



ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad y Computación

**“DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA CENTRALIZADO
DE MONITOREO DE TEMPERATURA PARA TERMOCUNAS DEL
ÁREA DE NEONATOS DE UN HOSPITAL.”**

TESINA DE SEMINARIO

Previo a la obtención del Título de:

INGENIERO EN TELEMÁTICA

INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

Presentado por:

Willam Marcel Ontaneda Yaguana.
Diana Lissette Vivanco Dávila

GUAYAQUIL – ECUADOR

Año: 2012

AGRADECIMIENTO

A Dios y mi familia por estar siempre conmigo, a Diana Vivanco por su aporte en esta tesina, a mis tíos Oswaldo Yaguana y Sandra Saavedra, quienes me acogieron en su hogar y me brindaron su cariño y confianza mientras realice mis estudios, a ellos ¡Gracias!

Willam Ontaneda Yaguana.

AGRADECIMIENTO

A Dios por sobre todas las cosas. A mis padres por su sacrificio constante y su manera única de trascender en mi vida. A mi compañero Willam por su paciencia y dedicación en este proyecto.

Al Ing. Miguel Yapur Auad por la ayuda brindada en el desarrollo de las ideas y sugerencias para la culminación de este trabajo y a cada una de las personas que siempre mostraron su apoyo y creyeron en mí.

Diana Vivanco Dávila.

DEDICATORIA

A mis Padres Edgar Ontaneda y Bertha Yaguana, quienes a la distancia me bendijeron siempre, a mis hermanos Vanessa, Yulissa, Anabel y Renson. A Dios.

A la memoria de (+) Teresa Herrera y Padre (+) Marcos Gonzales.

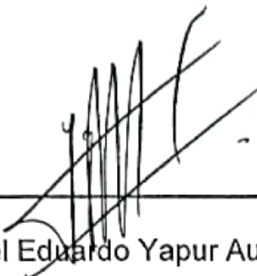
Willam Ontaneda Yaguana.

DEDICATORIA

A mi madre Marcia por su apoyo incondicional, a mis hermanas y hermanos quienes siempre me ayudaron aun sin ellos saber que lo hacían, ustedes son mi fuente de inspiración. A mis tías, tíos, primos y primas, por estar pendientes de mí. Y muy en especial a mi padre Luis Vivanco nada de esto hubiese sido posible sin ti. ¡Gracias a ustedes!

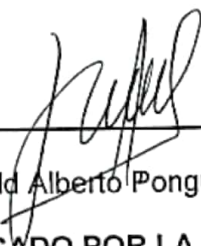
Diana Vivanco Dávila.

TRIBUNAL DE SUSTENTACIÓN



Ing. Miguel Eduardo Yapur Auad, M.Sc.

PROFESOR DEL SEMINARIO DE GRADUACIÓN



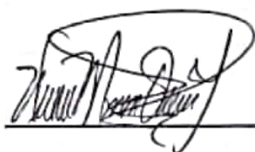
Ing. Ronald Alberto Ponguillo Intriago.

PROFESOR DELEGADO POR LA UNIDAD ACADÉMICA

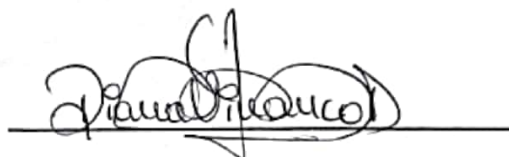
DECLARACIÓN EXPRESA

"La responsabilidad del contenido de esta Tesina de Seminario, nos corresponde exclusivamente; y el patrimonio intelectual de la misma a la Escuela Superior Politécnica del Litoral".

(Reglamento de exámenes y títulos profesionales de la ESPOL)



Willam Marcel Ontaneda Yaguana



Diana Lissette Vivanco Dávila

RESUMEN

En los primeros días de vida del neonato, el ambiente donde se encuentre será un factor muy importante para sobrevivir; es por ello la debida precaución y frecuente monitoreo de su hábitat.

La enfermera verifica la temperatura corporal del neonato cada determinado tiempo, pero estudios realizados demuestran que la temperatura corporal, depende mucho del ambiente en el que se encuentre; he ahí la importancia de tener una mayor monitorización sobre la temperatura ambiental del neonato.

Este trabajo se lo realizó con la finalidad de tener un control sobre la monitorización de la temperatura del ambiente donde se encuentre el neonato, ayudando así a que la temperatura corporal del recién nacido no sufra cambios bruscos.

Con esto se logrará dejar un buen precedente, sobre la influencia que hoy en día tiene la ingeniería en la medicina y el aporte que la tecnología brinda para el mejoramiento de la salud.

ÍNDICE GENERAL

RESUMEN	VII
ABREVIATURAS	XI
ÍNDICE DE FIGURAS	XIII
ÍNDICE DE TABLAS	XV
INTRODUCCIÓN	XVI
CAPÍTULO 1	1
1. MARCO TEÓRICO.....	1
1.1. Parámetros ambientales del neonato	1
1.2. Medidas para disminuir el impacto ambiental	4
1.2.2. Disminución del sonido	5
1.3. Temperatura del recién nacido	7
1.4. Hipotermia	12
1.5. Hipertermia	14
1.6. Modos de pérdida y ganancia de temperatura en el recién nacido.	16
1.6.1. Conducción	17
1.6.2. Convección	17
1.6.3. Evaporación	18
1.6.4. Radiación	18
1.7. Alternativas para el control del ambiente térmico de un recién nacido	19
1.7.1. Incubadora	20
1.7.2. Termocuna o cunas de calor radiante	21

1.7.3. Colchón térmico	22
CAPÍTULO 2	23
2. PRELIMINARES DEL DISEÑO	23
2.1. Descripción del circuito integrado LM35	23
2.2. Pantalla de cristal líquido (LCD)	25
2.3. Transmisor – Receptor inalámbrico (HM - TR 434)	29
2.3.1. Distribución de pines	30
2.3.2. Modo de configuración	32
2.4. Tipo de comunicación	32
2.5. El Microcontrolador	33
2.5.1. Componentes	34
2.5.2. PIC16F887 (Para el sensor de temperatura y Transmisión – Recepción inalámbricas)	41
2.5.3. PIC18F4550 (Para la conexión USB)	47
CAPÍTULO 3	50
3. IMPLEMENTACIÓN DE LA CENTRAL DE MONITOREO	50
3.1. Introducción	50
3.2. Desarrollo del software	53
3.3. Implementación de los circuitos	53
CAPÍTULO 4	58
4. PRUEBAS DE FUNCIONAMIENTO	58
CONCLUSIONES	64
RECOMENDACIONES	67
BIBLIOGRAFÍA	69
ANEXOS	72

Anexo A1.- PIC 16F887	72
Anexo A2.- PIC 18F4550	76
Anexo A3.- LM35	83
Anexo A4.- Código de control de temperatura y transmisión.....	86
Anexo A5: Recepción inalámbrica	90
Anexo A6: Conexión USB - PC.....	93
Anexo A7: Aplicación De Escritorio Desarrollada Con Lenguaje Java.....	97

ABREVIATURAS

ALU: Unidad Aritmético-Lógica.

ASCII: American Standard Code for Information Interchange

°C: Grado centígrado.

CG RAM: Character Generator RAM.

CISC: Complex Instruction Set Computer.

D/A: Digital/Analógico.

dB: Decibelio.

DDRAM: Data Display RAM.

I/O: Entrada/Salida.

EEPROM: Electrically Erasable Programmable Read Only Memory.

EPROM: Erasable Programmable ROM.

FDMA: Frequency Division Multiple Access.

HS: High Speed.

HID: Dispositivo de Interfaz Humana

LCD: Liquid Cristal Display.

LP: Low Power.

Mb/s: Megabits por Segundo.

OSC: Oscilador.

PCB: Printed Circuit Board

PIC: Programmable Interrupt Controller.

PROM: Memoria Programable ROM.

PWM: Pulse Width Modulation.

RAM: Random Access Memory.

RC: Resistor/Capacitor.

RISC: Reduced Instruction Set Computer.

RS232: Recommended Standard-232C.

SRAM: Static Random Access Memory.

TTL: Transistor-Transistor Logic.

ÍNDICE DE FIGURAS

FIGURA 1.1 Distribución de la grasa parda en el cuerpo del neonato.....	9
FIGURA 1.2 Efectos de la temperatura ambiental sobre el consumo de oxígeno.....	11
FIGURA 2.1 Encapsulado TO – 92. LM35.....	25
FIGURA 2.2 LCD 2X16, compuesta de 2 líneas de 16 caracteres.	26
FIGURA 2.3 Transmisor–Receptor HM–TR 434, diagrama de pines	31
FIGURA 2.4 Parámetros T y W para una señal periódica	33
FIGURA 2.5 PIC PIC16F887- Diagrama de pines	43
FIGURA 2.6 PIC16F887- Diagrama de bloques	44
FIGURA 2.7 Diagrama de pines del PIC18F4550.....	48
FIGURA 2.8 PIC18F4550- Diagrama de bloques	49
FIGURA 3.1 Imágenes tomadas de la herramienta Proteus instalado en Windows 7.....	51
FIGURA 3.2 Imágenes tomadas de la herramienta Pic C nstalado en Windows 7.....	52
FIGURA 3.3 Imágenes tomadas de la herramienta Netbeans instalado en Windows 7.	52

FIGURA 3.4 Circuito de sensores de temperatura LM-35	54
FIGURA 3.5 Circuito de sistemas de Botoneras	54
FIGURA 3.6 Circuito Central de transmisión y Visualización	55
FIGURA 3.7 Circuito de recepción inalámbrica	56
FFIGURA 3.8 Circuito de conexión USB	57
FIGURA 4.1 Central de monitoreo vista frontal.....	59
FIGURA 4.2 Central de monitoreo vista interna.....	59
FIGURA 4.3 Central de monitoreo transmisión inalámbrica	60
FIGURA 4.4 Aplicación Java para la conexión USB	60
FIGURA 4.5 Central de recepción inalámbrica vista exterior	61
FIGURA 4.6 Central de recepción inalámbrica vista interior	61
FIGURA 4.7 Central de recepción inalámbrica	62
FIGURA 4.8 Recepción – Transmisión funcionando.....	62

ÍNDICE DE TABLAS

TABLA 1.1 Rango en dB de los sonidos más comunes de una UCIN	6
TABLA 1.2 Mortalidad relacionada con las medidas de protección contra las pérdidas de calor.....	8
TABLA 2.1 Simbología y descripción de pines en la LCD	27
TABLA 2.2 Códigos de control del LCD.....	29
TABLA 2.3 Descripción de los pines del módulo HM – TR 434	31

INTRODUCCIÓN

Este trabajo presenta el diseño y construcción de una central de monitoreo de temperatura para termocunas, la cual cumple con las siguientes características:

- Central: Un módulo electrónico que se ubica en las termocunas; obtiene del neonato su temperatura ambiente, procesa la información y crea paquetes de datos que se envían inalámbricamente a un módulo receptor, ubicado en la central de enfermeras.
- Adicionalmente se crea otro paquete de datos que son enviados, a través de un puerto USB, a una aplicación creada en Windows 7.
- Recepción: Recibe la información de las temperaturas y las muestra en pantalla, LCD externa.
- Aplicación de Windows: Muestra la información de las temperaturas.

En el primer capítulo, se definen las condiciones neonatales necesarias para un mejor control del neonato; en el segundo capítulo se detallan los componentes utilizados, software y hardware necesarios para la construcción; el tercer capítulo se detalla paso a paso la construcción de la central de monitoreo; el cuarto capítulo muestra su funcionamiento, para finalmente concluir y recomendar sobre posibles mejoras al diseño.

CAPÍTULO 1

1. MARCO TEÓRICO

1.1. Parámetros ambientales del neonato

La Unidad de Cuidados Intensivos Neonatales (UCIN) debe ser en un lugar menos ruidoso, sin exceso de iluminación, con un clima controlado, menos agresivo para el neonato, con lo cual será un lugar más acogedor en sus primeros días fuera del seno materno, el doctor Manuel Aceves en Cuidados Intensivos Pediátricos, Ruza F. [1] lo manifiesta; así también señala que el recién nacido tiene características que lo hacen diferente al niño mayor o al adulto, al no ser capaz de tolerar los mismos límites

de temperatura. Los ambientes que pueden ser cómodos o normales para el personal que los atiende, pueden ser peligrosos para el prematuro al estar fuera de su capacidad de regulación.

Con los progresos de la neonatología se ha logrado aumentar la supervivencia del recién nacido; sin embargo, este cambio en el tema de la salud, tiene un precio en las consecuencias a largo plazo de los neonatos que regresan a sus casas.

A los bebés prematuros o a los recién nacidos de bajo peso y a quienes han sufrido enfermedades relacionadas a su prematuridad, se les ha presentado una variedad de dificultades en las áreas: cognitiva, académica, sensorio-motriz, social y emocional.

Los recién nacidos en las Unidades de Cuidados Intensivos Neonatales (UCIN), están expuestos a un ambiente indispensable para mantenerlos vivos, pero no tan apropiados en términos de soportar un desarrollo normal y natural.

Es entonces cuando se considera el aporte de la ecología neonatal, la cual se encarga del estudio del recién nacido y su interacción con el ambiente físico y biológico.

Dentro del ambiente físico podemos considerar a la luz, el calor o radiación solar, la humedad, el viento, el oxígeno, el dióxido de carbono, el agua y la atmósfera, entre otros. Mientras que el medio ambiente biológico está formado por los organismos vivos, principalmente plantas y animales.

El doctor Hernando Villamizar en su estudio sobre neonatos [2], indica que está comprobado que el ambiente de una UCIN puede afectar o aumentar, adversamente, las deficiencias de un recién nacido en su desarrollo neuro-psicológico.

Actualmente es difícil que las UCIN, con su avanzada tecnología brinden el ambiente adecuado y confortable para el desarrollo de los neonatos. Los pacientes hospitalizados en estas unidades, afrontan un significativo trastorno de su ecología neonatal en un ambiente contaminado.

Se debe destacar que el ambiente intrauterino y el que se ofrece en las UCIN, son diferentes. El medio intrauterino es líquido, caliente, oscuro, con movimientos rítmicos y con sonidos monótonos y amortiguados. La UCIN ofrece un ambiente ruidoso, muy luminoso, con continuas interrupciones del sueño e intervenciones dolorosas.

Hoy en día se conoce que los bebés prematuros son niños con riesgo de alteraciones del comportamiento, del aprendizaje, de la integración social, alteraciones visuales, auditivas y neurológicas. Actualmente se ha sugerido que las continuas agresiones ambientales para un cerebro en desarrollo, podrían ser las responsables de gran parte de estas secuelas. [3]

Para prevenir los inconvenientes mencionados, el personal encargado debe considerar cada aspecto perjudicial de una UCIN. El ambiente que se tratar de proporcionar al neonato, tiene que ser el adecuado para su desarrollo y organización de su sistema nervioso. El ambiente creado debe proveerle seguridad, comodidad, estimulación, respuesta a las necesidades que expresa y descanso, entre otras.

1.2. Medidas para disminuir el impacto ambiental

Los Doctores Carlos Mercau, Oscar Pérez, Leticia López y Carlos Stalteri, en la revista del Hospital J.M Ramos Mejía, Volumen XIII, N° 2, sugieren tomar a consideración las siguientes medidas para la disminución del impacto ambiental:

1.2.1. Disminución de la luz

Un niño ingresado en una UCIN se encuentra expuesto permanentemente a la luz brillante. El uso de

fototerapias, lámparas auxiliares, luces fluorescentes, etc., incrementa la intensidad de la luz hasta cuatro veces por encima del nivel recomendado para el ojo adulto.

Sin embargo, los niveles de luz en una UCIN no están establecidos. Las recomendaciones actuales sugieren una luminosidad entre 10 lux y 600 lux al lado de la cuna, tanto de fuentes naturales como eléctricas ajustables, impidiendo luces directas. [2]

1.2.2. Disminución del sonido

Ruidos de alarmas, objetos depositados sobre la incubadora o termocuna y conversaciones alrededor del paciente, reducen considerablemente el nivel de oxigenación arterial del paciente; incluso alteran el patrón respiratorio.

Considerando las normas de diseño de una UCIN, se indica que no se recomiendan ruidos que superen los 55 dB en cualquier parte del área de la Unidad, con un máximo transitorio de 70 dB. [3]

En una incubadora o termocuna, los recién nacidos están permanentemente expuestos a un nivel de ruido entre 50 – 90 dB. En la tabla 1.1 se detalla el rango en decibeles de los sonidos más comunes en una UCIN.

TABLA 1.1 Rango en dB de los sonidos más comunes de una UCIN. [4]

Alarmas de rango de temperatura	60 – 78 dB
Golpear con los dedos sobre la incubadora	70 – 95 dB
Cerrar la puerta de la incubadora	70 – 95 dB
Sonido de llamada de un teléfono celular	120 dB
Cerrar las ventanillas de la incubadora	80 – 110 dB

La polución por ruido produce en el recién nacido una pérdida en el desarrollo auditivo, además de un alto grado de estrés que se ve reflejado en las fluctuaciones del flujo cerebral.

Dentro de las medidas para mejorar las condiciones acústicas en una UCIN, se tiene como principales el uso de materiales absorbentes de ruido y elementos de la barrera, disminución de la intensidad de las alarmas, celulares apagados, entre otros.

1.3. Temperatura del recién nacido

Uno de los cuidados neonatales de mayor importancia es aquel que corresponde a la termorregulación, que es el control de la temperatura corporal del recién nacido.

Se define como termorregulación a la capacidad de regular la energía calórica del cuerpo. El doctor Manuel Aceves [5], manifiesta que durante el embarazo, los mecanismos maternos mantienen la temperatura y, después del nacimiento el neonato debe adaptarse a los cambios de temperatura; es decir, a los ambientes fríos, a través de una producción metabólica de calor, porque ellos no son capaces de producir calor por la actividad muscular.

Teniendo en cuenta los posibles desequilibrios que se presenten en la primera fase tras el parto, debido al estado delicado de neonato, se vuelve de vital importancia mantener un ambiente adecuado y cuidar de manera precisa al recién nacido, para que así mantenga los niveles apropiados de calor en su organismo.

Estudios realizados por el doctor Pierre Budin a comienzos del siglo XX, observó que los prematuros de menos de 1500g que

se enfriaban, tenían una mortalidad significativamente alta; ver tabla 1.2.

TABLA 1.2 Mortalidad relacionada con las medidas de protección contra las pérdidas de calor. [6]

Autores	Mayor Protección		Menor Protección	
	Número	Muertes (%)	Número	Muertes (%)
Silverman (1985)	91	16	91	32
Beatow (1964)	89	42	69	54
Day (1964)	60	23	65	37
Perlstein (1976)	105	22	105	35
Totales	345	26(media)	330	40 (media)

Posteriormente, el doctor William Silverman, en estudios controlados dejó establecida la importancia de la temperatura del ambiente en la supervivencia de los recién nacidos, demostrando así la supervivencia mayor de los niños que son cuidados dentro de ambientes térmicos más adecuados, a la cantidad de calor que se les proporcionaba.

En las primeras horas de vida es donde hay mayor riesgo de enfriamiento para el recién nacido. Su piel es muy delgada, con poca grasa subcutánea y vasos sanguíneos muy próximos a la piel, lo que favorece la pérdida de calor.

El recién nacido produce calor principalmente por la termogénesis química, llamada termogénesis sin estremecimiento, teniendo una forma eficiente de regular su temperatura a través del metabolismo llamado “grasa parda”.

La grasa parda es un tejido graso especial en el recién nacido, el cual tiene una alta capacidad para producir calor a través de reacciones químicas. Como se puede observar en la Figura 1.1, la grasa parda se encuentra localizada en la nuca, el cuello, la región interescapular, el mediastino y alrededor de los riñones.

Luego de metabolizarse, el calor producido se transmite a los vasos sanguíneos mayores y de ahí, a los vasos sanguíneos periféricos del resto del cuerpo, los cuales al mismo tiempo se estrechan para evitar la pérdida de calor. Normalmente la grasa parda desaparece al tercer mes de vida.

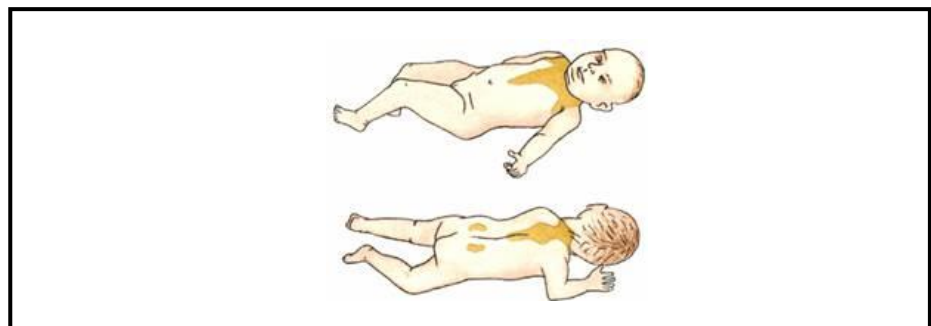


FIGURA 1.1 Distribución de la grasa parda en el cuerpo del neonato. [7]

Los prematuros e inmaduros, que presentan un descenso en la grasa subcutánea y menor capacidad de aislamiento, tienen además desventajas al presentar una superficie corporal mayor con relación a su peso y sobre todo una disminución importante en las reservas de grasa parda, por lo que son menos capaces de movilizar norepinefrina y ácidos grasos libres. Ellos son incapaces de recibir las suficientes calorías para proveer la termogénesis y el crecimiento. El consumo de oxígeno generalmente está asociado frecuentemente con enfermedades pulmonares.

Los efectos de la temperatura ambiental sobre la producción de calor se muestran a continuación en la Figura 1.2. Si se mantiene al neonato dentro de una zona de temperatura ambiental, en donde el consumo de oxígeno y por lo tanto la producción de calor sea mínima y la necesaria para mantener una temperatura corporal normal, se llamará zona de termoneutralidad. Si la temperatura ambiente cae hasta ciertos límites, al igual que la temperatura corporal, el consumo de oxígeno aumentará para mantener al niño dentro de la temperatura normal, lo cual significará un sacrificio de las calorías necesarias para otras funciones, como serían las de crecimiento. Si esto se prolonga, en el caso de rebasar los

límites, se llevará a cabo una parálisis de frío en los centros de regulación de temperatura, y el consumo de oxígeno también caerá, colocando al pequeño en peligro de muerte.

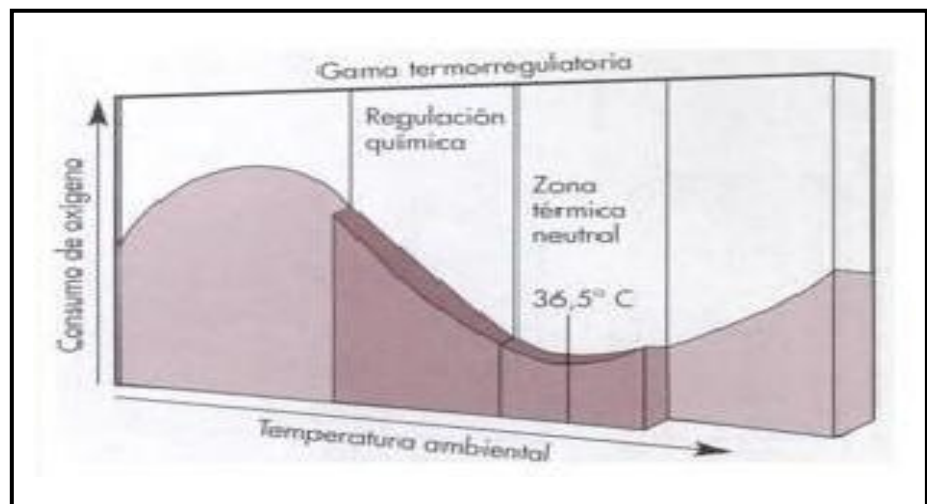


FIGURA 1.2 Efectos de la temperatura ambiental sobre el consumo de oxígeno. [8]

El recién nacido tiene una temperatura similar a la del adulto, alrededor de los 37°C, y dependerá del sitio donde se mida y el método que se esté usando.

Se considera que el bebé debe estar en un ambiente donde su temperatura central esté entre los 36,7°C y los 37,3°C.

Dentro de los valores normales de temperatura en el recién nacido a término, se tiene:

- ❖ Temperatura corporal central: Se refiere a la temperatura axilar y rectal $36.5^{\circ}\text{C} - 37.5^{\circ}\text{C}$.
- ❖ Temperatura de piel: Se refiere a la temperatura abdominal entre $36.0^{\circ}\text{C} - 36.5^{\circ}\text{C}$.

1.4. Hipotermia

En condiciones de estrés por frío, la temperatura corporal central es inicialmente normal a expensas de un gran costo energético. Cuando el recién nacido pierde la capacidad para mantener su temperatura corporal normal, cae en hipotermia.

Cuando las pérdidas de calor superan a la producción de calor, el organismo del bebé responde con mecanismos que disminuyen las pérdidas: vasoconstricción y postura corporal.

A los pocos días, el mecanismo de vasoconstricción está bien desarrollado en los recién nacidos que completaron su tiempo de gestación, pero en los prematuros no es tan efectivo. Una manera de comprobar que se efectúa el mecanismo es porque se puede ver que el bebé está pálido y aumenta el número de respiraciones por minuto.

Por otra parte, la postura es un mecanismo de defensa frente al frío. El recién nacido a término, puede cambiar su posición de

flexión de las cuatro extremidades y, el prematuro, por el contrario, presenta una posición con todos sus miembros extendidos.

En adición se exponen otros signos de hipotermia:

- ❖ Cuerpo frío al tacto
- ❖ Cianosis central y/o acrocianosis
- ❖ Aumento de requerimiento de oxígeno
- ❖ Respiraciones irregulares y/o apnea
- ❖ Intolerancia alimentaria
- ❖ Distensión abdominal, aumento del residuo gástrico
- ❖ Bradicardia
- ❖ Mala perfusión periférica
- ❖ Disminución de la actividad
- ❖ Letargo/Irritabilidad
- ❖ Disminución de reflejos
- ❖ Llanto débil

- ❖ Succión débil
- ❖ Hipoglucemia
- ❖ Edema
- ❖ Dificultad para descansar

1.5. Hipertermia

La exposición a temperaturas superiores al ambiente térmico neutro da lugar a una lesión por calor, que se manifiesta por aumento de la temperatura.

Mediciones simultáneas de la piel, del abdomen, rectal o del pie, ayudan a distinguir si el recién nacido está febril, porque ha sido iatrogénicamente sobrecalentado o, tiene un estado hipermetabólico provocado por una infección o por drogas. En el niño sin fiebre, la diferencia de temperatura abdominal de la rectal no es mayor a 1°C ó 2°C; y la temperatura del pie no será más de 2°C a 3 °C con respecto del abdomen [9]. En el niño febril con hipermetabolismo, estas diferencias serán mayores a las anteriormente mencionadas.

Cuando la causa es la temperatura ambiental, el tronco y las extremidades se encontrarán a la misma temperatura y se

identificará la vasodilatación. Si el aumento es excesivo, puede dar lugar a un golpe de calor o a un síndrome de encefalopatía y shock hemorrágico.

Una fuente de calor servocontrolada puede generar un exceso de calor y provocar hipertermia grave.

Entre los signos de hipertermia más comunes se tienen:

- ❖ Taquipnea
- ❖ Apnea
- ❖ Taquicardia
- ❖ Hipotensión
- ❖ Rubor
- ❖ Extremidades calientes
- ❖ Irritabilidad
- ❖ Alimentación irregular
- ❖ Letárgia
- ❖ Hipotonía
- ❖ Postura en extensión

- ❖ Llanto débil o ausente
- ❖ Temperatura de piel mayor que la temperatura central

1.6. Modos de pérdida y ganancia de temperatura en el recién nacido.

El individuo produce calor de dos maneras: el que proviene de las reacciones metabólicas producidas por todas las células del organismo y, el procedente del metabolismo producido por la actividad muscular. El recién nacido tiene muy limitada su capacidad de producir calor a través de la actividad muscular, por lo cual sólo quedará la posibilidad de hacerlo por medio del metabolismo de las células de la grasa parda, que tienen un complejo sistema mitocondrial y numerosas vacuolas grasas.

El recién nacido tiene limitada la capacidad para defenderse no sólo de los ambientes fríos, sino también de los cálidos.

Teniendo en cuenta que cuanto mayor es la superficie de contacto, mayor es la transferencia de calor. Existen cuatro mecanismos por los cuales el calor es transferido desde y hacia la superficie corporal. Estos son:

1.6.1. Conducción

La conducción es la transmisión de energía térmica entre dos cuerpos que se encuentran en contacto y, está en relación a la magnitud de la superficie unida, la diferencia de temperatura y al coeficiente de conducción.

Se da por la pérdida o ganancia del calor corporal a una superficie fría o caliente en contacto directo con el recién nacido. Entre éstos podemos destacar el contacto con objetos no precalentados (balanzas, colchones, placas radiológicas y estetoscopios).

1.6.2. Convección

La convección equivale a la conducción en movimiento y esto es aplicable para los gases y líquidos; el intercambio estará acorde tanto a la diferencia de temperaturas como al tamaño de los objetos, así como a la velocidad del fluido.

Este mecanismo se presenta por la pérdida o ganancia de calor corporal hacia corrientes de aire o agua, las que envuelven al recién nacido; por ejemplo, la circulación de

aire frío hacia un área expuesta o, la inmersión en el agua con una temperatura inadecuada.

1.6.3. Evaporación

La evaporación es un mecanismo por el cual se pierde energía a razón de 0.58 calorías por gramo evaporado.

En el recién nacido ocurre cuando el niño está bañado en el líquido amniótico, y se debe a la falta de queratinización de la epidermis y la vasodilatación de los vasos sanguíneos subcutáneos.

1.6.4. Radiación

La radiación es la transmisión del calor a distancia por medio de rayos infrarrojos, en donde la magnitud de los cuerpos, la superficie expuesta y la distancia, desempeñan un papel importante en el fenómeno; al mismo tiempo, emiten o reflejan el calor.

Al nacer, el neonato se encuentra rodeado de objetos o superficies más frías. La radiación se presenta por la pérdida de calor corporal hacia un objeto más frío, que no está en contacto directo con el niño.

1.7. Alternativas para el control del ambiente térmico de un recién nacido

Según recomendaciones de organismos internacionales, los termómetros de mercurio no deben ser utilizados porque el mercurio es un contaminante peligroso para el medio ambiente.

La medición de temperatura puede realizarse prácticamente en cualquier parte del cuerpo y por diversos instrumentos, siendo el más usado el termómetro clínico el cual debe tener un rango de medición desde los 32°C, con el fin de que puedan calificarse los grados de hipotermia.

La temperatura axilar es la recomendada como una prueba de detección de fiebre en el recién nacido.

La temperatura timpánica no está indicada en el recién nacido debido a la dificultad de lecturas por la posición del conducto auditivo externo.

Se recomienda además que el tiempo de permanencia del termómetro en el sitio de medición pueda llegar hasta los 9 minutos para considerar una exacta lectura.

Cuando se realice la monitorización de la temperatura con los sensores de temperatura de la incubadora o termocuna, los

sensores deben estar bien adheridos a la piel, sobre una superficie no lisa y no ósea, procurando no cubrir al sensor con ropa y no recostando al niño sobre el sensor.

1.7.1. Incubadora

La Incubadora es el elemento de uso común para calentar al neonato. La mayoría de los procedimientos invasivos se deberían realizar a través de las ventanas de ésta.

La incubadora más conocida en el medio es la de pared sencilla de acrílico, con un sistema de calentamiento de aire que mantiene la temperatura corporal principalmente por convección.

Los factores que intervienen para que ésta se mantenga en una zona de termoneutralidad son: los patrones de velocidad del aire, la humedad y las superficies en las cuales el calor se intercambia por radiación.

En la actualidad, la mayoría de los fabricantes de incubadoras incluye su modelo de doble pared. Éstas proporcionan una temperatura de operación y de la pared, mayor que las de pared simple.

El recién nacido debe estar desnudo para permitir que el calor de la incubadora sea efectivo.

1.7.2. Termocuna o cunas de calor radiante

Estos equipos son de fácil accesibilidad sin interrumpir la fuente de calor directa al niño.

Antes de colocar al recién nacido en la Termocuna, es necesario encender el equipo anticipadamente para que el colchón y la sábana donde se apoya el niño estén calientes. El calor por radiación será tendrá un nivel adecuado a las necesidades del niño.

Debido a que no se puede controlar la humedad se debe considerar el número de pérdidas insensibles de agua en el manejo de líquidos del recién nacido. En la práctica, el uso de las termocunas solo se aconseja para el nacimiento.

En adición, se sugiere tener levantadas las paredes de la cuna de calor radiante para crear un microclima y evitar corrientes de aire.

1.7.3. Colchón térmico

Se lo utiliza como alternativa a las cunas de calor radiante para mantener a recién nacidos con pesos menores a los 1500g, más calientes y contribuyendo a reducir la incidencia de hipotermia.

Sin embargo, su utilización más frecuente consiste en evitar la demora en la transición de incubadora a una cuna.

CAPÍTULO 2

2. PRELIMINARES DEL DISEÑO

A continuación se presenta en detalle, cada uno de los componentes necesarios para el diseño y construcción de la central de monitoreo; sus características y su correcto funcionamiento, para evitar fallas de implementación.

2.1. Descripción del circuito integrado LM35

El integrado LM35 es un sensor de temperatura producido por el fabricante National Semiconductor. Cuenta con una precisión

calibrada de 1°C . Su rango de medición va desde -55°C hasta 150°C , la salida es lineal y cada grado centígrado equivale a 10mV , para efectos de explicación se presenta el siguiente ejemplo:

❖ $150^{\circ}\text{C} = 1500\text{mV}$

❖ $-40^{\circ}\text{C} = -400\text{mV}$

Dentro de las características se destacan las siguientes:

- ❖ Está calibrado directamente en grados Celsius
- ❖ La tensión de salida es proporcional a la temperatura
- ❖ Tiene una precisión garantizada de 0.5°C a 25°C
- ❖ Opera entre los 4 y 30 V de alimentación
- ❖ Baja impedancia de salida
- ❖ Baja corriente de alimentación ($60\mu\text{A}$)

Además, el LM35 no requiere de circuitos adicionales para calibrarlo externamente.

Dentro de los diversos tipos de encapsulado, se destaca al más común el TO-92, el cual se utiliza en transistores de baja potencia. Este cuenta con tres pines: Vs, Vout y Gnd. Vs es el

pin de alimentación; generalmente este integrado es alimentado con 5V. Vout es el pin que entrega el valor de temperatura sensado en mV. Gnd es el pin de conexión a tierra.

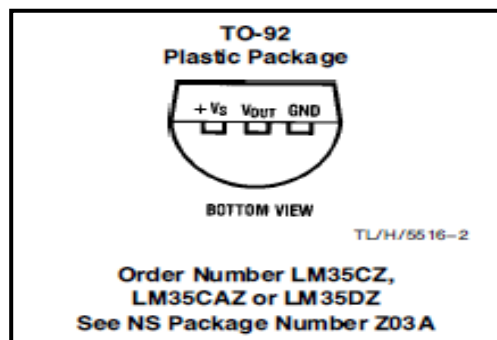


FIGURA 2.1 Encapsulado TO – 92. LM35. [10]

2.2. Pantalla de cristal líquido (LCD)

Las pantallas de cristal líquido LCD (Liquid Cristal Display) tienen la capacidad de mostrar cualquier carácter alfanumérico, permitiendo representar la información que genera cualquier equipo electrónico de una forma fácil.

La pantalla consta de una matriz de caracteres distribuidos en una, dos, tres o cuatro líneas de 16 hasta 40 caracteres cada línea.

El proceso de visualización es gobernado por un microcontrolador incorporado a la pantalla, siendo el Hitachi

44780 el modelo de controlador más utilizado; entre las principales características tenemos:

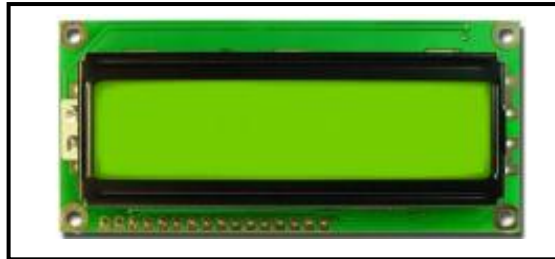


FIGURA 2.2 LCD 2X16, compuesta de 2 líneas de 16 caracteres. [11]

- ❖ El Voltaje de alimentación es de 5V, con un consumo menor de 5mA
- ❖ Pantalla de caracteres ASCII
- ❖ Desplazamiento de los caracteres de izquierda a derecha y viceversa
- ❖ Memoria de 40 caracteres por línea de pantalla, visualizándose 16 caracteres por línea

Pueden ser gobernados de 2 formas principales:

- ❖ Conexión con bus de 4 bits
- ❖ Conexión con bus de 8 bits

Diagrama de pines.

En la siguiente tabla se detallan las señales empleadas por el módulo, así como también el PIN al que corresponden:

TABLA 2.1 Simbología y descripción de pines en la LCD. [12]

PIN N.-	Simbología	Descripción
1	Vss	GND
2	Vdd	+ 5V
3	Vo - Vee	Alimentación del cristal líquido (contraste)
4	Rs	Selección de registro
5	R/W	Selección de lectura/escritura
6	E	Señal de activación del módulo
7	DB0	I/O (Bit 0 del dato)
8	DB1	I/O (Bit 1 del dato)
9	DB2	I/O (Bit 2 del dato)
10	DB3	I/O (Bit 3 del dato)
11	DB4	I/O (Bit 4 del dato)
12	DB5	I/O (Bit 5 del dato)
13	DB6	I/O (Bit 6 del dato)
14	DB7	I/O (Bit 7 del dato)

DDRAM (Data Display RAM)

El LCD cuenta con una zona de memoria RAM llamada DDRAM donde se almacenan los caracteres que se van a mostrar en la pantalla.

Tiene una capacidad de 80 bytes, 40 por cada línea, de los cuales 32 se pueden visualizar a la vez (16 por línea).

Las dos direcciones más importantes de la DDRAM son:

- ❖ 00h, comienzo de la primera línea
- ❖ 40h, comienzo de la segunda línea

CG RAM (Character Generator RAM)

La CG RAM es la memoria que contiene los caracteres definibles por el usuario. Está formada por 64 posiciones, con direcciones 00h – 3Fh. Cada posición es de 5 bits.

La memoria está dividida en 8 bloques, correspondiendo cada bloque a un carácter definible por el usuario.

Modo de funcionamiento

El LCD tiene 3 modos de funcionamiento principales:

- ❖ Modo comando
- ❖ Modo carácter o dato
- ❖ Modo de lectura del busy flag ó LCD ocupada

Comandos de control

Los comandos de control son diversos códigos que se transmiten a través del bus de datos del módulo LCD, a continuación los más destacados:

TABLA 2.2 Códigos de control del LCD. [13]

Instrucciones		Código									
		RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Instrucciones de comando	Clear Display	0	0	0	0	0	0	0	0	0	1
	Return Home	0	0	0	0	0	0	0	0	1	x
	Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
	Display ON/OFF Control	0	0	0	0	0	0	1	D	C	B
	Cursor or Display Shift	0	0	0	0	0	1	S/C	R/L	x	x
	Function Set	0	0	0	0	1	DL	N	F	x	x
	Set CG RAM Address	0	0	0	1	Puntero de RAM (CG RAM)					
	Set DD RAM Address	0	0	1	Puntero de RAM (DD RAM)						
	Read Busy Flag & RAM Pointer	0	1	BF	Puntero de RAM (DD RAM o CG RAM)						
Instrucciones de Datos	Write to CG RAM or DDRAM	1	0	Escribir Dato							
	Read from CG RAM or DDRAM	1	1	Leer Dato							

2.3. Transmisor – Receptor inalámbrico (HM - TR 434)

El módulo de enlace de datos inalámbricos HM-TR 434 pertenece a la familia de los trans - receptores HM-TR de la empresa Hope Microelectronics Co. Ltd.

Entre las características generales del módulo se destaca la alta tasa de transmisión de datos y, una cobertura de transmisión amplia; es poco susceptible a la interferencia. Tiene la capacidad de que el usuario pueda configurar la frecuencia de operación, por lo que puede ser utilizado en aplicaciones FDMA.

La desviación de la frecuencia de transmisión y el ancho de banda del receptor pueden ser seleccionados. El protocolo de transmisión es controlado por el dispositivo en sí y es fácil de usar. La velocidad de transmisión de datos puede ser escogida de un amplio rango.

El módulo provee un pin habilitador para el control de ciclos de trabajo con el fin de satisfacer los requerimientos de diferentes aplicaciones.

Tiene la capacidad de seleccionar los niveles lógicos de la interfaz estándar entre UART, TTL o RS232. Además el módulo es altamente confiable, pequeño y de fácil montaje.

2.3.1. Distribución de pines.

En el gráfico se presenta el diagrama de pines para el transmisor – receptor HM – TR 434.



FIGURA 2.3 Transmisor–Receptor HM–TR 434,
diagrama de pines. [14]

En la tabla a continuación, se detalla la descripción de cada uno de los pines del módulo HM – TR 434.

TABLA 2.3 Descripción de los pines del módulo
HM – TR 434. [15]

IN N.-	NOMBRE	DESCRIPCION
1.- VDD	Fuente de alimentación	+ 5V
2.- DTX	Transmisión de datos	Transmisión de datos por el módulo
3.- GND	Conexión a tierra	0V
4.- DRX	Recepción de datos	Recepción de datos por el módulo
5.- CONFIG	Modo de Configuración	Si el pin CONFIG está e ALTO al encender, el módulo entra en modo de configuración. Si el pin de CONFIG está en BAJO al encender, el módulo entra en el modo normal para la transmisión de datos.
6.- ENABLE	Habilitación	Si el pin ENABLE está en ALTO, el módulo se enciende. Si el pin ENABLE está en BAJO, el módulo entra en modo dormir.

2.3.2. Modo de configuración

Si el pin CONFIG está en ALTO al momento de encender, el módulo entrará en el modo de configuración automáticamente. En este modo, el módulo se comunica con el host en un formato serial fijo (9600, 8, N, 1).

El módulo puede trabajar apropiadamente con la configuración por defecto (9600, 8, N, 1) correspondientes a velocidad de transmisión, número de bits por dato, chequeo de paridad y número de bits de parada respectivamente.

2.4. Tipo de comunicación

El tipo de comunicación escogida para la generación de la señal a transmitir, es la modulación por ancho de pulsos (PWM).

La modulación PWM (Pulse Width Modulation), se basa en modificar el ciclo de una señal periódica (generalmente se emplea sobre señales sinusoidales o cuadradas).

El ciclo de trabajo de una señal periódica D se define a través del cociente entre el ancho relativo de su parte positiva W y el período de la señal T .

$$D = \frac{T}{W}$$

Donde,

D es el ciclo de trabajo (se define en porcentaje), W es el tiempo de la función positiva (ancho del pulso) y T es el período de la señal.

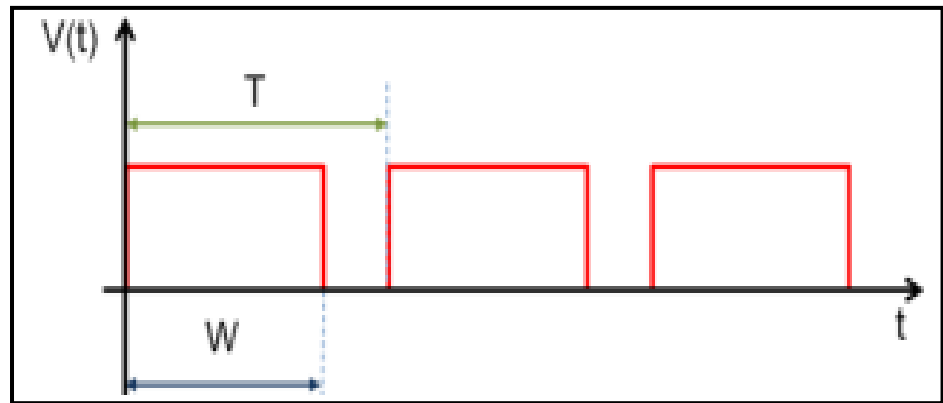


FIGURA 2.4 Parámetros T y W para una señal periódica. [16]

Las aplicaciones de la modulación PWM están orientadas fundamentalmente al control de fuentes conmutadas, velocidad de motores, elementos termoelectrónicos, y convertidores D/A.

2.5. El Microcontrolador.

Un microcontrolador es un circuito integrado que incluye en su interior las tres unidades funcionales de una computadora:

unidad central de procesamiento, memoria y unidades de E/S (entrada/salida).

Un microcontrolador cuenta con un generador de reloj integrado y una pequeña cantidad de memoria RAM y PROM/EPROM/EEPROM/FLASH; para que éste funcione, necesita algunos programas de control y un cristal de sincronización. Los microcontroladores, también disponen de dispositivos de entrada/salida, como convertidores analógico a digital, temporizadores, UART's y buses de interfaz serial especializados.

2.5.1. Componentes.

Registros

Los registros son usados para almacenar los resultados de la ejecución de instrucciones, cargar datos desde la memoria externa o almacenarlos en ella.

Mientras mayor sea el número de bits de los registros de datos del procesador, mayores serán sus prestaciones, en cuanto al poder de cómputo y velocidad de ejecución, de modo que éste parámetro determina la potencia que se puede incorporar al resto de los componentes del sistema.

De esta manera, un procesador de 16 bits puede realizar una suma de 16 bits en un solo ciclo de máquina, mientras que uno de 8 bits deberá ejecutar varias instrucciones antes de tener el resultado, aun cuando ambos procesadores tengan la misma velocidad de ejecución para sus instrucciones.

Unidad de control (UC)

La unidad de control es que cuenta con la lógica necesaria para la decodificación y ejecución de las instrucciones, el control de los registros, la ALU, los buses y demás módulos que formen parte del procesador.

Además es el elemento más complejo de un procesador, generalmente está dividido en unidades pequeñas, las cuales trabajan en conjunto. Contiene componentes tales como la unidad de decodificación, unidad de ejecución, controladores de memoria cache, controladores de buses, controlador de interrupciones, pipelines, entre otros elementos, los cuales siempre dependen del tipo de procesador.

Unidad Aritmética – Lógica (ALU)

Es la unidad donde se realizan las sumas, restas y operaciones lógicas típicas del álgebra de Boole. Hoy en día los procesadores más modernos tienen varias ALU, especializadas en la realización de operaciones complejas.

Buses

Los buses son el medio de comunicación que utilizan los diferentes componentes del procesador para intercambiar información entre sí. En el caso de los microcontroladores, no es común que los buses estén reflejados en el encapsulado del circuito, ya que éstos se destinan básicamente a las entradas/salidas (E/S) de propósito general y periféricos del sistema.

Conjunto de instrucciones

Son el conjunto de operaciones básicas que puede realizar el procesador, las cuales forman el software. El conjunto de instrucciones conforman el elemento básico del lenguaje.

Existen dos tipos básicos de instrucciones que determinan la arquitectura del procesador: CISC y RISC.

CISC, del inglés Complex Instruction Set Computing. Los microprocesadores CISC tienen un conjunto de instrucciones que se caracteriza por ser muy amplio, y que permiten realizar operaciones complejas entre operandos situados en la memoria o en los registros internos.

En la actualidad, la mayoría de los sistemas CISC de alto rendimiento, convierten las instrucciones complejas en varias instrucciones simples del tipo RISC, llamadas generalmente microinstrucciones.

RISC, del inglés Reduced Instruction Set Computer. Se centra en la obtención de procesadores con las siguientes características fundamentales:

Instrucciones de tamaño fijo

- ❖ Pocas instrucciones
- ❖ Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos

- ❖ Número relativamente elevado de registros de propósito general
- ❖ Los procesadores de los microcontroladores PIC son de tipo RISC.

Memoria

La memoria RAM es la encargada del almacenamiento de información temporal, que será utilizada por el procesador para realizar cálculos o cualquier tipo de operaciones lógicas. En el espacio de direcciones de memoria RAM se ubican además, los registros de trabajo del procesador, de configuración y los distintos periféricos del microcontrolador.

El tipo de memoria utilizada en las memorias RAM de los microcontroladores es SRAM. A pesar de que la memoria SRAM es más costosa que la DRAM, es el tipo adecuado para los microcontroladores porque éstos poseen pequeñas cantidades de memoria RAM.

Cuando se requiere aumentar la cantidad de memoria de datos, lo más frecuente es colocar dispositivos de memoria externa en forma de periféricos; de esta forma

se pueden utilizar memorias RAM, FLASH o incluso discos duros.

Periféricos

Los periféricos son dispositivos externos que se encargan de intercambiar los datos con el procesador.

La comunicación entre el procesador y el periférico está regulada en el procesador por los siguientes métodos:

- ❖ Polling.- El procesador revisa ordenadamente todos los periféricos para atender a cada uno de ellos secuencialmente
- ❖ Interrupciones.- El periférico que está listo para ser atendido por el procesador solicita una “interrupción” de la ejecución del programa para que el procesador lo atienda

Entradas y salidas de propósito general

También conocidos como puertos de E/S, permiten leer datos del exterior o escribir en ellos desde el interior del microcontrolador. Generalmente se agrupan en puertos de 8 bits.

Algunos puertos de E/S tienen características especiales que incorporan mecanismos especiales de interrupción para el procesador.

Temporizadores y contadores

Son circuitos sincrónicos para el conteo de los pulsos que llegan a su poder para conseguir la entrada de reloj.

Si la fuente de un gran conteo es el oscilador interno del microcontrolador, es común que no tengan un pin asociado, y en este caso trabajan como temporizadores. Por otra parte, cuando la fuente de conteo es externa, entonces tienen asociado un pin configurado como entrada; éste es el modo contador.

Los temporizadores son usados para la medición de frecuencia, implementación de relojes, para el trabajo de conjunto con otros periféricos que requieren una base estable de tiempo, entre otras funcionalidades.

Generalmente los tamaños de los registros de conteo son 8 y 16 bits.

Convertidor Analógico/Digital

El convertidor analógico/digital es el dispositivo que recibe una señal analógica y la muestrea con cierta frecuencia, para generar un valor digital representativo de la señal, al momento de la toma de la muestra.

El microcontrolador lo utiliza para tomar datos de varias entradas diferentes, que se seleccionan mediante un multiplexor.

Las resoluciones más frecuentes son 8 y 10 bits. También es posible conectar un convertidor externo, en caso de necesidad.

2.5.2. PIC16F887 (Para el sensor de temperatura y Transmisión – Recepción inalámbricas)

El PIC16F887 pertenece a la subfamilia de microcontroladores PIC (Peripheral Interface Controller) son fabricados por Microchip Technology Inc.

El PIC16F887 es un microcontrolador de tipo RISC, cuenta con memoria de programa de tipo EEPROM Flash mejorada. Esta característica facilita el diseño de

proyectos, ya que reduce el tiempo empleado en programar los microcontroladores.

Características principales del PIC16F887

Se destacan las siguientes características

- ❖ Procesador de arquitectura RISC avanzada
- ❖ 35 instrucciones con 14 bits de longitud. Estas se ejecutan en un ciclo de instrucción, menos las de salto que se ejecutan en dos ciclos de instrucción
- ❖ Hasta 368 bytes de memoria de datos RAM
- ❖ Hasta 256 bytes de memoria de Datos EEPROM
- ❖ Hasta 14 fuentes de interrupción internas y externas
- ❖ Pila de 8 niveles
- ❖ Modos de direccionamiento directo e indirecto
- ❖ Perro Guardián (WDT)
- ❖ Código de protección programable
- ❖ Modo SLEEP de bajo consumo

Diagrama de pines del PIC16F887

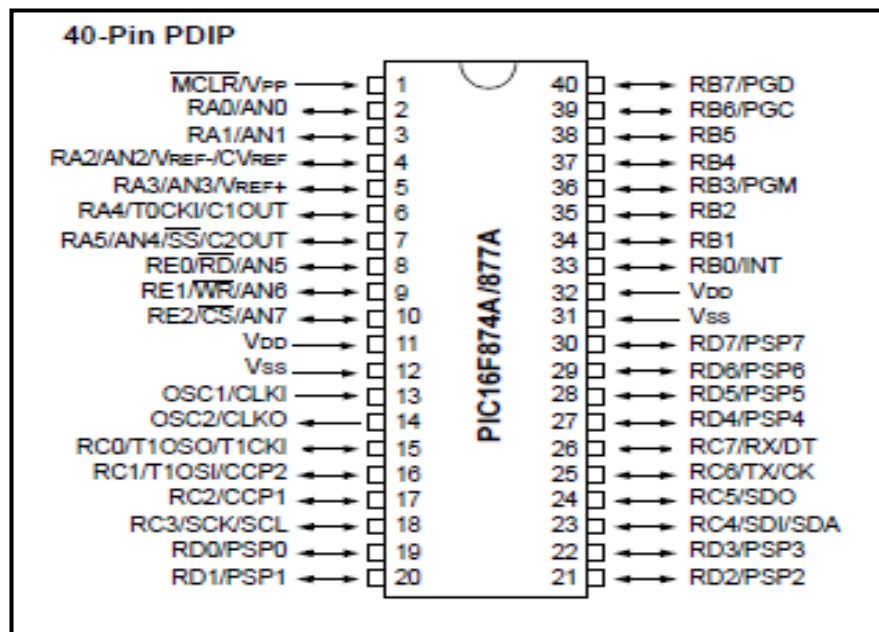


FIGURA 2.5 PIC PIC16F887- Diagrama de pines.

[18]

Descripción de los pines del PIC16F887

En el anexo A1 podemos tener una mejor visualización de los pines de este pic, los cuales han sido utilizados en su totalidad, convertidores analógicos / digitales, I/O, puertos de comunicación TX/RX, y los de alimentación.

Diagrama de bloques del PIC16F887

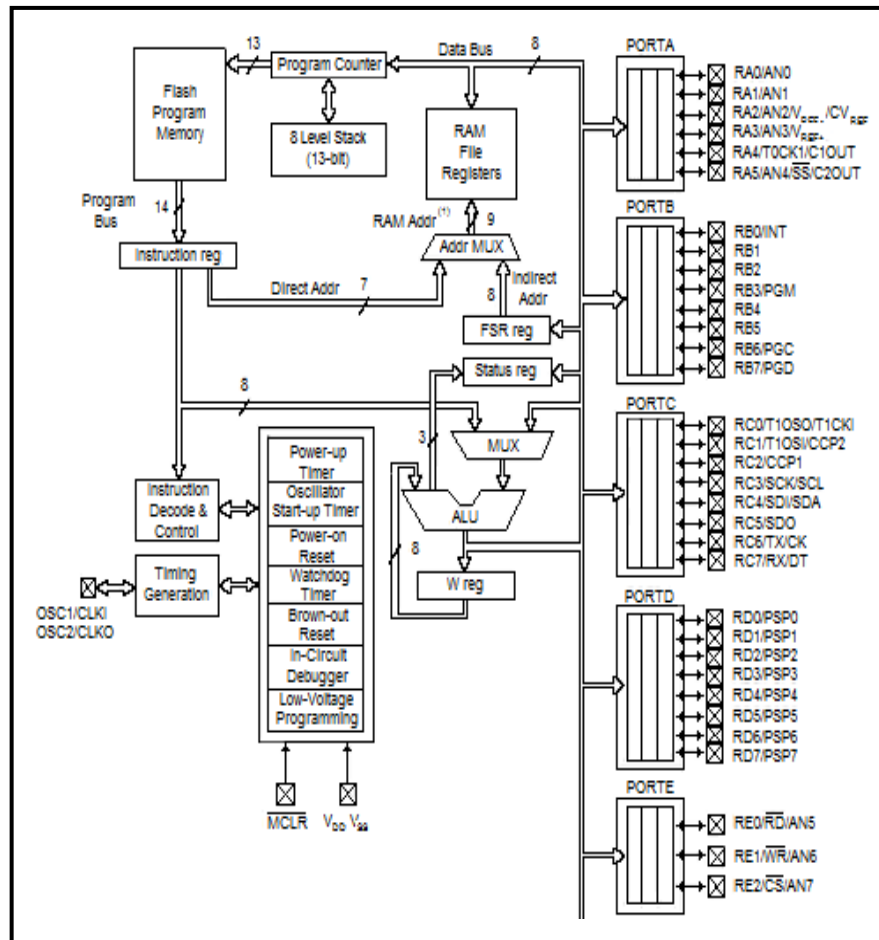


FIGURA 2.6 PIC16F887- Diagrama de bloques. [19]

El reloj del PIC16F887

El reloj u oscilador se utiliza para generar la base de tiempo del microcontrolador. Para la conexión del oscilador se emplean los terminales OSC1 y OSC2 del dispositivo.

Los microcontroladores PIC16F8XX emplean, por cada ciclo de instrucción cuatro ciclos de reloj.

La serie PIC16F87X puede trabajar a una frecuencia de reloj máxima de 20 MHz.

Estos microcontroladores permiten escoger entre cinco tipos distintos de osciladores:

- ❖ LP (Low Power): reloj de bajo consumo, estable, con frecuencia de oscilación de hasta 200 KHz
- ❖ XT (Cristal): estable, frecuencia de oscilación de hasta 4 MHz
- ❖ HS (High Speed): estable, frecuencia de oscilación de hasta 20 MHz
- ❖ RC (Resistor/Capacitor): frecuencia de oscilación dependiente de resistencia, condensador, voltaje de alimentación y temperatura de trabajo. Es el tipo más económico, pero también el más inestable
- ❖ Externo: cuando la señal de reloj es externa, generada por otro circuito

Los modos LP, XT y HS suponen la conexión de un cristal de cuarzo o resonador cerámico entre los OSC1/CLKIN y OSC2/CLKOUT del dispositivo, mientras que el modo RC y Externo sólo ocupan el pin OSC1/CLKIN.

Convertidor analógico digital del PIC16F887

El convertidor analógico-digital que llevan incorporados los microcontroladores de la subfamilia PIC16F8XX es de 10 bits.

El PIC16F887 tiene 8 canales de conversión disponibles. La técnica que utilizan estos dispositivos para la conversión es la de “incremento y comparación”. Consiste en usar un registro auxiliar, cuyo valor se compara con la entrada analógica. Si es menor se incrementa el registro y se vuelve a comparar. De esta manera, hasta que el valor del registro sea lo más aproximado a la entrada analógica sin desbordarse de éste valor.

El rango de voltaje aceptado para conversión de la señal analógica es de 0 V a 5 V.

2.5.3. PIC18F4550 (Para la conexión USB)

Este tipo de microcontrolador es fabricado por la empresa MICROCHIP Technology INC. Se destaca por su versatilidad, velocidad y disponibilidad de herramientas para su programación entre otros.

Tiene incluido una memoria Flash USB y control de flujo de datos. Soporta USB low speed (1.5Mb/s) y full speed (12Mb/s).

Características principales

Se destacan las siguientes:

- ❖ Rango de voltaje de operación: 2V – 5V
- ❖ Timer0: Timer/Counter de 8 ó 16 bits
- ❖ Timer1: Timer/Counter de 16 bits
- ❖ Timer2: Timer de 8 bits
- ❖ Timer3: Timer/Counter de 16 bits
- ❖ 35 Pines I/O (Entrada/Salida)
- ❖ 48 MHz como máxima frecuencia de trabajo
- ❖ 35 Instrucciones

- ❖ Núcleo de 8 bits
- ❖ 256 bytes de memoria EEPROM
- ❖ 2KB de memoria RAM
- ❖ Oscilador interno y externo

Diagrama de pines del PIC18F4550

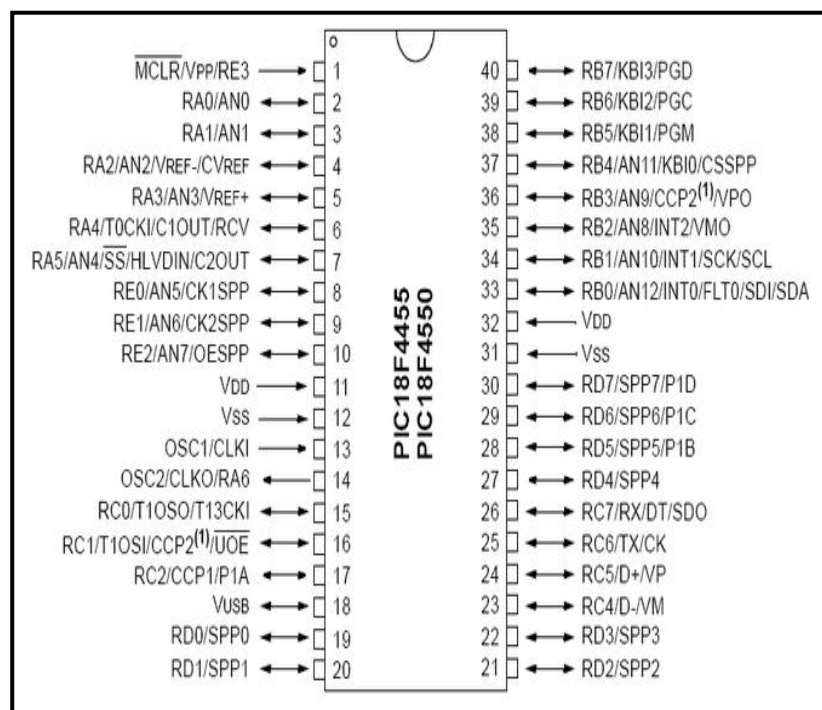


FIGURA 2.7 Diagrama de pines del PIC18F4550. [20]

Descripción de los pines del PIC18F4550

En el anexo A2 podemos tener una mejor visualización de los pines de este pic, los cuales han sido utilizados en

su totalidad, convertidores analógicos / digitales, I/O, y la comunicación USB.

Diagrama de bloques del PIC18F4550

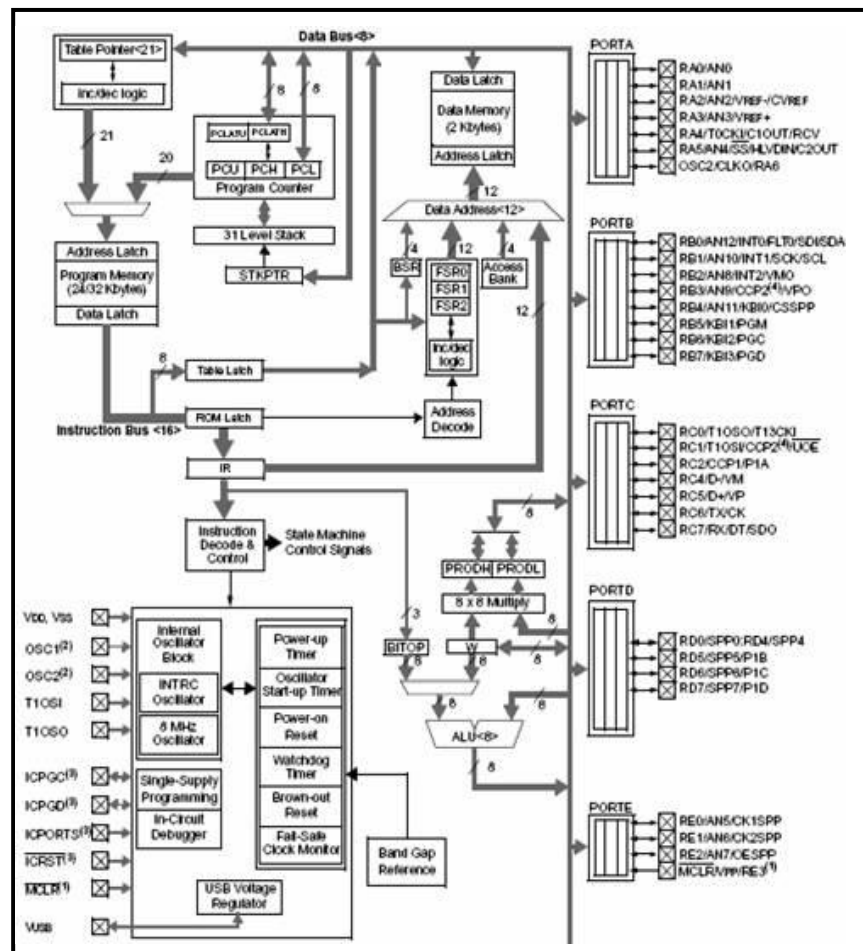


FIGURA 2.8 PIC18F4550- Diagrama de bloques. [21]

CAPÍTULO 3

3. IMPLEMENTACIÓN DE LA CENTRAL DE MONITOREO

3.1. Introducción

En la actualidad se pueden encontrar algunas herramientas de desarrollo para la simulación de circuitos electrónicos, los cuales son importantes para detectar y evitar conexiones erróneas. En la implementación del sistema electrónico se utiliza PROTEUS, desarrollado por Labcenter Electronics.

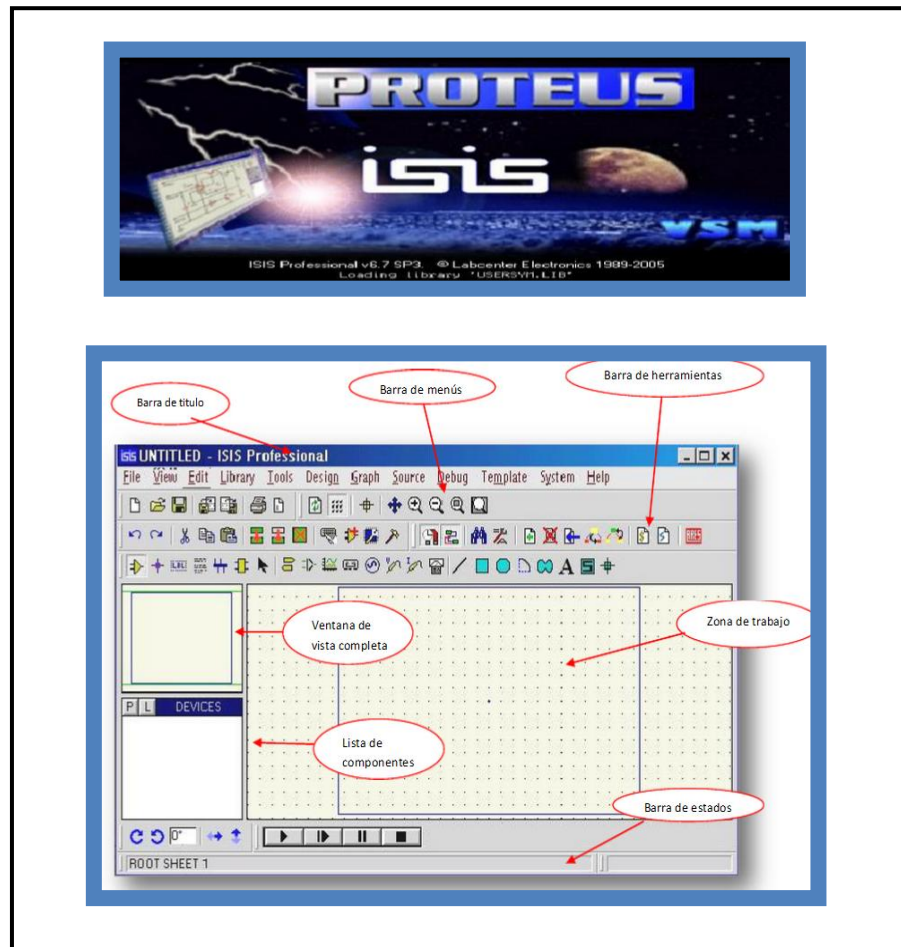


FIGURA 3.1 Imágenes tomadas de la herramienta Proteus, instalado en Windows 7.

La central de monitoreo, se implementa el código fuente de los microcontroladores del transmisor y receptor con uso de la herramienta Pic C, así como también para la comunicación USB.

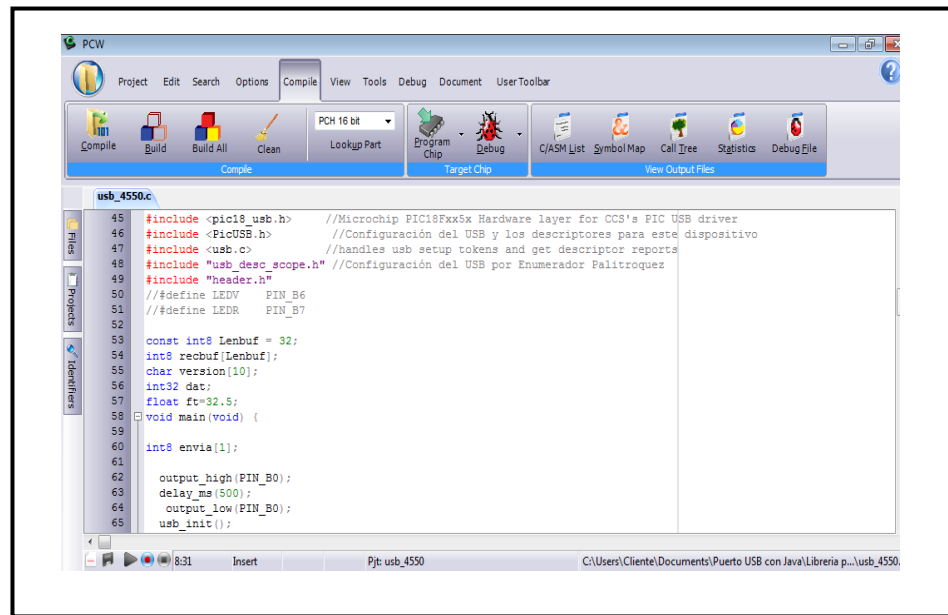


FIGURA 3.2 Imágenes tomadas de la herramienta Pic C instalado en Windows 7.

Las aplicaciones de escritorio son desarrolladas con Netbeans de la Plataforma Java de Sun Microsystems.

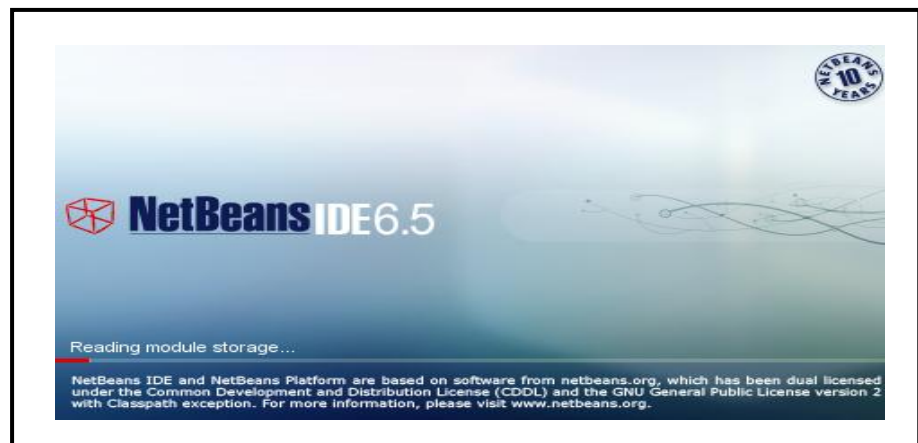


FIGURA 3.3 Imágenes tomadas de la herramienta Netbeans instalado en Windows 7.

3.2. Desarrollo del software

Control de temperatura y transmisión.

En esta sección se procede a desarrollar el código de temperatura y transmisión de los datos sensados. Ver Anexo A3.

Recepción inalámbrica

En esta sección se procede a desarrollar el código de la recepción inalámbrica. Ver Anexo A4.

Conexión USB – PC

En esta sección se procede a desarrollar el código de la conexión USB - PC. Ver Anexo A5.

Aplicación de escritorio desarrollada con lenguaje Java

En esta sección se procede a desarrollar el código de la aplicación de escritorio, la cual es desarrollada en base al lenguaje de programación Java. Ver Anexo A6.

3.3. Implementación de los circuitos

Central de monitoreo

En la central de monitoreo se encuentran los sensores de temperatura LM35; un filtro activo pasa bajo permite atenuar interferencias, causadas por la distancia en que está el sensor o

por ruidos audibles externos y así mejorar la calidad de recepción en el Pic y tener estabilidad de la medición.

El sistema de botoneras envía un bajo cuando no se ha presionado y un alto cuando se presiona, todas excepto Resetn, que es lo contrario.

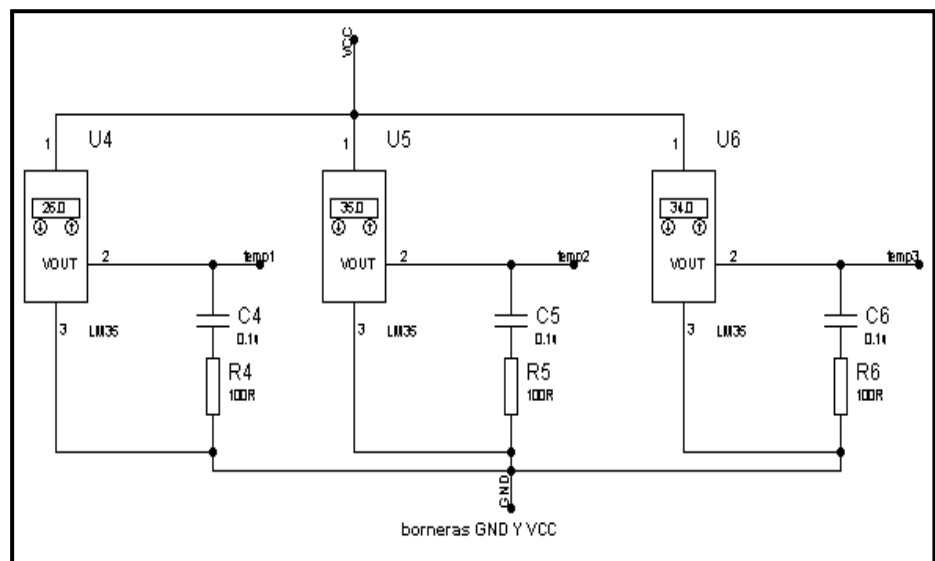


FIGURA 3.4 Circuito de sensores de temperatura LM35

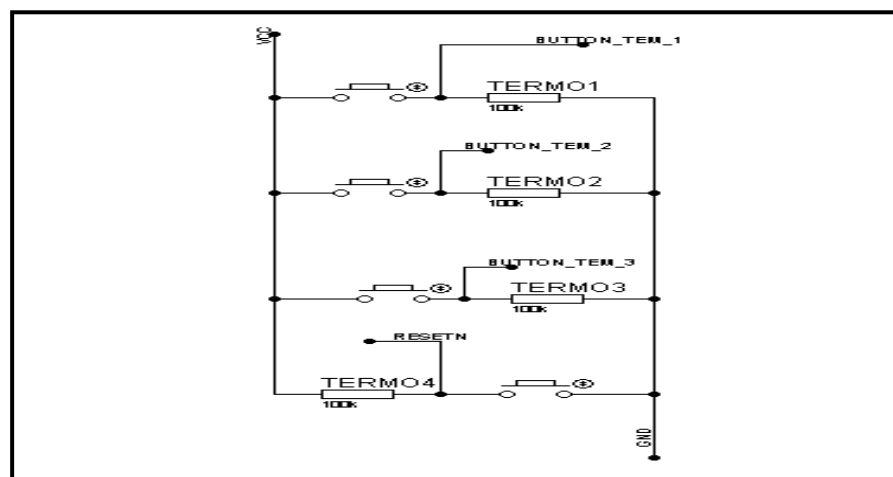


FIGURA 3.5 Circuito de sistemas de botoneras

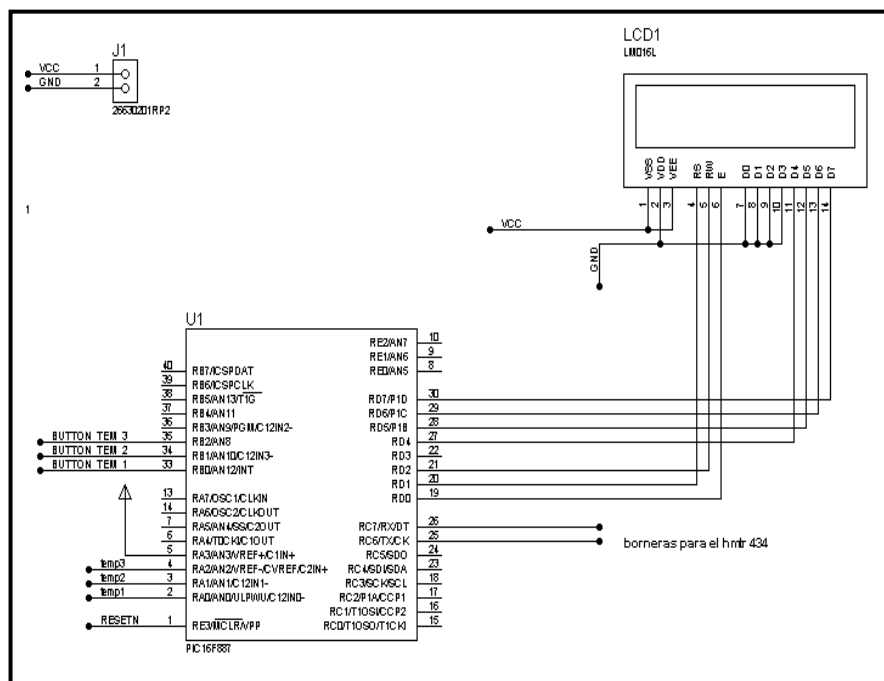


FIGURA 3.6 Circuito central de transmisión y visualización

Recepción inalámbrica

En la Recepción, el módulo HTMR-434 recibe los datos de la central de Monitoreo, los cuales son decodificados y guardados en una variable interna del banco de memoria del PIC 16F887; los datos se convierten en valor de temperatura y se visualizan en una LCD.

Los datos se muestran en forma cíclica de dos segundos para cada termocuna.

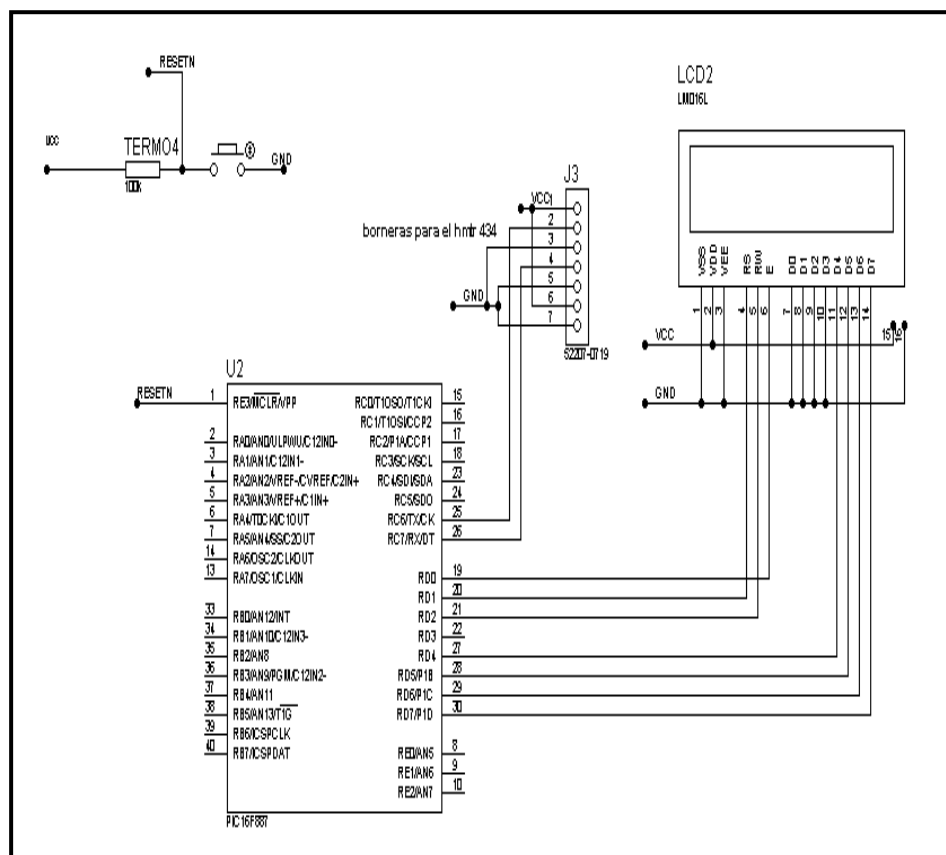


FIGURA 3.7 Circuito de recepción inalámbrica

Conexión USB

La conexión USB se realiza a través del PIC 18F4550, el cual tiene como característica principal una alta velocidad de transmisión. Recibe datos de los convertidores analógicos y los convierte en temperatura; una vez que tiene los valores, los guarda en los bancos de memoria interna del PIC y espera que la autenticación HID sea aceptada.

Realizada la autenticación PIC – PC, se envían los datos obtenidos a una aplicación de la plataforma Java.

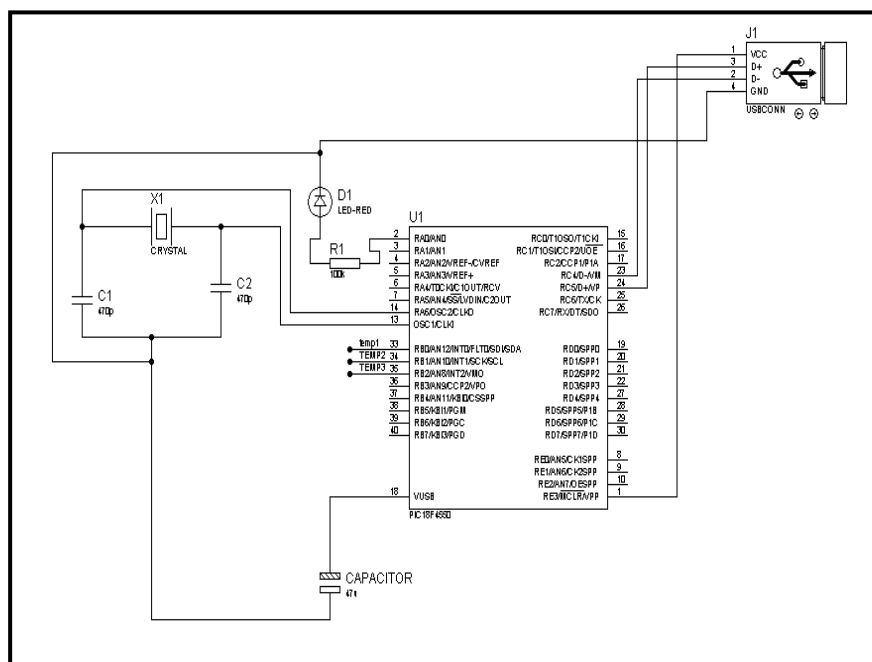


FIGURA 3.8 Circuito de conexión USB

Aplicación de escritorio Java

Instalados los drivers del PIC 18F4550, se tiene la facilidad de comunicación con una interfaz de cualquier sistema operativo, en este caso se ha elegido Windows 7.

La información obtenida por las entradas analógicas es convertida en valores de temperatura que son codificados en paquetes de datos de trama UTF-8. Cuando se autentica la comunicación PIC – PC, se decodifica la trama en números enteros y se los visualiza en una caja de texto de la aplicación creada en Netbeans.

CAPÍTULO 4

4. PRUEBAS DE FUNCIONAMIENTO

Una vez realizadas todas las pruebas en los simuladores, se realiza el cableado y ensamblaje de todos los elementos en el protoboard, para finalmente, constatados todos los posibles errores, se procede a la construcción de las tarjetas de circuito impreso (PCB).

Se utilizan cajas de acrílico transparente para que se pueda tener una mejor apreciación de los componentes utilizados.



FIGURA 4.1 Central de monitoreo vista frontal

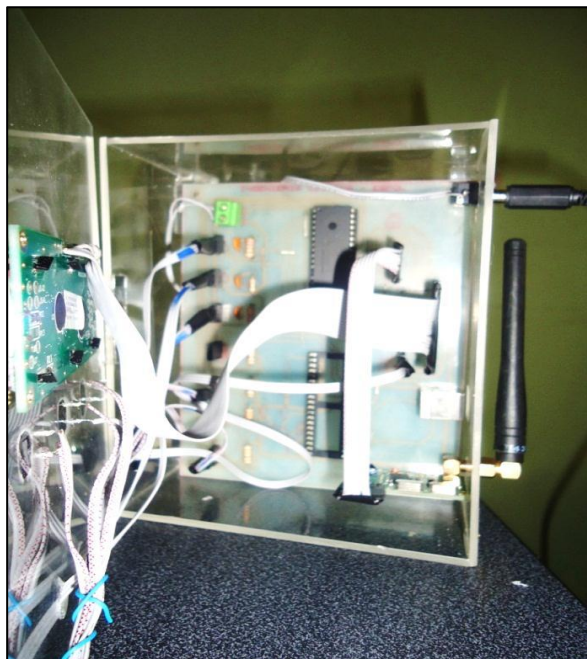


FIGURA 4.2 Central de monitoreo vista interna

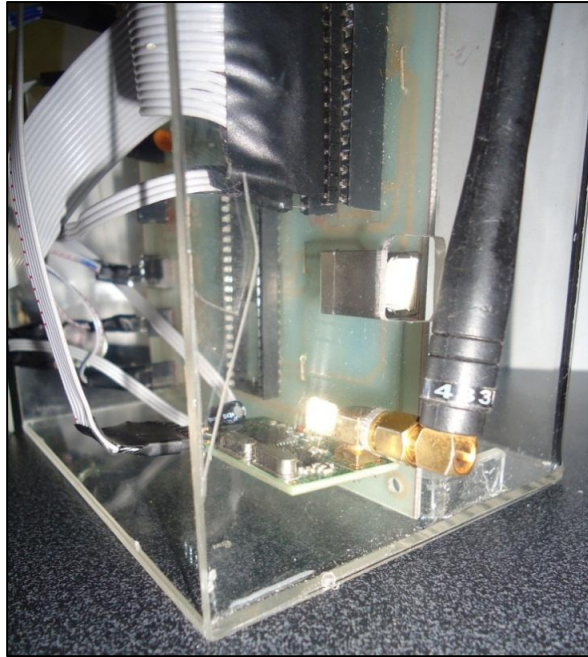


FIGURA 4.3 Central de monitoreo transmisión inalámbrica

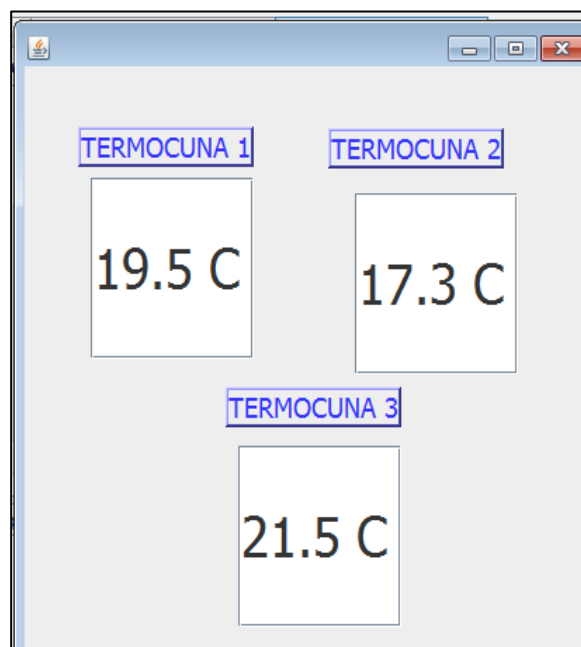


FIGURA 4.4 Aplicación Java para la conexión USB

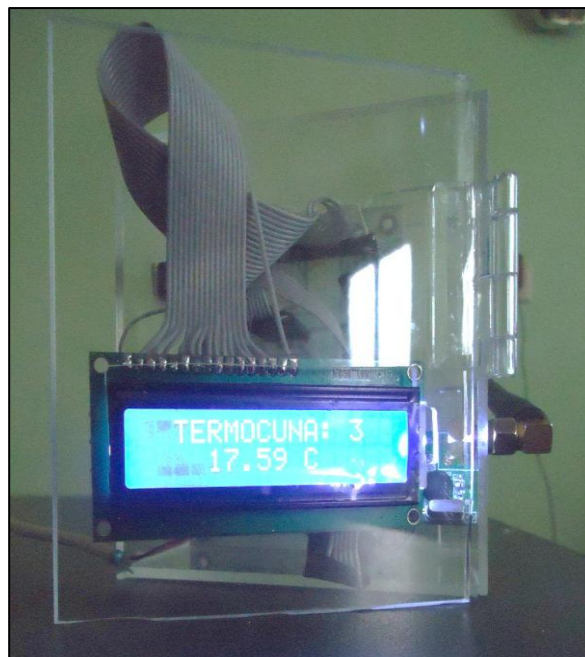


FIGURA 4.5 Central de recepción inalámbrica vista exterior

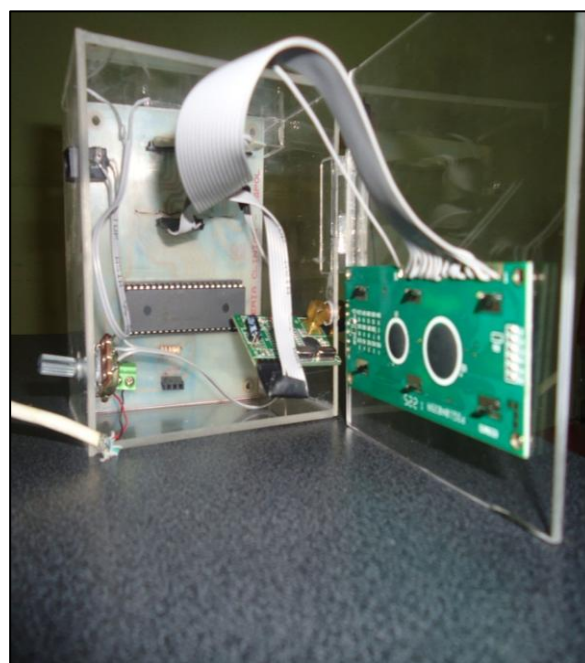


FIGURA 4.6 Central de recepción inalámbrica vista interior



FIGURA 4.7 Central de recepción inalámbrica



FIGURA 4.8 Recepción – Transmisión funcionando

CONCLUSIONES

1. Se ha construido un sistema de monitoreo electrónico y automático para ser colocado en la central de enfermeras, el cual es capaz de facilitar la supervisión de la temperatura del ambiente de las termocunas de neonatos.
2. Con el uso de esta central de monitoreo, basado en un método de medición de temperatura no invasivo, se ha minimizado la interacción de la enfermera – neonato, a menos que ésta sea necesaria.

3. Se ha podido demostrar con la construcción del proyecto, el necesario uso de la tecnología en la medicina.
4. Mediante el uso de los transceivers HM –TR 434, se logra establecer una comunicación inalámbrica de manera eficiente, obteniendo resultados exactos en la transmisión y recepción de datos, haciendo de este sistema independiente y libre de cables en el ambiente de las Unidades de Cuidados Intensivos.
5. La correcta programación de los PIC 16F887 y 18F4550, maximiza la funcionalidad del equipo. Estableciendo líneas de código correctas y rectificando los errores en programación, se demuestra que los valores sensados por el LM35, así como también la comunicación USB, se realizan con márgenes de error reducidos, aumentando de esta manera la correcta funcionalidad de la programación y el diseño del proyecto.

6. Mediante la aplicación de normas de diseño del PCB, se reducen los errores de construcción, eliminando la presencia de capacitancias parásitas y corrientes de fuga en el circuito.
7. El uso del bus de datos para establecer la conexión de los elementos, otorga una mejor estética al diseño del equipo, además de facilitar el cambio de piezas o elementos en caso de errores en el funcionamiento de los mismos.
8. Al usar acrílico para la caja ensamblada del equipo, se amplía la capacidad de observar la PCB en funcionamiento, así como también la conexión de los diferentes elementos con el fin de prever posibles errores en el funcionamiento.

RECOMENDACIONES

1. Para obtener datos lo más exactos posibles se recomienda tener un buen sensor de temperatura. El sensor LM35 tiene un aumento lineal de ± 0.5 grados, que es parcialmente aceptado debido a que sensa el ambiente del neonato.
2. Teniendo en cuenta que el desarrollo de una buena simulación garantiza en un 80% la funcionalidad del equipo al momento de realizar la implementación, se recomienda ser lo más estricto posible al momento de realizar la misma, con el fin de minimizar el tiempo dedicado a la detección de posibles errores en el desarrollo de la implementación.

3. En el desarrollo de la programación se debe realizar un backup de las líneas de código; así como también un correcto uso de comentarios entre las líneas, con el fin de mejorar la comprensión y corrección, en caso de requerirlo.

BIBLIOGRAFÍA

- [1] RUZA, Francisco. 2002. Cuidados Intensivos Pediátricos. España: Norma-Capitel, 2002. Capítulo 15, pág. 196.
- [2] VILLAMIZAR, Hernando A. 2007. Estudio Ecología Neonatal. Colombia. Congreso Colombiano de Pediatría. Modulo 1, pág. 32.
- [3] VILLAMIZAR, Hernando A. 2007. Estudio Ecología Neonatal. Colombia. Congreso Colombiano de Pediatría. Modulo 1, pág. 33.
- [4] Revista del Hospital J.M Ramos Mejía, edición electrónica – Volumen XIII N°2 – 2008
- [5] RUZA, Francisco. 2002. Cuidados Intensivos Pediátricos. España: Norma-Capitel, 2002. Capítulo 15, pág. 196.
- [6] RUZA, Francisco. 2002. Cuidados Intensivos Pediátricos. Capítulo 15, pág. 196.
- [7] Regulación de la temperatura en el bebé recién nacido, <http://www.infogen.org.mx/Infogen1/servlet/CtrlImpArt?clvarticulo=10864>, fecha de consulta: 14 de Mayo del 2012.

- [8] RUZA, Francisco. 2002. Cuidados Intensivos Pediátricos. España: Norma-Capitel, 2002. Capítulo 15, pág. 198
- [9] RUZA, Francisco. 2002. Cuidados Intensivos Pediátricos. España: Norma-Capitel, 2002. Capítulo 15, pág. 199.
- [10] Datasheet del LM-35, <https://www.national.com/ds/LM/LM35.pdf>, fecha de consulta: 24 de Junio del 2012.
- [11] PALACIOS, Enrique. 2004. Microcontrolador PIC16F84 desarrollo de proyectos. México: AlfaOmega, 2004. Capítulo 13, pág. 187
- [12] Modo Visualizador del LCD
<http://www.disca.upv.es/aperles/web51/modulos/lcd/index.html>, Fecha de consulta: 21 de Junio del 2012.
- [13] PALACIOS, Enrique. 2004. Microcontrolador PIC16F84 desarrollo de proyectos. México: AlfaOmega, 2004. Capítulo 13, pág. 192
- [14] Datasheet del HTMR-434, <http://roboeq.com/PDF/0501018.pdf>, Fecha de consulta: 21 de Junio del 2012.

- [15] Datasheet del HTMR-434, <http://roboeq.com/PDF/0501018.pdf>, Fecha de consulta: 21 de Junio del 2012.
- [16] Generación De Señales Senoidales Mediante PWM Y Filtros Activos De Segundo Orden, <http://www.redalyc.org/src/inicio/ArtPdfRed.jsp?iCve=84921327003>, Fecha de consulta: 21 de Junio del 2012.
- [17] Diferencias entre microcontroladores y microprocesadores, <http://www.monografias.com/trabajos27/microcontroladores/microcontroladores.shtml>, Fecha de consulta: 23 de Junio del 2012.
- [18] <http://ww1.microchip.com/downloads/en/DeviceDoc/41291F.pdf>, Fecha de consulta: 25 de Junio del 2012.
- [19] <http://ww1.microchip.com/downloads/en/DeviceDoc/41291F.pdf>, Fecha de consulta: 25 de Junio del 2012.
- [20] <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>, Fecha de consulta: 25 de Junio del 2012.
- [21] <http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>, Fecha de consulta: 25 de Junio del 2012.

ANEXOS

Anexo A1.- PIC 16F887



PIC16F882/883/884/886/887 Data Sheet

28/40/44-Pin, Enhanced Flash-Based 8-Bit
CMOS Microcontrollers with
nanoWatt Technology

PIC16F882/883/884/886/887

TABLE 1-1: PIC16F882/883/886 PINOUT DESCRIPTION

Name	Function	Input Type	Output Type	Description
RA0/AN0/ULPWU/C12IN0-	RA0	TTL	CMOS	General purpose I/O.
	AN0	AN	—	A/D Channel 0 input.
	ULPWU	AN	—	Ultra Low-Power Wake-up input.
	C12IN0-	AN	—	Comparator C1 or C2 negative input.
RA1/AN1/C12IN1-	RA1	TTL	CMOS	General purpose I/O.
	AN1	AN	—	A/D Channel 1 input.
	C12IN1-	AN	—	Comparator C1 or C2 negative input.
RA2/AN2/VREF-/CVREF/C2IN+	RA2	TTL	CMOS	General purpose I/O.
	AN2	AN	—	A/D Channel 2.
	VREF-	AN	—	A/D Negative Voltage Reference input.
	CVREF	—	AN	Comparator Voltage Reference output.
	C2IN+	AN	—	Comparator C2 positive input.
RA3/AN3/VREF+/C1IN+	RA3	TTL	—	General purpose I/O.
	AN3	AN	—	A/D Channel 3.
	VREF+	AN	—	Programming voltage.
	C1IN+	AN	—	Comparator C1 positive input.
RA4/T0CKI/C1OUT	RA4	TTL	CMOS	General purpose I/O.
	T0CKI	ST	—	Timer0 clock input.
	C1OUT	—	CMOS	Comparator C1 output.
RA5/AN4/SS/C2OUT	RA5	TTL	CMOS	General purpose I/O.
	AN4	AN	—	A/D Channel 4.
	SS	ST	—	Slave Select input.
	C2OUT	—	CMOS	Comparator C2 output.
RA6/OSC2/CLKOUT	RA6	TTL	CMOS	General purpose I/O.
	OSC2	—	XTAL	Master Clear with internal pull-up.
	CLKOUT	—	CMOS	Fosc/4 output.
RA7/OSC1/CLKIN	RA7	TTL	CMOS	General purpose I/O.
	OSC1	XTAL	—	Crystal/Resonator.
	CLKIN	ST	—	External clock input/RC oscillator connection.
RB0/AN12/INT	RB0	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN12	AN	—	A/D Channel 12.
	INT	ST	—	External interrupt.
RB1/AN10/P1C/C12IN3-	RB1	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN10	AN	—	A/D Channel 10.
	P1C	—	CMOS	PWM output.
	C12IN3-	AN	—	Comparator C1 or C2 negative input.
RB2/AN8/P1B	RB2	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN8	AN	—	A/D Channel 8.
	P1B	—	CMOS	PWM output.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

PIC16F882/883/884/886/887

TABLE 1-1: PIC16F882/883/886 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RB3/AN9/PGM/C12IN2-	RB3	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN9	AN	—	A/D Channel 9.
	PGM	ST	—	Low-voltage ICSP™ Programming enable pin.
	C12IN2-	AN	—	Comparator C1 or C2 negative input.
RB4/AN11/P1D	RB4	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN11	AN	—	A/D Channel 11.
	P1D	—	CMOS	PWM output.
RB5/AN13/T1G	RB5	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	AN13	AN	—	A/D Channel 13.
	T1G	ST	—	Timer1 Gate input.
RB6/ICSPCLK	RB6	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPCLK	ST	—	Serial Programming Clock.
RB7/ICSPDAT	RB7	TTL	CMOS	General purpose I/O. Individually controlled interrupt-on-change. Individually enabled pull-up.
	ICSPDAT	ST	CMOS	ICSP™ Data I/O.
RC0/T1OSO/T1CKI	RC0	ST	CMOS	General purpose I/O.
	T1OSO	—	CMOS	Timer1 oscillator output.
	T1CKI	ST	—	Timer1 clock input.
RC1/T1OSI/CCP2	RC1	ST	CMOS	General purpose I/O.
	T1OSI	ST	—	Timer1 oscillator input.
	CCP2	ST	CMOS	Capture/Compare/PWM2.
RC2/P1A/CCP1	RC2	ST	CMOS	General purpose I/O.
	P1A	—	CMOS	PWM output.
	CCP1	ST	CMOS	Capture/Compare/PWM1.
RC3/SCK/SCL	RC3	ST	CMOS	General purpose I/O.
	SCK	ST	CMOS	SPI clock.
	SCL	ST	OD	I ² C™ clock.
RC4/SDI/SDA	RC4	ST	CMOS	General purpose I/O.
	SDI	ST	—	SPI data input.
	SDA	ST	OD	I ² C data input/output.
RC5/SDO	RC5	ST	CMOS	General purpose I/O.
	SDO	—	CMOS	SPI data output.
RC6/TX/CK	RC6	ST	CMOS	General purpose I/O.
	TX	—	CMOS	EUSART asynchronous transmit.
	CK	ST	CMOS	EUSART synchronous clock.
RC7/RX/DT	RC7	ST	CMOS	General purpose I/O.
	RX	ST	—	EUSART asynchronous input.
	DT	ST	CMOS	EUSART synchronous data.
RE3/MCLR/VPP	RE3	TTL	—	General purpose input.
	MCLR	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
Vss	Vss	Power	—	Ground reference.
VDD	VDD	Power	—	Positive supply.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
HV = High Voltage XTAL = Crystal

PIC16F882/883/884/886/887

TABLE 1-2: PIC16F884/887 PINOUT DESCRIPTION (CONTINUED)

Name	Function	Input Type	Output Type	Description
RD7/P1D	RD7	TTL	CMOS	General purpose I/O.
	P1D	AN	—	PWM output.
RE0/AN5	RE0	TTL	CMOS	General purpose I/O.
	AN5	AN	—	A/D Channel 5.
RE1/AN6	RE1	TTL	CMOS	General purpose I/O.
	AN6	AN	—	A/D Channel 6.
RE2/AN7	RE2	TTL	CMOS	General purpose I/O.
	AN7	AN	—	A/D Channel 7.
RE3/MCLR/VPP	RE3	TTL	—	General purpose input.
	MCLR	ST	—	Master Clear with internal pull-up.
	VPP	HV	—	Programming voltage.
Vss	Vss	Power	—	Ground reference.
VDD	VDD	Power	—	Positive supply.

Legend: AN = Analog input or output CMOS = CMOS compatible input or output OD = Open Drain
 TTL = TTL compatible input ST = Schmitt Trigger input with CMOS levels
 HV = High Voltage XTAL = Crystal



PIC18F2455/2550/4455/4550

Data Sheet

28/40/44-Pin, High-Performance,
Enhanced Flash, USB Microcontrollers
with nanoWatt Technology

PIC18F2455/2550/4455/4550

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	PDIP, SOIC			
MCLR/VPP/RE3 MCLR VPP RE3	1	I P I	ST ST	Master Clear (input) or programming voltage (input). Master Clear (Reset) input. This pin is an active-low Reset to the device. Programming voltage input. Digital input.
OSC1/CLKI OSC1 CLKI	9	I I	Analog Analog	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. External clock source input. Always associated with pin function OSC1. (See OSC2/CLKO pin.)
OSC2/CLKO/RA6 OSC2 CLKO RA6	10	O O I/O	— — TTL	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In select modes, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. General purpose I/O pin.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels I = Input
O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.
2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

PIC18F2455/2550/4455/4550

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	PDIP, SOIC			
RA0/AN0	2	I/O	TTL	PORTA is a bidirectional I/O port.
RA0		I	Analog	Digital I/O.
AN0				Analog input 0.
RA1/AN1	3	I/O	TTL	Digital I/O.
RA1		I	Analog	Analog input 1.
AN1				
RA2/AN2/VREF-/CVREF	4	I/O	TTL	Digital I/O.
RA2		I	Analog	Analog input 2.
AN2		I	Analog	A/D reference voltage (low) input.
VREF-		I	Analog	Analog comparator reference output.
CVREF		O	Analog	
RA3/AN3/VREF+	5	I/O	TTL	Digital I/O.
RA3		I	Analog	Analog input 3.
AN3		I	Analog	A/D reference voltage (high) input.
VREF+		I	Analog	
RA4/T0CKI/C1OUT/RCV	6	I/O	ST	Digital I/O.
RA4		I	ST	Timer0 external clock input.
T0CKI		O	—	Comparator 1 output.
C1OUT		I	TTL	External USB transceiver RCV input.
RCV				
RA5/AN4/SS/HLVDIN/C2OUT	7	I/O	TTL	Digital I/O.
RA5		I	Analog	Analog input 4.
AN4		I	TTL	SPI slave select input.
SS		I	Analog	High/Low-Voltage Detect input.
HLVDIN		O	—	Comparator 2 output.
C2OUT				
RA6	—	—	—	See the OSC2/CLKO/RA6 pin.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels I = Input
 O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

PIC18F2455/2550/4455/4550

TABLE 1-2: PIC18F2455/2550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	PDIP, SOIC			
RB0/AN12/INT0/FLT0/SDI/SDA	21			PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0		I/O	TTL	Digital I/O.
AN12		I	Analog	Analog input 12.
INT0		I	ST	External interrupt 0.
FLT0		I	ST	PWM Fault input (CCP1 module).
SDI		I	ST	SPI data in.
SDA		I/O	ST	I ² C™ data I/O.
RB1/AN10/INT1/SCK/SCL	22			
RB1		I/O	TTL	Digital I/O.
AN10		I	Analog	Analog input 10.
INT1		I	ST	External interrupt 1.
SCK		I/O	ST	Synchronous serial clock input/output for SPI mode.
SCL		I/O	ST	Synchronous serial clock input/output for I ² C mode.
RB2/AN8/INT2/VMO	23			
RB2		I/O	TTL	Digital I/O.
AN8		I	Analog	Analog input 8.
INT2		I	ST	External interrupt 2.
VMO		O	—	External USB transceiver VMO output.
RB3/AN9/CCP2/VPO	24			
RB3		I/O	TTL	Digital I/O.
AN9		I	Analog	Analog input 9.
CCP2 ⁽¹⁾		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
VPO		O	—	External USB transceiver VPO output.
RB4/AN11/KBI0	25			
RB4		I/O	TTL	Digital I/O.
AN11		I	Analog	Analog input 11.
KBI0		I	TTL	Interrupt-on-change pin.
RB5/KBI1/PGM	26			
RB5		I/O	TTL	Digital I/O.
KBI1		I	TTL	Interrupt-on-change pin.
PGM		I/O	ST	Low-Voltage ICSP™ Programming enable pin.
RB6/KBI2/PGC	27			
RB6		I/O	TTL	Digital I/O.
KBI2		I	TTL	Interrupt-on-change pin.
PGC		I/O	ST	In-Circuit Debugger and ICSP programming clock pin.
RB7/KBI3/PGD	28			
RB7		I/O	TTL	Digital I/O.
KBI3		I	TTL	Interrupt-on-change pin.
PGD		I/O	ST	In-Circuit Debugger and ICSP programming data pin.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels I = Input
O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

PIC18F2455/2550/4455/4550

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
RA0/AN0	2	19	19	I/O	TTL	PORTA is a bidirectional I/O port.
RA0				I	Analog	Digital I/O.
AN0						Analog input 0.
RA1/AN1	3	20	20	I/O	TTL	Digital I/O.
RA1				I	Analog	Analog input 1.
AN1						
RA2/AN2/REF-/CVREF	4	21	21	I/O	TTL	Digital I/O.
RA2				I	Analog	Analog input 2.
AN2				I	Analog	A/D reference voltage (low) input.
VREF-				O	Analog	Analog comparator reference output.
CVREF						
RA3/AN3/REF+	5	22	22	I/O	TTL	Digital I/O.
RA3				I	Analog	Analog input 3.
AN3				I	Analog	A/D reference voltage (high) input.
VREF+						
RA4/T0CKI/C1OUT/RCV	6	23	23	I/O	ST	Digital I/O.
RA4				I	ST	Timer0 external clock input.
T0CKI				O	—	Comparator 1 output.
C1OUT				I	TTL	External USB transceiver RCV input.
RCV						
RA5/AN4/SS/HLVDIN/C2OUT	7	24	24	I/O	TTL	Digital I/O.
RA5				I	Analog	Analog input 4.
AN4				I	TTL	SPI slave select input.
SS				I	Analog	High/Low-Voltage Detect input.
HLVDIN				O	—	Comparator 2 output.
C2OUT						
RA6	—	—	—	—	—	See the OSC2/CLKO/RA6 pin.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels I = Input
 O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

PIC18F2455/2550/4455/4550

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
RC0/T1OSO/T13CKI RC0 T1OSO T13CKI	15	34	32	I/O O I	ST — ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI/CCP2/ UOE RC1 T1OSI CCP2 ⁽²⁾ UOE	16	35	35	I/O I I/O O	ST CMOS ST —	Digital I/O. Timer1 oscillator input. Capture 2 input/Compare 2 output/PWM2 output. External USB transceiver OE output.
RC2/CCP1/P1A RC2 CCP1 P1A	17	36	36	I/O I/O O	ST ST TTL	Digital I/O. Capture 1 input/Compare 1 output/PWM1 output. Enhanced CCP1 PWM output, channel A.
RC4/D-/VM RC4 D- VM	23	42	42	I I/O I	TTL — TTL	Digital input. USB differential minus line (input/output). External USB transceiver VM input.
RC5/D+/VP RC5 D+ VP	24	43	43	I I/O I	TTL — TTL	Digital input. USB differential plus line (input/output). External USB transceiver VP input.
RC6/TX/CK RC6 TX CK	25	44	44	I/O O I/O	ST — ST	Digital I/O. EUSART asynchronous transmit. EUSART synchronous clock (see RX/DT).
RC7/RX/DT/SDO RC7 RX DT SDO	26	1	1	I/O I I/O O	ST ST ST —	Digital I/O. EUSART asynchronous receive. EUSART synchronous data (see TX/CK). SPI data out.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels I = Input
O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

PIC18F2455/2550/4455/4550

TABLE 1-3: PIC18F4455/4550 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number			Pin Type	Buffer Type	Description
	PDIP	QFN	TQFP			
RE0/AN5/CK1SPP RE0 AN5 CK1SPP	8	25	25	I/O I O	ST Analog —	PORTC is a bidirectional I/O port. Digital I/O. Analog input 5. SPP clock 1 output.
RE1/AN6/CK2SPP RE1 AN6 CK2SPP	9	26	26	I/O I O	ST Analog —	Digital I/O. Analog input 6. SPP clock 2 output.
RE2/AN7/OESPP RE2 AN7 OESPP	10	27	27	I/O I O	ST Analog —	Digital I/O. Analog input 7. SPP output enable output.
RE3	—	—	—	—	—	See MCLR/VPP/RE3 pin.
VSS	12, 31	6, 30, 31	6, 29	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	7, 8, 28, 29	7, 28	P	—	Positive supply for logic and I/O pins.
VUSB	18	37	37	P	—	Internal USB 3.3V voltage regulator output, positive supply for the USB transceiver.
NC/ICCK/ICPGC ⁽³⁾ ICCK ICPGC	—	—	12	I/O I/O	ST ST	No Connect or dedicated ICD/ICSP™ port clock. In-Circuit Debugger clock. ICSP programming clock.
NC/ICDT/ICPGD ⁽³⁾ ICDT ICPGD	—	—	13	I/O I/O	ST ST	No Connect or dedicated ICD/ICSP port clock. In-Circuit Debugger data. ICSP programming data.
NC/ICRST/ICVPP ⁽³⁾ ICRST ICVPP	—	—	33	I P	— —	No Connect or dedicated ICD/ICSP port Reset. Master Clear (Reset) input. Programming voltage input.
NC/ICPORTS ⁽³⁾ ICPORTS	—	—	34	P	—	No Connect or 28-pin device emulation. Enable 28-pin device emulation when connected to VSS.
NC	—	13	—	—	—	No Connect.

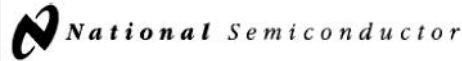
Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels I = Input
O = Output P = Power

Note 1: Alternate assignment for CCP2 when CCP2MX Configuration bit is cleared.

2: Default assignment for CCP2 when CCP2MX Configuration bit is set.

3: These pins are No Connect unless the ICPRT Configuration bit is set. For NC/ICPORTS, the pin is No Connect unless ICPRT is set and the DEBUG Configuration bit is cleared.

Anexo A3.- LM35



December 1994

LM35/LM35A/LM35C/LM35CA/LM35D 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°C 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 $60\text{ }\mu\text{A}$ from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55°C to $+150^{\circ}\text{C}$ temperature range, while the LM35C is rated for a -40°C to $+110^{\circ}\text{C}$ range (-10°C with improved accuracy). The LM35 series is

available packaged 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-202 package.

Features

- Calibrated directly in ° Celsius (Centigrade)
- Linear $+10.0\text{ mV}/^{\circ}\text{C}$ scale factor
- 0.5°C accuracy guaranteeable (at $+25^{\circ}\text{C}$)
- Rated for full -55°C 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 $60\text{ }\mu\text{A}$ current drain
- Low self-heating, 0.08°C in still air
- Nonlinearity only $\pm 1/4^{\circ}\text{C}$ typical
- Low impedance output, $0.1\text{ }\Omega$ for 1 mA load

Connection Diagrams

TO-46
Metal Can Package*



TL/H/5516-1

*Case is connected to negative pin (GND)
Order Number LM35H, LM35AH,
LM35CH, LM35CAH or LM35DH
See NS Package Number H03H

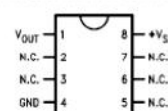
TO-92
Plastic Package



TL/H/5516-2

Order Number LM35CZ,
LM35CAZ or LM35DZ
See NS Package Number Z03A

SO-8
Small Outline Molded Package



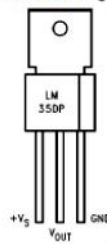
TL/H/5516-21

Top View

N.C. = No Connection

Order Number LM35DM
See NS Package Number M08A

TO-202
Plastic Package

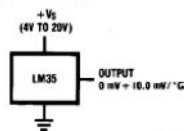


TL/H/5516-24

Order Number LM35DP
See NS Package Number P03A

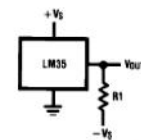
TRI-STATE® is a registered trademark of National Semiconductor Corporation.

Typical Applications



TL/H/5516-3

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



TL/H/5516-4

Choose $R_1 = -V_G/50\text{ }\mu\text{A}$

$V_{OUT} = +1,500\text{ mV}$ at $+150^{\circ}\text{C}$
 $= +250\text{ mV}$ at $+25^{\circ}\text{C}$
 $= -550\text{ mV}$ at -55°C

FIGURE 2. Full-Range Centigrade
Temperature Sensor

Absolute Maximum Ratings (Note 10)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	+35V to -0.2V
Output Voltage	+6V to -1.0V
Output Current	10 mA
Storage Temp., TO-46 Package,	-60°C to +180°C
TO-92 Package,	-60°C to +150°C
SO-8 Package,	-65°C to +150°C
TO-202 Package,	-65°C to +150°C

Lead Temp.:

TO-46 Package, (Soldering, 10 seconds)	300°C
TO-92 Package, (Soldering, 10 seconds)	260°C
TO-202 Package, (Soldering, 10 seconds)	+230°C

SO Package (Note 12):

Vapor Phase (60 seconds) 215°C

Infrared (15 seconds) 220°C

ESD Susceptibility (Note 11) 2500V

Specified Operating Temperature Range: T_{MIN} to T_{MAX}

(Note 2)

LM35, LM35A -55°C to +150°C

LM35C, LM35CA -40°C to +110°C

LM35D 0°C to +100°C

Electrical Characteristics (Note 1) (Note 6)

Parameter	Conditions	LM35A			LM35CA			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy (Note 7)	$T_A = +25^\circ\text{C}$	± 0.2	± 0.5		± 0.2	± 0.5	± 1.0	$^\circ\text{C}$
	$T_A = -10^\circ\text{C}$	± 0.3			± 0.3			$^\circ\text{C}$
	$T_A = T_{MAX}$	± 0.4	± 1.0		± 0.4	± 1.0		$^\circ\text{C}$
	$T_A = T_{MIN}$	± 0.4	± 1.0		± 0.4		± 1.5	$^\circ\text{C}$
Nonlinearity (Note 8)	$T_{MIN} \leq T_A \leq T_{MAX}$	± 0.18		± 0.35	± 0.15		± 0.3	$^\circ\text{C}$
Sensor Gain (Average Slope)	$T_{MIN} \leq T_A \leq T_{MAX}$	$+10.0$	$+9.9$, $+10.1$		$+10.0$		$+9.9$, $+10.1$	mV/ $^\circ\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1$ mA	$T_A = +25^\circ\text{C}$ $T_{MIN} \leq T_A \leq T_{MAX}$	± 0.4 ± 0.5	± 1.0	± 3.0	± 0.4 ± 0.5	± 1.0	± 3.0	mV/mA mV/mA
Line Regulation (Note 3)	$T_A = +25^\circ\text{C}$ $4V \leq V_S \leq 30V$	± 0.01 ± 0.02	± 0.05	± 0.1	± 0.01 ± 0.02	± 0.05	± 0.1	mV/V mV/V
Quiescent Current (Note 9)	$V_S = +5V$, $+25^\circ\text{C}$	56	67		56	67		μA
	$V_S = +5V$	105		131	91		114	μA
	$V_S = +30V$, $+25^\circ\text{C}$	56.2	68		56.2	68		μA
	$V_S = +30V$	105.5		133	91.5		116	μA
Change of Quiescent Current (Note 3)	$4V \leq V_S \leq 30V$, $+25^\circ\text{C}$	0.2	1.0		0.2	1.0		μA
	$4V \leq V_S \leq 30V$	0.5		2.0	0.5		2.0	μA
Temperature Coefficient of Quiescent Current		$+0.39$		$+0.5$	$+0.39$		$+0.5$	$\mu\text{A}/^\circ\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	$+1.5$		$+2.0$	$+1.5$		$+2.0$	$^\circ\text{C}$
Long Term Stability	$T_J = T_{MAX}$, for 1000 hours	± 0.08			± 0.08			$^\circ\text{C}$

Note 1: Unless otherwise noted, these specifications apply: $-55^\circ\text{C} \leq T_J \leq +150^\circ\text{C}$ for the LM35 and LM35A; $-40^\circ\text{C} \leq T_J \leq +110^\circ\text{C}$ for the LM35C and LM35CA; and $0^\circ\text{C} \leq T_J \leq +100^\circ\text{C}$ for the LM35D. $V_S = +5V$ dc and $I_{QAO} = 50 \mu\text{A}$, in the circuit of Figure 2. These specifications also apply from $+2^\circ\text{C}$ to T_{MAX} in the circuit of Figure 1. Specifications in **boldface** apply over the full rated temperature range.

Note 2: Thermal resistance of the TO-46 package is $40^\circ\text{C}/\text{W}$, junction to ambient, and $24^\circ\text{C}/\text{W}$ junction to case. Thermal resistance of the TO-92 package is $180^\circ\text{C}/\text{W}$ junction to ambient. Thermal resistance of the small outline molded package is $220^\circ\text{C}/\text{W}$ junction to ambient. Thermal resistance of the TO-202 package is $85^\circ\text{C}/\text{W}$ junction to ambient. For additional thermal resistance information see table in the Applications section.

Electrical Characteristics (Note 1) (Note 6) (Continued)

Parameter	Conditions	LM35			LM35C, LM35D			Units (Max.)
		Typical	Tested Limit (Note 4)	Design Limit (Note 5)	Typical	Tested Limit (Note 4)	Design Limit (Note 5)	
Accuracy, LM35, LM35C (Note 7)	$T_A = +25^{\circ}\text{C}$	± 0.4	± 1.0		± 0.4	± 1.0		$^{\circ}\text{C}$
	$T_A = -10^{\circ}\text{C}$	± 0.5			± 0.5		± 1.5	$^{\circ}\text{C}$
	$T_A = T_{\text{MAX}}$	± 0.8	± 1.5		± 0.8		± 1.5	$^{\circ}\text{C}$
	$T_A = T_{\text{MIN}}$	± 0.8		± 1.5	± 0.8		± 2.0	$^{\circ}\text{C}$
Accuracy, LM35D (Note 7)	$T_A = +25^{\circ}\text{C}$				± 0.6	± 1.5		$^{\circ}\text{C}$
	$T_A = T_{\text{MAX}}$				± 0.9		± 2.0	$^{\circ}\text{C}$
	$T_A = T_{\text{MIN}}$				± 0.9		± 2.0	$^{\circ}\text{C}$
Nonlinearity (Note 8)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.3		± 0.5	± 0.2		± 0.5	$^{\circ}\text{C}$
Sensor Gain (Average Slope)	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	$+10.0$	$+9.8, +10.2$		$+10.0$		$+9.8, +10.2$	mV/ $^{\circ}\text{C}$
Load Regulation (Note 3) $0 \leq I_L \leq 1 \text{ mA}$	$T_A = +25^{\circ}\text{C}$	± 0.4	± 2.0		± 0.4	± 2.0		mV/mA
	$T_{\text{MIN}} \leq T_A \leq T_{\text{MAX}}$	± 0.5		± 5.0	± 0.5		± 5.0	mV/mA
Line Regulation (Note 3)	$T_A = +25^{\circ}\text{C}$	± 0.01	± 0.1		± 0.01	± 0.1		mV/V
	$4\text{V} \leq V_S \leq 30\text{V}$	± 0.02		± 0.2	± 0.02		± 0.2	mV/V
Quiescent Current (Note 9)	$V_S = +5\text{V}, +25^{\circ}\text{C}$	56	80		56	80		μA
	$V_S = +5\text{V}$	105		158	91		138	μA
	$V_S = +30\text{V}, +25^{\circ}\text{C}$	56.2	82		56.2	82		μA
	$V_S = +30\text{V}$	105.5		161	91.5		141	μA
Change of Quiescent Current (Note 3)	$4\text{V} \leq V_S \leq 30\text{V}, +25^{\circ}\text{C}$	0.2	2.0		0.2	2.0		μA
	$4\text{V} \leq V_S \leq 30\text{V}$	0.5		3.0	0.5		3.0	μA
Temperature Coefficient of Quiescent Current		$+0.39$		$+0.7$	$+0.39$		$+0.7$	$\mu\text{A}/^{\circ}\text{C}$
Minimum Temperature for Rated Accuracy	In circuit of Figure 1, $I_L = 0$	$+1.5$		$+2.0$	$+1.5$		$+2.0$	$^{\circ}\text{C}$
Long Term Stability	$T_J = T_{\text{MAX}}$, for 1000 hours	± 0.08			± 0.08			$^{\circ}\text{C}$

Note 3: Regulation is measured at constant junction temperature, using pulse testing with a low duty cycle. Changes in output due to heating effects can be computed by multiplying the internal dissipation by the thermal resistance.

Note 4: Tested Limits are guaranteed and 100% tested in production.

Note 5: Design Limits are guaranteed (but not 100% production tested) over the indicated temperature and supply voltage ranges. These limits are not used to calculate outgoing quality levels.

Note 6: Specifications in **boldface** apply over the full rated temperature range.

Note 7: Accuracy is defined as the error between the output voltage and $10\text{mV}/^{\circ}\text{C}$ times the device's case temperature, at specified conditions of voltage, current, and temperature (expressed in $^{\circ}\text{C}$).

Note 8: Nonlinearity is defined as the deviation of the output-voltage-versus-temperature curve from the best-fit straight line, over the device's rated temperature range.

Note 9: Quiescent current is defined in the circuit of Figure 1.

Note 10: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions. See Note 1.

Note 11: Human body model, 100 pF discharged through a 1.5 k Ω resistor.

Note 12: See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" or the section titled "Surface Mount" found in a current National Semiconductor Linear Data Book for other methods of soldering surface mount devices.

Anexo A4.- Código de control de temperatura y transmisión.

```
/******  
  
*****  
  
**      INGENIERÍA CLÍNICA  
  
**      Profesor de Cátedra:      Ing. Miguel Yapur.  
  
**      Integrantes de grupo:      Willam Ontaneda Yaguana.  
**                                      Diana Vivanco Dávila.  
  
**      Título: Transmisión de datos de la central de monitoreo  
  
**      Descripción:  
  
**      Sensa los valores de las entradas analógicas AN1,AN2,  
**      AN3 y convierte su valor en la temperatura, mostrando los  
**      datos en una LCD para luego ser transmitidos con el  
**      uso del transmisor inalámbrico  
  
*****  
  
*****/  
  
// DECLARACIÓN DE LOS FUSES DEL  
MICROCONTROLADOR  
  
#include <16f887.h>    // Agrega la librería del microcontrolador  
  
#device adc=10         // Usa resolución de 10 bits.  
  
#fuses INTRC_IO        // Oscilador interno.  
  
#fuses NOWDT           // No utilizamos el perro guardián.  
  
// Seleccionamos la frecuencia interna  
  
#use delay(clock=4000000) #use
```

```

// velocidad de transmisión

rs232(baud=9600,xmit=pin_c6,rcv=pin_c7)

#BYTE TRISB=0X00          // configuramos los puertos

#BYTE PORTB=0X00

#include <LCD.C>          // Agrega la librería del LCD

// FUNCIÓN PRINCIPAL DEL MAIN

void main ()

{

    // Declaramos las variables internas

    int val1;

    int opcion=1;

    float tem,tem1;

    int numTem=0;

    lcd_init();

    // Seleccionamos las entradas analógicas

    set_tris_b(0xff);

    // velocidad de conversión analógica

    setup_adc (adc_clock_div_32);

    //DESPLIEGUE VISUAL EN LA LCD

    lcd_gotoxy(1,1);

    printf(lcd_putc," INGENIERIA\n");

    printf(lcd_putc," CLINICA\n");

```

```

delay_ms (2000);

lcd_gotoxy(1,1);

printf(lcd_putc," W. ONTANEDA\n");

printf(lcd_putc," D. VIVANCO\n");

delay_ms (2000);

printf(lcd_putc,"\f \n");

while(1)
{
    if(numTem==3)

        numTem=0;

        set_adc_channel (numTem); //Elige canal a medir RA0

        tem=read_adc (); //Hace conversión AD

tem1=(tem/2); //Pasa binario a °C

val1=(int)tem1;

// llamamos a la función de transmisión de información

put(val1);

// Verificamos si se ha seleccionado una botonera

if(!input(PIN_B0)==0)

    opcion=1;

else if(!input(PIN_B1)==0)

    opcion=2;

else if(!input(PIN_B2)==0)

```

```

        opcion=3;

else

        opcion=opcion;

switch(opcion)

{

        case 1:

set_adc_channel (0); //Elige canal a medir RA0

tem = read_adc (); //Hace conversión AD

lcd_gotoxy(1,1);

printf(lcd_putc," TERMOCUNA: 1\n");

// tasa de conversión A/D

printf(lcd_putc,"    %f\n",tem*0.48875);

break;

        case 2:

set_adc_channel (1); //Elige canal a medir RA0

tem=read_adc (); //Hace conversión AD

lcd_gotoxy(1,1);

printf(lcd_putc," TERMOCUNA: 2\n");

printf(lcd_putc,"    %f\n",tem*0.48875);

break;

        case 3:

set_adc_channel (2); //Elige canal a medir RA0

```

```

// delay_us (20);

tem=read_adc (); //Hace conversión AD

lcd_gotoxy(1,1);

printf(lcd_putc," TERMOCUNA: 3\n");

printf(lcd_putc,"    %f\n",tem*0.48875);

break;

}

delay_ms (1000);

numTem++;

}

}

```

Anexo A5: Recepción inalámbrica

```

/*****
*****

**      INGENIERÍA CLÍNICA

**      Profesor de Cátedra:          Ing. Miguel Yapur.

**      Integrantes de grupo:      Willam Ontaneda Yaguana.

**                                      Diana Vivanco Dávila.

**      Título: Recepción de datos de la central de monitoreo

**      Descripción:

**      Sensa los valores de las entradas inalámbricas

**      y convierte su valor en la temperatura, mostrando los

```

****** datos en una LCD

*****/

#include <16F887.h> // Descripción De Las Librerías A Usar

#include <LCD.C>

// Declarando Los Fuses De Configuración

#FUSES INTRC,NOWDT

#use delay (clock=4000000)

// Velocidad De Comunicación

#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7)

// Declarando Puertos De Salida (Alarmas)

#BYTE TRISB=0X00

// Asignando Salidas En Bajo

#BYTE PORTB=0X00

int valor,val;

float data;

int cont=1;

int opcion;

#int_RDA

RDA_isr()

{

 valor=GETC(); // Receptando Valores Inalámbricos

```

}

void main ()

{
    set_tris_b(0x00); // Salidas Digitales

    lcd_init();

    // Interrupciones De Comunicación

    enable_interrupts(INT_RDA);

    enable_interrupts(GLOBAL);

    // Despliegue Visual Del Lcd

    lcd_gotoxy(1,1);

    printf(lcd_putc," INGENIERIA\n");

    printf(lcd_putc," CLINICA\n");

    delay_ms (2000);

    lcd_gotoxy(1,1);

    printf(lcd_putc," W. ONTANEDA\n");

    printf(lcd_putc," D. VIVANCO\n");

    delay_ms (2000);

    printf(lcd_putc,"\f \n");

    while(valor!=0)

    {

        if(cont==4)

            cont=1;

```

```

    lcd_gotoxy(1,1);

    printf(lcd_putc,"  TERMOCUNA: %d\n",cont);

    printf(lcd_putc,"    %3d,%d C\n",(valor/10)+19,valor%10);

    printf(lcd_putc,"    %2.2f C\n",valor*2*0.48875);

    // Conversión A Grados Centígrados

    data=valor*2*0.48875*10;

    delay_ms (1000);

    cont++;

    }

}

```

Anexo A6: Conexión USB - PC

```

/*****
*****

**      INGENIERÍA CLÍNICA

**      Profesor de Cátedra:      Ing. Miguel Yapur.

**      Integrantes de grupo:      Willam Ontaneda Yaguana.

**                                      Diana Vivanco Dávila.

**      Título: Conexión USB-PC

**      Descripción:

**      Censa los valores de las entradas analógicas del Im-35

**      y convierte su valor en la temperatura, enviando los

```

** datos a la PC vía USB,

*****/

#include <18F4550.h>

// Driver Microchip Pic18fxx5x Hardware

#include <pic18_usb.h>

// Configuración del USB y los descriptores para este dispositivo

#include "header.h"

#include <usb.c> // Drivers Para La Comunicación USB

#include <stdlib.h>

//Usa resolución de 10 bits./ Se Define Después De Las
Librerías

#device adc=10

// Habilitamos el uso de las directivas HID

#define USB_HID_DEVICE TRUE

// Puertos De Comunicación De Datos De Entrada Al PIC

#define USB_EP1_TX_ENABLE USB_ENABLE_BULK

// Puertos De Comunicación De Datos De Salida Al Pic

#define USB_EP1_RX_ENABLE USB_ENABLE_BULK

// Numero De Paquetes De Recepción (Máximo 32)

#define USB_EP1_TX_SIZE 32

```
// Numero De Paquetes De Envió (Máximo 32)
```

```
#define USB_EP1_RX_SIZE 32
```

```
#fuses XTPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG,  
USBDIV, PLL1, CPUDIV1, VREGEN
```

```
#use delay(clock=48000000)
```

```
#BYTE TRISB=0X00
```

```
#BYTE PORTB=0X00
```

```
// Declaración De Variables Globales
```

```
const int8 Lenbuf = 32;
```

```
int8 recbuf[Lenbuf];
```

```
char version[5];
```

```
int32 dat;
```

```
float ft=32.5;
```

```
// Función Principal
```

```
void main(void) {
```

```
int8 envia[1];
```

```
FLOAT tem;
```

```
set_tris_b(0xff);
```

```
setup_adc (adc_clock_div_32);
```

```
delay_ms(500);
```

```
usb_init(); // Inicializa La Conexión Usb
```

```
usb_task(); // Enviando Autenticación

// Esperando Respuesta De Autenticacion

usb_wait_for_enumeration();

// Interrupciones Necesarias De Comunicación

enable_interrupts(global);

while (TRUE){

    // Obteniendo El Valor De Los Sensores De Temperatura

    set_adc_channel (12); //Elige canal a medir

    tem=read_adc (); //Hace conversión AD

    dat=(int32)(tem*100*0.4850); // Conversión AD


    if(usb_enumerated()){ // Verifica Comunicación Serial Con Pc
        if (usb_kbhit(1)){ // Verifica El Bits De Autenticación De
            Comunicación

            // Recibe Paquete De Datos

            usb_get_packet(1, recbuf, Lenbuf);

            // Convierte Valor De La Temperatura A Un Int32

            itoa(dat,10,version);

            // Envía Paquete De Datos Con Las Temperaturas

            usb_put_packet(1, version, 4, USB_DTS_TOGGLE);
```

```

if(recbuf[0]==127){
    // Enciende Led De Comunicación Exitosa
    output_high(PIN_A0);
}
delay_ms(500);
if(recbuf[0]==126){
    output_low(PIN_A0);
    // APAGA EL DE COMUNICACIÓN EXITOSA
    delay_ms(500);
    } } } } }

```

Anexo A7: Aplicación De Escritorio Desarrollada Con Lenguaje Java

```

// Declarando librerías y paquetes

import jPicUsb.*; // librería para la comunicación con el PIC

import java.io.UnsupportedEncodingException;

import java.util.logging.Level;

import java.util.logging.Logger;

public class nuevojPic extends javax.swing.JFrame {

    public byte prender =127;

    public byte apagar=127;

```

```
/** Crea nuevo form nuevojPic */

public nuevojPic() {

    initComponents();

    try{

        iface.load();

    }

    catch(Exception e){ System.out.println(e);}

    iface.set_instance(0);

    iface.set_vidpid("vid_04d8&pid_000b");

}

private void initComponents() {

    jButton1 = new javax.swing.JButton();

    jButton2 = new javax.swing.JButton();

    jTextField1 = new javax.swing.JTextField();

    jPanel1 = new javax.swing.JPanel();

    jLabel1 = new javax.swing.JLabel();

}
```

```
jTextField2 = new javax.swing.JTextField();
```

```
jPanel2 = new javax.swing.JPanel();
```

```
jLabel2 = new javax.swing.JLabel();
```

```
jTextField3 = new javax.swing.JTextField();
```

```
jPanel3 = new javax.swing.JPanel();
```

```
jLabel3 = new javax.swing.JLabel();
```

```
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
```

```
setCursor(new  
java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
```

```
setLocationByPlatform(true);
```

```
setResizable(false);
```

```
jButton1.setText("Prender Led");
```

```
jButton1.addActionListener(new java.awt.event.ActionListener()
```

```
{
```

```
public void actionPerformed(java.awt.event.ActionEvent evt)
```

```
{    jButton1ActionPerformed(evt);    }    });
```

```
jButton2.setText("Apagar Led");
```

```

jButton2.addActionListener(new java.awt.event.ActionListener()

{

public void actionPerformed(java.awt.event.ActionEvent evt) {

    jButton2ActionPerformed(evt);    }    });

jTextField1.setBackground(new java.awt.Color(255, 255, 255));

jTextField1.setEditable(false);

jTextField1.setFont(new java.awt.Font("Tahoma", 0, 50));

jTextField1.setText("00,00");

jTextField1.setBorder(javax.swing.BorderFactory.createEtched
Border());

jTextField1.setSelectedTextColor(new java.awt.Color(153,
153, 255));

jTextField1.addActionListener(new
java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt)
    {        jTextField1ActionPerformed(evt);        }    });

jPanel1.setBackground(new java.awt.Color(102, 102, 255));

```

```
jPanel1.setBorder(javax.swing.BorderFactory.createBevelBo  
rder(javax.swing.border.BevelBorder.RAISED));
```

```
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 14));
```

```
jLabel1.setText("TERMOCUNA 1");
```

```
javax.swing.GroupLayout jPanel1Layout =
```

```
new javax.swing.GroupLayout(jPanel1);
```

```
jPanel1.setLayout(jPanel1Layout);
```

```
jPanel1Layout.setHorizontalGroup(  
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING).addGroup(javax.swing.GroupLayout.Alignme  
nt.TRAILING, jPanel1Layout.createSequentialGroup()
```

```
.addGap(10, 10, 10)
```

```
.addComponent(jLabel1,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
```

```
.addContainerGap() );
```

```
jPanel1Layout.setVerticalGroup(  
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Ali  
gnment.LEADING)
```

```

        .addGroup(jPanel1Layout.createSequentialGroup()

        .addContainerGap()

        .addComponent(jLabel1)

        .addContainerGap(17, Short.MAX_VALUE))

    );

    jTextField2.setBackground(new java.awt.Color(255,
        255, 255));

    jTextField2.setEditable(false);

    jTextField2.setFont(new java.awt.Font("Tahoma", 0, 50));

    jTextField2.setText("00,00");

    jTextField2.setBorder(javax.swing.BorderFactory.createEt
chedBorder());

    jTextField2.setSelectedTextColor(new java.awt.Color(153,
        153, 255));

    jTextField2.addActionListener(new
    java.awt.event.ActionListener() {      public void
actionPerformed(java.awt.event.ActionEvent evt) {

        jTextField2ActionPerformed(evt);  }    });

```

```
jPanel2.setBackground(new java.awt.Color(102, 102, 255));

jPanel2.setBorder(javax.swing.BorderFactory.createBevelBorder(
    javax.swing.border.BevelBorder.RAISED));

jLabel2.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N

jLabel2.setText("TERMOCUNA 1");

javax.swing.GroupLayout jPanel2Layout =
    new javax.swing.GroupLayout(jPanel2);

jPanel2.setLayout(jPanel2Layout);

    jPanel2Layout.setHorizontalGroup(
        jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
                jPanel2Layout.createSequentialGroup()
                    .addGap(10, 10, 10)
                    .addComponent(jLabel2,
                        javax.swing.GroupLayout.DEFAULT_SIZE,
                        javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                    .addContainerGap(10)
                )
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jPanel2Layout.createSequentialGroup()
                    .add(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(jLabel2)
                        .add(jLabel1)
                    )
                    .addGap(10, 10, 10)
                )
            )
    );

jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .add(jPanel2Layout.createSequentialGroup()
            .add(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .add(jLabel2)
                .add(jLabel1)
            )
            .addGap(10, 10, 10)
        )
    );
```

```
gment.LEADING)
```

```
.addGroup(jPanel2Layout.createSequentialGroup()
```

```
    .addContainerGap()
```

```
    .addComponent(jLabel2)
```

```
    .addContainerGap(17, Short.MAX_VALUE))
```

```
);
```

```
jTextField3.setBackground(new java.awt.Color(255, 255, 255));
```

```
jTextField3.setEditable(false);
```

```
jTextField3.setFont(new java.awt.Font("Tahoma", 0, 50));
```

```
jTextField3.setText("00,00");
```

```
jTextField3.setBorder(javax.swing.BorderFactory.createEtchedBorder());
```

```
jTextField3.setSelectedTextColor(new java.awt.Color(153, 153, 255));
```

```
jPanel3.setBackground(new java.awt.Color(102, 102, 255));
```

```
jPanel3.setBorder(javax.swing.BorderFactory.createBevelBorder(javax.swing.border.BevelBorder.RAISED));
```

```
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 14));

jLabel3.setText("TERMOCUNA 1");

javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);

jPanel3.setLayout(jPanel3Layout);

jPanel3Layout.setHorizontalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
jPanel3Layout.createSequentialGroup()

.addGap(10, 10, 10)

.addComponent(jLabel3,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addContainerGap()) );

jPanel3Layout.setVerticalGroup(
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Ali
gnment.LEADING)

.addGroup(jPanel3Layout.createSequentialGroup()
```

```
.addContainerGap()
```

```
.addComponent(jLabel3)
```

```
.addContainerGap(17, Short.MAX_VALUE) );
```

```
javax.swing.GroupLayout layout = new
```

```
javax.swing.GroupLayout(getContentPane());
```

```
getContentPane().setLayout(layout);
```

```
layout.setHorizontalGroup(
```

```
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
```

```
LEADING).addGroup(layout.createSequentialGroup()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
```

```
.Alignment.LEADING)
```

```
.addGroup(layout.createSequentialGroup()
```

```
.addContainerGap()
```

```
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
```

```
.Alignment.LEADING, false)
```

```
.addComponent(jTextField1,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE, 123,
```

```
Short.MAX_VALUE).addComponent(jPanel1, 0,
```

```
javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
Short.MAX_VALUE)))  
  
.addGroup(layout.createSequentialGroup()  
.addGap(20, 20, 20).addComponent(jButton1)))  
  
.addGap(28, 28, 28)  
  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING)  
.addGroup(layout.createSequentialGroup()  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING)  
.addComponent(jTextField2,  
javax.swing.GroupLayout.DEFAULT_SIZE, 129,  
Short.MAX_VALUE).addComponent(jPanel2,  
javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
.addGap(18, 18, 18)  
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout  
.Alignment.LEADING, false)  
.addComponent(jTextField3)  
.addComponent(jPanel3,  
javax.swing.GroupLayout.DEFAULT_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE,
```

```
Short.MAX_VALUE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))

.addGroup(layout.createSequentialGroup()

.addComponent(jButton2).addContainerGap())) );

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup().addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addGap(34, 34, 34).addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGroup(layout.createSequentialGroup()
.addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addGap(34, 34, 34)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
.addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE, 103,
javax.swing.GroupLayout.PREFERRED_SIZE))))
.addGap(18, 18, 18)
.addGroup(layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING).addComponent(jButton1)
.addComponent(jButton2)).addContainerGap(38,
Short.MAX_VALUE)) );

pack(); }

```

```
private void
```

```
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
comando(iniciar); }
```

```
private void
```

```
jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
comando(terminar); }
```

```
private void
```

```
jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
}
```

```
private void
```

```
jTextField2ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
}
```

```
public void comando(byte envio)
```

```
{
```

```
int cantidad=0,n;
```

```
String temp;
```

```
float cant=0;
```

```
char val[];
```

```
int cnt,cnt2;

try {

byte[] out = {envio};

iface.QWrite(out, 1, 1000);

byte[] respuesta = iface.QRead(64,500);

String data_in = new String(respuesta, "UTF-8");

if (data_in.length() == 0) {

    System.out.print("No se recibieron datos: " );

    // Se muestran los valores en los labels

    System.out.print("data22:" + data_in + ">>Datos  
recibidos: "+ respuesta);

} else {

    cantidad= Integer.parseInt(data_in);

    cnt= (int)(data_in.charAt(0)-48)*10 +

        (int)(data_in.charAt(1)-48);

    cnt2= (int)(data_in.charAt(2)-48)*10 +

        (int)(data_in.charAt(3)-48);
```

```
        jTextField1.setText(cnt+","+cnt2);
```

```
    }
```

```
    System.out.println(data_in);
```

```
    } catch (UnsupportedEncodingException ex) {
```

```
        Logger.getLogger(nuevojPic.class.getName()).log(Level.
```

```
        SEVERE, null, ex);
```

```
    }}
```

```
// FUNCIÓN PRINCIPAL
```

```
public static void main(String args[]) {
```

```
    java.awt.EventQueue.invokeLater(new Runnable() {
```

```
        public void run() {
```

```
            new nuevojPic().setVisible(true); } }); }
```

```
private javax.swing.JButton jButton1;
```

```
private javax.swing.JButton jButton2;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;  
  
private javax.swing.JPanel jPanel1;  
  
private javax.swing.JPanel jPanel2;  
  
private javax.swing.JPanel jPanel3;  
  
private javax.swing.JTextField jTextField1;  
  
private javax.swing.JTextField jTextField2;  
  
private javax.swing.JTextField jTextField3;  
  
}
```