ver figura 6.6).



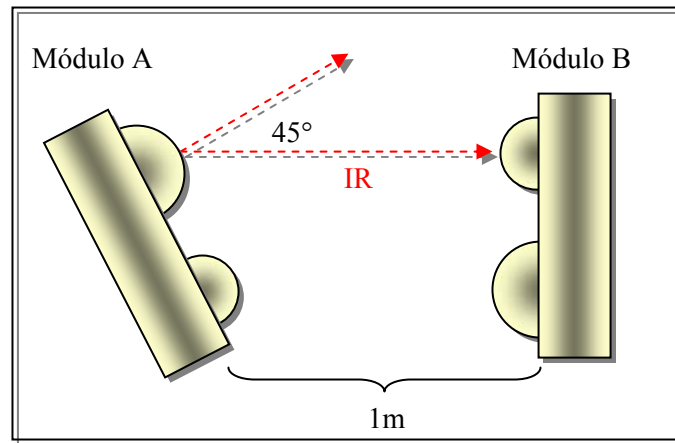
**Figura 6.6** Recepción de datos utilizando la herramienta Hyper Terminal

Se realizó la misma prueba mencionada anteriormente de enviar cadenas de datos de un módulo a otro a diferentes distancias para determinar el alcance del enlace de comunicación. Por medio de esta prueba se definió que el alcance del enlace infrarrojo es de aproximadamente 2m (ver tabla 6.2).

**Tabla 6.2** Valores de distancia de prueba para la recepción de datos entre módulos

Distancia entre módulos (m)	Recepción de datos
0.40	No se presentan errores
0.80	No se presentan errores
1.20	No se presentan errores
1.60	No se presentan errores
1.90	No se logra establecer comunicación

Para determinar el ángulo crítico del enlace de comunicación, se colocaron los módulos a un metro de distancia y se enviaban cadenas de datos a diferentes ángulos (girando el módulo A) hasta observar el punto en que se no lograba establecer la comunicación (ver Figura 6.4). Para un ángulo de aproximadamente 45° se determinó que los módulos no podían establecer la comunicación (ver tabla 6.3).



**Figura 6.4** Esquema del ángulo máximo entre transceivers IR

**Tabla 6.3** Valores de ángulos de prueba para establecer el alcance

Ángulo entre módulos (°)	Recepción de datos
0	No se presentan errores
15	No se presentan errores
30	No se presentan errores
45	No se logra establecer comunicación

Al comparar los resultados con las hojas de datos de los componentes se comprueba el buen funcionamiento del HSDL 3612 y el MCP2120.



### **6.1.3 Microcontrolador Pic16F873**

El correcto funcionamiento de este dispositivo fue comprobado durante las pruebas mencionadas anteriormente. A continuación se presentan los elementos del microcontrolador que fueron probadas:

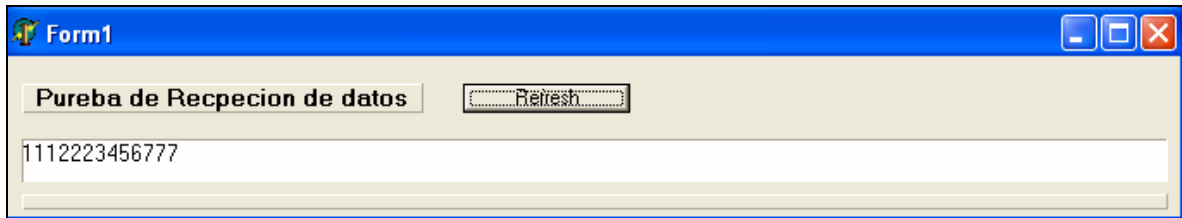
- Convertidor Análogo-Digital (ADC): durante la prueba del sensor de temperatura descrita en la sección 6.1.1, se utilizó el puerto ADC (A0) para capturar los valores provenientes del módulo de recolección de datos. En el momento en que los datos fueron desplegados en la computadora, se comprobó el correcto funcionamiento del ADC, ya que el microcontrolador fue capaz de procesar los datos recolectados.
  
- UART: el correcto funcionamiento de este elemento del microcontrolador fue comprobado tanto en las pruebas del sensor de temperatura (ver sección 6.1.1) como de los sensores infrarrojos (ver sección 6.1.2), en los cuales la comunicación se logró establecer con la PC y entre los módulos respectivamente
  
- Puertos de control: en la sección 5.1.4.2 sobre la descripción del microcontrolador, se mencionó la necesidad de utilizar puertos para configurar diferentes dispositivos y además se describió los pines utilizados para dicho caso. En las pruebas de los sensores infrarrojos, se comprobó el correcto funcionamiento de estos puertos ya una vez configurado el dispositivo MCP2120, se logró establecer la comunicación entre los módulos de prueba.

## 6.2 Comunicación entre microcontrolador y PC

Como primer método se utilizó el cable USB como medio para establecer la comunicación entre el microcontrolador y la PC. Para ello se utilizó el driver conocido como FTDI, el cual crea un puerto virtual, que le permite al usuario configurar el puerto USB como un puerto serial con protocolo RS232.

Durante las pruebas se presentaron problemas de reconocimiento del dispositivo USB por parte de la PC, por lo que había que intentar reconectar el cable en los diversos puertos de la computadora hasta que lo reconociera o se debía de reiniciar la misma. Para verificar que el problema no se encontraba en el sistema operativo, se realizaron pruebas en la versión Windows XP y en Windows 98, obteniendo los mismos resultados. En los casos en que la PC reconocía el puerto USB, el sistema completo funcionaba sin ningún problema. Debido a esta inestabilidad se decidió optar por el uso del puerto serie convencional de la PC.

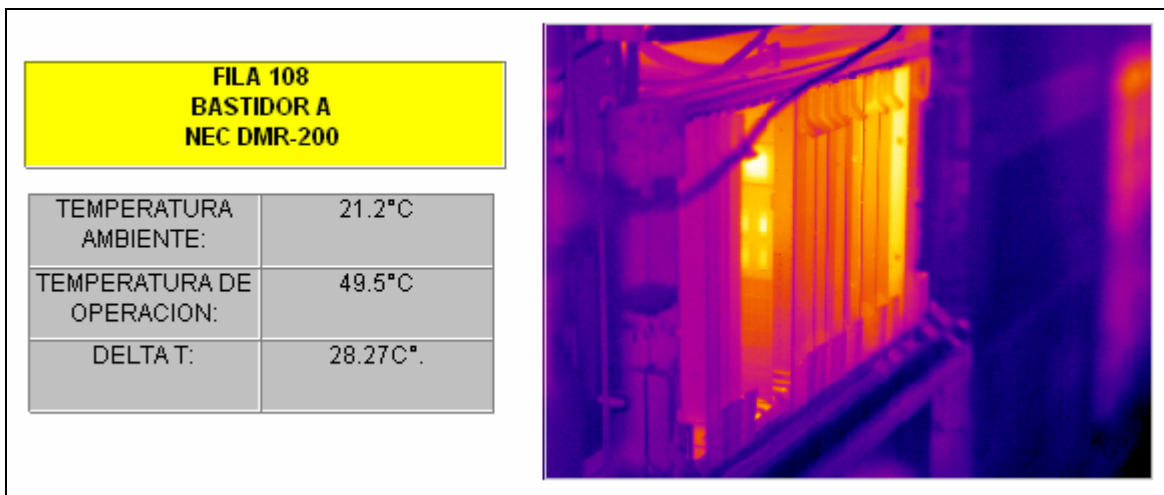
Utilizando tanto la UART de la PC como la del microcontrolador, fue muy sencillo establecer la comunicación serial utilizando el protocolo RS232. El lenguaje de programación Delphi cuenta con una componente que reconoce el puerto serie y permite utilizarlo con unos cuantos comando. De igual manera, el microcontrolador también cuenta con funciones específicas que permiten utilizar el puerto de una manera sencilla. Para probar el correcto funcionamiento de la conexión, se implementó un código en el cual, el usuario digitaba cualquier cantidad de caracteres, estos eran enviados al microcontrolador y este último los devolvía a la PC para que fueran desplegados. En la figura 6.5 se observa una prueba en donde se escribió a través de la computadora una secuencia de caracteres con los valores 1112223456777, los cuales eran enviados al microcontrolador, éste último los devolvía a la PC y eran desplegados en una interfaz realizada en el lenguaje de programación Delphi.



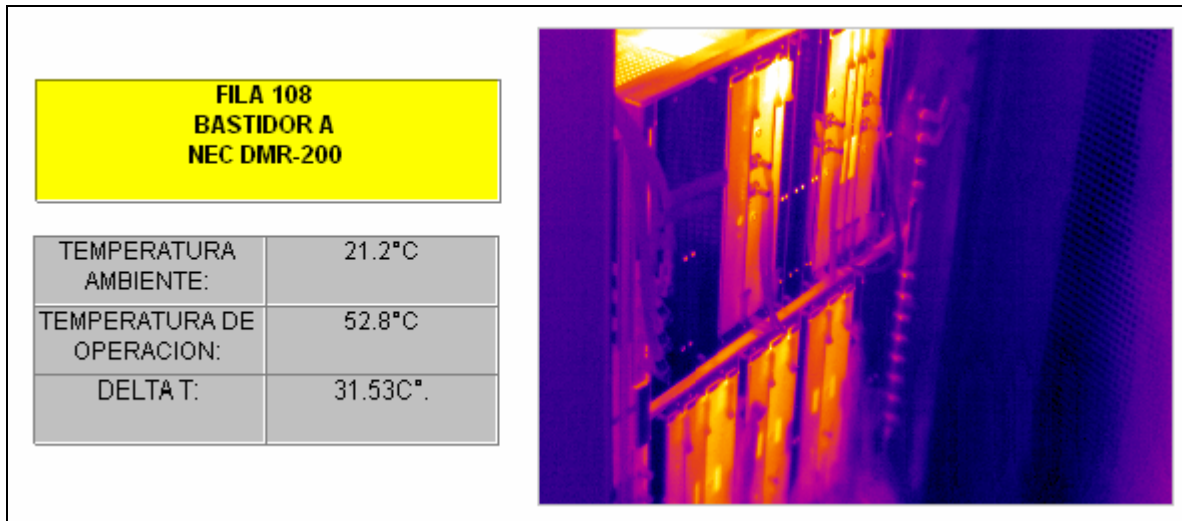
**Figura 6.5.** Interfaz de prueba de comunicación entre PC y microcontrolador

### 6.3 Cuantificación de la temperatura en los equipos

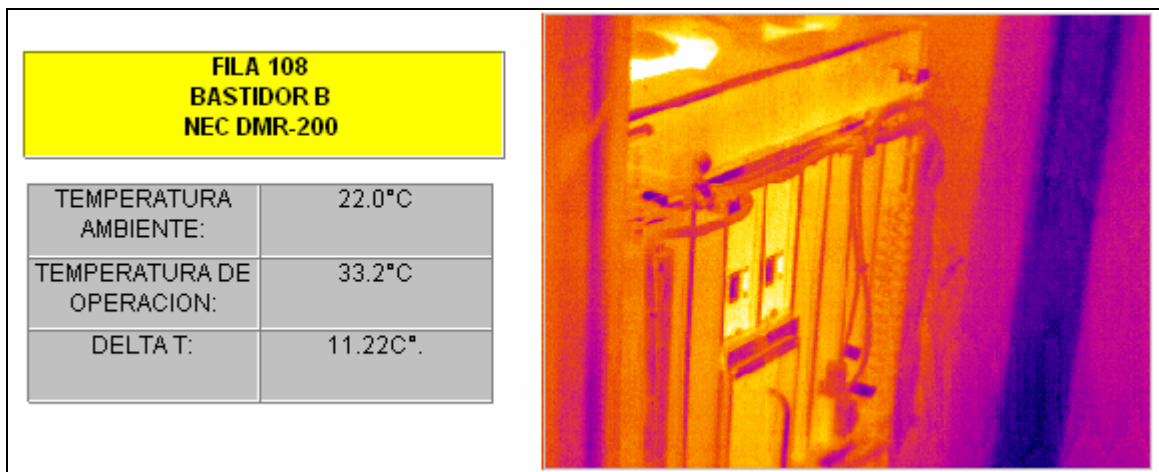
Para determinar la razón por la cuál los equipos se estaban dañando, se utilizó como base un estudio que se realizó mediante el uso de una cámara termográfica, en donde se tomaron fotos de los equipos que se encuentran en la sala en estudio. A continuación se muestran los resultados (solo se presentan unos cuantos resultados para efecto de análisis):



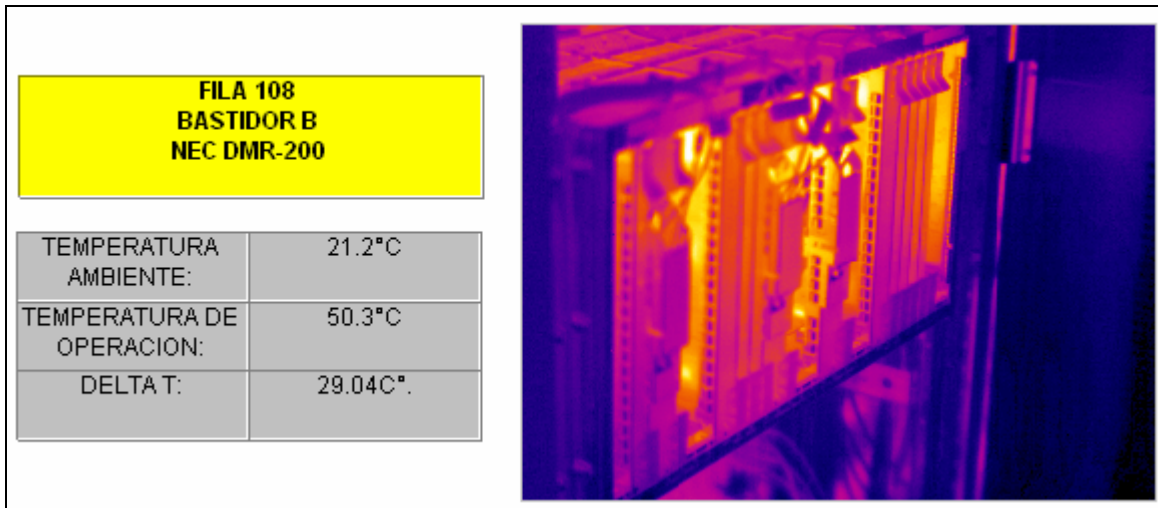
**Figura 6.6** Gráfica térmica del equipo NEC DMR-200 A



**Figura 6.7** Gráfica térmica del equipo NEC DMR-200 B



**Figura 6.8** Gráfica térmica del equipo NEC DMR-200 C



**Figura 6.9** Gráfica térmica del equipo NEC DMR-200 D

En las figuras de la 6.6 a la 6.9 se observan los resultados que se obtuvieron en un estudio termográfico realizado en la sala de telecomunicaciones. Se escogieron estas termográficas debido a que corresponden a equipos que son del mismo modelo (NEC DMR-200) para simplificar el análisis (los demás resultados presentan un comportamiento semejante). Según las especificaciones del fabricante de este equipo, el punto de operación máximo de temperatura debería de encontrarse en 45°C (ver anexo), y al compararlo con los resultados experimentales, se observa que en la mayoría de los casos se sobrepasa. El sistema actual solo cuenta con un sensor general de temperatura que indica la temperatura ambiente, debido a ello se concluyó que el sistema de enfriamiento no cuenta con la cantidad suficiente de sensores para mantener una temperatura uniforme.

#### 6.4 Modelado matemático

Para determinar el comportamiento de temperatura de la sala mediante un modelado matemático, se realizaron mediciones de temperatura utilizando como escalón un cambio temperatura de aproximadamente  $2^{\circ}\text{C}$  (mediante el control remoto del sistema). Es decir, el punto inicial de temperatura se tomó cuando la sala se encontraba a  $22.2^{\circ}\text{C}$  y la lectura final se realizó cuando se encontraba a  $20.3^{\circ}\text{C}$ . En la tabla 6.4 se presentan los tiempos que se tomaron para cada cambio de  $0.1^{\circ}\text{C}$  en la temperatura una vez que se estableció el escalón en el control remoto del aire acondicionado.

Según los valores de la tabla 6.4, actualmente el tiempo necesario para realizar un cambio de  $2^{\circ}\text{C}$  es de aproximadamente 9 minutos (540 segundos).

Mediante el uso de la herramienta Matlab, se puede observar en la Figura 6.10 el comportamiento gráfico de los valores experimentales que se encuentran en la tabla 6.4, además del comportamiento simulado que proporciona dicha herramienta, el cual tiene una correlación 94,5364% con el comportamiento experimental-

Mediante el uso de la herramienta Matlab se obtuvo la función de transferencia del sistema con el comando `ident`, ya que se cuenta con la entrada escalón (de dos grados centígrados) y la salida del sistema (ver figura 6.10).

En el gráfico que se presenta en la Figura 6.11 se observa que la respuesta no presenta sobrepaso máximo y además tiene un comportamiento estable. El error de estado estacionario es de apenas un 1.96%, el cual se considera bastante aceptable.

Debido al comportamiento que presenta el sistema, se puede definir que el único factor que se debería de considerar cambiar sería el tiempo que tarda el sistema en estabilizarse, el cual es de aproximadamente 19.3 minutos. Para modificar este parámetro se debe considerar las limitaciones físicas del sistema actual, ya que cabe la posibilidad de que el sistema esté entregando su máxima

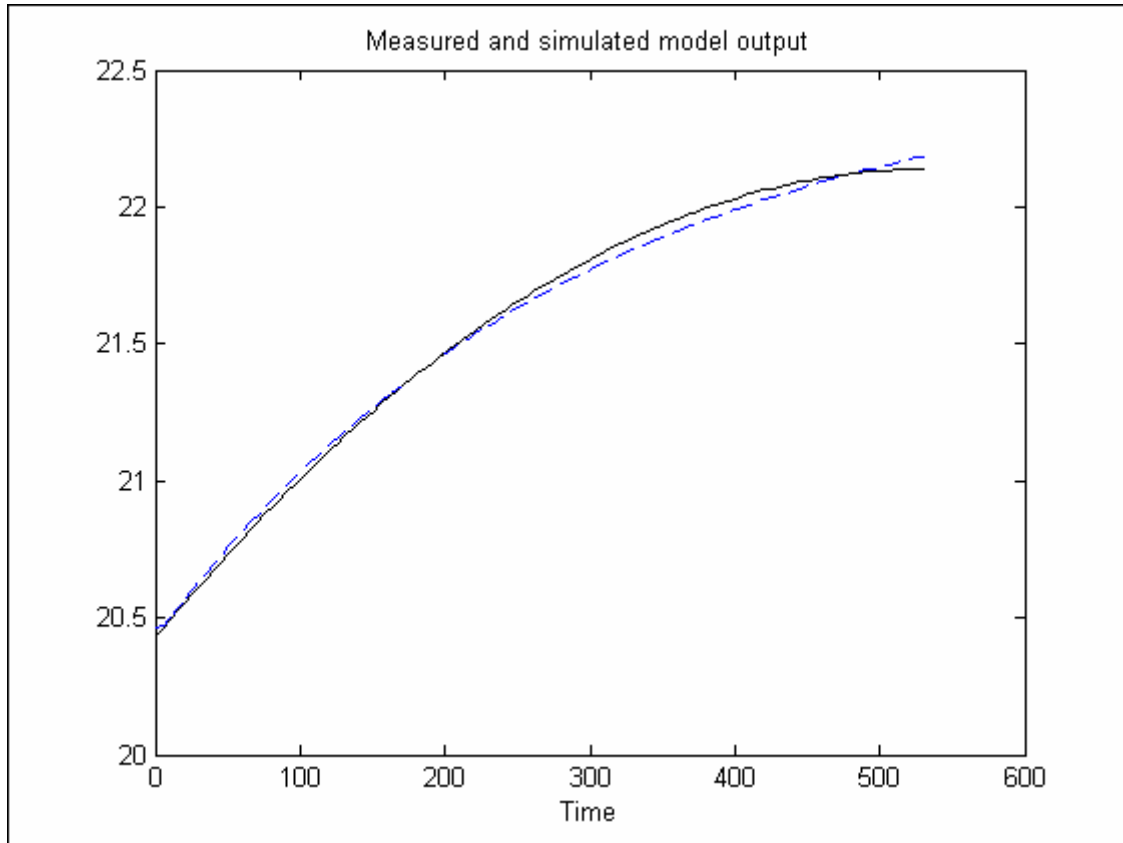
capacidad para enfriar en el momento en que se le solicita un cambio de temperatura.

Como parámetro para el programa que se encarga automatizar el cambio de temperatura cuando se presente una zona con alta temperatura, se consideró el tiempo de estabilización, el cual es de 19.3 minutos para un cambio de dos grados. Por lo que para realizar un cambio de un grado se necesitarían 9.65, el cual al ser redondeado tendríamos aproximadamente 10 minutos por grado centígrado.

**Tabla 6.4** Valores experimentales del tiempo y temperatura del sistema de enfriamiento

Tiempo (segundos)	Temperatura (°C)
0	22.2
12	22.1
17	22
25	21.9
33	21.8
40	21.7
61	21.6
87	21.5
116	21.4
137	21.3
156	21.2
179	21.1
211	21
233	20.9
261	20.8
310	20.7
347	20.6
389	20.5
440	20.4
540	20.3

**- Comportamiento de un cambio de temperatura respecto al tiempo**



**Figura 6.10** Temperatura vs Tiempo

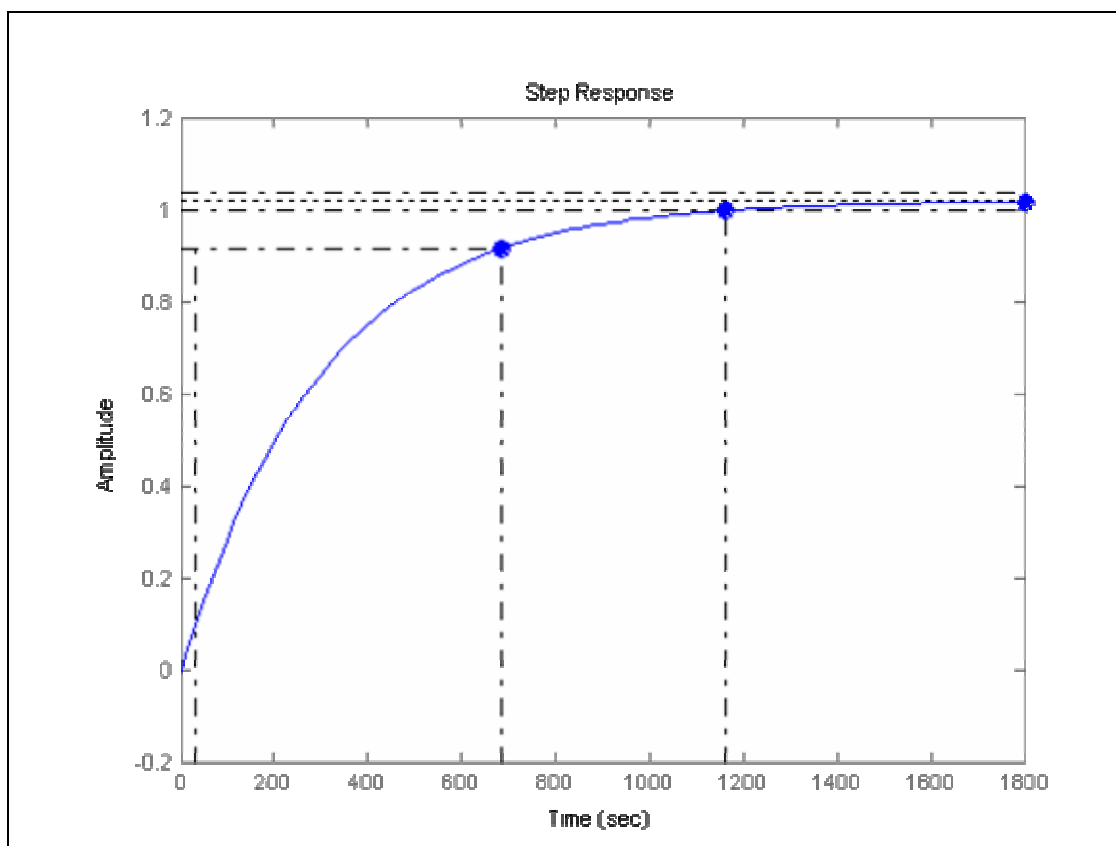
-- Azul punteada: Curva experimental

-- Negra: Curva aproximada por la herramienta Matlab

Porcentaje de Correlación: 94,5364%



### - Respuesta al escalón



**Figura 6.11** Respuesta al escalón

Valores obtenidos utilizando la herramienta Matlab:

- Tiempo de levantamiento: 652s
- Tiempo de asentamiento: 1160 s.
- Valor final: 1.02 con respecto a un escalón unitario
- Error de estado estacionario: 1.96%

## 6.5 Interfaz con el usuario

En la sección 5.2.1 sobre criterios de diseño del software, se mencionó que la interfaz con el usuario debía caracterizarse por su facilidad de interacción con el usuario (es decir, una interfaz amigable). Por esta razón se decidió realizar el programa de manera que fuera lo más “intuitivo” para el usuario, lo cual se logró mediante la automatización del sistema, disminuir la cantidad de opciones al máximo y eliminar el acceso a opciones que no pueden ser solicitadas en un momento específico (se inhabilitan por medio de software).

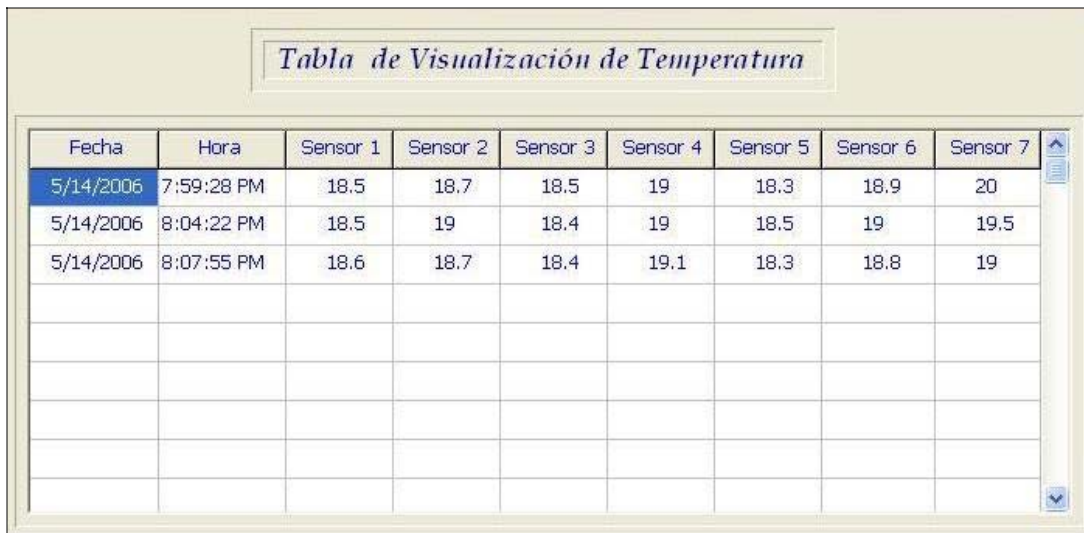
En la Figura 6.12 se puede observar el menú de opciones implementado en el lenguaje de programación Delphi. Esta interfaz le permite al usuario seleccionar el puerto por el cual está conectado el cable serial de la computadora al sistema, si le quiere indicar al sistema que puede comenzar a recolectar los datos de temperatura (de lo contrario lo hará cada 15min automáticamente), las opciones de visualizar los datos en una tabla o en un gráfico 3D, crear un hoja de datos tipo Excel donde se tienen almacenados los valores de temperatura por sensor o buscar ayuda sobre el funcionamiento del programa (manual de usuario).

Las opciones de visualización de datos (tabla o gráfico) y la creación del archivo Excel no pueden ser solicitadas sin antes tener valores de temperatura válidos, por lo tanto estas opciones podrán ser utilizadas hasta 15 segundos después (tiempo necesario para recolectar la información) de haber solicitado los datos de temperatura.



**Figura 6.12** Menú de opciones de la interfaz con el usuario

En la Figura 6.13 se tienen los valores obtenidos de temperatura de los módulos de prueba utilizando el comando de visualizar datos.

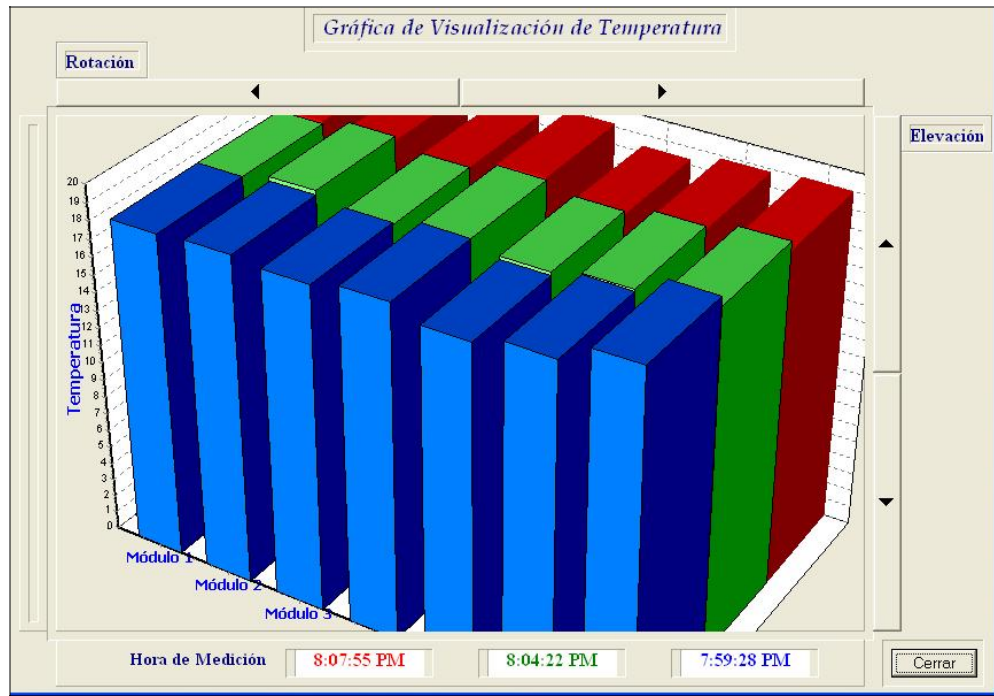


Fecha	Hora	Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7
5/14/2006	7:59:28 PM	18.5	18.7	18.5	19	18.3	18.9	20
5/14/2006	8:04:22 PM	18.5	19	18.4	19	18.5	19	19.5
5/14/2006	8:07:55 PM	18.6	18.7	18.4	19.1	18.3	18.8	19

**Figura 6.13** Tabla con los valores de temperatura capturados por el sistema

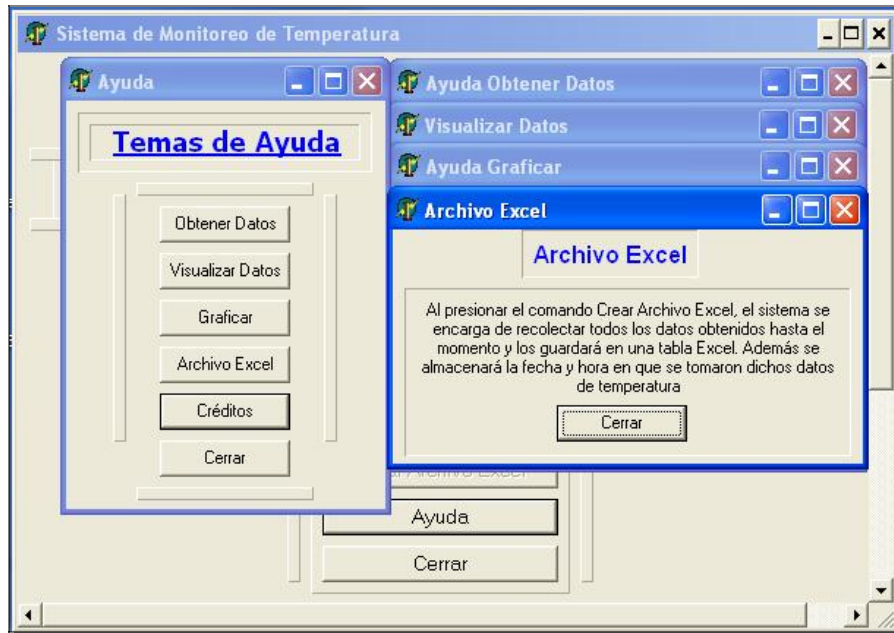
Utilizando el comando de graficar, se pueden visualizar los datos como se muestra en la Figura 6.14 (los valores son los mismos que los de la tabla de la

Figura 6.13), donde además se tienen las opciones de rotar y elevar el gráfico para observar mejor la tendencia de la temperatura a través del tiempo.



**Figura 6.14** Gráfica 3D de los valores capturados por el sistema

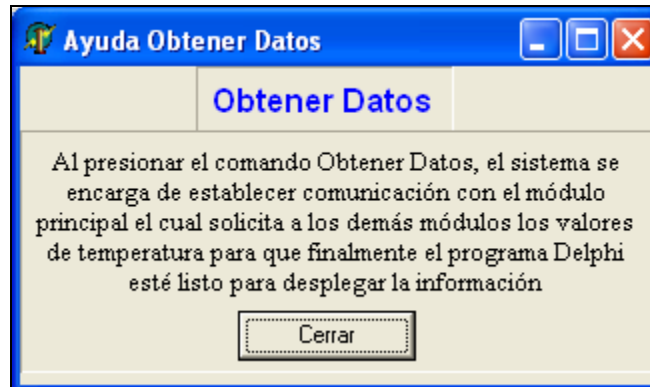
En la Figura 6.15 se observan las opciones que permite el comando de Ayuda, el cual tiene el propósito de brindarle al usuario un manual de cómo utilizar el software desarrollado.



**Figura 6.15** Ayuda del sistema (manual de usuario)

A continuación se presentan las descripciones de cada uno de los comandos de Ayuda:

- Obtener Datos: Al presionar el comando Obtener Datos, el sistema se encarga de establecer la comunicación con el módulo principal el cual solicita a los demás módulos los valores de temperatura para que finalmente el programa Delphi esté listo para desplegar la información (ver figura 6.16).



**Figura 6.16** Comando de ayuda Obtener Datos

- Visualizar Datos: Este comando despliega una tabla con todos los valores tomados desde el momento en que se corrió el programa. Además presenta la hora y fecha en que fueron recolectados los datos.

Debido a que existen dos niveles de sensores, se tienen dos opciones de visualizar datos: inferior y superior, lo cual permite una mayor visualización de los mismos (ver figura 6.17).

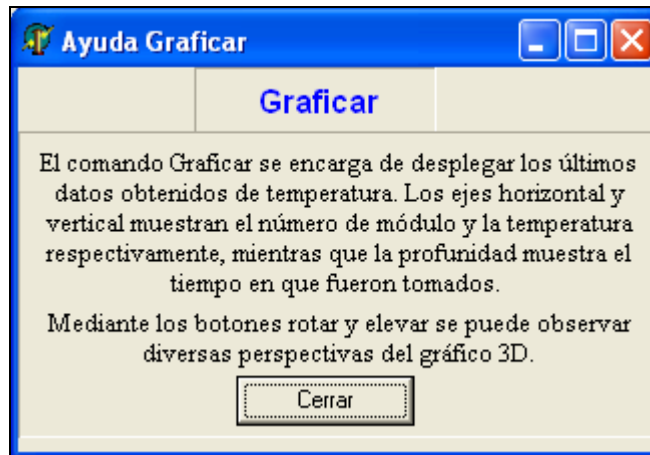


**Figura 6.17** Comando de ayuda Visualizar datos

- Graficar: El comando Graficar se encarga de desplegar los últimos datos obtenidos de temperatura. Los ejes horizontal y vertical muestran el número de

módulo y la temperatura respectivamente, mientras que la profundidad muestra el tiempo en que fueron tomados.

Mediante los botones rotar y elevar se pueden observar diferentes perspectivas del gráfico 3D (ver figura 6.18).



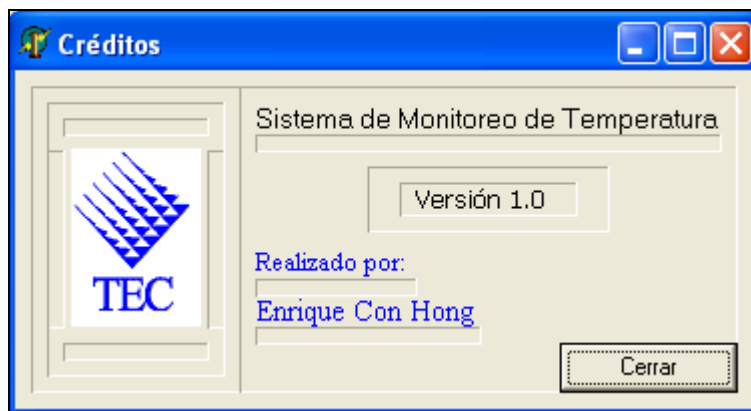
**Figura 6.18** Comando de ayuda Graficar

- Archivo Excel: Al presionar el comando Crear Archivo Excel, el sistema se encarga de recolectar todos los datos obtenidos hasta el momento y los guardará en una tabla tipo Excel. Además se almacenará la fecha y hora en que se tomaron dichos datos de temperatura (ver figura 6.19).



**Figura 6.19** Comando de ayuda Archivo Excel

- Créditos: ver figura 20



**Figura 6.20** Comando de ayuda Créditos



## Capítulo 7: Conclusiones y recomendaciones

### 7.1 Conclusiones

- El pic16F873 requiere de una etapa de acondicionamiento de señal para lograr procesar los datos provenientes del sensor de temperatura LM35.
- Mediante pruebas se determinó que el alcance máximo de los transceivers infrarrojos es de aproximadamente 1.90m.
- Mediante pruebas se determinó que el ángulo de entre los módulos para lograr establecer una comunicación, no debe de sobrepasar los 45°.
- El análisis termográfico demuestra la existencia de un problema por sobrecalentamiento de los equipos de telecomunicaciones y su necesidad de supervisión.
- Mediante el modelado matemático se determinó que el tiempo de levantamiento del sistema es de 652 segundos y el de asentamiento es de 1160 segundos ante un señal escalón (de dos grados centígrados).
- En caso de un error físico, el sistema implementado es capaz de detectar el módulo que presentó el fallo.
- El protocolo implementado permite detectar errores por pérdida de información en el enlace inalámbrico y solicitar su retransmisión en caso necesario.
- Cada módulo tiene la capacidad de identificar si la información recibida debe ser ignorada (en caso que no le corresponda) o si debe ser procesada.
- El software desarrollado como interfaz para el usuario es considerado amigable debido a la escasa interacción que requiere del usuario y a su manejo intuitivo.
- Se optó por el uso de un puerto serial convencional en lugar del puerto USB por las inestabilidades presentadas con el driver FTDI.

## 7.2 Recomendaciones

- Durante las pruebas realizadas a las placas con los circuitos impresos, se presentaron problemas con el sensor de temperatura, los cuales se dañaron a la hora de soldar dicho componente, por que se recomienda tener sumo cuidado de no calentar mucho el dispositivo o utilizar un encapsulado tipo TO-220, el cual cuenta con un disipador de calor.
- Como se mencionó en la sección 6.2, se decidió utilizar un puerto serial como medio de comunicación entre el sistema implementado y la PC debido a que el puerto USB no siempre era reconocido por la PC utilizando el driver FTDI. Por lo tanto, se recomienda buscar un driver más actualizado que no presente tal inestabilidad.
- En el caso de que se desee implementar el programa de automatización de ajuste de temperatura, es necesario considerar las limitaciones físicas del sistema de enfriamiento.

## Bibliografía

- [1] ADR Infor (s.f). *“Casting o transformación de tipos”*.  
Revisado el 7 de enero del 2007, de  
<http://www.adrformacion.com/cursos/java/leccion2/tutorial3.html>
- [2] Coughlin, Robert; Driscoll, Frederick. *Amplificadores Operacionales y Circuitos Integrados Lineales*. 4<sup>ta</sup> Edición. Prentice Hall. México. 1993.
- [3] Kuo, Benjamín. *Sistemas de control automático*. 7<sup>ta</sup> Edición. Prentice Hall. México. 1996.
- [4] Ogata, Katsuhiko. *Ingeniería de Control moderna*. Prentice Hall. Madrid. 1992.
- [5] Rentarías Toledo, Héctor (2004). *“Protocolo RS232”*  
Revisado el 7 de enero del 2007, de  
<http://mx.geocities.com/pcmuseo/mecatronica/puertors232.htm>
- [6] (s.f). *“Puertos infrarrojos”*. Revisado el 7 de enero del 2007, de  
<http://www.tecnotopia.com.mx/mecatronica/puerto-infrarrojo.htm>
- [7] Aguirre, Jorge Eduardo (s.f). *Redes Inalámbricas*  
Revisado el 7 de enero del 2007, de  
[www.monografias.com/trabajos/redesinalam/redesinalam.shtml](http://www.monografias.com/trabajos/redesinalam/redesinalam.shtml)
- [8] Mayné, Jordi (2003). *“Sensores acondicionadores y procesadores de señal”*. Revisado el 7 de enero del 2007, de  
[http://www.bairesrobotics.com.ar/data/Sensores\\_2003.pdf](http://www.bairesrobotics.com.ar/data/Sensores_2003.pdf)
- [9] Gil, S y Rodríguez, E (s.f). Física re-Creativa. *“Termometría, sensores de temperatura”*. Revisado el 7 de enero del 2007, de  
<http://www.fisicarecreativa.com/guias/sensores-temp.pdf>
- [10] (s.f). *“Tipos de sensores de contacto”*.  
Revisado el 7 de enero del 2007, de  
<http://www.temperatures.com/csensors.html>

- [11] Durda, Frank (13 de enero, 1996). *“Transmisión serial UART”*.  
Revisado el 7 de enero del 2007, de  
[http://www.freebsd.org/doc/en\\_US.ISO8859-1/articles/serial-uart/](http://www.freebsd.org/doc/en_US.ISO8859-1/articles/serial-uart/)

## **Apéndices**

## **A.1 Glosario, abreviaturas y simbología**

- ADC: Convertidor Analógico – Digital
- CMOS: Complementary Metal-Oxide Semiconductor
- DCE: Data Communications Equipment
- DTE: Data Terminal Equipment
- ICE: Instituto Costarricense de Electricidad
- ID: identificación
- IR: Infrarrojo
- IrDA: Infrared Developers Association
- ITCR: Instituto Tecnológico de Costa Rica
- Mbps: mega bits por segundo
- NTC: Coeficiente de temperatura negativo
- PC: Personal Computer
- PPM: Pulse Position Modulation (modulación de pulso por posición)
- PTC: Coeficiente de temperatura positivo
- RF: Radio Frecuencia
- RTD: Detector de resistencia por temperatura
- RS232: protocolo utilizado para el control del puerto serial
- RZI: Return-to-Zero Invert (retorno a cero invertido)
- Termometría: Estudio de los sensores de temperatura
- Transceivers: Transmisor/receptor

- UART: Universal Asynchronous Receiver/transmitter (transmisor/receptor sincrónico universal)
- USART: Universal Synchronous-asynchronous Receiver/transmitter (transmisor/receptor sincrónico-asincrónico universal)
- USB: Universal Serial Bus

## **A.2. Información sobre la empresa/institución**

### **A.2.1 Descripción de la empresa**

El Instituto Costarricense de Electricidad (ICE) fue creado por el Decreto - Ley No.449 del 8 de abril de 1949 como una institución autónoma, con personalidad jurídica y patrimonio propio. Está dotado de plena autonomía e independencia administrativa, técnica y financiera. Al ICE le corresponde, por medio de sus empresas, desarrollar, ejecutar, producir y comercializar todo tipo de servicios públicos de electricidad y telecomunicaciones, así como actividades o servicios complementarios a estos.

Como objetivos primarios el ICE debe desarrollar, de manera sostenible, las fuentes productoras de energía existentes en el país y prestar el servicio de electricidad. A su vez, se encarga de desarrollar y prestar los servicios de telecomunicaciones, con el fin de promover el mayor bienestar de los habitantes del país y fortalecer la economía nacional.

Posteriormente, en 1963 y por medio de la Ley No. 3226, la Asamblea Legislativa le confirió al ICE un nuevo objetivo: el establecimiento, mejoramiento, extensión y operación de los servicios de comunicaciones telefónicas, radiotelegráficas y radiotelefónicas en el territorio nacional. Tres años más tarde, instaló las primeras centrales telefónicas automáticas y, a partir de entonces, las telecomunicaciones iniciaron su desarrollo.

Con el devenir del tiempo, ha evolucionado como un grupo corporativo de empresas estatales, integrado por el propio ICE (Sectores Electricidad y Telecomunicaciones) y sus empresas: Radiográfica Costarricense S.A. (RACSA) y la Compañía Nacional de Fuerza y Luz S.A. (CNFL), las cuales han trazado su trayectoria, mediante diversos proyectos de modernización desarrollados en las últimas décadas.



### **A.2.2 Descripción del área donde se realizará el proyecto**

El proceso de Transmisión del ICE pertenece a la llamada UEN (Gestión de Red y Mantenimiento) donde el mismo encaja dentro del nivel de "Gestión de Averías", "Conmutación", "Electromecánica y Civil" y "Planta Eterna", entre otros. A saber el Departamento de Transmisión cuenta con dos instalaciones físicas principales: el Edificio Administrativo, localizado al costado sur del Edificio Central del ICE en San Pedro y el Proceso de Transmisión en el sexto piso del Edificio Central. En el cuarto piso del mismo edificio se le ha asignado un espacio físico para el equipo propio de radioenlaces de baja capacidad, radiotelefonía, microondas, múltiplex de fibra óptica y sincronización y el laboratorio de reparación.

Dentro del amplio margen de actividades y tareas propias del Departamento en cuanto a operaciones y mantenimiento de equipos se enumeran: pruebas de aceptación (del equipo entrante), reparación de averías, mantenimiento preventivo y trabajos especializados, así también como la operación de los equipos que suplen la telefonía fija nacional.

### **A.2.3 Antecedentes prácticos**

En el año 2000 se realizó un proyecto semejante elaborado como proyecto de graduación por el Ing. Alonso Piedra Gutiérrez, el cual se denominó “Diseño e implementación de un sistema de monitoreo de temperaturas para la sala de transmisión del ICE San Pedro”. El Ing. Piedra fue asesorado de parte del Instituto Tecnológico de Costa Rica y de parte de ICE por el Ing. Julio Stradi.

Dicho proyecto no logró alcanzar las expectativas debido a que los módulos que se diseñaron contaban con una conexión alamburada (el problema de ello fue mencionado en el apartado anterior), carencia de escalabilidad en el chasis del prototipo desarrollado y problemas de interfase con el software.

## Apéndice A.3 Programas

### Apéndice A.3.1 Programa genérico de los nodos del sistema

```
include <16F873a.h>
#use delay(clock=20000000)
#fuses HS,NOWDT
#fuses NOPROTECT,NOPUT,NOLVP,NOBROWNOUT,NOCPD,NOWRT
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7)
#use fast_io(A)
#use fast_io(B)
#use fast_io(C)

*****VARIABLES DEL PROGRAMA*****
int cont; //Contador
int CMAX; //Indica un valor maximo de un contador
int i; //Variable contador para enviar CADENA
int suma = 0; //Variable para checksum
int modulo; //Variable para checksum
int checksum; //Variable para checksum
INT cont_A=0;
INT CMAX_A=0;

BYTE TOKEN; //Almacena valor de estado para TOKEN
BYTE TX_REPLY=0x33; //Almacena valor de estado para REPLY
BYTE DATO;
BYTE TEMP; //Almacena valor de temperatura del modulo actual
BYTE ESTADO; //Indica la funcion que se debe de realizar
BYTE COD_ERROR; /*Almacena valor de codigo de error para
identificar modulo con problemas*/
BYTE DATOS[11]; //Cadena con los datos recibidos
/*
DATOS[0] Estado Indica funcion a realizar
DATOS[1] Error Identifica modulo con error
DATOS[2] ID Identifica el modulo q envio datos
DATOS[3] TEM8 Temperatura del modulo 8
DATOS[4] TEM7 Temperatura del modulo 7
DATOS[5] TEM6 Temperatura del modulo 6
DATOS[6] TEM5 Temperatura del modulo 5
DATOS[7] TEM4 Temperatura del modulo 4
DATOS[8] TEM3 Temperatura del modulo 3
DATOS[9] TEM2 Temperatura del modulo 2
DATOS[10] CHECKSUM Almacena codigo de deteccion de error (cde)
*/
BYTE DATOS_A[6];
BYTE DATOS_B[6];
BYTE ESTADO_A;

BOOLEAN REC=FALSE; //Indica que recibio un dato
BOOLEAN TX_TOKEN=FALSE; /*Indica que se debe de transmitir el TOKEN al
modulo siguiente superior*/
BOOLEAN TX_CADENA=FALSE; /*Indica que se debe de transmitir la CADENA al
modulo siguiente superior*/
BOOLEAN TX_ERROR=FALSE; /*Indica que se debe de mandar error al modulo
anterior inferior*/
BOOLEAN TRANS=FALSE; /*Indica que ya se puede empezar el proceso de
transmitir*/
BOOLEAN REPLY=FALSE; //Indica que se recibio respuesta
BOOLEAN GEN_ERROR=FALSE; //Indica que se genero un error
BOOLEAN CHECK_ID=FALSE; /*Indica si el mensaje es para este modulo,
de lo contrario se descarta*/
BOOLEAN CHECK_ERROR=FALSE; //Indica si el operacion matematica coincide con cde
```

Proyecto de Graduación  
Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```

*****IDENTECH*****
*
* NOMBRE DE PROCEDIMIENTO: IDENTECH                                VERSIÓN 7
*
* FUNCION QUE REALIZA: Identifica de qué módulo proviene la información recibida
*
* ENTRADA DE VARIABLES: DATOS[2], TX_CADENA, TX_ERROR, TX_TOKEN
*
* SALIDA DE VARIABLES: CHECK_ID
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                                FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****

void IDENTIFICACION (){

IF (DATOS[2] == 0x33 && (TX_CADENA==TRUE|| TX_ERROR==TRUE)){
CHECK_ID=TRUE;}

ELSE IF (DATOS[2] == 0x31 && TX_TOKEN==TRUE){
CHECK_ID=TRUE;}
}

*****REPLYECH*****
*
* NOMBRE DE PROCEDIMIENTO: REPLYECH                                VERSIÓN 7
*
* FUNCION QUE REALIZA: Espera a que se reciba respuesta una vez que se enviaron los
* datos a un módulo específico para continuar con el siguiente procedimiento, en
* caso de que no reciba respuesta activa una bandera que indica que se produjo una
* falla
*
* ENTRADA DE VARIABLES:TRANS, TX_TOKEN, TX_CADENA, TX_ERROR, REPLY,
*
* SALIDA DE VARIABLES: GEN_ERROR, DATOS[0], DATOS[1], DATOS[2], TX_TOKEN,
* TX_CADENA, TX_ERROR, REPLY
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                                FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****

void esperando_REPLY(){
a=0;
WHILE(a<100){
IF(TRANS){
IF (TX_TOKEN){
IF(REPLY){
REPLY=FALSE;           //Reinica Reply
TX_TOKEN=FALSE;       //Reinica TX_TOKEN
a=200;                //Sale del ciclo de espera
TRANS=FALSE;
}ELSE
DATOS[0] = 0X30;
DATOS[1] = 0X30;
DATOS[2] = 0X32;
FOR (i=0;i<3;++i){
PUTC(DATOS[i]);       //Enviar TOKEN
}
}
}
}
}
}
}

```

Proyecto de Graduación  
 Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```

        DELAY_ms(250);}          //Reenvia TOKEN
    }ELSE IF (TX_CADENA){
        IF(REPLY){
            REPLY=FALSE;
            TX_CADENA=FALSE;
            a=200;
            TRANS=FALSE;
            DELAY_ms(1000);
        }ELSE{
            DELAY_ms(350);
            DATOS[0] = 0x31;
            DATOS[2] = 0x32;
            FOR (i=0;i<11;++i){
                PUTC(DATOS[i]);
                Delay_ms(100);}
        }          //Reenvia CADENA
    }ELSE IF (TX_ERROR){
        IF (REPLY){
            REPLY=FALSE;
            TX_ERROR=FALSE;
            a=200;
            TRANS=FALSE;
        }ELSE
        {
            DATOS[0] = 0x32;
            FOR (i=0;i<3;++i){
                PUTC(DATOS[i]);
                Delay_ms(100);}
        }
    }          //Reenvia ERROR
    DELAY_ms(100);          //
    a=a+1;
    }}
    IF (a >= 100 && a <= 199){
        GEN_ERROR=TRUE;}    /*Se genera codigo de error al no recibir
                             respuesta */
}

*****TxCADECH*****
*
* NOMBRE DE PROCEDIMIENTO: TxCADECH          VERSIÓN 7
*
* FUNCION QUE REALIZA: Actualiza los valores del arreglo conocido como CADENA con
* los valores del módulo en cuestión y los envía al módulo siguiente.
*
* ENTRADA DE VARIABLES: ninguna
*
* SALIDA DE VARIABLES: DATOS[0], DATOS[1], DATOS[2], DATOS[3], DATOS[4]
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong          FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****

void enviar_CADENA(){
    DATOS[0] = 0x31;          //Estado: Cadena
    DATOS[1] = 0x30;          //No afecta porque no se ha producido error
    DATOS[2] = 0x32;          //Le asigno el ID del modulo
    DATOS[4] = READ_ADC();    //Inserta valor de temperatura
    FOR (i=0;i<11;++i){
        PUTC(DATOS[i]);          //ENVIAR TODA LA CADENA
        DELAY_ms(250);}
}

```

Proyecto de Graduación  
 Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```
*****TOKENECH*****
*
* NOMBRE DE PROCEDIMIENTO: TOKENECH                VERSIÓN 7
*
* FUNCION QUE REALIZA: Actualiza los valores del arreglo conocido como TOKEN con
* los valores del módulo en cuestión y los envía al módulo siguiente.
*
* ENTRADA DE VARIABLES: ninguna
*
* SALIDA DE VARIABLES: DATOS[0], DATOS[1], DATOS[2]
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                 FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*****
```

```
VOID enviar_TOKEN(){
DATOS[0] = 0X30;
DATOS[1] = 0X30;
DATOS[2] = 0X32;
FOR (i=0;i<3;++i){
    PUTC(DATOS[i]);           //Envia TOKEN
    DELAY_ms(250);}}
*****
```

```
*****ERRORDECH*****
*
* NOMBRE DE PROCEDIMIENTO: ERRORECH                VERSIÓN 7
*
* FUNCION QUE REALIZA: Actualiza los valores del arreglo conocido como ERROR con
* los valores del módulo en cuestión y los envía al módulo siguiente.
*
* ENTRADA DE VARIABLES: ninguna
*
* SALIDA DE VARIABLES: DATOS[0], DATOS[2]
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                 FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*****
```

```
VOID enviar_ERROR(){
DATOS[0] = 0x32;
DATOS[2] = 0x33;
FOR (i=0;i<11;++i){
    PUTC(DATOS[i]);
    DELAY_ms(250);}}
*****
```

Proyecto de Graduación  
 Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```

*****REPLYECH*****
*
* NOMBRE DE PROCEDIMIENTO: TxCADECH                VERSIÓN 7
*
* FUNCION QUE REALIZA: Envía el valor 0x33 definido como el valor que indica que
* se recibieron los datos correctamente
*
* ENTRADA DE VARIABLES: ninguna
*
* SALIDA DE VARIABLES: TX_REPLY
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                 FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****
VOID enviar_REPLY(){
PUTC(TX_REPLY);
DELAY_ms(250);}

*****LED__ECH*****
*
* NOMBRE DE PROCEDIMIENTO: LED__ECH                VERSIÓN 7
*
* FUNCION QUE REALIZA: enciende y apaga un led del puerto B4 para indicar que el
* módulo se encuentra energizado
*
* ENTRADA DE VARIABLES: ninguna
*
* SALIDA DE VARIABLES: ninguna
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                 FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****
VOID BLINK_A(){
    OUTPUT_HIGH(PIN_B4);
    DELAY_ms(300);
    OUTPUT_LOW(PIN_B4);
    DELAY_ms(300);}

*****INTERECH*****
*
* NOMBRE DE PROCEDIMIENTO: INTERECH                VERSIÓN 7
*
* FUNCION QUE REALIZA: Interrupción que se realiza en el momento en que se reciben
* datos de algún módulo y define que función debe de realizarse. Además se realiza
* la operación checksum para ver si los datos no presentan ningún error
*
* ENTRADA DE VARIABLES: DATOS []
*
* SALIDA DE VARIABLES: REC, TX_TOKEN, TX_CADENA, TX_ERROR, REPLY, CHECK_ERROR
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                 FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****
    
```

Proyecto de Graduación  
Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```
#int_rda
rda_isr() {

    DATOS[cont]=GETC();           //Captura dato entrante
    enable_interrupts(INT_RDA);
    enable_interrupts(global);   //Habilita interrupciones
    ESTADO=DATOS[0];            /*Copia primer dato de la trama (identifica
                                funcion)*/

    IF(ESTADO==0x30){           //Indica q debe de enviar el TOKEN al modulo 4
        REC=TRUE;
        TX_TOKEN=TRUE;
        CMAX=3;
        REPLY=FALSE;

    }ELSE IF (ESTADO==0x31){   //Indica q se debe enviar cadena al modulo 2
        REC=TRUE;
        TX_CADENA=TRUE;
        CMAX=11;
        REPLY=FALSE;

    }ELSE IF (ESTADO==0x32){   //Indica q se debe enviar error al modulo 2
        REC=TRUE;
        TX_ERROR=TRUE;
        CMAX=11;
        REPLY=FALSE;

    }ELSE IF (ESTADO==0x33 && (TX_TOKEN || TX_CADENA || TX_ERROR)){ /*Indica
                                                                    q recibió reply del módulo 2 o 4*/
        REPLY=TRUE;
        CONT=0;
        REC=FALSE;

    }ELSE{
        REC=FALSE;
    }
}
IF(REC){
    CONT=CONT+1;
    IF(CONT==CMAX){

        IF (TX_CADENA==TRUE && REPLY==FALSE){
            for (i = 3; i < 10; i++ )
                {suma += DATOS[i]};

            modulo = suma % 256;
            checksum = (256 - modulo);
            suma=0;
            modulo=0;}

        IF (checksum == DATOS[10]){
            CHECK_ERROR=TRUE;}

        TRANS=TRUE;
        CONT=0;

    }
}
}
} //end int_rda
```



Proyecto de Graduación  
 Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```
*****MAINECH*****
*
* NOMBRE DE PROCEDIMIENTO: MAINECH                VERSIÓN 7
*
* FUNCION QUE REALIZA: procedimiento principal en donde se configuran los
* dispositivos periféricos, identifica las funciones que deben de realizarse y
* a dichas funciones hace un llamado a los procedimientos correspondientes
*
* ENTRADA DE VARIABLES: TRANS, TX_CADENA, TX_TOKEN, TX_ERROR, REPLY, CHECK_ERROR
* CHECK_ID, GEN_ERROR
*
* SALIDA DE VARIABLES: REC, TX_TOKEN, TX_CADENA, TX_ERROR, REPLY, CHECK_ID, DATOS[0],
* DATOS[1], DATOS[2], GEN_ERROR
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: BLINK_A, IDENTIFICACION(), enviar_REPLY(),
* enviar_TOKEN, esperando_REPLY(), enviar_CADENA(), enviar_ERROR, READ_ADC()
*
* REALIZADO POR: Enrique Con Hong                FECHA: 23/02/06
*
* ULTIMA MODIFICACION: 14/4/06
*
*****
```

```
void main() {

    setup_adc_ports(RA0_ANALOG);
    setup_adc(ADC_CLOCK_INTERNAL);
    set_adc_channel(0);
    setup_spi(FALSE);
    setup_counters(RTCC_INTERNAL,RTCC_DIV_2);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_ccp1(CCP_OFF);
    setup_ccp2(CCP_OFF);
    enable_interrupts(INT_RDA);
    enable_interrupts(global);
    SET_TRIS_B(0x00);
    SET_TRIS_C(0x80);

//CONFIGURA EL MCP2120
    OUTPUT_HIGH(PIN_C4);
    DELAY_ms(50);
    OUTPUT_LOW(PIN_C5);
    DELAY_ms(50);
    OUTPUT_HIGH(PIN_B5);
    DELAY_ms(50);

//ACTIVA SENSOR 2
    OUTPUT_LOW(PIN_B0);
    OUTPUT_LOW(PIN_B1);
//DESACTIVA SENSOR 1
    OUTPUT_HIGH(PIN_B2);
    OUTPUT_HIGH(PIN_B3);

    DELAY_ms(100);

    WHILE (TRUE) {
        BLINK_A();

        IF(TRANS){
            IF (TX_CADENA==TRUE && REPLY==FALSE && CHECK_ERROR==FALSE){
                TRANS=FALSE;
                TX_CADENA=FALSE;
                Delay_ms(250);           //Llega cadena pero hay error en checksum, espera datos
                IDENTIFICACION();
            }
        }
    }
}
```

Proyecto de Graduación  
Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```
IF ((TX_TOKEN==TRUE && REPLY==FALSE && CHECK_ID==TRUE) || (TX_CADENA==TRUE &&
REPLY==FALSE && CHECK_ERROR==TRUE && CHECK_ID==TRUE) || (TX_ERROR==TRUE &&
REPLY==FALSE && CHECK_ID==TRUE)){
    enviar_REPLY();
    //ACTIVA SENSOR 1
    OUTPUT_LOW(PIN_B2);
    OUTPUT_LOW(PIN_B3);
    //DESACTIVA SENSOR 2
    OUTPUT_HIGH(PIN_B0);
    OUTPUT_HIGH(PIN_B1);
}

IF(CHECK_ID==TRUE && TX_TOKEN==TRUE){
    enviar_TOKEN();
    esperando_REPLY();}

}ELSE IF(CHECK_ID==TRUE && TX_CADENA==TRUE){
TEMP = READ_ADC(); // Almacena valor de temperatura dado por el sensor
DATOS[8] = TEMP; // Le asigno el valor de temperatura del modulo 3
for (i = 3; i < 10; i++)
    {suma += DATOS[i];};
    modulo = (suma % 256);
    checksum = (256 - modulo);
    DATOS[10]=checksum;
    suma=0;
    modulo=0;
//ACTIVA SENSOR 2
    OUTPUT_LOW(PIN_B0);
    OUTPUT_LOW(PIN_B1);
//DESACTIVA SENSOR 1
    OUTPUT_HIGH(PIN_B2);
    OUTPUT_HIGH(PIN_B3);
    enviar_CADENA();
    esperando_REPLY();}

ELSE IF(CHECK_ID==TRUE && TX_ERROR==TRUE){
//ACTIVA SENSOR 2
    OUTPUT_LOW(PIN_B0);
    OUTPUT_LOW(PIN_B1);
//DESACTIVA SENSOR 1
    OUTPUT_HIGH(PIN_B2);
    OUTPUT_HIGH(PIN_B3);
    enviar_ERROR();
    esperando_REPLY();
}
    IF (GEN_ERROR){
//ACTIVA SENSOR 2
        OUTPUT_LOW(PIN_B0);
        OUTPUT_LOW(PIN_B1);
//DESACTIVA SENSOR 1, derecha, del impreso si el reg esta arriba
        OUTPUT_HIGH(PIN_B2);
        OUTPUT_HIGH(PIN_B3);
        DATOS[0] = 0x32; //Indica en estado de que hubo un error
        DATOS[1] = 0x32; //Indica que el error se dio en el modulo 2
        DATOS[2] = 0x32; //ID de la fuente de la CADENA
        FOR (i=0;i<3;++i){
            PUTC(DATOS[i]);}
        TRANS=FALSE;
        CHECK_ID=FALSE;
        GEN_ERROR=FALSE;
    }
}
}
}
```

### Apéndice A.3.2 Programa de control de temperatura

```
*****MAINECH*****
*
* NOMBRE DE PROCEDIMIENTO: MAINECH                                VERSIÓN 7
*
* FUNCION QUE REALIZA: Toma cada uno de los valores de temperatura de los módulos y
* los compara con el valor de temperatura del sistema de enfriamiento, en caso de
* de que este ultimo sea mayor, se tomar el valor de la diferencia y el tiempo
* necesario para equiparar ambos valores y son enviados al sistema de enfriamiento
*
* ENTRADA DE VARIABLES: valor_del_sistema, temp[],
*
* SALIDA DE VARIABLES: timer, nueva_temp
*
* PROCEDIMIENTOS QUE SON EMBOCADOS: ninguno
*
* REALIZADO POR: Enrique Con Hong                                FECHA: 21/5/06
*
* ULTIMA MODIFICACION: 21/5/06
*
*****

while (true){
valor_minimo:=0;

for i:=0 to i == 18 {
    if (temp[i] > valor_del_sistema) then {
        Ajustar_temp := true;
        if (temp[i] > valor_minimo){
            valor_minimo := temp[i];
        }
    }
}

if (Ajustar_temp) {
    diferencia_de_temp := (valor_minimo - valor_del_sistema);
    Ajustar_calculos := true;
    Ajustar_temp := false;
}

if (Ajustar_calculos) {
    tiempo_de_espera=(int)(diferencia_de_temp/0.1);           // Obtiene cambios de
0.1C necesarios para                                       temperatura
deseada
    nueva_temp := valor_del_sistema - diferencia_de_temp;     // Se obtiene
valor de temperatura deseado
    Ajustar_tiempo := true;
    Realizar_calculos:=false;
}

if (Ajustar_tiempo) {
    if (tiempo_de_espera==0){
        timer:=60}                                           // Se asigna un timer que espera
un minuto                                                  para el cambio de
temperatura, en este caso                                   seria un
cambio menor a 0.1C
    else {
        timer:=tiempo_de_espera*60}                          // Se asigna un timer que espera
un minuto por cada 0.1C que haya q ajustar
}
```

## Proyecto de Graduación

### Desarrollo de un sistema de monitoreo para la sala de equipos de telecomunicaciones del ICE

```
        Ajustar_tiempo := false;
        Activar_timer := true;
        Send := nueva_temp;           // Envía nuevo valor de
temperatura al                       sistema de
aire acondicionado
    }

    if (Activar_timer){
        wait:= timer}
    }                               // END del while (true)
```

## Apéndice A.4 Circuitos impresos

### Apéndice A.4.1 Nodo principal

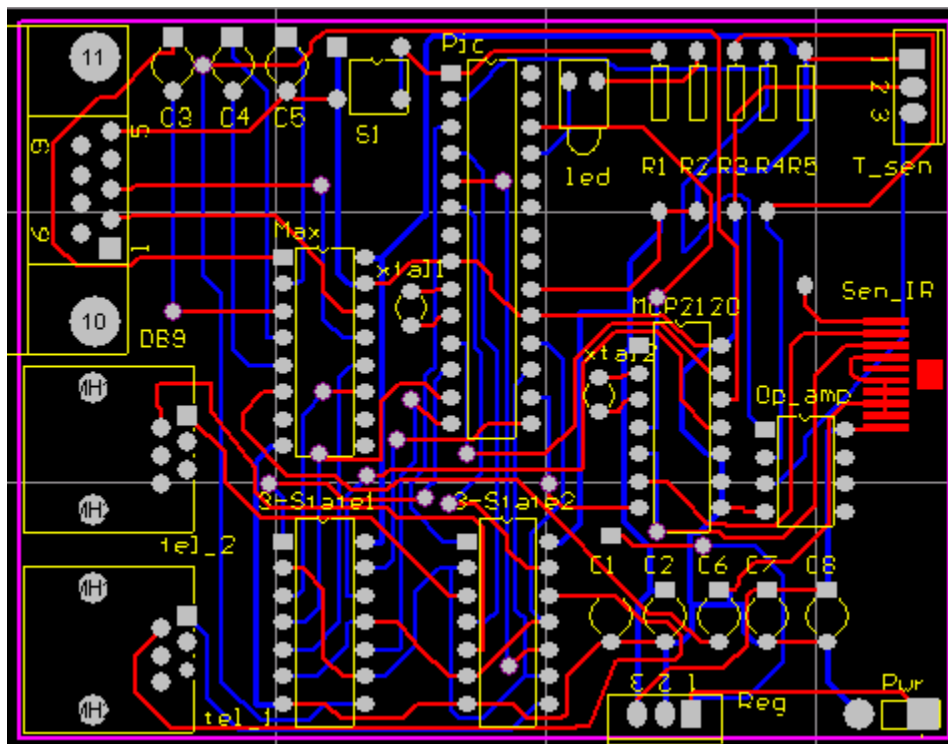


Figura A.4.1 Impreso del nodo principal

### Apéndice A.4.2 Nodo intermedio

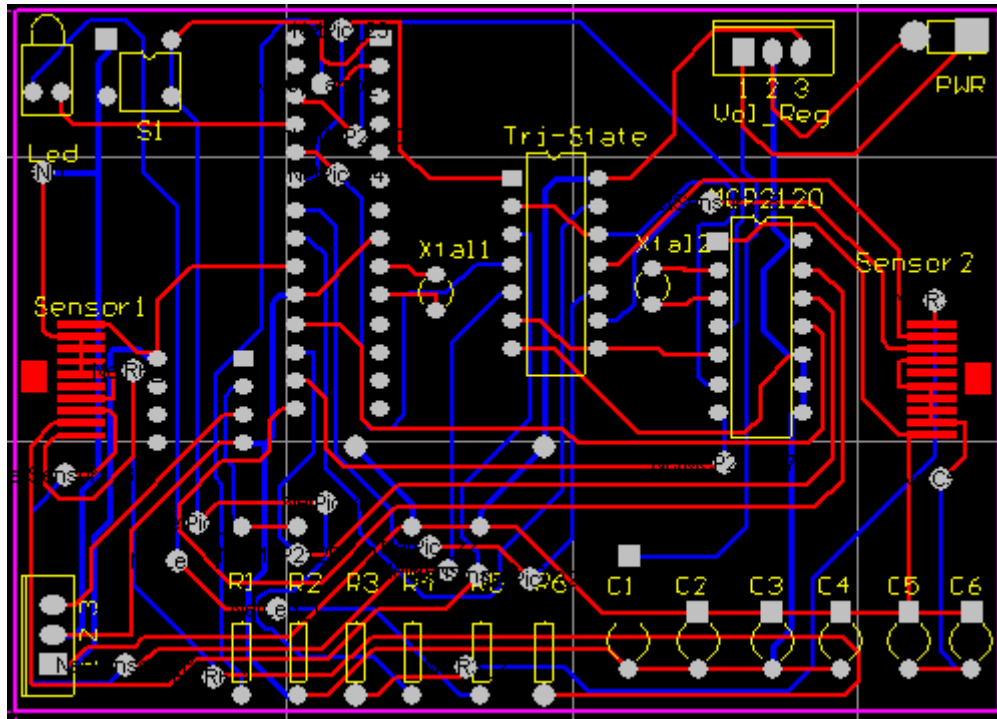


Figura A.4.2 Impreso nodo intermedio

### Apéndice A.4.3 Nodo extremo final

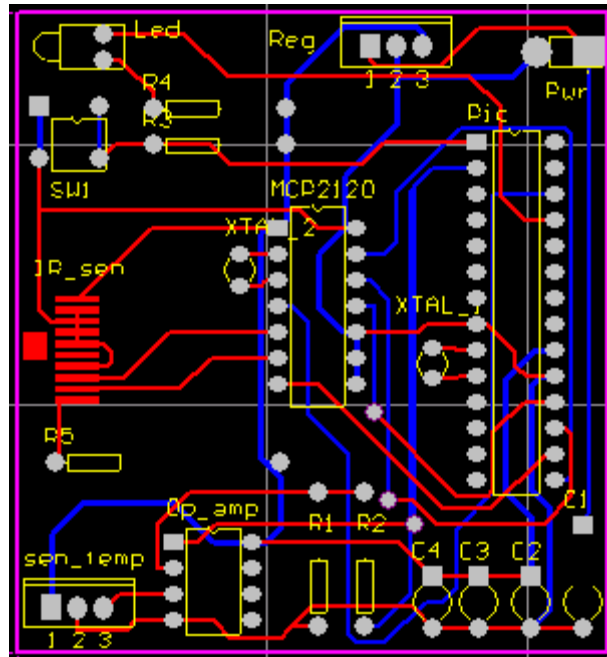


Figura A.4.3 Impreso nodo extremo final

## **ANEXOS**





# MCP2120

## Infrared Encoder/Decoder

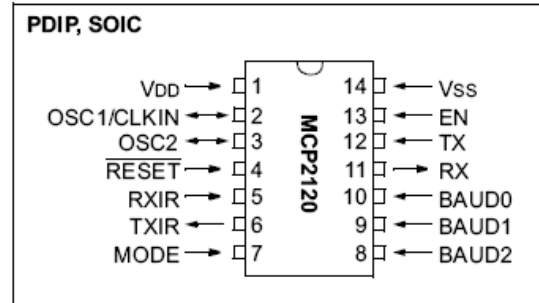
### FEATURES

- Supports with IrDA® Physical Layer Specification (version 1.3)
- UART to IR Encoder/Decoder
  - Interfaces with IrDA Compliant Transceivers
  - Used with any UART, including standard 16550 UART and microcontroller UART
- Transmit/Receive formats supported:
  - 1.63  $\mu$ s
- Hardware or Software Baud rate selection
  - Up to IrDA standard 115.2 kbaud operation
  - Up to 312.5 kbaud operation (at 20 MHz)
  - Low power mode

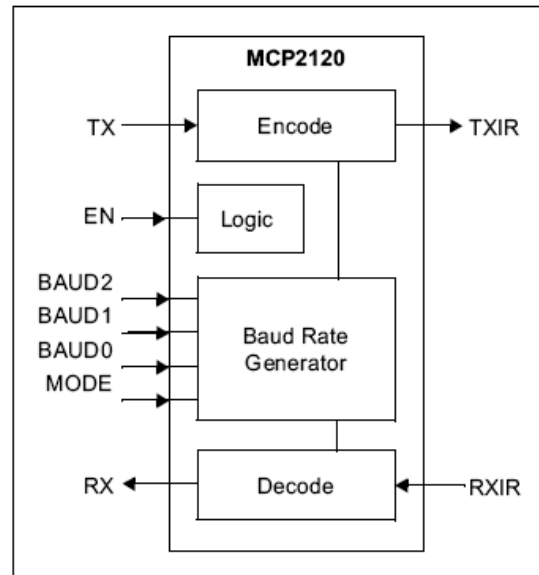
### CMOS TECHNOLOGY

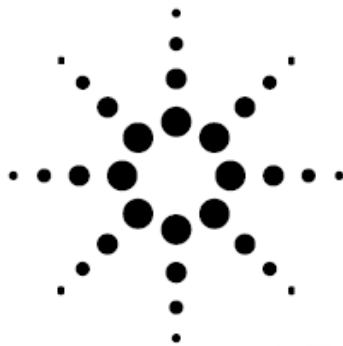
- Low-power, high-speed CMOS technology
- Fully static design
- Low voltage operation
- Commercial and Industrial temperature ranges
- Low power consumption
  - < 1 mA @ 3.3V, 8 MHz (typical)
  - 3  $\mu$ A typical @ 5.0V when disabled

### PIN DIAGRAMS



### BLOCK DIAGRAM





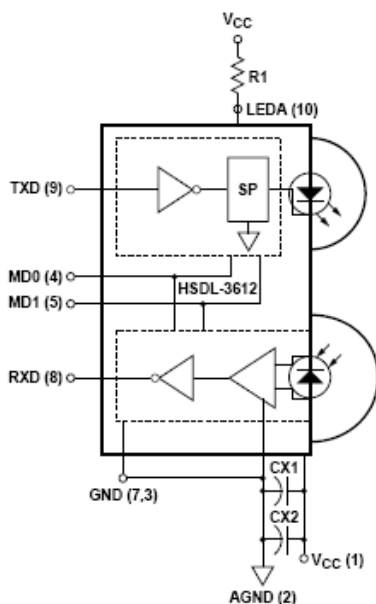
## Agilent HSDL-3612 IrDA® Data Compliant 115.2 kb/s 3 V to 5 V Infrared Transceiver Data Sheet



### Description

The HSDL-3612 is a low-profile infrared transceiver module that provides interface between logic and IR signals for through-air, serial, half-duplex IR data link. The module is compliant to IrDA Data Physical Layer Specifications 1.4 and IEC825-Class 1 Eye Safe.

### Functional Block Diagram



### Applications

- **Digital imaging**
  - Digital still cameras
  - Photo-imaging printers
- **Data communication**
  - Notebook computers
  - Desktop PCs
  - Win CE handheld products
  - Personal Digital Assistants (PDAs)
  - Printers
  - Fax machines, photocopiers
  - Screen projectors
  - Auto PCs
  - Dongles
  - Set-Top box
- **Telecommunication products**
  - Cellular phones
  - Pagers
- **Small industrial & medical instrumentation**
  - General data collection devices
  - Patient & pharmaceutical data collection devices

### Features

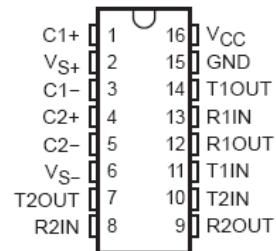
- **Fully compliant to IrDA 1.0 physical layer specifications**
  - 9.6 kb/s to 115.2 kb/s operation
- **Typical link distance > 1.5 m**
- **IEC825-Class 1 eye safe**
- **Low power operation range**
  - 2.7 V to 5.25 V
- **Small module size**
  - 4.0 x 12.2 x 5.1 mm (HxWxD)
- **Complete shutdown**
  - TXD, RXD, PIN diode
- **Low shutdown current**
  - 10 nA typical
- **Adjustable optical power management**
  - Adjustable LED drive-current to maintain link integrity
- **Integrated EMI shield**
  - Excellent noise immunity
- **Edge detection input**
  - Prevents the LED from long turn-on time
- **Interface to various super I/O and controller devices**
- **Designed to accommodate light loss with cosmetic window**
- **Only 2 external components are required**

## MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- $\mu$ F Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- $\pm$ 30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- $\mu$ F Charge-Pump Capacitors is Available With the MAX202
- Applications
  - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE  
 MAX232I . . . D, DW, OR N PACKAGE  
 (TOP VIEW)



### description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept  $\pm$ 30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

### ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
SOP (NS)	Reel of 2000	MAX232NSR	MAX232	
-40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at [www.ti.com/sc/package](http://www.ti.com/sc/package).



# PIC16F87XA

## 28/40/44-Pin Enhanced Flash Microcontrollers

### Devices Included in this Data Sheet:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

### High-Performance RISC CPU:

- Only 35 single-word instructions to learn
- All single-cycle instructions except for program branches, which are two-cycle
- Operating speed: DC – 20 MHz clock input  
DC – 200 ns instruction cycle
- Up to 8K x 14 words of Flash Program Memory, Up to 368 x 8 bytes of Data Memory (RAM), Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to other 28-pin or 40/44-pin PIC16CXXX and PIC16FXXX microcontrollers

### Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during Sleep via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I<sup>2</sup>C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) – 8 bits wide with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

### Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter (A/D)
- Brown-out Reset (BOR)
- Analog Comparator module with:
  - Two analog comparators
  - Programmable on-chip voltage reference (VREF) module
  - Programmable input multiplexing from device inputs and internal voltage reference
  - Comparator outputs are externally accessible

### Special Microcontroller Features:

- 100,000 erase/write cycle Enhanced Flash program memory typical
- 1,000,000 erase/write cycle Data EEPROM memory typical
- Data EEPROM Retention > 40 years
- Self-reprogrammable under software control
- In-Circuit Serial Programming™ (ICSP™) via two pins
- Single-supply 5V In-Circuit Serial Programming
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving Sleep mode
- Selectable oscillator options
- In-Circuit Debug (ICD) via two pins

### CMOS Technology:

- Low-power, high-speed Flash/EEPROM technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Commercial and Industrial temperature ranges
- Low-power consumption

Device	Program Memory		Data SRAM (Bytes)	EEPROM (Bytes)	I/O	10-bit A/D (ch)	CCP (PWM)	MSSP		USART	Timers 8/16-bit	Comparators
	Bytes	# Single Word Instructions						SPI	Master I <sup>2</sup> C			
PIC16F873A	7.2K	4096	192	128	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F874A	7.2K	4096	192	128	33	8	2	Yes	Yes	Yes	2/1	2
PIC16F876A	14.3K	8192	368	256	22	5	2	Yes	Yes	Yes	2/1	2
PIC16F877A	14.3K	8192	368	256	33	8	2	Yes	Yes	Yes	2/1	2



November 2004

## LM158/LM258/LM358/LM2904 Low Power Dual Operational Amplifiers

### General Description

The LM158 series consists of two independent, high gain, internally frequency compensated operational amplifiers which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage.

Application areas include transducer amplifiers, dc gain blocks and all the conventional op amp circuits which now can be more easily implemented in single power supply systems. For example, the LM158 series can be directly operated off of the standard +5V power supply voltage which is used in digital systems and will easily provide the required interface electronics without requiring the additional  $\pm 15V$  power supplies.

The LM358 and LM2904 are available in a chip sized package (8-Bump micro SMD) using National's micro SMD package technology.

### Unique Characteristics

- In the linear mode the input common-mode voltage range includes ground and the output voltage can also swing to ground, even though operated from only a single power supply voltage.
- The unity gain cross frequency is temperature compensated.
- The input bias current is also temperature compensated.

### Advantages

- Two internally compensated op amps
- Eliminates need for dual supplies
- Allows direct sensing near GND and  $V_{OUT}$  also goes to GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

### Features

- Available in 8-Bump micro SMD chip sized package, (See AN-1112)
- Internally frequency compensated for unity gain
- Large dc voltage gain: 100 dB
- Wide bandwidth (unity gain): 1 MHz (temperature compensated)
- Wide power supply range:
  - Single supply: 3V to 32V
  - or dual supplies:  $\pm 1.5V$  to  $\pm 16V$
- Very low supply current drain (500  $\mu A$ )—essentially independent of supply voltage
- Low input offset voltage: 2 mV
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Large output voltage swing



November 2004

## LM340/LM78XX Series 3-Terminal Positive Regulators

### General Description

The LM140/LM340A/LM340/LM78XXC monolithic 3-terminal positive voltage regulators employ internal current-limiting, thermal shutdown and safe-area compensation, making them essentially indestructible. If adequate heat sinking is provided, they can deliver over 1.0A output current. They are intended as fixed voltage regulators in a wide range of applications including local (on-card) regulation for elimination of noise and distribution problems associated with single-point regulation. In addition to use as fixed voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents.

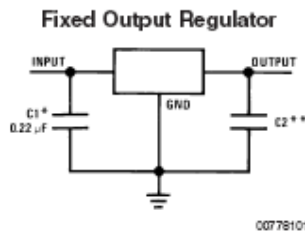
Considerable effort was expended to make the entire series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the output, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

The 5V, 12V, and 15V regulator options are available in the steel TO-3 power package. The LM340A/LM340/LM78XXC series is available in the TO-220 plastic power package, and the LM340-5.0 is available in the SOT-223 package, as well as the LM340-5.0 and LM340-12 in the surface-mount TO-263 package.

### Features

- Complete specifications at 1A load
- Output voltage tolerances of  $\pm 2\%$  at  $T_j = 25^\circ\text{C}$  and  $\pm 4\%$  over the temperature range (LM340A)
- Line regulation of 0.01% of  $V_{\text{OUT}}/\text{V}$  of  $\Delta V_{\text{IN}}$  at 1A load (LM340A)
- Load regulation of 0.3% of  $V_{\text{OUT}}/\text{A}$  (LM340A)
- Internal thermal overload protection
- Internal short-circuit current limit
- Output transistor safe area protection
- P\* Product Enhancement tested

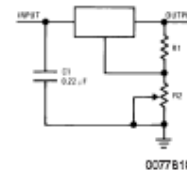
### Typical Applications



\*Required if the regulator is located far from the power supply filter.

\*\*Although no output capacitor is needed for stability, it does help transient response. (If needed, use 0.1  $\mu\text{F}$ , ceramic disc).

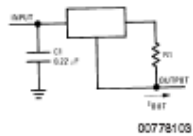
### Adjustable Output Regulator



$$V_{\text{OUT}} = 5V + (5V/R1 + I_Q) R2 \quad 5V/R1 > 3 I_Q$$

load regulation ( $L_r$ ) =  $[(R1 + R2)/R1]$  ( $L_r$  of LM340-5).

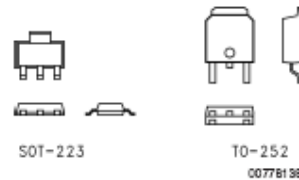
### Current Regulator

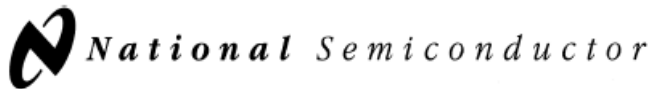


$$I_{\text{OUT}} = \frac{V_{2-3}}{R1} + I_Q$$

$\Delta I_Q = 1.3 \text{ mA}$  over line and load changes.

### Comparison between SOT-223 and D-Pak (TO-252) Packages





## LM35 Precision Centigrade Temperature Sensors

### General Description

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 does not require any external calibration or trimming to provide typical accuracies of  $\pm 1/4^\circ\text{C}$  at room temperature and  $\pm 3/4^\circ\text{C}$  over a full  $-55$  to  $+150^\circ\text{C}$  temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only  $60\ \mu\text{A}$  from its supply, it has very low self-heating, less than  $0.1^\circ\text{C}$  in still air. The LM35 is rated to operate over a  $-55^\circ$  to  $+150^\circ\text{C}$  temperature range, while the LM35C is rated for a  $-40^\circ$  to  $+110^\circ\text{C}$  range ( $-10^\circ$  with improved accuracy). The LM35 series is available pack-

aged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D is also available in an 8-lead surface mount small outline package and a plastic TO-220 package.

### Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guaranteeable (at +25°C)
- Rated for full  $-55^\circ$  to  $+150^\circ\text{C}$  range
- Suitable for remote applications
- Low cost due to wafer-level trimming
- Operates from 4 to 30 volts
- Less than  $60\ \mu\text{A}$  current drain
- Low self-heating,  $0.08^\circ\text{C}$  in still air
- Nonlinearity only  $\pm 1/4^\circ\text{C}$  typical
- Low impedance output,  $0.1\ \Omega$  for 1 mA load

### Typical Applications

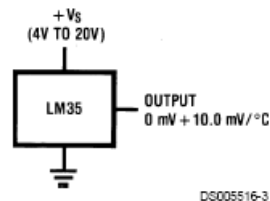
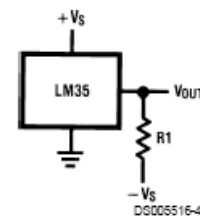


FIGURE 1. Basic Centigrade Temperature Sensor  
( $+2^\circ\text{C}$  to  $+150^\circ\text{C}$ )



Choose  $R_1 = -V_S/50\ \mu\text{A}$   
 $V_{OUT} = +1,500\ \text{mV}$  at  $+150^\circ\text{C}$   
 $= +250\ \text{mV}$  at  $+25^\circ\text{C}$   
 $= -550\ \text{mV}$  at  $-55^\circ\text{C}$

FIGURE 2. Full-Range Centigrade Temperature Sensor

DOI-N08050  
DESCRIPCIÓN

### REQUERIMIENTOS AMBIENTALES

Temperatura:	5°C a +45°C (estándar)
Humedad relativa:	hasta 95% a 35°C
EMI:	EN55022 <b>Precaución:</b> Este es un producto de clase A. En un ambiente doméstico, puede causar interferencia de radio en cuyo caso el usuario debe tomar medidas adecuadas.
EMS:	EN50082-1

*Resumen técnico*  
18-23