UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

SISTEMA DE ADQUISICIÓN DE DATOS (SAD) DE PROPÓSITO GENERAL

T E S I S

QUE PARA OBTENER EL TÍTULO DE

INGENIERO ELÉCTRICO - ELECTRÓNICO

P R E S E N T A

PATRICIO RAMÍREZ JOSÉ TOMÁS

DIRECTOR DE TESIS:

ING. RAFAEL FLORES GARCÍA.

MÉXICO. D. F. **NOVIEMBRE DE 2006**

A través de los años he aprendido que no tenemos que cambiar a los amigos, si entendemos que los amigos cambian, que la amistad verdadera continua creciendo, aún más allá de la distancia. He aprendido que cuesta mucho tiempo y trabajo llegar a ser la persona que uno quiere ser, pero que también uno puede seguir adelante mucho después de que ha caído. He aprendido que somos responsables por lo que hacemos, no importa cómo nos sintamos.

A veces, cuando estoy enojado y triste, he aprendido que no importa qué tanto está roto tu corazón, el mundo no se detiene por tu pena, y no siempre es suficiente ser perdonado por los demás. A veces tienes que aprender a perdonarte a ti mismo, pues es verdad que nuestro pasado y nuestras circunstancias pueden haber influenciado en quiénes somos, pero somos responsables por quién llegamos a ser. Cada vez entiendo más que la madurez tiene más que ver con las experiencias que has tenido y lo que has aprendido de ellas, que con el número de cumpleaños que has celebrado.

Gracias papá, gracias mamá, gracias hermanos, y gracias a todos a aquellos amigos y profesores de la Facultad de Ingeniería, que me han ayudado a levantarme por más caídas que he tenido. Una vez en los primeros años de la universidad, un amigo me mostró un fragmento de un pensamiento escrito por él, ese pensamiento me gusto mucho porque describe de una forma muy positiva el verdadero sentir, al menos en mi caso, de cuando se nos dificulta alcanzar una meta, este es un fragmento de aquella reflexión:

Muchas veces me he sentido solo....

es casi media noche y renace la esperanza
como el día gris espera su luz
y la tristeza espera su alegría
entonces yo sé que puedo continuar.....

como un arco iris se impone entre las nubes
y un ave Fénix resurge de sus cenizas
aquel pensamiento
no imaginado por mí
espera para decirme que he triunfado

y que así debo continuar
pues siempre tuve su compañía
y entonces
poder llorar con su sonrisa.....

Sergio Escutia

Con agradecimiento para:
El ***Ingeniero Rafael Flores García***. Por su enorme ayuda en la revisión y conclusión de esta tesis.

A los integrantes del sínodo por sus sugerencias:
Ing. Eduardo Carranza Torres, ***Ing. Rodolfo Peters Lammel***, ***Ing. Francisco Rodríguez Ramírez*** y al ***Ing. Sabino Ortega Monjarás*** por sus atenciones y conocimientos gracias.

Dedicado a la memoria de mi abuelo
Don Domingo Ramírez Bernal (1929-2000).

Gracias por tus consejos viejo, te llevare siempre en mi corazón.
El recorrido ha concluido. La historia apenas empieza….

ÍNDICE

Pág.

1. Prólogo……..1

1.1 Objetivo……..……2

1.2 Alcance……...……3

1.3 Introducción……3

2. Conceptos básicos e Instrumentación……...5

2.1 Sistemas de medida. ……..…5

2.1.1 Transductores, sensores y acondicionamiento……..…..7

2.1.2 Dominio de datos y conversiones……8

2.2 Tipos de sensores……..10

2.3 Exactitud, precisión, sensibilidad……11

2.4. Linealidad y resolución. ……13

2.4.1 Errores sistemáticos. ……14

2.4.2 Errores aleatorios……...…15

3. Medición de la Temperatura……...16

3.1. La Temperatura……...16

3.2 Escalas de temperatura. ……...16

3.3 Sensores de temperatura. ……17

3.4 Sensores de Temperatura con Termopares……...18

3.4.1 Acondicionadores de señal para Termopares. ……..21

3.5. Sensores de Temperatura con elementos Resistivos……..…21

3.5.1 Los termistores……..…25

3.6 Transductores de temperatura con semiconductores……28

3.7 Consideraciones para la selección……32

4. Microcontroladores…….37

4.1. Controlador y microcontrolador. ……38

4.2 Recursos comunes a todos los microcontroladores. ……39

4.3. Arquitectura del Microcontrolador AT 89C51……42

4.4. Memoria fuera del μc 89c51……49

5. Diseño del circuito de medición de temperatura…….57

5.1 Amplificador de instrumentación. ……...57

5.1.2 Rechazo de Modo Común…….58

5.2 Sensor de Temperatura con Salida en Corriente (AD590)….61

5.3 Configuración básica del circuito sensor de temperatura AD590. ……62

5.3.1 Reducción de interferencias en la medida de temperatura con el circuito integrado AD590. …….62

6. Diseño del Dispositivo de adquisición y registro de información……...64

6.1 Convertidor Analógico/digital…….65

6.2 Resolución y velocidad del Convertidor ADC (TC 109A) ……68

6.3 Parámetros específicos del circuito CI TC 109A..……..…..71

6.4 Almacenamiento de información en memoria SRAM externa……..…..76

6.5 El microcontrolador AT 89C51 como microprocesador..……81

6.6 Sección de comunicaciones…….….84

6.7 Despliegue de la información. …….…87

7. Resultados……..90

7.1 El método de los mínimos cuadrados……...90

7.2 Experimento de calibración. ……90

7.2.2 Pruebas de caracterización para el transductor AD590, usado en el Sistema de adquisición de datos. ……..…91

7.3 Análisis de regresión lineal. ……99

8. Conclusiones. ……104

Bibliografía……...…106

ANEXOS

DIAGRAMAS DE FLUJO (ANEXO A)……..107
HOJAS DE ESPECIFICACIONES (ANEXO B)……110

1. Prólogo.

En la actualidad el vertiginoso desarrollo de la electrónica ha motivado que todas las esferas de la vida humana se estén automatizando, por ejemplo: la industria, el hogar, los comercios, la agricultura, la ganadería, el transporte, las comunicaciones, etc. En todo ese proceso de automatización el microprocesador y el microcontrolador juegan un papel de suma importancia. Ellos han permitido el desarrollo de sistemas inteligentes que resuelven los más diversos problemas, estos son los llamados Sistemas de Adquisición de Datos.

El objetivo básico de los "Sistemas de Adquisición de Datos"(S.A.D.) es la integración de los diferentes recursos que lo componen: Transductores de diferentes tipos y naturaleza, multiplexores, amplificadores, conversores A/D y D/A, junto con el uso del microcontrolador, utilizando de este todas sus prestaciones: interrupciones, temporizadores, comunicación serie así como memorias y puertos externos. Todo ello sirve para crear un sistema que se encargue de una aplicación específica, como es monitorear una o varias variables (PH, humedad relativa, temperatura, iluminación, concentración, etc.) para una posterior utilización de la misma ya sea con fines docentes, científicos, de almacenamiento o control.

Una importante aplicación de los métodos de instrumentación, utilizando sensores conjuntamente con sistemas de adquisición de datos, son las actividades de inspección, en las centrales nucleares de componentes como los generadores de vapor, las tuberías de los distintos sistemas de las plantas, o las turbinas. Estos componentes, requieren el uso de sistemas automatizados de adquisición de datos, de las señales que se obtienen durante las inspecciones. En este ejemplo la adquisición automática y remota de datos se realiza por las condiciones de contaminación, radiación, accesos a los componentes y también por el gran volumen de datos que se adquieren. Además, es necesario adquirir información de las coordenadas de la posición de los desperfectos observados para su correcta localización y todas las señales necesarias para el análisis de los mismos.

Otra importante aplicación se encuentra en la investigación científica, ya que con frecuencia centros de investigación oceanográfica y meteorológica realizan experimentos de campo para observar una región en forma intensiva o por periodos cortos de tiempo para alimentar modelos probabilísticos de eventos climáticos que requieren equipos de monitoreo de diversos tipos de sensores y sistemas de adquisición de datos que proporcionen información especifica de varios fenómenos en la naturaleza.

Para medir variables que intervienen ya sea en procesos industriales y de investigación, el sensor debe estar en contacto con el medio ambiente, este protegido o no. Este es uno de los factores, junto con el uso continuo, que más afectan en la contaminación y degradación del sensor en varios niveles, para el sensado constante de

una variable monitoreada, por lo que el desarrollo de sistemas de adquisición de datos y sensores es un campo que ha estado perfeccionándose por mucho tiempo. Hoy en día el mercado de la instrumentación electrónica han alcanzado un nivel de competitividad muy alto, y por ello en la mayoría de los casos el precio de estos sistemas va de la mano con su calidad.

Existen decenas de marcas de fabricantes que construyen sistemas de sensores con un nivel de calidad aceptable, sin embargo la mayoría de estas compañías son extranjeras y a menudo presentan un precio demasiado alto en sus equipos para algunas instituciones de investigación o industrias que desean adquirirlos, de manera tal que, en muchas ocasiones, esta situación no permite la implantación de redes de sensado, y por ende un óptimo control en procesos industriales o de investigación a baja y mediana escala de producción.

En el mercado existen los sistemas HoBo Stowaway, modelos U12-011, que miden directamente temperatura (T) y humedad relativa (RH), a doce bits, con precisiones de ±0.35 ºC y ± 2.5%. Con capacidad para 21500 lecturas de temperatura y 21500 de humedad relativa. Los HoBo tienen la ventaja de una interfaz directa a puerto USB para recuperar los datos obtenidos en una computadora y continuar midiendo. Sin embargo su precio rebasa los mil dólares, y es por demás decir que su implementación en redes de sensado es muy costosa, La empresa National Instruments desarrolla sistemas de adquisición de datos de señales diversas, así sistemas de evaluación de los datos adquiridos, que se caracterizan por su alta velocidad de adquisición y tratamiento, pero su precio también resulta muy alto aunque no llegan a rebasar los mil dólares

Ante una situación como esta seria deseable obtener un sistema de adquisición de datos de calidad y funcionalidad, comparable con la tecnología de las mejores marcas que fabrican estos dispositivos, y a un precio mucho menor de los mismos, que tenga además repuestos fáciles de conseguir en nuestro país y que sea funcional en cuanto a su utilidad para distintos tipos de sensores.

1.1 Objetivo

El objetivo primario en este trabajo será incorporar un estudio detallado acerca del desarrollo de un sistema de adquisición de datos de propósito general a bajo costo, que se encargue de monitorear cualquier tipo de variable, así como exponer la importancia de estos sistemas en el desarrollo de la electrónica de control en procesos industriales o de investigación. También se hará hincapié en las partes que componen los Sistemas de Adquisición de datos.

El prototipo de Sistema de Adquisición de datos presentado en este trabajo, proporciona altas prestaciones, para una correcta y eficaz evaluación de los datos adquiridos. Este sistema se caracteriza por su alta velocidad de adquisición y tratamiento de datos, es un sistema portátil y tiene la posibilidad de emplear diversos sensores, la evaluación mediante gráficas de los datos adquiridos con respecto del tiempo, así como el almacenamiento de la información histórica y actual, es una característica vital en la toma de decisiones y análisis, lo que hace de este prototipo un sistema integral y funcional.

Uno de los aspectos destacables del Sistema de Adquisición de Datos, desarrollado en este trabajo, es el diseño e integración que incluye la programación del despliegue de datos en una computadora personal, mediante un sistema amigable que nos servirá para ajustar y decidir el tiempo de muestreo, aspecto reseñable ya que, incluso, en equipos comerciales, es habitual tener que modificar y completar la programación adquirida para adaptarla a las necesidades del usuario.

1.2 Alcance

Esta tesis trata sobre el desarrollo de un prototipo de sistema de adquisición de datos (SAD), de propósito general, empleando en especifico el sensor comercial de temperatura AD 590, el desarrollo permite entender el funcionamiento primario de los dispositivos con sensores desde su etapa transductora, el acondicionamiento de su señal de salida, y los principios de transducción en la conversión de una magnitud a otra.

Otra aplicación será, presentar un análisis acerca del estudio de los datos obtenidos, en la medición de la temperatura así como el desarrollo, diseño y construcción del sistema de adquisición de datos. Esto no quiere decir que este sistema solo sirva para la medición de la temperatura, ya que su desarrollo es de propósito general, puede ser usado con cualquier tipo de transductor, como por ejemplo de presión, salinidad ó fluorescencia, al que previamente, se le haya acondicionado la señal a su salida, para que esta sea en voltaje proporcional a la magnitud medida, y compatible con la tarjeta de adquisición de datos. Cabe aclarar que se ha elegido al sensor de temperatura en este trabajo, para ejemplificar el uso del prototipo del sistema propuesto.

1.3 Introducción

Una de las características que deberá mostrar el Sistema de Adquisición de Datos en este trabajo, será proveer gráficas del comportamiento de la variable monitoreada con respecto del tiempo. En el circuito del SAD se utiliza el microcontrolador AT89C51. Este microcontrolador, de 8 bits, almacena los datos medidos en dos memorias RAM externas, para luego poder descargar esa información en una computadora personal y graficarla con respecto del tiempo. Este sistema también permitirá graficar los datos adquiridos al mismo tiempo que son muestreados. La resolución del sistema será de $\pm 0.002^{0}C$, dada por el fabricante del convertidor ADC TC 109A, con un rango de medición de temperatura de -50 ^{0}C a 100 ^{0}C. En la estructura interna del SAD se usa un convertidor analógico/digital de 12 bits mas el signo, y un bit de sobre rango que indicará que los datos muestreados han sobrepasado la capacidad del sistema, en el rango de medición.

Finalmente con el mismo software de programación en lenguaje C++ builder, que se utiliza para graficar los datos adquiridos, se podrá desplegar un menú de opciones, donde el usuario, además de tener la opción de graficar podrá elegir el tiempo de muestreo de la magnitud medida, esto significa que el rango de muestreo puede variar de 10 en 10 segundos hasta llegar a un minuto, o si se prefiere, incrementar el

rango de muestreo hasta llegar a un máximo de una muestra cada minuto. El sistema en modo autónomo podrá adquirir información durante un máximo de 9 horas continuas.

El desarrollo del Sistema de Adquisición de Datos propuesto, implica ahondar en la instrumentación electrónica necesaria, de acuerdo a las características antes mencionadas. De esta manera se documenta el funcionamiento de los Circuitos Integrados utilizados en la integración del sistema de adquisición de datos, como el transductor de temperatura AD590, el microcontrolador AT89C51, las dos memorias RAM HL62256 externas, el convertidor analógico/digital, así como el circuito integrado, implicado en la comunicación del sistema con la computadora personal, y los utilizados en el acondicionamiento de la señal del transductor.

2. Conceptos básicos e Instrumentación

Las aplicaciones de la electrónica, presentes actualmente en innumerables aspectos de nuestra vida cotidiana no serían posibles sin los sensores. Sin la capacidad que estos ofrecen de medir las magnitudes físicas para su conocimiento o control, muchos de los dispositivos electrónicos no serían más que simples curiosidades de laboratorio. La utilización de sensores es indispensable en la automatización de industrias de proceso y manufacturados, incluida la robótica en ingeniería experimental, en sectores productivos como son el ahorro energético y el control ambiental, en automóviles y electrodomésticos, en agricultura y medicina,…

En la actualidad la instrumentación de los sensores se ha desarrollado de manera sofisticada en diseño y nivel de construcción. Esto no hace sino aumentar su importancia al extender sus campos de aplicación [4]. Sin embargo, una gran mayoría de los dispositivos de medición utiliza algún principio eléctrico básico para su operación o cuentan con un dispositivo electrónico para las etapas intermedia, de modificación y de lectura final.

En la presentación de resultados finales del presente trabajo se utilizarán algunos conceptos básicos, por lo que en este capítulo se tratarán primero algunos aspectos generales de los transductores, como su terminología, característicos de linealidad y resolución.

En el siguiente capítulo 3 se mencionan algunos de los más importantes dispositivos eléctricos, que se emplean comúnmente, en los procesos de medición de la temperatura, también se tratará el acondicionamiento de señal en el transductor de temperatura (AD590) utilizado para este proyecto, todo ello mediante el previo estudio de los conceptos básicos presentados en este capítulo.

2.1 Sistemas de medida.

Se denomina sistema a la combinación de dos o más elementos, subconjuntos y partes necesarias para realizar una o varias funciones. En los sistemas de medida, esta función es la asignación objetiva y empírica de un número a una propiedad o cualidad de un objeto o evento, de tal forma que la describa. Es decir, el resultado de la medida debe ser: independiente del observador (objetiva), basada en la experimentación (empírica), y de tal forma que exista una correspondencia entre las relaciones numéricas y entre las propiedades descritas.

Los objetivos de la medida pueden ser: la vigilancia o seguimiento de procesos, como es el caso de la temperatura ambiente, de los contadores de gas y de agua, de monitorización clínica. El control de un proceso, como en el caso de un termostato o el control de nivel de un dispositivo, pero también puede ser una necesidad de la

ingeniería experimental, como sucede con el estudio de la distribución de temperaturas en el interior de una pieza irregular, o de las fuerzas sobre el conductor simulado de un vehículo cuando este choca contra un objeto. Las medidas en prototipos son además necesarias para verificar los resultados de los modelos desarrollados en una computadora.

En un sentido amplio la realización de una medida para adquisición de datos, implica también el procesamiento de dicha información y la presentación de resultados, de forma que puedan ser percibidos por nuestros sentidos. Cualquiera de estas funciones puede ser local o remota, implicando ello, en este segundo caso, la necesidad de transmitir la información. En el diagrama de la figura 2.1 se muestra la trayectoria por etapas que sigue una señal X que se desea medir

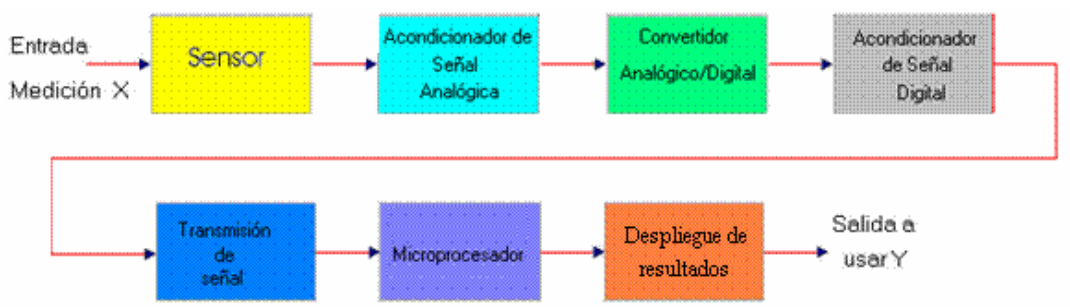


Figura 2.1 Estructura básica de un sistema de adquisición de datos.

El elemento esencial de un sistema de adquisición de datos es el transductor, el cual, en la mayoría de los casos, presenta una señal eléctrica que es una indicación de la variable física que se está midiendo, la señal puede ser de corriente, voltaje, frecuencia o pulsos eléctricos. En este trabajo, el análisis presentado se supone para un transductor de temperatura AD590, que convierten las señales físicas de interés en señales eléctricas. En la figura 2.1 se muestra de manera compacta, la topología básica de un sistema de adquisición de datos, del cual se analiza a lo largo de este trabajo cada uno de los bloques que lo componen.

El sistema general de adquisición de datos consta de tres etapas:

1. La etapa de entrada, la cual consiste en los transductores apropiados y los circuitos acondicionadores de señal (amplificadores, filtros, etc.).
2. Una etapa de conversión de señal, la cual convierte la señal de entrada en un voltaje y a continuación expresa este voltaje en forma digital. La conversión a forma digital se efectúa cambiando la señal de voltaje a señal de frecuencia, la cual se utiliza a su vez para activar circuitos contadores electrónicos. La conversión es esencialmente un proceso de modulación en frecuencia.
3. Una etapa de salida, la cual maneja la señal digital y la expresa en forma impresa o gráfica, mediante una computadora personal u otro dispositivo de salida, debe incluir circuitos de acoplamiento adecuados, en la interfaze de comunicación entre el procesador y la computadora personal.

2.1.1 Transductores, sensores y acondicionamiento

Un transductor es, simplemente, un dispositivo que proporciona una salida utilizable en respuesta a una magnitud física, propiedad o condición específica que se desea medir. Se denomina transductor, en general, a todo dispositivo que convierte una señal de una forma física en una señal correspondiente pero de otra forma física distinta. Es, por tanto, un dispositivo que convierte un tipo de energía en otro. Esto significa que la señal de entrada, de interés, es siempre una energía o potencia. Y muchas veces esta salida representa la suma de muchas energías, pero al medir una de las componentes de la señal suele ser tan pequeña que puede despreciarse, y se interpreta que se mide sólo la otra componente.

Al medir una fuerza, por ejemplo, se supone que el desplazamiento del transductor es despreciable, es decir, no se carga al sistema. Pero en la transducción siempre se extrae una cierta energía del sistema donde se mide, por lo que es importante garantizar que esto no lo perturba.

Dado que hay seis tipos de señales: mecánicas, térmicas, magnéticas, eléctricas, ópticas y moleculares (químicas), cualquier dispositivo que convierta una señal de un tipo en una señal de otro tipo, se considera un transductor, y la señal de salida podría ser de cualquier forma física. En la práctica, no obstante, se consideran transductores por antonomasia aquellos que ofrecen una señal de salida eléctrica. Ello se debe a que los sistemas de medida electrónicos ofrecen, entre otras las siguientes ventajas:

1. Debido a la estructura electrónica de la materia, cualquier variación de un parámetro no eléctrico de un material viene acompañada por la variación de un parámetro eléctrico. Eligiendo el material adecuado, esto permite realizar transductores con salida eléctrica para cualquier magnitud física no eléctrica.

2. Dado que en el proceso de medida no conviene extraer energía del sistema donde se mide, lo mejor es amplificar la señal de salida del transductor. Con amplificadores electrónicos se pueden obtener fácilmente ganancias de potencia de 10^{10} en una sola etapa, a baja frecuencia.

3. Además de la amplificación, hay una gran variedad de recursos, en forma de circuitos integrados, para acondicionar o modificar las señales eléctricas. Incluso hay transductores que incorporan físicamente en un mismo encapsulado parte de estos recursos.

4. Existen también numerosos recursos para presentar o registrar información si se hace electrónicamente, pudiéndose manejar no solo datos numéricos, sino también textos, gráficos y diagramas.

5. La transmisión de señales eléctricas es más versátil que la de señales mecánicas, hidráulicas o neumáticas, aunque si bien no hay que olvidar que estas pueden ser más convenientes en determinadas circunstancias, como pueden ser la presencia de radiaciones ionizantes o atmósferas explosivas, en muchos casos estos sistemas han sido sustituidos por otros eléctricos. De hecho, mientras en industrias de proceso (química, petróleo, gas, alimentación textil,...), donde se introdujeron en seguida los sistemas automáticos, se encuentran actualmente sistemas neumáticos junto

a sistemas eléctricos más recientes, en cambio en las industrias de manufacturados, donde hay una serie de procesos discontinuos y que son de automatización más reciente, apenas hay sistemas neumáticos.

Un sensor es un dispositivo que, a partir de la energía del medio donde se mide, proporciona una señal de salida transducible que es función de la variable medida.

Sensor y transductor se emplean a veces como sinónimos, pero sensor sugiere un significado más extenso: la ampliación de los sentidos para adquirir un conocimiento de cantidades físicas que, por su naturaleza o tamaño, no pueden ser percibidas directamente por los sentidos. Transductor, en cambio, sugiere que la señal de entrada y la de salida son convertidas de una variable física a otra.

Los acondicionadores de señal, adaptadores o amplificadores, en sentido amplio, son los elementos del sistema de medida que ofrecen, a partir de la señal de salida de un sensor electrónico, una señal apta para ser presentada, registrada o simplemente permita un proceso posterior mediante un equipo o instrumento estándar. Consisten normalmente en circuitos electrónicos que ofrecen, entre otras funciones, las siguientes: amplificación, filtrado, adaptación de impedancias, etc.

Se considera, por ejemplo, el caso en que una de las etapas de tratamiento de la señal de medida es digital, si la salida del sensor es analógica, que es lo más frecuente, hará falta un convertidor A/D. Estos tienen una impedancia de entrada limitada, exigen que la señal aplicada sea continua y que su amplitud este entre unos límites determinados, que no suelen exceder de 10V. Todas estas exigencias obligan a interponer un acondicionador de señal entre el sensor y el convertidor A/D, aunque muchas veces el acondicionador ofrece señales de apenas unos milivoltios.

La presentación de los resultados puede ser de forma analógica o digital (óptica, acústica o táctil). El registro puede ser magnético o sobre papel, e incluso electrónico (memorias eléctricas), y exige siempre que la información de entrada esté en forma eléctrica.

2.1.2 Dominio de datos y conversiones

En los sistemas de medida, las funciones de transducción, acondicionamiento, procesamiento y presentación, no siempre se pueden asociar a elementos físicos distintos. Además, la separación entre el acondicionamiento y el procesamiento puede ser a veces difícil de definir. Pero, en general, siempre es necesaria una acción sobre la señal del sensor antes de su utilización final. Con el término interfase se designa, en ocasiones, el conjunto de elementos que modifican las señales, cambiando incluso de dominio de datos, pero sin cambiar su naturaleza, es decir, permaneciendo siempre en el dominio eléctrico.

Se denomina dominio de datos al nombre de una magnitud mediante la que se representa o transmite información. El concepto de dominios de datos y el de conversiones entre dominios, es de gran interés para describir los transductores y los circuitos electrónicos asociados. En la figura 2.2 se representa un diagrama con algunos de los posibles dominios, detallando en particular ciertos dominios eléctricos [4].

En el dominio analógico, la información está en la amplitud de la señal, bien se trate de carga, corriente, tensión o potencia. En el dominio temporal, la información no está en las amplitudes de las señales, sino en las relaciones temporales: periodo o frecuencia, anchura de pulsos o fase. En el dominio digital las señales tienen sólo dos niveles. La información puede estar en el número de pulsos, y venir representada por palabras serie o paralelo.

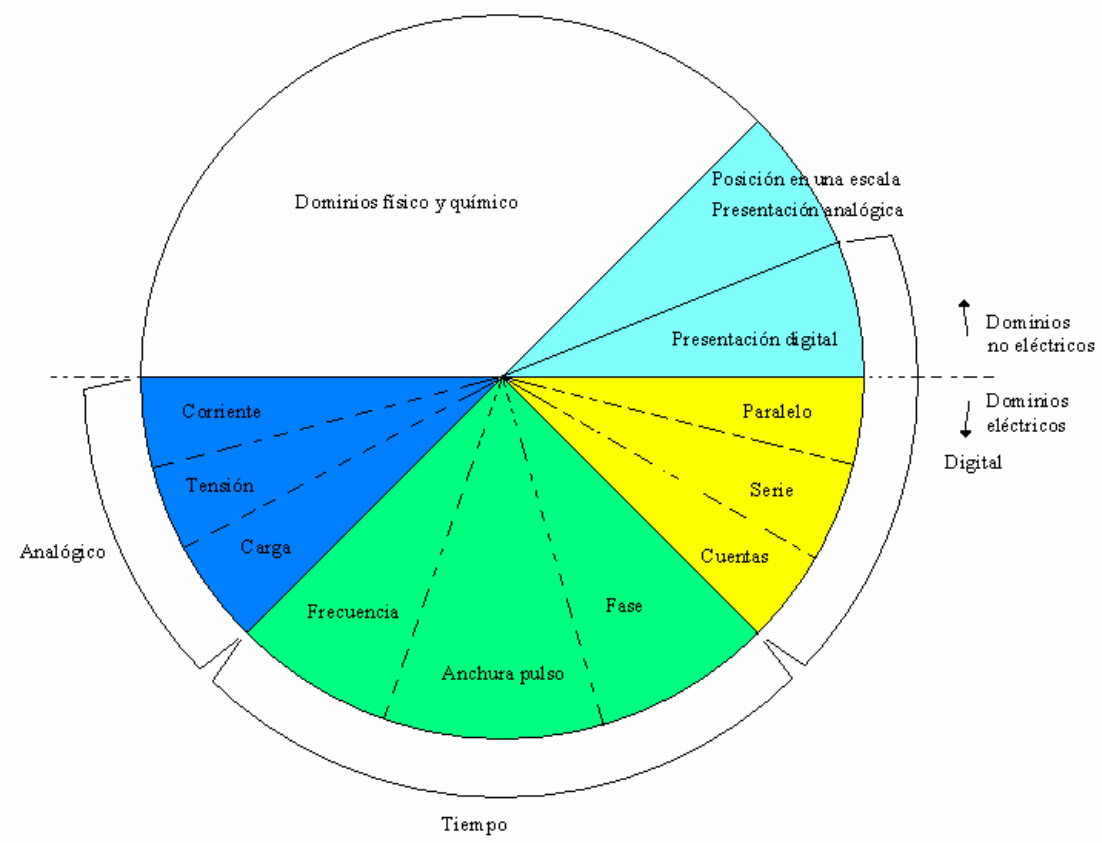


Figura 2.2 Dominios de datos

La estructura de un sistema de medida refleja las conversiones entre dominios que se realizan, e influye particularmente en ella el que se trate de una medida directa o indirecta.

Una medida física es directa cuando se deduce información cuantitativa acerca de un objeto físico o acción mediante comparación directa con una referencia. A veces se puede hacer simplemente de forma mecánica, como en el caso de una balanza clásica.

En las medidas indirectas la cantidad de interés se calcula a partir de otras medidas y de la aplicación de la ecuación que describe la ley que relaciona dichas magnitudes. Los métodos empleados suelen ser siempre eléctricos. Es el caso, por ejemplo, de la medida de la potencia transmitida por un eje de un motor a partir de la medida del par y de la medida de su velocidad de rotación.

2.2 Tipos de sensores

El número de sensores disponibles para las distintas magnitudes físicas es tan elevado que no se puede proceder racionalmente a su estudio sin clasificarlos previamente de acuerdo con algún criterio.

Según el requerimiento de energía, los sensores se pueden dividir en autogeneradores y no autogeneradores. Los sensores autogeneradores no requieren de una fuente externa. Los sensores no autogeneradores requieren de fuente externa de energía.

Según la señal de salida, los sensores se clasifican en analógicos o digitales. En los analógicos la salida varía a nivel macroscópico de forma continua. La información esta en la amplitud. En los sensores digitales la salida varía en forma de saltos o pasos discretos y su información esta en la representación de los cambios discretos. Los sensores digitales tienen también mayor fidelidad y mayor fiabilidad.

Atendiendo al modo de funcionamiento, los sensores pueden ser de deflexión o de comparación. En los sensores que funcionan por deflexión, la magnitud medida produce algún efecto físico, que engendra algún efecto similar, pero opuesto, en alguna parte del instrumento, y que esta relacionado con alguna variable útil. Un dinamómetro para la medida de fuerzas es un sensor de este tipo en el que la fuerza aplicada deforma un muelle hasta que la fuerza de recuperación de este, proporcional a su longitud, iguala la fuerza aplicada.

En los sensores que funcionan por comparación, se intenta mantener nula la deflexión mediante la aplicación de un efecto bien conocido, opuesto al generado por la magnitud a medir. Hay un detector del desequilibrio y un medio para reestablecerlo. En una balanza manual, por ejemplo, la colocación de una masa de un platillo provoca un desequilibrio, indicado por una aguja sobre una escala. El operario coloca entonces una o varias masas en el otro platillo hasta alcanzar el equilibrio, que se juzga por la posición de la aguja.

Cualquiera de estas clasificaciones es exhaustiva, y cada una tiene interés particular para diferentes situaciones de medida. Ahora bien, para el estudio de un gran número de sensores se suele acudir a su clasificación de acuerdo con la magnitud medida. Se habla, en consecuencia, de sensores de temperatura, presión, caudal, humedad, posición, velocidad, aceleración, fuerza o par. Sin embargo, esta clasificación difícilmente puede ser exhaustiva ya que la cantidad de magnitudes que se pueden medir es prácticamente inagotable.

Desde el punto de vista de la ingeniería electrónica, es más atractiva la clasificación de los sensores de acuerdo con el parámetro variable de transducción: resistencia, capacitancia, inductancia, añadiendo luego los sensores generadores de tensión, carga o corriente. Este tipo de clasificación poco frecuente, permite reducir el número de grupos a unos pocos y se presta bien al estudio de los acondicionadores de señal asociados. En el cuadro 2.1 se recogen los sensores y métodos de detección ordinarios para las magnitudes más frecuentes.

Cuadro 2.1 Clasificación de los sensores de acuerdo a los parámetros de transducción (resistencia, capacitancia, inductancia,)

Sensores	Magnitudes								
	Posición Distancia Desplazamiento	Velocidad	Aceleración Vibración	Temperatura	Presión	Caudal Flujo	Nivel	Fuerza	Humedad
Resistivos	Potenciómetros Galgas Magnetorresistencias		Galgas + masa-resorte	RTD Termistores	Potenciómetros +tubo Bourdon	Anemómetros de hilo caliente Galgas + voladizo Termistores	Potenciómetro + flotador Termistores LDR	Galgas	Humistor
Capacitivos	Condensador Diferencial				Condensador variable + diafragma		Condensador variable	Galgas Capacitivas	Dieléctrico variable
Inductivos y electromagnéticos	LVDT Corrientes Foucault Resolver Inductosyn Efecto Hall	Ley Faraday LVT Efecto Hall Corrientes Foucault	LVDT +masa-resorte		LVDT + diafragma Reluctancia variable + diafragma	LVDT + rota metro Ley Faraday	LVDT + flotador Corrientes Foucault	Magneto-elástico LVDT + célula carga	
Generadores			Piezo eléctricos +masa-resorte	Termopares Piroeléctricos	Piezoeléctricos			Potencio-eléctricos	
Digitales	Codificadores Incrementales y absolutos	Codificadores incrementales		Osciladores de cuarzo	Codificador + tubo Bourdon	Vórtices			SAW
Uniones p-n	Fotoeléctricos			Diodo Transistor Convertidores T/I			Fotoeléctricos		
Ultrasonidos	Reflexión	Efecto Doppler				Efecto Doppler Tiempo tránsito Vórtices	Reflexión Absorción		

2.3 Exactitud, precisión, sensibilidad

La exactitud es la cualidad que caracteriza la capacidad de un instrumento de medida de dar indicaciones que se aproximen al verdadero valor de la magnitud medida. El valor exacto, verdadero o ideal, es el que se obtendría si la magnitud se midiera con un método ejemplar. Se considera como tal aquel método de medida en el que los expertos coinciden que es suficientemente exacto para la finalidad pretendida con los resultados que se obtengan. La exactitud de un sensor se determina mediante la denominada calibración del mismo. Consiste en mantener todas las entradas excepto una a un valor constante, y se va anotando los valores que toma la salida. La representación de estos valores en función de los de la entrada define la curva de calibración. Para poder conocer el valor de la magnitud de entrada, esta debe tener un valor bien conocido, constituyendo lo que se denomina un patrón de referencia. Su valor debe conocerse con una exactitud al menos diez veces mayor que la del sensor que se calibra.

La discrepancia entre la indicación del instrumento y el valor verdadero de la magnitud medida se denomina error. La diferencia entre la indicación del instrumento y el valor verdadero se denomina error absoluto.

$$\text{Error absoluto} = \left| resultado - valor \quad verdadero \right| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2.1)$$

Lo más común es especificar el error como cociente entre el error absoluto y el valor verdadero de la magnitud medida, cociente que se denomina error relativo, Este

suele tener dos términos: uno dado como porcentaje (tanto por ciento) de la lectura, y otro constante.

$$error \quad relativo = \frac{/\, error \quad absoluto\, /}{valor \quad verdadero} \qquad (2.2)$$

Generalmente para los sensores se especifica un error relativo como porcentaje de la lectura exclusivamente. Todos los sensores de una misma clase tienen un error en la medida, dentro de su alcance nominal y en unas condiciones establecidas, que no supera un valor concreto, denominado índice de clase.

El valor medido y su incertidumbre deben darse con valores numéricos compatibles, de forma que el resultado numérico de la medida no debe tener más cifras, de las que se puedan considerar validas a la luz de la incertidumbre sobre dicho resultado. Por ejemplo, al medir la temperatura ambiente, un resultado de la forma 20 ^{0}C ± 1 ^{0}C esta expresado correctamente, mientras que las expresiones 20 ^{0}C ± 0.1 ^{0}C, 20.5 ^{0}C ± 1 ^{0}C, 20.5 ^{0}C ± 10% son todas incorrectas.

La precisión es la cualidad que caracteriza la capacidad de un instrumento de medida de dar el mismo valor de la magnitud medida, al medir varias veces en unas mismas condiciones determinadas (ambientales, operador, etc.), prescindiendo de su concordancia o discrepancia con el valor real de dicha magnitud. La precisión implica que se tenga simultáneamente una conformidad en las sucesivas lecturas y un número alto de cifras significativas y es, por tanto, una condición necesaria pero no suficiente para la exactitud. La figura 2.3 presenta distintas situaciones posibles.

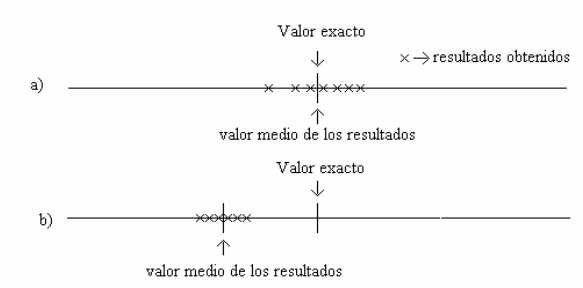


Figura 2.3. Distintas situaciones de medida que indican la diferencia entre exactitud y fidelidad. En el caso a) hay una gran exactitud y una baja fidelidad. En el caso b) la fidelidad es mayor pero hay una gran inexactitud.

En los sensores, cuando hay una variación de la salida a lo largo del tiempo se habla a veces de inestabilidad, y se dice que el sensor tiene derivas. En particular, se especifican a veces las denominadas derivas de cero y derivas del factor de escala. La deriva de cero expresa la variación de la salida con entrada nula. La deriva del factor de escala expresa la variación de la sensibilidad.

La sensibilidad o factor de escala es la pendiente de la curva de calibración, que puede ser o no constante a lo largo de la escala de medida. Para un sensor cuya salida

esté relacionada con la entrada x mediante la ecuación y = f(x), la sensibilidad en el punto x_a, $S(x_a)$, es

$$S(x_a) = \left.\frac{dy}{dx}\right|_{X=Xa} \quad \text{.....(2.3)}$$

En los sensores interesa tener una sensibilidad alta y, si es posible, constante. Para un sensor con respuesta

$$y = kx + b \quad \text{.....(2.4)}$$

La sensibilidad es S = k, para todo el margen de valores de x aplicables. Para uno cuya respuesta sea

$$y = kx^2 + b \quad \text{.....(2.5)}$$

La sensibilidad es S = 2kx, y varía a lo largo de todo el margen de medida.

2.4. Linealidad y resolución.

Características como exactitud, precisión y sensibilidad, son suficientes para describir el comportamiento de un sensor. Pero a veces se emplean, además en su lugar, otras que expresan características alternativas o de interés particular para ciertos casos, o bien son complementarias de cara a conocer la idoneidad de un sistema de medida para una aplicación dada.

La linealidad expresa el grado de coincidencia entre la curva de calibración y una línea recta determinada. Según cual sea dicha recta se habla de:

Linealidad independiente: la línea de referencia se define por el método de mínimos cuadrados. De esta forma, el máximo error positivo y el mínimo error negativo son iguales. Es la forma de especificación que suele dar mejor calidad.

Linealidad ajustada al cero: la recta se define también por el método de los mínimos cuadrados, pero con la restricción adicional de pasar por cero.

Linealidad Terminal: la recta se define por la salida sin entrada (o la menor del margen de medida) y la salida teórica máxima, correspondiente a la mayor entrada admitida.

Linealidad a través de los extremos: la recta se define mediante la salida real cuando la entrada es la menor del alcance especificado, y la salida real cuando la entrada es la máxima del alcance especificado.

Linealidad teórica: la recta es la definida por las provisiones teóricas formuladas al diseñar el sensor.

En la figura 2.4 se representan estas distintas rectas para un sensor con una curva de calibración dada. Resulta pues, que la linealidad expresa hasta que punto es constante la sensibilidad del sensor, pero para que un sensor sea válido no es indispensable que sea lineal. El interés de la linealidad esta en que la conversión lectura valor-medido es más fácil si la sensibilidad es constante, pues entonces basta multiplicar la indicación de salida por un factor constante para conocer el valor de la entrada. Además, en instrumentos lineales la no linealidad equivale a la inexactitud.

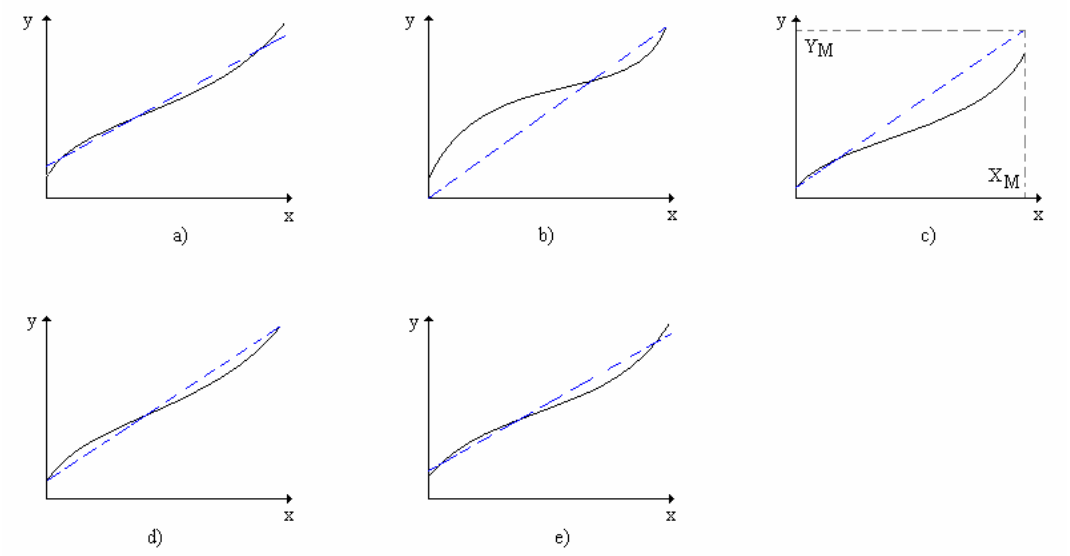


Figura 2.4. Rectas de referencia tomadas para definir la linealidad, a) mínimos cuadrados b) mínimos cuadrados ajustados al cero; c) Terminal d) a través de los extremos e) teórica.

2.4.1 Errores sistemáticos.

La calibración estática de un sensor permite detectar y corregir los denominados errores sistemáticos. Se dice de un error que es sistemático cuando en el curso de varias medidas de una magnitud de un denominado valor, hechas en las mismas condiciones, o bien permanece constante en valor absoluto y signo, o bien varía de acuerdo con una ley definida cuando cambian las condiciones de medida. Dado que también el tiempo es una condición de medida, estas deben de ser realizadas en un intervalo de tiempo breve. Los errores sistemáticos dan lugar, pues, a un sesgo en las medidas.

La posibilidad de estos errores se entiende si se considera que en el resultado de una medida influye no sólo el aparato empleado para efectuarla si no también el método, el operario (en algunos casos) y toda una serie de circunstancias (climáticas, mecánicas o eléctricas) que nunca son ideales, constantes y conocidas todas. La presencia de errores sistemáticos puede describirse, por tanto, midiendo la misma magnitud que en dos aparatos distintos, o con dos métodos distintos, o dando las lecturas dos operarios distintos, o cambiando de forma ordenada las condiciones de medida y viendo su efecto en el resultado. Para juzgar sobre la consistencia de los

resultados obtenidos hay que recurrir a criterios estadísticos. En cualquier caso siempre hay un cierto riesgo de que un error sistemático pase inadvertido, incluso en las medidas de mayor calidad. El objetivo será pues, tener un riesgo muy pequeño de que haya errores grandes no detectados.

2.4.2 Errores aleatorios

Los errores aleatorios son los que permanecen una vez eliminadas las causas de errores sistemáticos. Se manifiestan cuando se mide repetidamente la misma magnitud, con el mismo instrumento y el mismo método.

Los errores aleatorios se denominan también errores accidentales o fortuitos, y ello da a entender que pueden ser inevitables. la ausencia de variaciones de unas a otras lecturas cuando se están realizando una serie de medidas de la misma magnitud con el mismo sistema de medida, no es necesariamente una indicación de ausencia de errores aleatorios. Puede suceder, por ejemplo, que el instrumento no tenga suficiente resolución, es decir, que su capacidad para apreciar pequeños cambios en la magnitud medida sea muy limitada, de modo que no sean detectados por el operario en el dispositivo final de lectura.

La presencia de errores aleatorios hace que después de realizar una o varias medidas de una determinada magnitud se tenga una incertidumbre sobre el verdadero valor de ésta (valor exacto). Cuanto mayor sea dicha incertidumbre, evaluada mediante parámetros estadísticos, menos repetible es la medida. Si además hay errores sistemáticos, el resultado final diferirá del correcto y, por tanto, la medida será inexacta.

Si se hace la medida de varias lecturas, los errores aleatorios se cancelan por ser errores accidentales o fortuitos, y quedan sólo los errores sistemáticos. Ya que estos son reproducibles, se pueden conocer para unas condiciones de medida dadas y corregir la lectura cuando se mida en las mismas condiciones. Esta determinación de la diferencia entre el valor verdadero y el valor obtenido se realiza durante la calibración, en unas condiciones dadas, y normalmente durante este proceso se ajusta el instrumento para eliminar dicho error. Cuando se realice una medida aislada, en las mismas condiciones, quedará sólo la componente aleatoria del error.

Sucede en la práctica, sin embargo, que durante el proceso de calibración sólo se puede eliminar los errores sistemáticos en condiciones muy específicas, por lo que es posible que en otras condiciones se tengan errores de este tipo incluso superiores a los aleatorios que el fabricante recoge en las especificaciones. Aunque no hay obligatoriedad de hacerlo así, lo habitual es especificar el margen que es probable contenga el verdadero valor, dando de alguna forma la diferencia entre el valor máximo y el mínimo.

3. Medición de la Temperatura

Aunque existe cierta familiaridad con la temperatura como la medida de lo caliente o lo frío, no es fácil definirla exactamente. De acuerdo con las sensaciones fisiológicas el nivel de temperatura se expresa en un sentido cualitativo con palabras como congelado, frío, tibio o caliente. Sin embargo, no es posible asignar valores numéricos a las temperaturas solo con base en las sensaciones. Además los sentidos pueden estar equivocados. Una silla metálica, por ejemplo, se percibirá como mucho más fría que una de madera, aunque ambas estén a la misma temperatura.

Algunas propiedades de los materiales cambian con la temperatura, de manera repetitiva y predecible, cambio que constituye la base para una medición exacta de esta variable. En este capítulo no se discutirá a detalle cada uno de los aspectos ligados con la temperatura, como las relaciones existentes de ésta, y los niveles de energía de los líquidos y sólidos, pues hay un gran número de aspectos teóricos, dentro del campo de la termodinámica, relacionados con el concepto de la temperatura que requieren de un extenso y exhaustivo estudio. Por lo que únicamente se acotará una definición básica, acorde a las necesidades de comprensión, para poder definir los diferentes métodos que se utilizan en la medición de la temperatura, y así entender la función del transductor ocupado en este proyecto. En esta tesis sólo nos ocuparemos de las formas de medir la temperatura con métodos in situ (de contacto).

3.1. La Temperatura

Una característica importante de los sistemas a escala microscópica es que sus moléculas individuales se mueven por el con cierta velocidad, vibran unas en torno de otras y rotan alrededor de un eje durante su movimiento aleatorio. La velocidad promedio y el grado de actividad de las moléculas son proporcionales a la Temperatura

Por lo anterior, se puede definir a la Temperatura de un cuerpo como una medida de la energía cinética de las moléculas de una sustancia, debida a la agitación térmica.

También, podemos decir que la temperatura es el potencial de flujo de **calo**r. Ya que, **calor** es energía, específicamente energía en transferencia, debido a la diferencia de temperaturas entre un sistema y su entorno o entre dos sistemas, estancias o cuerpos.

3.2 Escalas de temperatura.

Las escalas de temperatura más usuales son la Fahrenheit y la centígrada. Estas escalas de temperatura se basan en dividir en un número de partes iguales el intervalo de temperatura comprendido entre los puntos de fusión y ebullición del agua a la presión atmosférica estándar. La escala de grados centígrados asigna 100 unidades entre ambos puntos, mientras que la escala Fahrenheit asigna 180. La escala en grados centígrados absoluta se llama también escala Kelvin; mientras que a la escala Fahrenheit absoluta se le denomina Rankine. Las dos escalas absolutas se han definido de tal manera que se apeguen lo más posible a la escala termodinámica de temperatura.

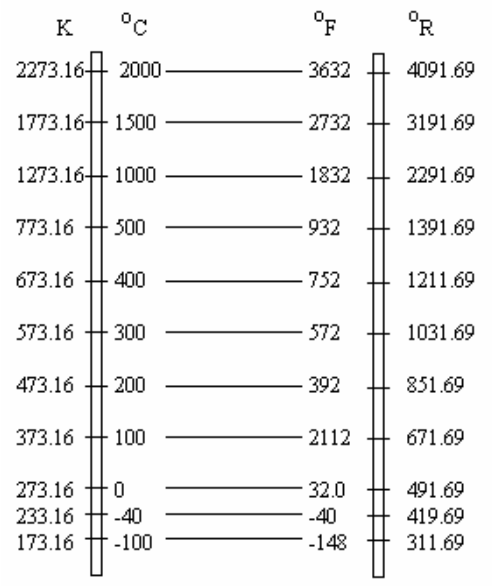


Figura 3.1. Relación entre las escalas Fahrenheit y centígrada

El punto de ebullición del agua se toma arbitrariamente como 100 en la escala de los grados centígrados, y 212 en la escala Fahrenheit. La relación entre ambas escalas se muestra en la figura 3.1. De la figura resultan evidentes las siguientes relaciones:

$$^{0}F = 32.0 + \frac{9}{5}\,^{0}C \quad \text{.....} \quad (3.1)$$

$$^{0}R = \frac{9}{5}\,K \quad \text{.....} \quad (3.2)$$

3.3 Sensores de temperatura.

Probablemente sea la temperatura el parámetro físico más común que se mide en una aplicación electrónica, incluso en muchos casos en que el parámetro de interés no es la temperatura, ésta se ha de medir para incluir indirectamente su efecto en la medida deseada. La diversidad de sus aplicaciones ha condicionado igualmente una gran proliferación de dispositivos sensores y transductores, desde la sencilla unión bimetálica de los termostatos, hasta los dispositivos semiconductores más complejos.

La temperatura es un factor de medida engañoso debido a su simplicidad. A menudo pensamos en ella como un simple número, pero en realidad es una estructura estadística cuya exactitud y repetitividad pueden verse afectadas por la masa térmica, el tiempo de medida, el ruido eléctrico y los algoritmos de medida.

Dicho de otra forma, la temperatura es difícil de medir con exactitud aún en circunstancias óptimas, y en las condiciones de prueba en entornos reales es aún más difícil. Entendiendo las ventajas y los inconvenientes de los diversos enfoques que existen para medir la temperatura, resultará más fácil evitar los problemas y obtener mejores resultados.

En los siguientes apartados se comparan, de manera brevemente cualitativa, los tipos más corrientes de transductores de temperatura que se usan en los sistemas de adquisición de datos: detectores de temperatura de resistencia (RTD), termistores, sensores de CI (circuitos integrados) y termopares. La elección de un transductor de temperatura adecuado y su correcta utilización puede marcar la diferencia entre unos resultados equívocos y unas cifras fiables, aunque esto también dependerá de la aplicación en la que se vaya a utilizar el transductor.

De ahora en adelante se comparan los cuatro tipos de transductores de temperatura básicos, in situ (de contacto), de los que derivan algunos otros. Una vez conocida la forma en que opera cada tipo de transductor de temperatura, se analizarán sus principales características para determinar cuales son los factores más importantes a considerar en la elección del transductor de temperatura, que utilizará para desarrollar el sistema de adquisición de datos.

1. Termopares: Los termopares utilizan la tensión eléctrica generada en la unión de dos metales en contacto térmico, debido a sus distintos comportamientos eléctricos.

2. Resistivos: Lo constituyen las RTD (Resistance Temperature Detector) basados en la dependencia de la resistividad de un conductor con la temperatura, están caracterizados por un coeficiente de resistividad positivo PTC (Positive Thermal Coefficient). Los sensores de temperatura de Pt (platino) operan basándose en el cambio de la resistencia eléctrica en función de la variación de temperatura.

3. Resistivos de coeficiente negativo como son las NTC (Negative Thermal Coefficient), que se llaman Termistores y están caracterizados por un coeficiente de temperatura negativo.

4. Semiconductores: Se basan en la variación de la conducción de una unión p-n polarizada directamente.

3.4 Sensores de Temperatura con Termopares

El método más común para medir la temperatura por efectos eléctricos es el que utiliza el termopar. Cuando se juntan dos metales diferentes, como se muestra en la figura 3.2, se produce una fuerza electromotriz entre los puntos A y B, la cual es, principalmente, función de la temperatura de la junta. Este fenómeno se conoce como efecto **Seebeck**. Si los dos materiales se conectan a un circuito externo, de manera que circule corriente, la Fem (fuente de fuerza electromotriz) puede alterarse ligeramente. Además si existe un gradiente de temperatura en uno o ambos materiales, la fem, también puede tener otra ligera alteración. Por lo tanto, en el circuito termoeléctrico se presentan tres fuentes de fuerza electromotriz: la fem de Seebeck, causada por la unión de dos metales diferentes; la fem de Peltier, ocasionada por el flujo de corriente en el circuito; y la fem de Thomson, que se debe a la existencia de gradientes de temperatura en el material. La fem de Seebeck es de primordial importancia, ya que depende de la temperatura de la junta. Si se mide cuidadosamente la fem generada en la junta de dos metales diferentes como función de la temperatura, dicha junta nos puede servir para la medición de la temperatura. El principal problema para lograr lo anterior es la medición del potencial, ya que cuando se conectan los metales al circuito medidor, se genera otra

fem en las uniones de estos con los alambres que sirven de conexión al circuito medidor. Esta fem dependerá de la temperatura de la última junta y, por lo tanto, se debe tener en cuenta este potencial adicional.

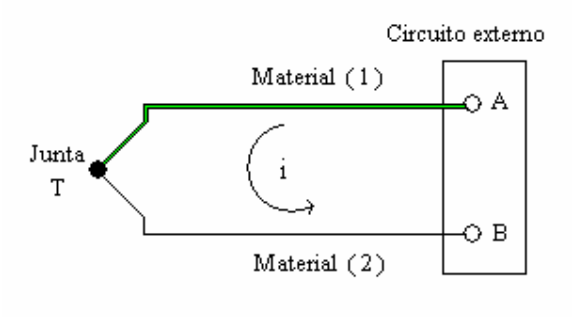


Figura 3.2 Junta de dos metales diferentes indicando el efecto termoeléctrico.

Para el análisis de circuitos termoeléctricos, se tienen dos leyes generales:

1. Se coloca un tercer metal en un circuito termoeléctrico como el mostrado en la figura 3.3, la fem neta del circuito no se vera alterada siempre y cuando las nuevas conexiones se mantengan a la misma temperatura. Este postulado se puede demostrar con la ayuda de la segunda ley de la termodinámica y se conoce como *la ley de los metales intermedios.*

2. En los arreglos mostrados en la Figura 3.4, los circuitos de termopares están construidos con los mismos materiales, pero operan entre dos límites de temperatura diferentes. El circuito de la figura 3.4a genera una fem de E1 entre las temperatura T1 y T2; el circuito de la figura 3.4b proporciona una fem de E2 entre las temperaturas T2 y T3. *La ley de temperaturas intermedias* establece que el mismo circuito generara una fem de E3 = E1 + E2 cuando opere entre las temperaturas T1 y T2, como se muestra en la figura 3.4c.

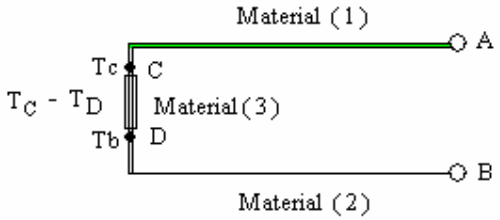


Figura 3.3 Influencia de un tercer metal en un circuito termo eléctrico; ley de los metales intermedios

Se puede observar que todos los circuitos de termopares involucran cuando menos dos juntas. Si se conoce la temperatura de una de ellas, entonces la temperatura

de la otra se puede calcular fácilmente conociendo las propiedades termoeléctricas de los materiales. [1]

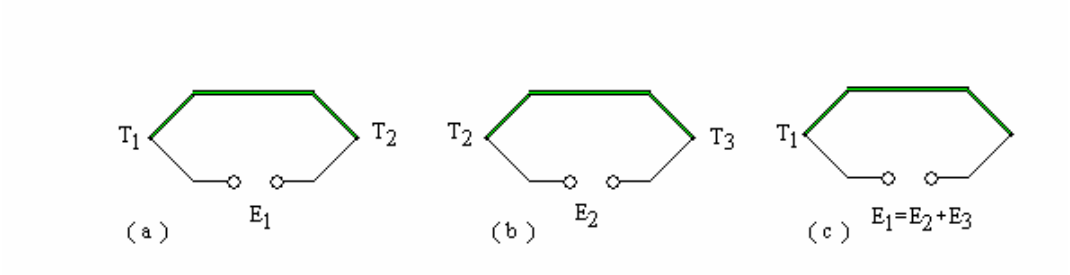


Figura 3.4 Circuitos que ilustran la ley de temperaturas intermedias

Un arreglo comúnmente usado para establecer la temperatura de referencia, consiste de un baño de hielo. Concretamente una mezcla en equilibrio de hielo y agua destilada saturada de aire produce una temperatura conocida de 32 ^{o}F. Es común expresar la fem obtenida en términos del potencial generado manteniendo la junta de referencia a 32 ^{o}F. Donde el voltaje E generado por un solo termopar, generalmente se expresa como:

$$E = AT + \frac{1}{2}BT^2 + \frac{1}{3}CT^3 \quad \text{.....(3.3)}$$

Donde T es la temperatura en ^{o}C. y E esta basada en una temperatura de referencia de 0 ^{o}C, las constates A, B y C dependen de los materiales de la junta.

La sensibilidad o tensión termoeléctrica de un termopar, esta dada por:

$$E = AT + \frac{1}{2}BT^2 + \frac{1}{3}CT^3 \quad \text{.....(3.4)}$$

Los termopares son baratos y robustos, tienen una estabilidad bastante buena a lo largo del tiempo. Debido a su pequeño tamaño, responden rápidamente a los cambios de temperatura. Tiene una linealidad y exactitud razonable. Debido a que el número de electrones libres en un metal depende de la temperatura y de la composición del metal, dos metales de desigual isoterma, dan una diferencia de potencial que es una función repetible de la temperatura, como se muestra en la figura 3.5. El voltaje resultante depende de las temperaturas, T1 y T2, de una manera repetible. [1]

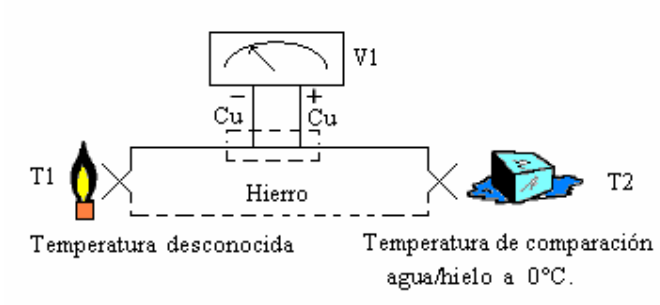


Figura 3.5 Voltaje del termopar con referencia a 0 ^{o}C

Puesto que el termopar es básicamente un dispositivo de medida diferencial, se necesita una temperatura de referencia conocida para una de las uniones, así la temperatura de la otra unión será deducida del voltaje de salida. Los termopares están fabricados de materiales especialmente seleccionados que se han caracterizado exhaustivamente en términos de voltaje con la temperatura de comparación, que normalmente es la del punto de agua/hielo de 0°C. En la tabla 3.1 se presenta un resumen de las juntas de mayor aplicación, y los rangos de operación, así como los valores aproximados de la sensibilidad de varios materiales

La tabla 3.1 muestra los distintos tipos de termopares con su rango típico, su sensibilidad y la designación estándar.

Material de la unión	Rango Típico (°C)	Sensibilidad (µV/°C)	Designación
Pt6%/Rodio – Pt(30%)/Rodio	38 a 1800	7.7	B
Tungsteno(5%)/Renio–Tungsteno(26%)/Renio	0 a 2300	16	C
Cromo- Constant	0 a 982	76	E
Hierro – Constant	0 a 760	55	J
Cromo – Aluminio	-184 a 1260	39	K
Pt(13%)/Rodio – Pt	0 a 1593	11.7	R
Pt(10%)/Rodio – Pt	0 a 1538	10.4	S
Cobre- Constant	-184 a 400	45	T

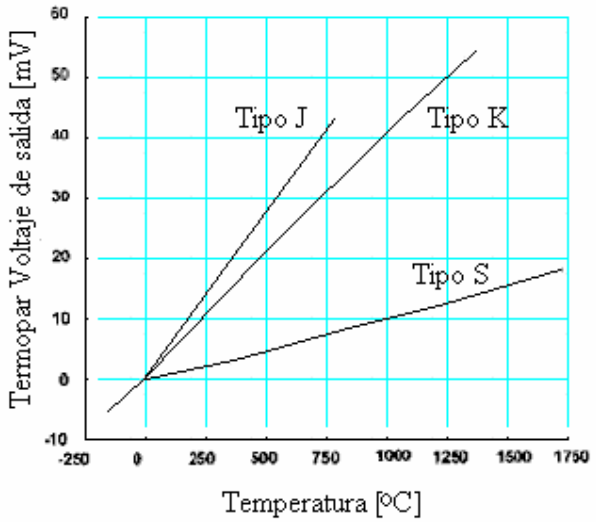


Figura 3.6. Curva de los termopares tipo J, K y S en mV vs °C

3.4.1 Acondicionadores de señal para Termopares.

El hecho de que los termopares sean dispositivos de bajo nivel de tensión, y a la vez de baja impedancia, hace que el acondicionador de señal sea totalmente imprescindible. Aun cuando la linealidad en algunos tipos es pobre el hecho de ser predecible y repetible su comportamiento permite que los termopares puedan ser compensados analógicamente y/o digitalmente.

Analog Devices dispone de unos circuitos integrados acondicionadores de señal para termopares, como el AD594, mostrado en la figura 3.7, para termopares tipo J que tienen un amplificador de instrumentación y un compensador lineal, una salida de alarma de rotura o desconexión del termopar, se alimenta a +5V y suministra una salida de 10mV/ºC.

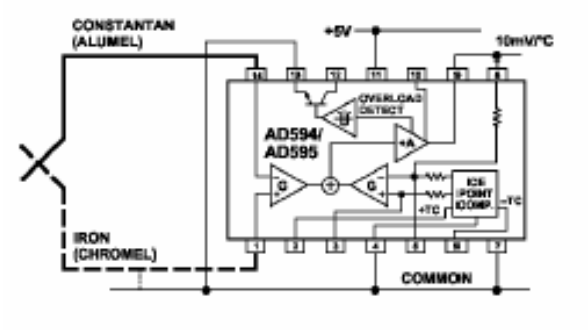


Figura 3.7 Acondicionador de señal de un termopar

3.5. Sensores de Temperatura con elementos Resistivos

El termómetro de resistencia eléctrica es un dispositivo muy exacto para la medición de temperaturas, que consiste en un elemento resistivo que se expone a la temperatura por medir. Existen dos tipos de termómetros basados en la propiedad de resistencia eléctrica; estos son los que tienen un coeficiente de resistividad positiva, y los de coeficiente de resistividad negativo, los RTD (Resistance Temperature Detector) y los termistores respectivamente. Independientemente del coeficiente de resistividad, el cambio en la resistencia eléctrica del material constituirá una indicación de la temperatura. Existen varios tipos de materiales que pueden utilizarse como elementos resistivos. En la tabla 3.2 se muestran las características de alguno de ellos. El coeficiente de resistencia por temperatura α se define por:

$$\alpha = \frac{R_2 - R_1}{R_1 T_2 - R_1 T_1} \quad \text{.....(3.5)}$$

Donde R_1 y R_2, son las resistencias del material a las temperaturas T_1 y T_2 respectivamente. La relación anterior se aplica generalmente en un rango limitado de temperaturas, en el que la variación de la resistencia con la temperatura sea aproximadamente lineal. [1]

Cuando se desea abarcar rangos mayores, la resistencia del material se expresa generalmente por la relación cuadrática siguiente.

$$R = R_0\left(1 + aT + bT^2\right) \quad \text{..... (3.6)}$$

Donde ***R*** es la resistencia a la temperatura T.

R₀ es la resistencia a 0 oC

a y ***b*** son constantes determinadas experimentalmente

Tabla 3.2 Coeficientes de resistencia por temperatura α a temperatura ambiente

Material	Coeficiente de Temperatura α
Níquel	0.0067
Hierro (aleaciones)	0.002 a 0.006
Tungsteno	0.0048
Aluminio	0.0045
Cobre	0.0043
Plomo	0.0042
Plata	0.0041
Oro	0.004
Platino	0.00392
Mercurio	0.00099
Magnanina	$\pm$ 0.00002
Carbón	-0.0007
Electrolitos	-0.02 a -0.09
Semiconductor (Termistores)	-0.068 a +0.14

La medida de temperatura utilizando sondas de resistencia depende de las características de la resistencia eléctrica, en función de la temperatura que son propias del elemento de detección. El elemento consiste usualmente en un arrollamiento de hilo muy fino, de conductor adecuado, bobinado entre capas de material aislante y protegido con un revestimiento de vidrio o de cerámica.

El material que forma el conductor se caracteriza por el llamado ***"coeficiente de temperatura de resistencia"***, como ya se ha mencionado, y que se expresa, a una temperatura especificada, en la variación de la resistencia en ohmios del conductor por cada grado que cambia su temperatura.

La relación entre estos factores puede verse en la expresión lineal siguiente:

$$R_t = R_0(1+\alpha t) \qquad (3.7)$$

Donde : $\mathbf{R_0}$ = Resistencia en ohms a 0°C, $\mathbf{R_t}$ = Resistencia en ohms t °C. y $\boldsymbol{\alpha}$ = Coeficiente de temperatura de la resistencia.

Los metales puros tienen un coeficiente de resistencia de temperatura positivo bastante constante. **El coeficiente de resistencia de temperatura**, generalmente llamado **coeficiente de temperatura** es la razón de cambio de resistencia al cambio de temperatura. Un coeficiente positivo significa que la resistencia aumenta a medida que aumenta la temperatura. Si el coeficiente es constante, significa que el factor de proporcionalidad entre la resistencia y la temperatura es constante y que la resistencia y la temperatura se graficarán en una línea recta.

Cuando se usa un alambre de metal puro para la medición de temperatura, se le refiere como detector resistivo de temperatura, o RTD (por las siglas en inglés de resistive temperature detector).

El detector de temperatura de resistencia (RTD) se basa en el principio, según el cual la resistencia de todos los metales depende de la temperatura. La elección del platino en los RTD de máxima calidad, permite realizar medidas más exactas y estables, hasta una temperatura de aproximadamente 500 ºC. Los RTD más económicos utilizan níquel o aleaciones de níquel, pero no son tan estables ni lineales como los que emplean platino.

Los dispositivos RTD más comunes están construidos con una resistencia de platino (Pt), llamadas también PRTD, aunque también se utilizan otros materiales cuyas características se recogen en la tabla 3.3 siguiente. Típicamente tienen una resistencia entre 20Ω y 20kΩ. La ventaja más importante es que son lineales dentro del rango de temperatura entre –200ºC y 850ºC.

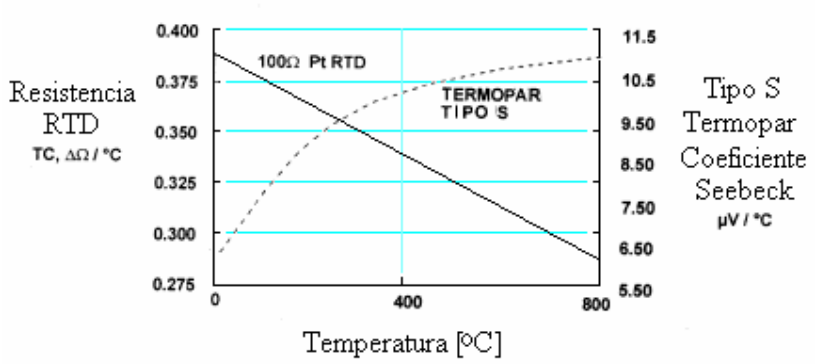
Tabla 3.3 Rangos de medición de temperatura para algunos dispositivos RTD

Material	Rango de temperatura (ºC)	Variación coef (%/ºC a 25ºC)
Platino	-200 a + 850	0.39
Niquel	-80 a 320	0.67
Cobre	-200 a +260	0.38
Niquel-acero	-200 a +260	0.46

En cuanto a las desventajas, el platino encarece los RTD, y otro inconveniente es el auto calentamiento. Para medir la resistencia hay que aplicar una corriente, que, por supuesto, produce una cantidad de calor que distorsiona los resultados de la medida.

Una tercera desventaja, que afecta al uso de este dispositivo para medir la temperatura, es la resistencia de los RTD. Al ser tan baja, la resistencia de los hilos conductores que conectan el RTD puede provocar errores importantes.

Cabe hacer notar que el termómetro de resistencia de platino se utiliza en la escala internacional de temperatura, para el rango definido por los puntos del oxigeno y del antimonio. Dependiendo de la aplicación, existen varios métodos para fabricar termómetros por resistencia. En todos los casos se debe estar seguro de que la resistencia esta libre de esfuerzos mecánicos, y esta colocada de tal manera que no tenga contacto con la humedad, ya que esta perturba la medición. En la Figura 3.8 se muestra de forma comparativa, la respuesta de un Termómetro RTD y de un termopar de tipo S



La figura 3.8 de la gráfica, muestra la respuesta de la resistencia de un RTD con la temperatura y la comparativa respecto a un Termopar tipo S.

Sensibilidad

Un termómetro RTD, cuya resistencia varía linealmente con la temperatura ambiente implica una función de la resistencia con la temperatura del a siguiente forma:

$$R = R_0\left[1 + \alpha\left(T - T_0\right)\right] \quad (3.8)$$

Donde R_0 es la resistencia a la temperatura de referencia T_0, la sensibilidad esta dada entonces por:

$$S = \frac{dR}{dT} = \alpha R_0 \quad (3.9)$$

R_0 dependerá de las dimensiones de la resistencia del material utilizado.

Acondicionadores de señal para las RTD

Hay muchas maneras de acondicionar la señal que se recibe de un RTD. La primera propuesta de Analog Devices es con un amplificador de instrumentación y su circuitería adicional, figura 3.9 a. La segunda propuesta es algo similar con el circuito ADT70 perfectamente adecuado para acondicionar los PRTD de Pt (de platino), que entrega una salida de 5mV/ºC cuando se utiliza una RTD de 1kΩ. Figura 3.9 b

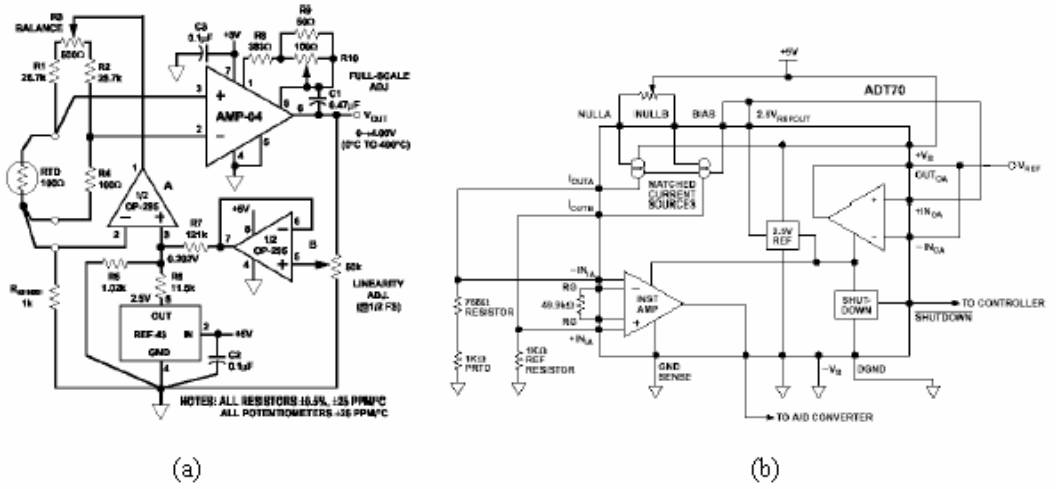


Figura 3.9 Las figuras anteriores son sólo dos ejemplos de la vasta cantidad de métodos de acondicionamiento de señal para los termómetros con RTD, los cuales se muestran como ejemplo para darse una idea de las posibilidades de acondicionamiento de señal en estos sensores.

3.5.1 Los termistores

Cuando se usan **óxidos metálicos** (materiales semiconductores de silicio) para la medición de temperatura, el material de oxido metálico es conformado en forma que se asemeja a un pequeño capacitor. El dispositivo formado así se llama Termistor. Por lo regular, **Los termistores** tienen coeficientes de temperatura negativos grandes que no

son constantes. En otras palabras, el cambio de resistencia por unidad de cambio de temperatura es mucho mayor que para el metal puro, y el cambio es en dirección contraria, a diferencia de los termómetros RTD, en los Termistores la resistencia disminuye a medida que se aumenta la temperatura. El hecho de que el coeficiente no sea constante, significa que el cambio en la resistencia, por unidad de cambio de temperatura es diferente a diferentes temperaturas.

La poca linealidad de los termistores, los hace poco apropiados para la medición de temperatura a través de rangos amplios. Sin embargo, para la medición de temperaturas dentro de bandas angostas, están muy bien dotados, pues dan una gran respuesta a un cambio de temperatura pequeño.

Como regla general, los termistores son preferibles cuando la banda de temperaturas esperada es angosta, mientras que los RTD (Detectores Resistivos de Temperatura) son preferibles cuando la banda de temperatura esperada es amplia. [2]

El termistor es un dispositivo semiconductor que tiene un coeficiente negativo de resistencia por temperatura, en contraste con el coeficiente negativo que muestran la mayoría de los metales. La variación de la resistencia con la temperatura sigue un comportamiento exponencial, en vez de una relación polinomial:

$$R = R_0 e^{\left[\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)\right]} \quad \text{.....(3.10)}$$

Donde R_0 es la resistencia a la temperatura de referencia T_0 y β es una constante determinada experimentalmente. El valor numérico de β esta comprendido, generalmente, entre 3500 y 4600 K, dependiendo del material del termistor y de la temperatura.

El termistor es un dispositivo extremadamente sensible y, por medio de una calibración adecuada, se pueden obtener precisiones del orden de 0.01 ^{0}C. Además, otra característica interesante del termistor es que se puede utilizar para compensar incrementos de resistencia por temperatura en circuitos eléctricos, lo cual se debe a su coeficiente negativo de temperatura...

La sensibilidad a la temperatura del termistor, se obtiene derivando la función del comportamiento de la resistencia con la temperatura:

Si la función que define el comportamiento de la resistencia del termistor con la temperatura es la ecuación 3.8, entonces la sensibilidad del termistor será la siguiente ecuación:

$$S = \frac{dR}{dT} = R_0 e^{\left[\beta\left(\frac{1}{T} - \frac{1}{T_0}\right)\right]}\left(\frac{-\beta}{T^2}\right) \quad \text{.....}$$
(3.11)

En la figura 3.10 se compara la resistividad de tres termistores con la del platino, y en la figura 3.11, se muestra una curva voltaje-corriente típica de un termistor obtenida en condiciones estáticas. [1]

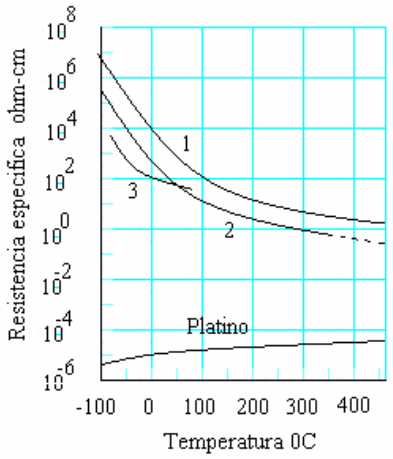


Figura 3.10 Resistividad de tres termistores comparadas con las del platino.

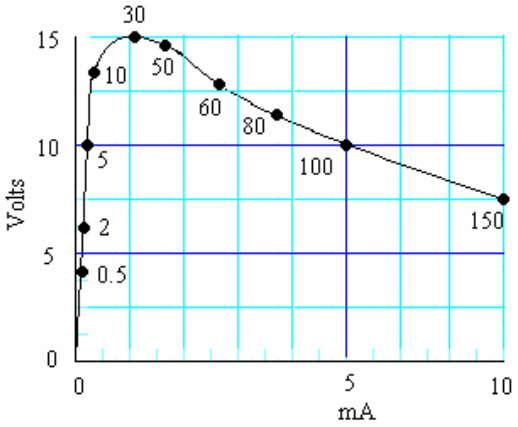


Figura 3.11 Características estáticas voltaje-corriente para un termistor típico

Los termistores NTC (Coeficiente de Temperatura Negativo) son resistencias muy sensibles a la temperatura, y están fabricados de una mezcla de óxidos de Mn, Ni, Co, Cu, Fe y están moldeados en un cuerpo cerámico de varios tamaños, típicamente tienen una resistencia entre 50Ω y 1M Ω a 25ºC y una sensibilidad del 4%/ºC a 25ºC. El efecto del Coeficiente Negativo, con la Temperatura puede resultar de un cambio externo de la temperatura ambiente, o un calentamiento interno debido al efecto Joule de una corriente que fluye a través del termistor. [1] La curva del termistor se puede linealizar con una resistencia montada en paralelo con la del termistor NTC. Figura 3.12

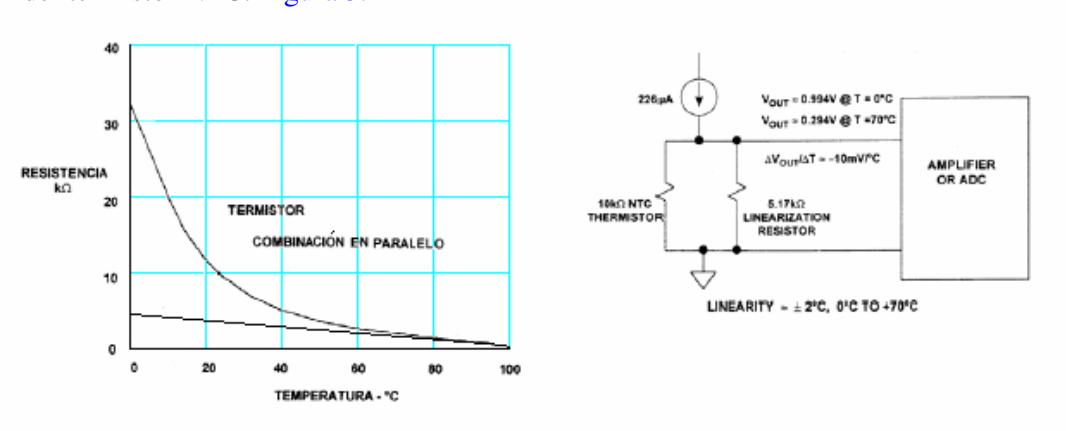


Figura 3.12 Curva del termistor con una resistencia montada en paralelo y circuito de linealización.

Los termistores se pueden conectar directamente a la entrada de un convertidor A/D y después a un microcontrolador cualquiera para procesar su señal, como se muestra en la figura 3.13, con los Microconverters de Analog Devices, la familia AduC8xx que incluye todo un sistema completo de adquisición de precisión, así como un microcontrolador con memoria Flash que puede procesar y visualizar los datos. Aunque el inconveniente de este sistema integrado es su costo.

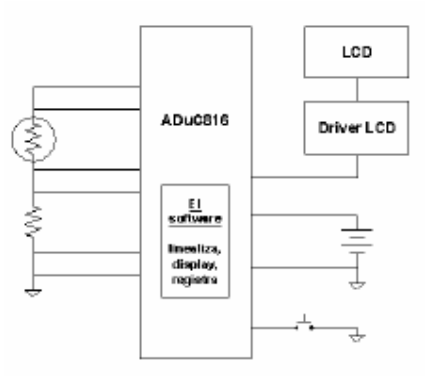


Figura 3.13. Los Microconverters de Analog Devices, la familia AduC8xx incluyen todo un sistema completo de adquisición de precisión, para las señales provenientes de los termistores.

3.6 Transductores de temperatura con semiconductores

La característica directa de un diodo depende de la temperatura, y si bien esto se considera un inconveniente en muchos casos, es posible utilizar dicha dependencia para medir la temperatura u otras magnitudes que se puedan asociar a un cambio de temperatura. Sucede, sin embargo que dicha dependencia ni es lineal, ni es suficientemente repetitiva de acuerdo con las exigencias de los sistemas de medida. Se prefiere por ello emplear la dependencia térmica de la tensión base emisor, V_{BE} de un transistor cuya corriente de colector sea constante. Figura 3.14

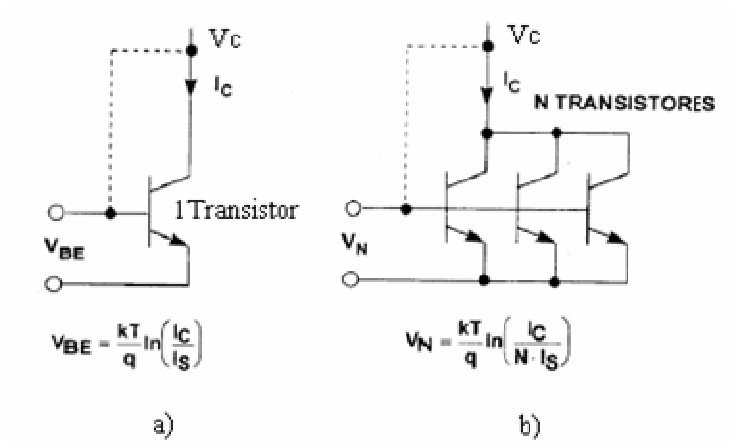


Figura 3.14. a) V_{BE} de un transistor b) V_N de un arreglo de transistores

De acuerdo con el modelo de Ebers-Moll, la corriente de colector de un transistor ideal es:

$$i_C = \alpha_F I_{ES}\left(e^{\frac{qV_{BE}}{kT}} - 1\right) - i_{cs}\left(e^{\frac{-qV_{CB}}{kT}} - 1\right) \quad (3.12)$$

donde:

- α_F es la relación de transferencia directa de corriente
- I_{ES} es la corriente de saturación de emisor
- q. es la carga del electrón, q = $1.60\text{x}10^{-19}$ C
- V_{BE} es la tensión de base-emisor
- K es la constante de Boltzman, k = 1.3807×10^{-23} J/K
- T es la temperature absoluta
- I_{CS} es la corriente de saturación de colector
- V_{CB} es la tensión colector-base.

El producto α_F I_{ES} se designa como I_S. En la zona activa, $i_C >> I_S$, y si además la tensión colector-base se hace cero, de la ecuación para la corriente de colector de un transistor ideal, se deduce que:

$$V_{BE} = \frac{kT}{q}\ln\frac{i_C}{i_S} \quad (3.13)$$

Esta ecuación muestra ya la dependencia de V_{BE} con la temperatura, pero I_S es también función de la temperatura T.

$$i_s = BT^3 e^{\frac{-qV_{go}}{kT}} \quad (3.14)$$

Donde: B es una constante que depende del nivel de dopaje y de la geometría, y es independiente de la temperatura y V_{go} es el ancho de la banda prohibida (1.12 v a 300 K para el silicio). Combinando las ecuaciones 3.13 y de 3.14 se obtiene:

$$V_{BE} = \frac{kT}{q}\ln\frac{i_C}{BT^3} + v_{gO} \quad (3.15)$$

Si la tensión base-emisor correspondiente a una corriente de colector constante I_{CO} a una temperatura dada To se denomina V_{BEO}, se tiene.

$$V_{BE} = \frac{kT}{q}\ln\frac{i_C}{I_{c0}}\left(\frac{T_0}{T}\right)^3 + \left(V_{BEO} - V_{g0}\right)\frac{T}{T_0} + v_{gO} \quad (3.16)$$

Resulta, pues, que la relación entre VBE y T es no lineal, y depende del valor de la corriente de colector. Para analizar la no linealidad, se puede derivar respecto a la temperatura en un punto de corriente de colector constante. Para $i_c = I_{co}$, se obtiene.

$$\left.\frac{dV_{BE}}{dT}\right|_{ic=Ico} = \frac{V_{BEO} - V_{go}}{T_0} - \frac{3k}{q}\left(1 + \ln\frac{T}{T_0}\right) \quad (3.17)$$

El primer término del segundo miembro corresponde a la sensibilidad, mientras que el segundo término indica la no linealidad. Para el silicio son, respectivamente, del orden de -2.1 mV/^{0}C y 0.34 mV/^{o}C.

Esta fuerte no linealidad y la necesidad de trabajar con una corriente de colector que debe mantenerse constante con el tiempo, la temperatura, etc., hacen que esta solución sea poco atractiva en sistemas de alta precisión. La alternativa preferida consiste en emplear dos transistores bipolares cuyas densidades de corriente de emisor tengan una relación constante: Un método para ello consiste en emplear dos transistores idénticos con corrientes de colector distintas, tal como se indica en la figura 3.15a. Suponiendo los dos transistores a la misma temperatura, la diferencia entre las tensiones base-emisor respectiva será:

$$V_d = V_{BE1} - V_{BE2} = \frac{kT}{q}\ln\frac{I_{C1}}{I_{S1}} - \frac{kT}{q}\ln\frac{I_{c2}}{I_{S2}} \quad (3.18)$$

Si los dos transistores se suponen idénticos, se tendrá que $I_{S1} \approx I_{S2}$, y queda

$$V_d = \frac{kT}{q}\ln\frac{I_{C1}}{I_{C2}} \quad (3.19)$$

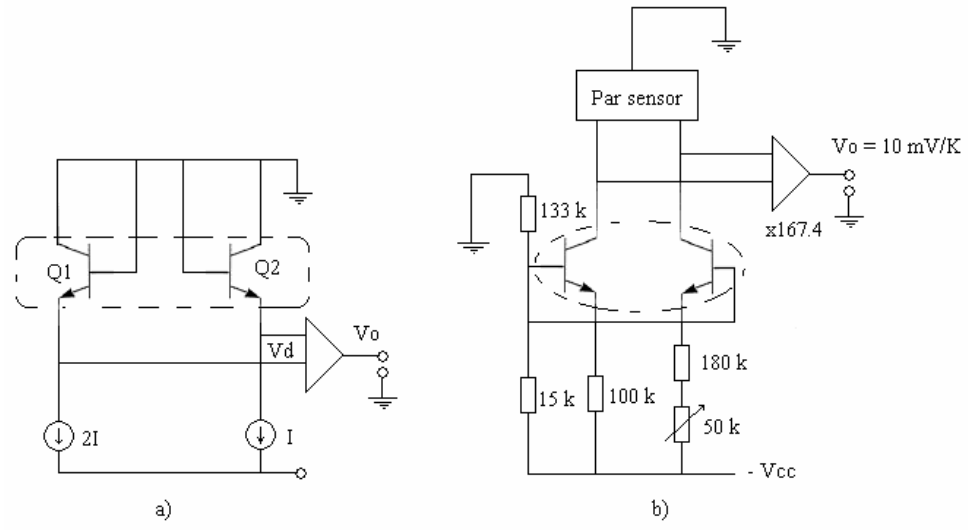


Figura 3.15. a) termómetro basado en la dependencia de la tensión base-emisor con la temperatura en un transistor bipolar. El empleo de dos fuentes de corriente con una relación dada permite prescindir de una referencia estable y confiere mayor linealidad. b) Esquema de las fuentes de corriente.

Así pues, si I_{c1}/I_{c2} es constante, y en la Figura 3.15a Vd será proporcional a T, sin necesidad de mantener una fuente de corriente perfectamente estable. Basta con que sea constante la relación entre dos fuentes de corriente. En la figura 3.15b, $I_{c1}/I_{c2} = 2$, de modo que Vd/T = 59.73 µV/k. Con un amplificador diferencial de ganancia 167.4, la tensión de salida es de 10 mV/K.
Otro método consiste en emplear dos transistores con áreas de emisor distintas, pero con la misma corriente de colector. En la figura 3.16 se presenta el esquema simplificado de un sensor de este tipo que es un convertidor de temperatura-corriente que ha encontrado un amplio uso. Su circuito equivalente es una fuente de corriente de dos terminales que deja pasar una corriente numéricamente igual (expresada en microamperios) a la temperatura absoluta. Los transistores Q3 y Q4 son iguales y constituyen un espejo de corriente, de modo que

$$I_{c1} = I_{C2} = I_T / 2 \quad (3.20)$$

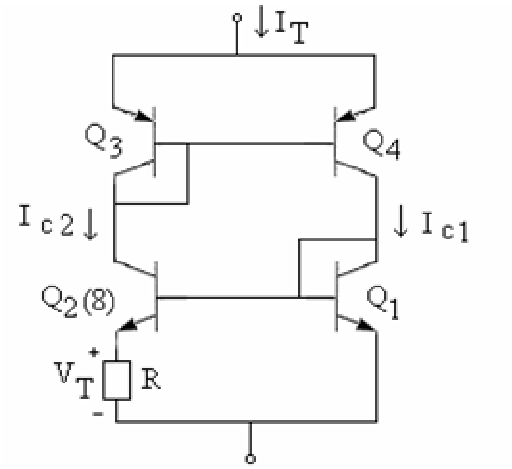


Figura 3.16 Esquema simplificado de un convertidor temperatura-corriente (Analog Devices)

Q2 esta constituido por 8 transistores en paralelo, iguales entre si y a Q1. De este modo la densidad de corriente de emisor es 8 veces mayor que en Q1. Si se designa como I_{C1} la corriente de colector del transistor Q1 y como I_{C2} la corriente de colector de cada uno de los transistores Q2, la tensión de salida es.

$$V_T = V_{BE1} - V_{BE2} = \frac{kT}{q}\ln\frac{I_1}{I_2} = \frac{k}{q}\ln(8)T = 179T \quad (3.21)$$

La corriente de entrada será:

$$I_T = 2I_{c2} = 2V_T/R \quad (3.22)$$

Si se ajusta R hasta que sea 358 Ω, se tendrá con independencia de la tensión aplicada (dentro de un cierto margen):

$$\frac{I_T}{I} = 1\mu A / k \quad (3.23)$$

Tener una salida en forma de corriente es una ventaja para las medidas remotas, ya que la longitud de los cables y las posibles tensiones de interferencias debidas a acoplamientos capacitivos no afectaran por que el circuito tiene baja impedancia equivalente. En la figura 3.17 se presentan 3 aplicaciones de un sensor de este tipo

AD590. La primera de ellas es un termómetro que ofrece una salida en forma de tensión. La segunda muestra varios sensores conectados en serie, de modo que la tensión de salida es proporcional a la temperatura mínima. La tercera consiste en la conexión de varios sensores en paralelo, con lo que se obtiene una tensión proporcional la temperatura media. [4]

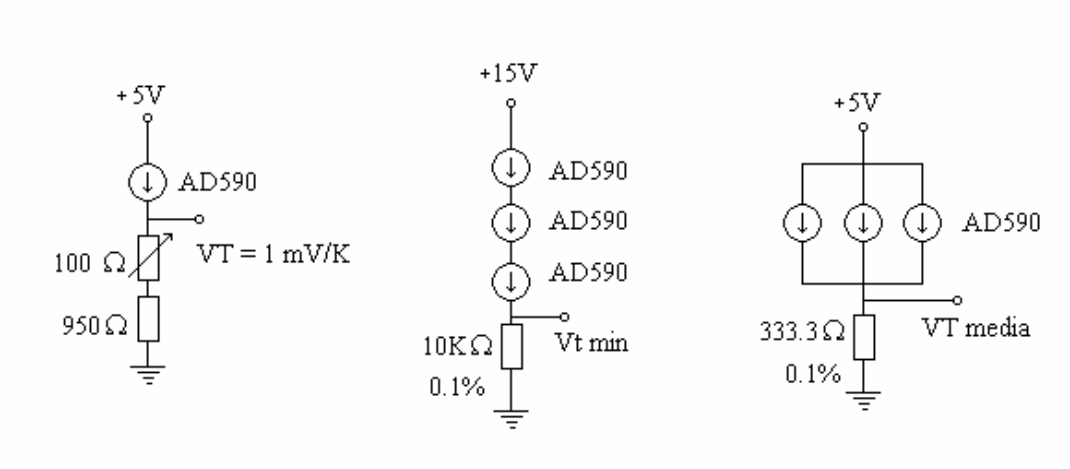


Figura 3.17. Distintas aplicaciones de un convertidor temperatura-corriente (Analog Devices)

3.7 Consideraciones para la selección

Como ya se ha visto, ningún transductor es el mejor en todas las situaciones de medida, por lo que tenemos que saber, cuándo debe utilizarse cada uno de ellos. Como podemos ver, en la Tabla 3.4 donde se comparan los tipos de transductores de temperatura, descritos en esta tesis. En esta tabla se exponen los factores que deben tenerse en cuenta al momento de elegir un transductor: las prestaciones, el alcance efectivo, el precio y la comodidad.

La selección de un sensor de temperatura implica una extensa matriz de consideraciones que es necesario evaluar en la selección del tipo de transductor. Principalmente, es necesario seleccionar un diseño de sensor cuyo elemento sensor alcance la temperatura del fluido o superficie a medir, y la alcance dentro de un tiempo permisible para realizar la medida. La salida de un sensor de temperatura en cualquier momento es simplemente la medida de la temperatura de su elemento sensor (transductor). En algunos casos es difícil asegurar que esta temperatura es la misma que la temperatura que se desea medir.

Tabla 3.4 Consideraciones de selección

	RTD	Termistor	Sensor de IC	Termopar
Ventajas	Más estable. Más preciso. Más lineal que los Termopares	Alto rendimiento Rápido Medida de dos hilos	El más lineal El de más alto rendimiento Económico	Autoalimentado Robusto Económico Amplia variedad de formas físicas Amplia gama de temperaturas
Desventajas	Caro. Lento. Precisa fuente de Alimentación. Pequeño cambio de resistencia. Medida de 4 hilos Autocalentable	No lineal. Rango de Temperaturas Limitado. Frágil. Precisa fuente de Alimentación. Autocalentable	Limitado a < 250 ºC Precisa fuente de alimentación Lento Autocalentable Configuraciones limitadas	No lineal Baja tensión Precisa referencia El menos estable El menos sensible

Para muchas aplicaciones los siguientes criterios de selección son de principal importancia: la naturaleza y las características del fluido o sólido a medir; el rango de medida (y posiblemente los limites de sobre rango); la constante de tiempo; y el tipo de acondicionador de señal asociado, el sistema de datos y el equipo de visualización utilizado. Por ejemplo en los fluidos, cuya temperatura se detecta con sondas de inmersión, estos pueden ser líquidos o gases; pueden ser corrosivos o no corrosivos; oxidantes o reductores; pueden estar estáticos o moviéndose a baja, media, alta o muy alta velocidad; pueden estar libres o contenidos en una tubería, un conducto, un tanque, una vasija o una cavidad, por esto, es indispensable tomar en cuenta todas estas consideraciones, al momento de elegir entre uno u otro transductor de temperatura.

Como ya se ha explicado antes, para cada bloque que compone a un sistema de adquisición de datos, existen varias alternativas que pueden ser exploradas para obtener un óptimo diseño. Para realizar el sistema de adquisición de datos, propuesto en este trabajo, era necesario decidir que tipo de transductor podríamos utilizar. En este caso, ya que el objetivo principal es la construcción y diseño de un sistema de adquisición de datos, cualquier tipo de transductor, puede servir para complementar el desarrollo de este sistema, sin embargo se ha elegido utilizar un transductor de temperatura, y a la temperatura en si misma, como variable a medir, por su importancia y características que definen muchos procesos en varias aplicaciones antes mencionadas. Para complementar de una manera mas apropiada y visualizar los alcances de este trabajo, se ha decidido utilizar un transductor que funcionará a base de dispositivos semiconductores, principalmente utilizando la característica directa del diodo que depende de la temperatura. Este tipo de transductores se pueden implementar, de manera teórica, con arreglos de transistores en la forma de corriente de espejo, sin embargo su funcionamiento practico es deficiente, debido a que debe existir una gran reciprocidad entre la constante β de los transistores utilizados en el arreglo, situación que en la realidad no sucede. Sin embargo el principio teórico mencionado en las figuras 3.15 y 3.16 es la base para la fabricación de Circuitos integrados que permiten definir dos temperaturas de consigna (alta/baja). Algunos de estos dispositivos son fabricados por la empresa Analog Devices, y sus características principales son mostradas en la tabla 3.5.

Tabla 3.5 Características de diversos sensores de temperatura basados en uniones semiconductoras.

Modelo	Sensibilidad	Margen	Exactitud
AD592CN	1 μA/K	-25 ^{0}C a + 105 ^{0}C	0.3 ^{0}C
LM35	± 10 mV/K	-55 ^{0}C a + 150 ^{0}C	± 0.25 ^{0}C
LM335A	10 mV/K	-40 ^{0}C a + 100 ^{0}C	± 1 ^{0}C
MMBTS 102	-2.25 mV/K	-40 ^{0}C a + 150 ^{0}C	± 2 ^{0}C
REF-02A	2.1 mV/K	-55 ^{0}C a + 125 ^{0}C	± 0.5 ^{0}C
TMP-01	5 mV/K	-55 ^{0}C a + 125 ^{0}C	± 0.6 ^{0}C

En este trabajo de tesis será usado el CI AD590, el cual es muy similar al AD592 CN y fácil de conseguir, proporciona un margen de salida de -50 ^{0}C a + 150 ^{0}C, con una sensibilidad de 1 μA/K. Este transductor de temperatura es muy útil para mediciones en sondas de medida, y por tener poca masa son rápidos (1.5 a 10s para cambios de

temperatura de 50 ^{0}C). Además si la sonda esta aislada eléctricamente, permite medir la temperatura de componentes activos en funcionamiento. El AD 590 introduce el menor ruido posible, debido a su arquitectura, es relativamente de bajo costo, ponderado con la calidad y eficiencia que ofrece en mediciones de instrumentación, y por último cuenta con un mayor rango de temperatura, a diferencia de los termopares o los termistores que algunas veces fungen mejor como dispositivos de tipo compuerta que como transductores de temperatura. En seguida se describen algunas características acerca de la estructura del CI AD590

Características generales del CI AD590:

- Corriente Lineal de salida: 1 mA/K.
- Rango de medición: –55.8C a + 150.8C
- Empaquetado Cerámico del Sensor
- El Dispositivo consta de dos terminales: Voltaje de Entrada/Corriente de salida
- Calibración con láser a 60.58^{O}C
- Excelente Linealidad: 60.38^{O} C Sobre todo el rango de lecturas (AD590M)
- Potencia suministrada entre el rango de: +4 V a +30V

1. Para calibrar el sensor de temperatura AD590 se requiere suministrar un voltaje de dc entre sus dos terminales (+4 V a +30 V). Es innecesario aplicar al dispositivo transmisores costosos, filtros, circuitos de compensación y de linealización.

2. State-of-the-art La forma de fabricación en el nivel de juntura, asegura que las unidades AD590 sean fácilmente intercambiables.

3. Interfase de rechazo superior en los resultados de salida. Los requisitos de potencia son bajos (1.5 mWs @ 5 V @ +25°C.). Estas características hacen del AD590 fácilmente aplicable en un sensor remoto.

4. La alta impedancia de entrada (>10 MΩ) proporciona un excelente rechazo de suministro de voltaje en derivación y onda CMRR. Para esta instancia el cambio de potencia de 5v a 10 v resulta en solo 1 μA máximo de cambio de corriente, o 1°C de error equivalente.

5. El AD590 es eléctricamente durable: resistirá un voltaje de hasta 44V y un voltaje inverso de 20v. Fuera de este rango, habrá irregularidades o se deberán revisar los pines para no dañar el dispositivo.

Especificaciones del sensor de temperatura.

Las especificaciones del AD590 son fáciles de implantar a una gran variedad de diferentes aplicaciones. Es importante comprender bien todas las especificaciones y los efectos de suministrar voltaje en el dispositivo, así como los efectos que provoca el medio ambiente en la exactitud.

El AD590 es básicamente un regulador de corriente proporcional a la temperatura absoluta (por sus siglas en ingles PTAT Proportional to Absolute Temperature o bien en español PTA). Esa corriente de salida, es al mismo tiempo igual a un factor de escala de

la temperatura del sensor en Kelvin. Este factor de escala es acotado a un 1µA/K por el fabricante, se ajusta la temperatura indicada en grados con la temperatura actual del medio.

El error en el factor de escala es particularmente fácil de ajustar. La figura 3.18 es la representación más elemental que se puede efectuar de este ajuste. Este circuito recorta la temperatura en el AD590 donde es medida por una referencia de temperatura en el sensor y R limita así a VT = 1mV/K a esa temperatura. Mientras este error es limitado en la temperatura de salida, el efecto es cero sobre todo el rango de temperatura. En más aplicaciones en cuanto a esto hay una conversión de corriente a voltaje en el resistor (Con una corriente de entrada ADC y una referencia) dicho voltaje puede ser recortado por un factor de escala ajustado.

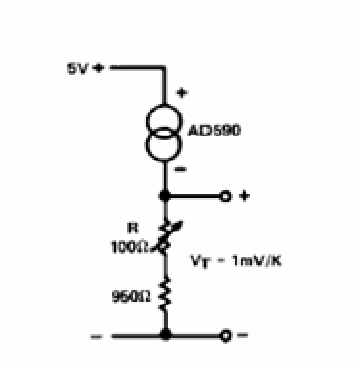


Figura 3.18. Ajuste básico de factor de escala.

El arreglo básico del circuito de la figura 3.18 será utilizado mas adelante en el capítulo 5 como parte del acondicionamiento de la señal de entrada a un amplificador de instrumentación y después a un convertidor analógico digital.

4. Microcontroladores

Los sistemas basados en microcontroladores constituyen la evolución natural de los sistemas digitales. Esta evolución ha sido posible gracias a los avances tecnológicos que el sector de la electrónica ha experimentado a lo largo de su corta vida.

La introducción de circuitos integrados de baja escala de integración (1960) sentó las bases para la realización de sistemas electrónicos. Más adelante los circuitos de alta escala de integración (LSI) permitieron la fabricación del primer microcontrolador integrado, el 4004 de Intel, que revolucionó el mundo electrónico. En la actualidad los circuitos de muy alta escala de integración son muy comunes y permiten la realización de funciones impensables hasta no hace demasiado tiempo. Estos circuitos permiten la integración en un solo dispositivo de más de un billón de transistores.

Los principios basados en microprocesadores se encuentran en la propia aparición del microcontrolador aunque con anterioridad ya se realizaban las funciones equivalentes con dispositivos discretos e incluso con válvulas electrónicas (ENIAC). Las bases de los sistemas basados en microcontroladores no son nuevas en si mismas sino en la forma de desarrollarlas.

En la actualidad los microcontroladores disponen de un amplísimo campo de aplicaciones: electrónica industrial, telecomunicaciones, automóvil, ordenadores, radio y televisión y en general, en todos los procesos en que sea necesario el control y/o tratamiento de señales eléctricas.

La gran difusión de los microcontroladores se debe a varios factores, entre ellos están los siguientes:

a) Gran capacidad funcional.
b) Flexibilidad para el diseño
c) Alta fiabilidad funcional
d) Bajo costo

Los microcontroladores presentan una amplia gama de posibilidades en función de los dispositivos empleados. Si nos centramos en la Unidad Central de Proceso (CPU) cuyo elemento principal es el microprocesador, disponemos de sistemas capaces de procesar información desde un tamaño de 4 bits hasta 128 bits. La velocidad varía desde unos cuantos MHz a 1400 MHz, y el consumo va desde algunos microamperios hasta decenas de miliamperios 5v. Para cada sistema en particular tendremos que elegir los elementos correspondientes a las necesidades en cuestión.

Aún con tanta diversidad de dispositivos, todos presentan una serie de rasgos comunes; lo que nos va a permitir un estudio detallado de los sistemas basados en microcontroladores sin tener que analizar uno a uno los casos particulares. La estructura básica de cada elemento que compone un sistema basado en microcontroladores es fija. Las diferencias existentes entre los distintos sistemas físicos reales son debidas a que se emplean distintos dispositivos para cada aplicación particular. Se supone que para cada aplicación usaremos los elementos más adecuados, y de esta forma obtenemos el rendimiento máximo del sistema para un costo mínimo.

4.1. Controlador y microcontrolador.

Recibe el nombre de controlador el dispositivo que se emplea para el gobierno de uno o varios procesos. Por ejemplo, el controlador que regula el funcionamiento de un horno dispone de un sensor que mide constantemente su temperatura interna y cuando traspasa los límites prefijados, genera las señales adecuadas que accionan los dispositivos que intentan llevar el valor de la temperatura dentro del rango estipulado.

Aunque el concepto de controlador ha permanecido invariable a través del tiempo su implementación física ha variado frecuentemente. Hace más de tres décadas, los controladores se construían exclusivamente con componentes de lógica discreta, posteriormente se emplearon los microprocesadores que se rodeaban con chips de memoria y, dispositivos de E/S sobre una tarjeta de circuito impreso. En la actualidad, todos los elementos del controlador se han podido incluir en un chip, el cual recibe el nombre de microcontrolador. Realmente consiste en un sencillo pero completo computador contenido en el corazón de un circuito integrado.

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador.

Un microcontrolador dispone normalmente de los siguientes componentes:

- Procesador o UCP (Unidad Central de Proceso).
- Memoria RAM para contener los datos.
- Memoria de programa tipo ROM/PROM/EPROM.
- Bus de líneas de E/S para comunicarse con el exterior.
- Diversos módulos para el control de periféricos: temporizadores, compuertas serie y paralelo, conversores Analógico/Digital y conversores Digital/Analógico, etc.
- Generador de impulsos de reloj que sincronizan el funcionamiento de todo el sistema.

Los procesos que para su regulación incorporan un microcontrolador disponen de las siguientes ventajas:

Aumento de prestaciones:
Un mayor control sobre un determinado elemento representa una mejora considerable en el mismo.

Aumento de la fiabilidad:
Al reemplazar el microcontrolador por un elevado número de elementos disminuye el riesgo de averías y se precisan menos ajustes.

Reducción del tamaño en el producto acabado:
La integración del microcontrolador en un chip disminuye el volumen y la mano de obra.

Mayor flexibilidad:

Las características de control están programadas, por lo que su modificación sólo necesita cambios en el programa de instrucciones.

El microcontrolador es en definitiva un circuito integrado que incluye todos los componentes de un computador. Debido a su reducido tamaño es posible montar el controlador en el propio dispositivo al que gobierna. En este caso el controlador recibe el nombre de controlador empotrado (embedded controller).

4.2. Recursos comunes a todos los microcontroladores.

Al estar todos los microcontroladores integrados en un chip, su estructura fundamental y sus características básicas son muy parecidas. Todos deben disponer de los bloques esenciales, procesador, memoria de datos y de instrucciones, líneas de E/S, oscilador de reloj y módulos controladores de periféricos. Sin embargo, cada fabricante intenta enfatizar los recursos más idóneos para las aplicaciones a las que se destinan preferentemente.

Aunque inicialmente todos los microcontroladores adoptaron la arquitectura clásica de von Neumann, en el momento presente se impone la arquitectura Harvard. La arquitectura de von Neumann se caracteriza por disponer de una sola memoria principal donde se almacenan datos e instrucciones de forma indistinta. A dicha memoria se accede a través de un sistema de buses único (direcciones, datos y control).

La arquitectura Harvard dispone de dos memorias independientes, una que contiene sólo instrucciones y otra sólo datos. Ambas disponen de sus respectivos sistemas de buses de acceso y es posible realizar operaciones de acceso (lectura o escritura) simultáneamente en ambas memorias. Figura 4.1.

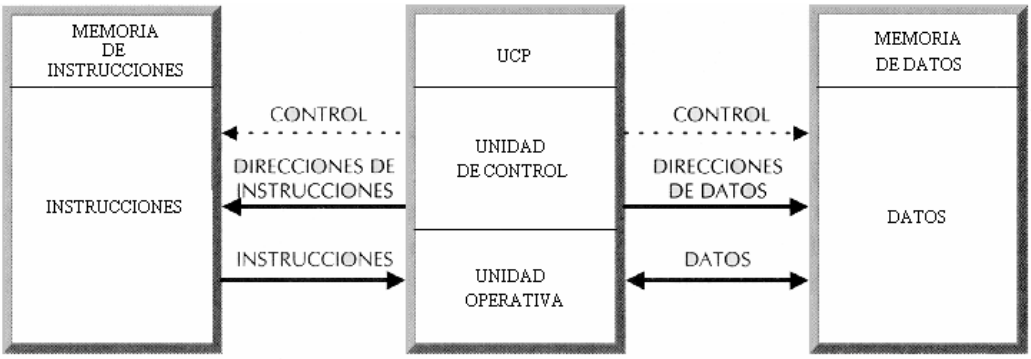


Figura 4.1. La arquitectura Harvard dispone de dos memorias independientes para datos y para instrucciones, permitiendo accesos simultáneos.

El procesador o UCP es el elemento más importante del microcontrolador y determina sus principales características, tanto a nivel hardware como software.

Se encarga de direccionar la memoria de instrucciones, recibir el código de la instrucción en curso, su decodificación y la ejecución de la operación que implica la instrucción, así como la búsqueda de los operandos y el almacenamiento del resultado.

Existen tres orientaciones en cuanto a la arquitectura y funcionalidad de los procesadores actuales.

CISC: Un gran número de procesadores usados en los microcontroladores están basados en la filosofía CISC (Computadores de Juego de Instrucciones Complejo). Disponen de más de 80 instrucciones máquina en su repertorio, algunas de las cuales son muy sofisticadas y potentes, requiriendo muchos ciclos para su ejecución.

Una ventaja de los procesadores CISC es que ofrecen al programador instrucciones complejas que actúan como macros o funciones.

RISC: Tanto la industria de los computadores comerciales como la de los microcontroladores están dirigiéndose hacia la filosofía RISC (Computadores de Juego de Instrucciones Reducido). En estos procesadores el repertorio de instrucciones máquina es muy reducido y las instrucciones son simples y, generalmente, se ejecutan en un ciclo.

La sencillez y rapidez de las instrucciones permiten optimizar el hardware y el software del procesador.

SISC: En los microcontroladores destinados a aplicaciones muy concretas, el juego de instrucciones, además de ser reducido, es "específico", o sea, las instrucciones se adaptan a las necesidades de la aplicación prevista. Esta filosofía se ha bautizado con el nombre de SISC (Computadores de Juego de Instrucciones Específico).

Memoria

En los microcontroladores la memoria de instrucciones y datos está integrada en el propio chip. Una parte debe ser no volátil, tipo ROM, y se destina a contener el programa de instrucciones que gobierna la aplicación. Otra parte de memoria será tipo RAM, volátil, y se destina a guardar las variables y los datos.

Hay dos peculiaridades que diferencian a los microcontroladores de las computadoras personales:

No existen sistemas de almacenamiento masivo como disco duro o disquetes.

Como el microcontrolador sólo se destina a una tarea en la memoria ROM, sólo hay que almacenar un único programa de trabajo.

La RAM en estos dispositivos es de poca capacidad pues sólo debe contener las variables y los cambios de información que se produzcan en el transcurso del programa. Por otra parte, como sólo existe un programa activo, no se requiere guardar una copia del mismo en la RAM pues se ejecuta directamente desde la ROM.

Los usuarios de computadoras personales están habituados a manejar Megabytes de memoria, pero los diseñadores con microcontroladores trabajan con capacidades de ROM comprendidas entre 512 bytes y 8 k bytes y de RAM comprendidas entre 20 y 512 bytes.

Según el tipo de memoria ROM que dispongan los microcontroladores, la aplicación y utilización de los mismos es diferente. Se describen las cinco versiones de memoria no volátil que se pueden encontrar en los microcontroladores del mercado.

1º. ROM con máscara

Es una memoria no volátil de sólo lectura cuyo contenido se graba durante la fabricación del chip. El elevado costo del diseño de la máscara sólo hace aconsejable el empleo de los microcontroladores con este tipo de memoria cuando se precisan cantidades superiores a varios miles de unidades.

2ª. OTP

El microcontrolador contiene una memoria no volátil de sólo lectura "programable una sola vez" por el usuario. OTP (One Time Programmable). Es el usuario quien puede escribir el programa en el chip mediante un sencillo grabador controlado por un programa desde una PC.

La versión OTP es recomendable cuando es muy corto el ciclo de diseño del producto, o bien, en la construcción de prototipos y series muy pequeñas.

Tanto en este tipo de memoria como en la EPROM, se suele usar la encriptación mediante fusibles para proteger el código contenido.

3ª EPROM

Los microcontroladores que disponen de memoria EPROM (Erasable Programmable Read OnIy Memory) pueden borrarse y grabarse muchas veces. La grabación se realiza, como en el caso de los OTP, con un grabador gobernado desde una PC. Si, posteriormente, se desea borrar el contenido, disponen de una ventana de cristal en su superficie por la que se somete a la EPROM a rayos ultravioleta durante varios minutos. Las cápsulas son de material cerámico y son más caros que los microcontroladores con memoria OTP que están hechos con material plástico.

4ª EEPROM

Se trata de memorias de sólo lectura, programables y borrables eléctricamente EEPROM (Electrical Erasable Programmable Read OnIy Memory). Tanto la programación como el borrado, se realizan eléctricamente desde el propio grabador y bajo el control programado de una PC. Es muy cómoda y rápida la operación de grabado y la de borrado. No disponen de ventana de cristal en la superficie.

Los microcontroladores dotados de memoria EEPROM una vez instalados en el circuito, pueden grabarse y borrarse cuantas veces se quiera sin ser retirados de dicho circuito. Para ello se usan "grabadores en circuito" que confieren una gran flexibilidad y rapidez a la hora de realizar modificaciones en el programa de trabajo.

El número de veces que puede grabarse y borrarse una memoria EEPROM es finito, por lo que no es recomendable una reprogramación continúa. Este tipo de memoria es relativamente lenta.

5ª FLASH

Se trata de una memoria no volátil, de bajo consumo, que se puede escribir y borrar. Funciona como una ROM y una RAM pero consume menos y es más pequeña.

A diferencia de la ROM, la memoria FLASH es programable en el circuito. Es más rápida y de mayor densidad que la EEPROM.

La alternativa FLASH está recomendada frente a la EEPROM cuando se precisa gran cantidad de memoria de programa no volátil. Es más veloz y tolera más ciclos de escritura/borrado.

Las memorias EEPROM y FLASH son muy útiles al permitir que los microcontroladores que las incorporan puedan ser reprogramados "en circuito", es decir, sin tener que sacar el circuito integrado de la tarjeta. Así, un dispositivo con este tipo de memoria incorporado al control del motor de un automóvil permite que pueda modificarse el programa durante la rutina de mantenimiento periódico, compensando los desgastes y otros factores tales como la compresión, la instalación de nuevas piezas, etc. La reprogramación del microcontrolador puede convertirse en una labor rutinaria dentro de la puesta a punto.

Puertos de Entrada y Salida

La principal utilidad de los pines que posee la cápsula que contiene un microcontrolador es soportar las líneas de E/S que comunican a la UCP interna con los periféricos exteriores.

Según los controladores de periféricos que posea cada modelo de microcontrolador, las líneas de E/S se destinan a proporcionar el soporte a las señales de entrada, salida y control.

Reloj principal

Todos los microcontroladores disponen de un circuito oscilador que genera una onda cuadrada de alta frecuencia, que configura los impulsos de reloj usados en la sincronización de todas las operaciones del sistema.

Generalmente, el circuito de reloj está incorporado en el microcontrolador y sólo se necesitan unos pocos componentes exteriores para seleccionar y estabilizar la frecuencia de trabajo. Dichos componentes suelen consistir en un cristal de cuarzo junto a elementos pasivos o bien un resonador cerámico o una red R-C.

Aumentar la frecuencia de reloj supone disminuir el tiempo en que se ejecutan las instrucciones pero lleva aparejado un incremento del consumo de energía.

4.3. Arquitectura del Microcontrolador AT 89C51

Intel diseñó en 1982 el miembro mas importante de la familia de los microcontroladores MCS-51 para aplicaciones industriales, el 80C51. Con el tiempo la patente fue liberada de la empresa Intel, y otras compañías como Texas Instrument, Motorola o Atmel, comenzaron a fabricar sus propias versiones, lo que provocó que se fueran añadiendo más miembros a la familia de los microcontroladores 80C51, como los µc 80C31, µc 80C57 o el µc 89C51 de Atmel. La utilización del µc 89C51 de Atmel, para este proyecto, no es una mera casualidad, ya que este microcontrolador cuenta con muchos recursos útiles para la implementación de un sistema de adquisición de datos, destacando de estos su memoria flash, esto permite usar al microcontrolador en más de una ocasión, borrar todo programa contenido en él y volverlo a programar hasta 10 000 veces, por lo que se facilitan las pruebas que se pudieran requerir.

El µC-89C51 está basado en los microprocesadores de 8 bits, contiene internamente un CPU de 8bits, 3 puertos de entrada y salida paralelos, un puerto de control, el cual a su vez contiene; un puerto serie, dos entradas para Timer/Contador de 16 bits, dos entradas para interrupciones externas, las señales de RD y WR para la toma o almacenamiento de datos externos en RAM, la señal de PSEN para la lectura de instrucciones almacenadas en EPROM externa. Gracias a estas tres señales el µC-89C51 puede direccionar 64 K de programa y 64K de datos separadamente, es decir un total de 128Kbytes. Además cuenta con 128 bytes de memoria RAM interna. El µC-89C51 puede generar la frecuencia (Baud Rate) de Transmisión/Recepción de datos por el puerto serie de manera automática partiendo de la frecuencia del oscilador general, por medio de la programación del Timer 1. Dicha frecuencia de transmisión puede ser cambiada en cualquier momento con solo cambiar el valor almacenado en el control o también se puede duplicar o dividir la frecuencia con solo escribir sobre el bit 7 (SMOD) del registro de control (PCON).

Algunas características del microcontrolador At 89C51 son las siguientes:

- Microcontrolador CISC de 8 bits
- Arquitectura tipo Harvard modificada
- 4K bytes de memoria interna de programa tipo Flash
- 128 bytes de memoria interna de datos SRAM
- 2 Temporizadores / contadores de 16 bits
- 1 Interfase serial asíncrona (UART)
- Frecuencia de operación desde unos pocos hertz hasta versiones que pueden correr a 24MHz
- Ejecución de instrucciones en 12 o 24 ciclos del oscilador

Una de las características más importantes de los microcontroladores de 8 bits, es que se facilita el manejo de muchos periféricos. En el proyecto del diseño de un Sistema de Adquisición de Datos, al utilizar el µc 89C51, se posibilita el control de numerosos periféricos externos; memorias, convertidores analógico/digital ó digital/analógico, display LCD, etc. En la Figura 4.2 se muestra de forma gráfica la estructura básica del microcontrolador AT89C51, referente a su arquitectura interna, junto con los 4 puertos bidireccionales que posee este circuito integrado

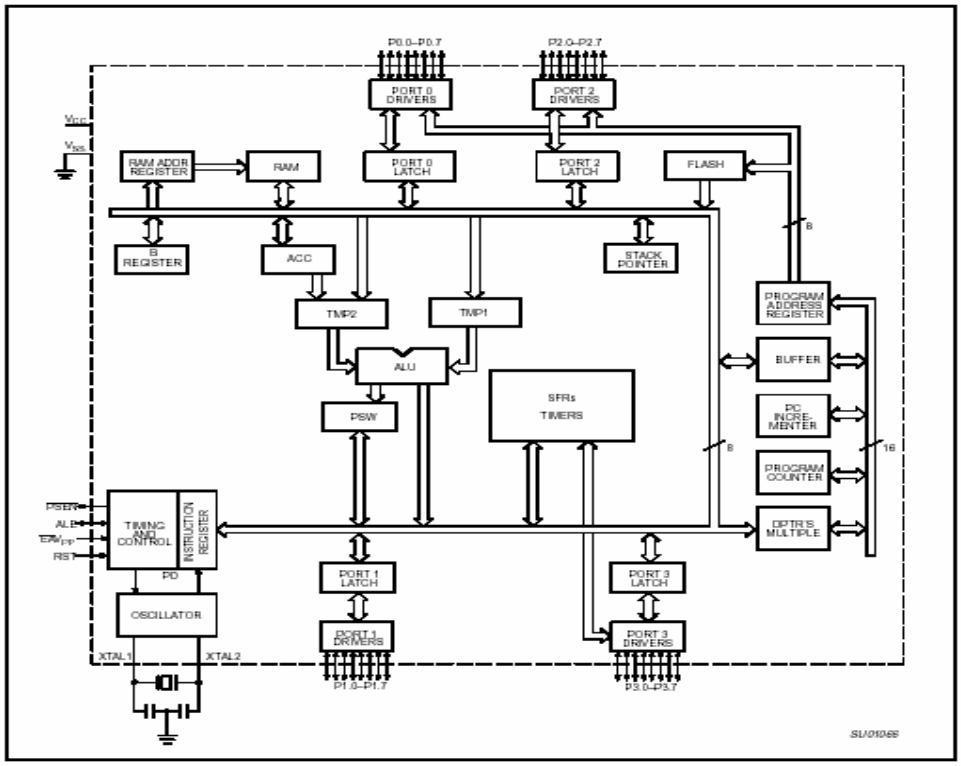


Figura 4.2. Arquitectura interna del µC 89C51

El encapsulado del 89C51 posee 40 pines como se indica en la figura 4.3 a), para este microcontrolador en particular, cada pin debe soportar numerosas funciones. Este hecho obliga a que cada Terminal posea dos funciones diferentes multiplexadas en el tiempo. Las funciones generales y las señales de dicho microcontrolador son:

- 32 líneas bidireccionales (4 puertos de E/S)
- Un canal serie Full duplex.
- Capacidad de control de 64 kB de memoria de código.
- Capacidad de control de 64 KB de memoria de datos.
- 2 contadores
- 2 líneas de interrupción.

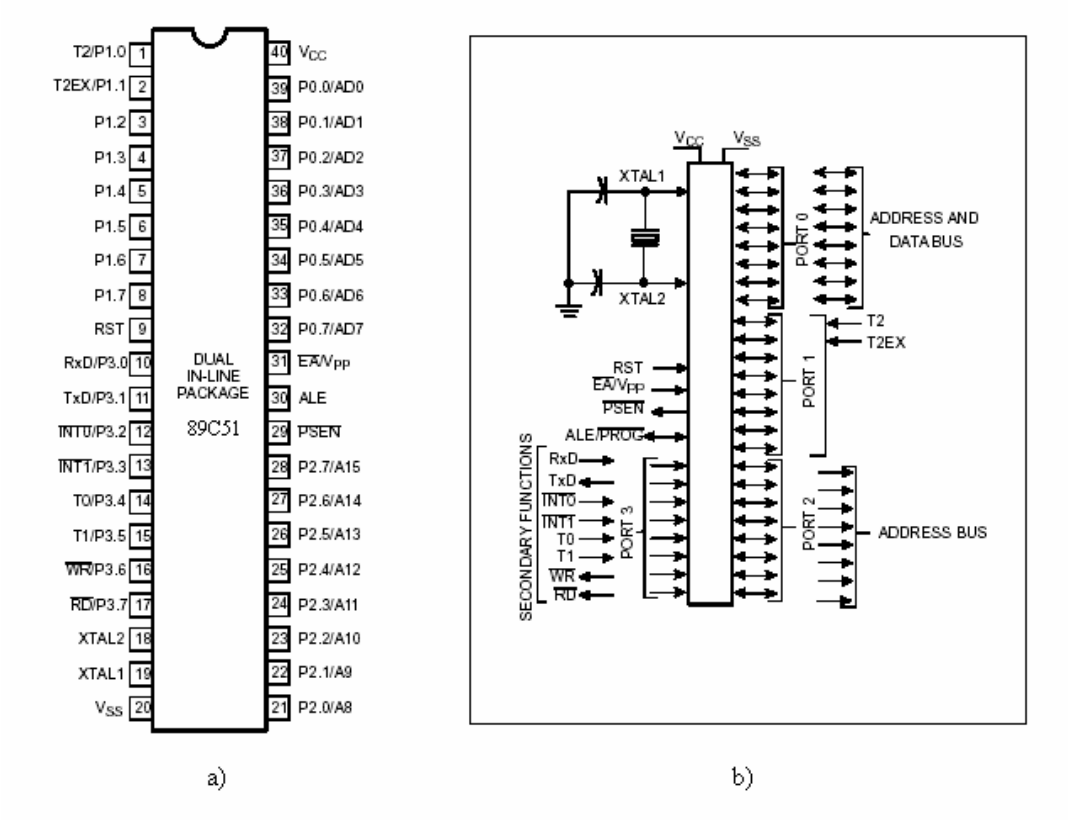


Figura 4.3 a) Configuración de pines para µC 89C51. b) Puertos del µC 89C51

La figura 4.3 a). Muestra la configuración de los pines, en total 40, para el microcontrolador AT89C51. En la figura 4.2 b. se esboza las aplicaciones específicas de los puertos, para transmitir o recibir datos (puerto 3) o para direccionar la información externa (puerto 0).

Descripción de los espacios de memoria para el µc AT 89C51.

El espacio de memoria gobernado por el µC 89c51 se divide en tres zonas fundamentales (figura 4.4):

La primera, llamada memoria de programa, en donde se encuentran todas las instrucciones que van a ser ejecutadas por el µC-89C51, es decir, el programa de trabajo. De 2 a 4 Kbyte de memoria se hallan implementados en el propio chip como ROM programable en fábrica. Cuando se requiere trabajar con una mayor memoria de programa, la memoria de programa (externa) es seleccionada mediante la activación de la señal PSEN (estado bajo) y se conectan exteriormente al puerto 60 Kbytes de

memoria como máximo, ya que para este microcontrolador, el máximo espacio de memoria de programa que se puede acceder es de 64 Kbytes.

El segundo espacio de memoria denominado, memoria de datos externo, es accesado mediante la activación de las señales RD y WR, durante la lectura o escritura de datos respectivamente. En este espacio del µC 89C51 toma todos valores que se encuentran en memoria como DATOS, es decir, el microcontrolador no puede ejecutar ninguna instrucción que se encuentre aquí almacenada. El 89C51 puede direccionar también 64 Kbytes de memoria de datos.

El tercer espacio de memoria es denominado como memoria RAM interna, el cual se subdivide en 128 bytes de memoria bajos y en 128 bytes de memoria altos. En los primeros 128, se encuentran 4 bancos de 7 registros cada uno. Estos registros son de gran ayuda para la simplificación de los programas, debido a que cada uno de ellos nos permite almacenar datos momentáneamente y realizar un vasto número de instrucciones del µC 8951. También dentro de este espacio, se encuentran 16 bytes (del 20_H al $2F_H$) que pueden ser direccionados directamente por bit. En la parte alta de la memoria RAM interna, se encuentran el contenido de los Registros de Funciones Especiales, formado por Puertos, Registros de Control, Acumuladores, Registros de interrupción, etc.

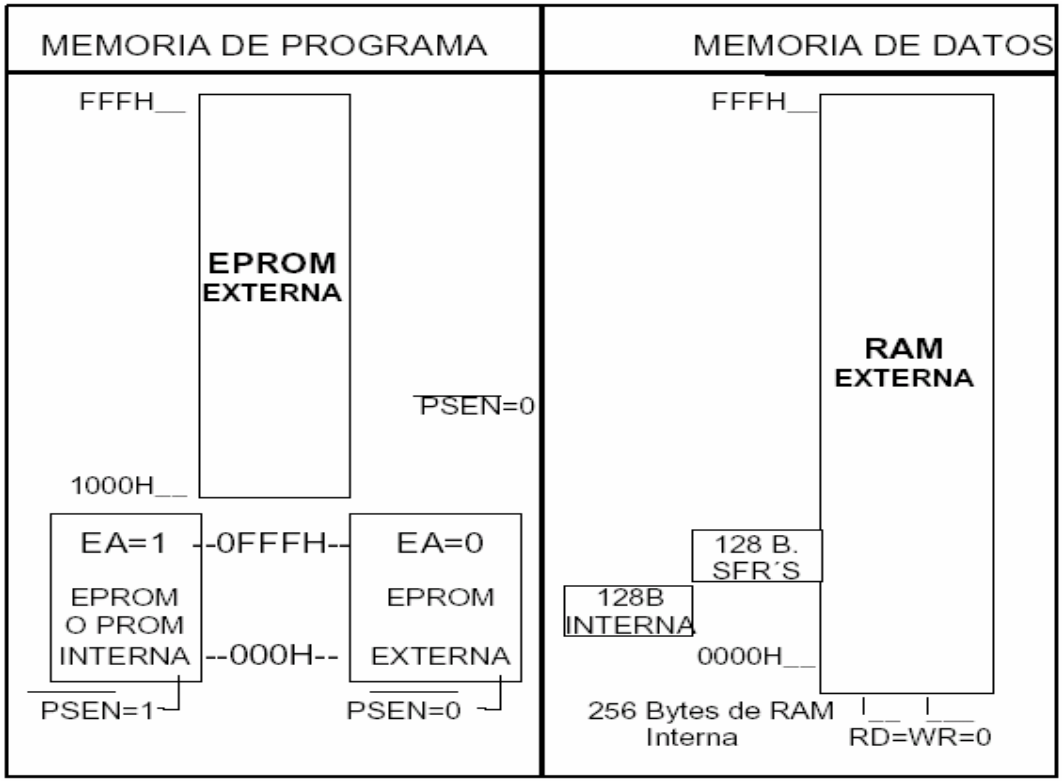


Figura 4.4 En la figura se observan, de forma esquemática, los espacios de memoria del µC 89C51. La memoria interna de datos: posee 256 bytes. En la Memoria interna de programa 4 KBytes inferiores son los que se hallan implementados en el propio chip como ROM programable en fábrica. Los 60 KB restantes hay que conectarles externamente. El µC 89C51 puede direccional hasta 64 KB de memoria externa de RAM.

Localidades asignadas a las interrupciones.

La tabla 4.1, muestra las localidades que han sido asignadas por el fabricante, para dar servicio a las rutinas de interrupción.

Tabla 4.1 Localidades de las rutinas de interrupción.

Fuente de interrupción	Vector de direcciones
IE0 (Interrupción 0 externa)	0003H
TF0 (Interrupción del timer 0)	000BH
IE1 (Interrupción 1 externa)	0013H
TF1 (Interrupción del timer 1)	001BH
R1 y T1 (Interrupción serial)	0023H
TF2 y EXF2 (Sólo para el 8052)	002BH

Una interrupción puede ser causada de manera externa o interna, es decir puede ser producida por un dispositivo periférico o por programación respectivamente. La interrupción con mayor orden es el RESET el cual no puede ser mascarable. Cuando el RESET ocurre el programa comienza a partir de la dirección 0000H del programa.

Cuando es provocada una interrupción, el Contador del Programa (PC) almacena su contenido temporalmente dentro del SP (apuntador de apilamiento o puntero de pila) y se carga con la dirección de la localidad donde se encuentra la rutina de servicio de la interrupción correspondiente. Una vez posicionado en esa localidad deberá de comenzar la ejecución de la rutina de servicio, hasta que encuentre la instrucción RETI, que le permitirá al PC recuperar nuevamente su valor original almacenado en el SP, y continuar con el programa anterior a la interrupción.

Por ejemplo a la interrupción 0, se le asigna la localidad 0003H, si la interrupción no se utiliza, esta localidad puede utilizarse para propósitos generales del programa, si la interrupción ha sido permitida, (estableciendo el bit correspondiente dentro del registro de control IE), en el momento que exista una activación de la interrupción (estado bajo en la línea INTO) el PC se cargará con 0003 y saltará a esa localidad para comenzar a ejecutar la rutina de servicio.

Estas localidades de memoria de los servicios de interrupción están separadas en intervalos de 8 bytes, entre sí. Cuando un servicio de interrupción es corto, éste puede estar contenido en los 8 bytes. En el caso de que fuese largo se puede ejecutar un salto a otra localidad de memoria para continuar con la secuencia de interrupción. El término del servicio de interrupción deberá de realizarse mediante la ejecución de la instrucción de la instrucción RETI.

Memoria de programa interna y externa.

Cuando se utilizan elementos de la familia del µC-51, con memoria interna ROM, esta puede ser accesada mediante la conexión de la línea EA =1 (Vcc) (External Access Enable). Si la memoria interna es de 4 Kbytes y EA = 1, el CPU seleccionará internamente el ROM, desde 0000H hasta 0FFFH y de manera externa automáticamente a partir de 1000H hasta FFFFH.

Por el contrario, si la línea EA = 0, el CPU seleccionará de forma externa el ROM, desde la dirección 0000H hasta FFFFH. En el caso del 8031 ésta línea se conecta siempre a 0 Volts (Vss), no así para el 89C51.

La línea PSEN (Program Store Enable), que sirve para leer el ROM externo, es activada en todas las búsquedas (**Fetche** lecturas de código de instrucción) del programa. PSEN NO SE ACTIVA en búsquedas (fetches) del ROM interno. La fig. 4.5 muestra la conexión a una EPROM externa, la cual es muy similar a la conexión de una memoria RAM externa.

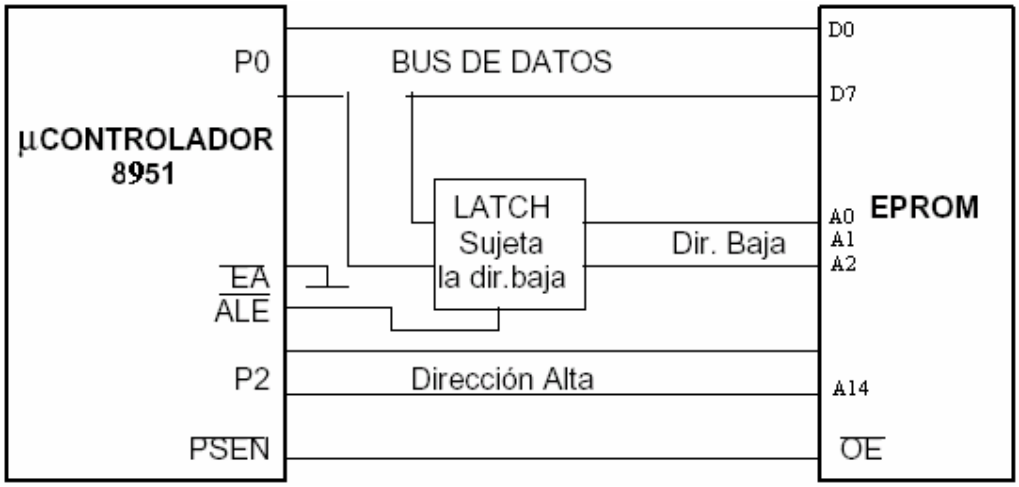


Figura 4.5. Conexión a una memoria de programa externa.

Memoria de datos (DATA MEMORY)

El espacio de memoria RAM interno está dividido en dos espacios, el primer bloque es referido como la parte baja de 128 bytes, el segundo es la parte alta de 128 bytes llamado espacio SFR (Registros de Funciones Especiales). Los primeros 128 bytes, son presentados en todos los dispositivos de la familia MCS-51, como se presenta en la Fig. 4.6.

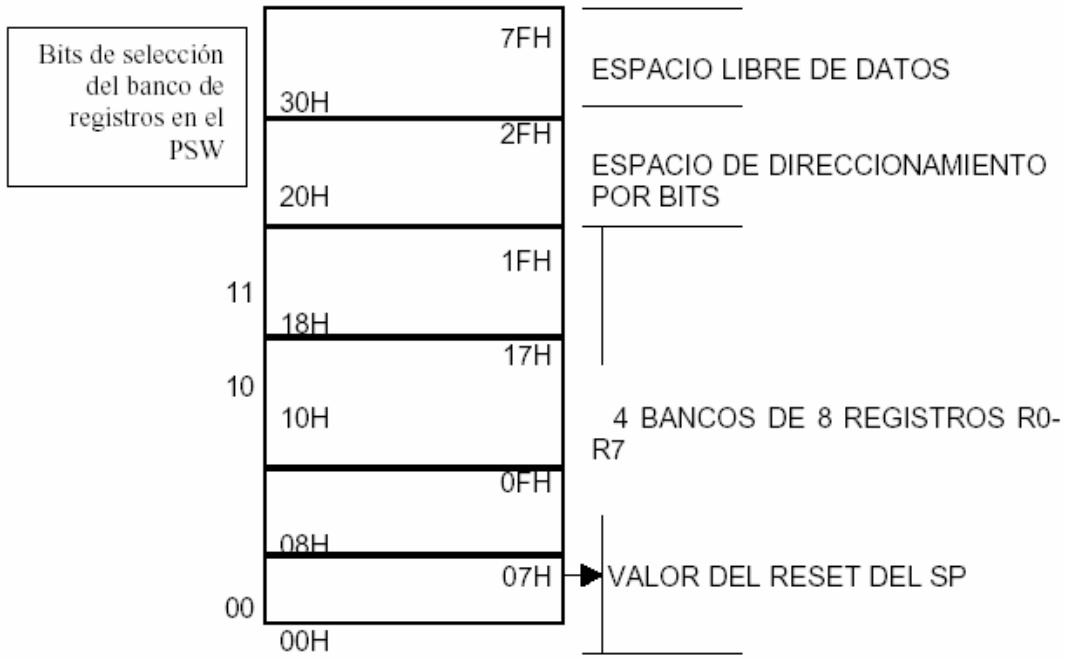


Fig. 4.6. Distribución de los 128 Bytes más bajos de la memoria RAM interna.

Como se puede observar en la figura anterior, los 128 bytes más bajos son divididos en 4 bloques de 8 registros cada uno, que contienen los valores de los registros R0 A R7, los bloques pueden ser seleccionados mediante la escritura en los bits 3 y 4 del registro PSW (palabra del estado del programa). La utilización de registros permite un uso más eficiente del espacio de códigos debido a que sus direccionamientos son de 8 bits únicamente.

Todas las versiones de los microcontroladores µC-51, incluyendo el AT 89C51, contienen un espacio de 128 bytes en la parte alta de la memoria que son direccionados directamente, en este espacio se localizan los Registros de Funciones Especiales (SFR). Estos registros especiales, tienen sus localidades bien establecidas, y son utilizados por el microcontrolador para realizar las distintas operaciones internas que ejecuta el microcontrolador, así como también para el control y acceso de los diferentes puertos de entrada y salida.

Memoria RAM interna.

El microcontrolador 89C51 posee 256 bytes de RAM interna para almacenaje de datos, esta es la memoria estática más rápida y que tiene los más variados modos de direccionamiento.

A la Memoria interna de datos se accede en base al tipo de direccionamiento empleado. El direccionamiento puede ser:

- Direccionamiento directo e indirecto: se puede emplear para la zona de RAM interna comprendida entre 00h y 7Fh, que suele ser donde se coloca también la pila.
- Direccionamiento sólo directo: se emplea en la zona de registros de función especial (SFR) situada entre la dirección 80h y FFh de la memoria interna
- Direccionamiento sólo indirecto: Cuando escribimos en la zona de SFR usando este direccionamiento, en realidad escribimos en una zona de RAM interna a la que sólo se accede mediante direccionamiento indirecto.
- Direccionamiento a nivel de bit: esta área tiene una longitud de 16 bytes (segmento 20h a 2Fh). Cada uno de los 1287 bits de este segmento se puede direccionar directamente (00h a 7Fh). Los bits pueden referirse de dos formas diferentes, bien por sus direcciones (bits 00h a bits 7Fh) , o por los bytes que los contienen (20h a 2Fh) seguidos por el número de bit: 20.0 a 20.7 serian los 8 bits de la posición 20h

Los Registros de Función Especial (*SFR Special function register*) son direcciones de memoria interna que controlan los periféricos disponibles en el Chip, como los puertos, temporizadores e interrupciones, así como otras características del procesador. Como ejemplo de registros de funciones especiales, podemos citar al acumulador (ACC, en dirección E0h), el registro B (en 0Fh) y el puntero de datos (DPH en 83h y DPL en 82h). La siguiente tabla (tabla 4.2) muestra los SFR del 89C51 estándar. Estos SFR figuran principalmente en las instrucciones.

Tabla 4.2 Referencia a los registros de funciones especiales del µC 89C51

Símbolo	Descripción	Direccionamiento Directo
P0	Puerto 0	80h
SP	Puntero de Pila	81h
DPTR	Puntero de datos (2 bytes)	
DPL	Parte Baja Puntero de Datos	82h
DPH	Parte alta Puntero de Datos	83h
PCON	Control de alimentación	87h
TCON	Control de temporizador	88h
TMOD	Modo del Temporizador	89h
TL0	Temporizador bajo 0	8Ah
TL1	Temporizador bajo 1	8Bh
TH0	Temporizador alto 0	8Ch
TH1	Temporizador alto 1	8Dh
P1	Puerto 1	90h

4.4. Memoria fuera del µC 89C51

Con al excepción de unos pocos miembros, los mas pequeños, toda la familia de los microcontroladores µC-51, pueden acceder a un espacio de memoria externa al chip, ya sea memoria RAM de datos o de programa ROM (ver figura 4.5). Para poder hacer esto los 8 bits más significativos de direcciones aparecen en el puerto 2. Las direcciones más bajas y los datos aparecen sobre el puerto 0. Para ahorrar pines o patillas el puerto cero primero se emplea como los 8 bits menos significativos del bus de direcciones y luego

como bus de datos. Esto se denomina como sabemos multiplexado. Empleamos menos pines pero se reduce la velocidad de los accesos a memoria y necesitamos un latch externo de direcciones. El acceso externo a memoria en el microcontrolador AT89C51 siempre es más lento por el multiplexado del bus de datos y direcciones menos significativos

Entrada /Salida

Si no tenemos la manera de intercambiar información con el mundo exterior la computadora seria inservible. Mientras que con los PC (Computadoras personales) hay formas estandarizadas de realizar las conexiones de E/S (Hiperterminal, teclado, etc.), los microcontroladores destacan proporcionando entradas y salidas mucho más adaptables. El termino puerto se usa para referirse a un bloque de entrada/salida. Hay dos tipos comunes de puertos, serie y paralelo.

Puertos paralelo. Los puertos paralelo son grupos de, normalmente, ocho bits sobre patillas individuales. Un puerto paralelo latchea o retiene su valor hasta que volvemos a mandar otro valor. Podemos realizar puertos de salida externos con latches de ocho bits (como el CI 74374). Un puerto paralelo de entrada pasa los estados de las patillas al microprocesador en el instante que se ejecuta la instrucción de copia para esa dirección. Podemos realizar puertos de entrada mediante buffers triestado octales (como el CI 74244).

La conexión del 89C51 al mundo exterior se realizar normalmente a través de puertos. Siempre tenemos una serie de patillas disponible directamente en el chip y podemos tener también puertos adicionales añadidos sobre el bus de expansión. Los puertos internos del 89C51 son inusuales porque sólo son parcialmente bidireccionales. Como se muestra en el esquema para una patilla o pin en la Figura 4.7.

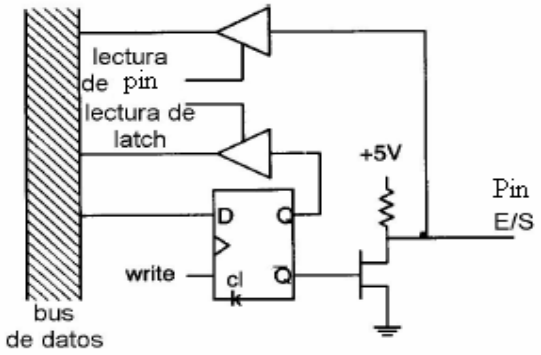


Figura 4.7 Patilla o pin del puerto interno básico

Sobre la salida podemos colocar un cero en cualquier momento pero, para realizar una entrada, la circuitería interna de salida debe estar desconectada. En otro caso leeríamos nuestra propia salida en lugar de la señal proveniente del exterior. El aspecto software clave es la necesidad de que en cualquier puerto que usemos como entrada debe tener 1 escrito en el registro correspondiente a él. Algunas de las patillas de los puertos pueden tener otras funciones especiales, en caso de que las usemos de esta forma, no debemos usar estas patillas como puertos de entrada o salida. Este es el caso del puerto 3, donde

tenemos las opciones de escritura y lectura en memoria externa, así como la transmisión y recepción de datos.

No hay forma de desactivar o fijar la dirección del puerto. Recordemos que los pines del puerto pueden absorber una corriente máxima de 1.6 mA pero sólo ofrecer decenas de microamperios. También, P0 cuando se usa como puerto de entrada/salida (opuesto al bus de datos para expansión de memoria externa) no dispone de resistencias de pull up, el dispositivo externo tendrá que proporcionar el nivel lógico alto.

Puertos Serie. Los puertos serie transfieren bits simples de datos uno a otro, tomando al menos ocho transferencias para intercambiar un byte. La forma más común de puerto serie emplea solo un pin para cada dirección de la transferencia. Normalmente el hardware interno maneja los detalles de temporización e incluye los registros de desplazamiento paralelo/serie. En conjunto, el hardware para esta función recibe el nombre de transmisor/receptor asíncrono universal (UART). Este elemento descarga al procesador del trabajo de gestionar la temporización y organizar los bits de forma que el procesador pueda hacer otras tareas. Es posible mandar información serie directamente sobre otras patillas de puertos ordinarios usando software y temporizadores, particularmente cuando la tasa de transmisión es baja, pero resulta mejor usar el hardware de la UART cuando sea posible.

Secuencia de ejecución de instrucciones en el µAT89C51.

La lectura de un código de instrucción requiere primero mandar la dirección de la instrucción (desde el contador del programa) habilitando un buffer triestado en la CPU para enviar el PC al bus de direcciones. En este momento, la parte baja del bus de direcciones permanece sólo hasta que la señal ALE desaparece puesto que estos mismos pines llevan la información del byte del código, cuando la memoria lo haya enviado. Cabe recordar que en el µC AT89C51 tanto el bus de datos bajo y la parte menos significativa del bus de direcciones se encuentran multiplexado. Durante el tiempo desde que la dirección esta en el bus, el dispositivo de memoria externa que contiene la instrucción puede decodificar la dirección y reconocer que se le esta pidiendo una respuesta.

Para el ciclo de lectura de un código de instrucción, tras un retardo controlado por el reloj el procesador coloca una señal, habilitación del almacenamiento del programa (PSEN). Esto provoca que el dispositivo de almacenamiento de código habilite sus salidas triestado y dirija el bus de datos con el contenido de la dirección particular de la instrucción que se esta pidiendo. Al final de la señal PSEN, una vez que el byte de código esta en el bus, la CPU almacena el código desde el bus de datos.

A continuación, la CPU decodifica la instrucción de forma que se lleve a cabo la secuencia correcta de operaciones para ejecutar la instrucción, copiar un byte de datos a una dirección especifica, comparar dos valores, o incluso cambiar la dirección de la próxima búsqueda de instrucción (un salto).

Tomemos una instrucción concreta, por ejemplo, un comando para copiar el contenido de la dirección de memoria 2045h en el registro puntero de datos (MOV DPTR,#2045h). Cuando el primer byte de la instrucción haya sido decodificado, implicara que son necesarios dos bytes mas de instrucción, los dos bytes que van al

registro. El próximo paso para ejecutar esta instrucción es incrementar el contador de programa en uno y buscar el segundo byte de instrucción. Finalmente, se recoge el segundo byte de instrucción, los bytes entrantes en este caso van directamente a DPTR. Una mirada a la tabla de instrucciones nos dice que este proceso emplea 24 ciclos de reloj para la instrucción MOV DPTR.

Consideremos una instrucción que transfiere datos a dispositivos externos. En este caso, las líneas de control RD o WR entran en juego. Tomemos la instrucción copiar a dato externo (MOVX). Entonces se necesita que el operando se encuentre ya en el acumulador (quizás con MOV A,#dato) y la dirección de memoria externa que se encuentre en el puntero de datos (DPTR). Los pasos para ejecutar la instrucción (MOVX @DPTR,A) involucran buscar el byte de instrucción, decodificarlo, poner el valor DPTR en el bus de direcciones, poner el valor del acumulador en el bus de datos, y luego activar la señal WR (escritura en memoria). Esta es la finalidad de la CPU Todo lo que hacen los dispositivos de memoria es reconocer la dirección con su circuitería de decodificación y guardar los datos cuando la señal WR termine.

Señales de control del bus.

Las señales dentro del AT89C51 no están descritas en el libro del fabricante y no nos interesan cuando usemos el dispositivo en la práctica. Más importantes son las señales externas para expandir la memoria fuera del chip, añadir puertos o realizar interfaces con otros dispositivos, que es justamente lo que necesitamos conocer para implantar el sistema de adquisición de datos de este proyecto. Todas estas señales pertenecen a la expansión fuera del chip.

ALE

Esta señal, habilitación del latch de direcciones (ALE, address latch enable), se usa para activar un latch externo que capture la parte baja de direcciones colocado sobre P0 antes de que el dato sea transferido desde o hacia la memoria externa.

PSEN

Habilitación del almacenamiento del programa (/PSEN, program store enable) indica que la ROM debe colocar el código en el bus de datos. /PSEN sólo se activa si se produce acceso a memoria externa. Si colocamos la patilla /EA a masa, todos los accesos se realizan a la ROM externa. Si dejamos /EA flotante alto, los accesos se realizan en el rango dentro del chip que no provocan que la señal /PSEN pase a nivel bajo.

Temporización del ciclo de lectura de un código de instrucción (ciclo FETCH)

La forma de la figura 4.8 muestra un ciclo de lectura de código de instrucción externo (ciclo FETCH externo). Los bordes sugieren que existen tiempo de subida y de bajada asociados con las señales mientras que las señales en medio (ni 1 ni 0) para P0 muestran el bus en triestado en los intermedios entre la colocación de los datos. Si enganchamos un osciloscopio en las líneas del bus, podremos observar algo mucho más definido, especialmente durante los tiempos de triestado.

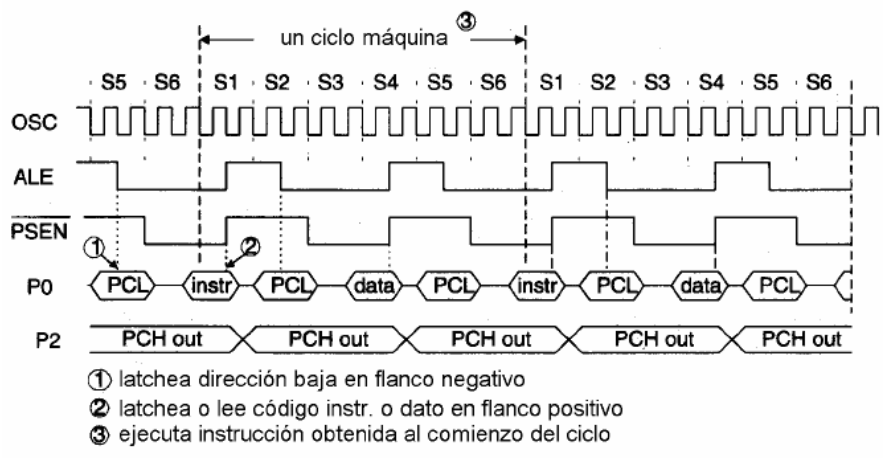


Figura 4.8 Temporización del ciclo de lectura de código FETCH.

Hay que tener en cuenta que un microprocesador concreto tiene una velocidad máxima debido a estas transiciones no instantáneas. Mientras más pongamos el chip a su límite superior de velocidad, más redondeadas aparecerán las señales y más probable será que el microprocesador no funcione.

RD Y WR

Estas señales sólo se emplean para los dispositivos de almacenamiento externo y puertos externos. El siguiente cronograma muestra la temporización de una instrucción MOVX. Si usamos la instrucción MOVX con el acumulador como destino, generaremos una señal de lectura y con el acumulador como fuente, una señal de escritura. La forma de estas señales, así como sus secuencias se muestran en la figura 4.9.

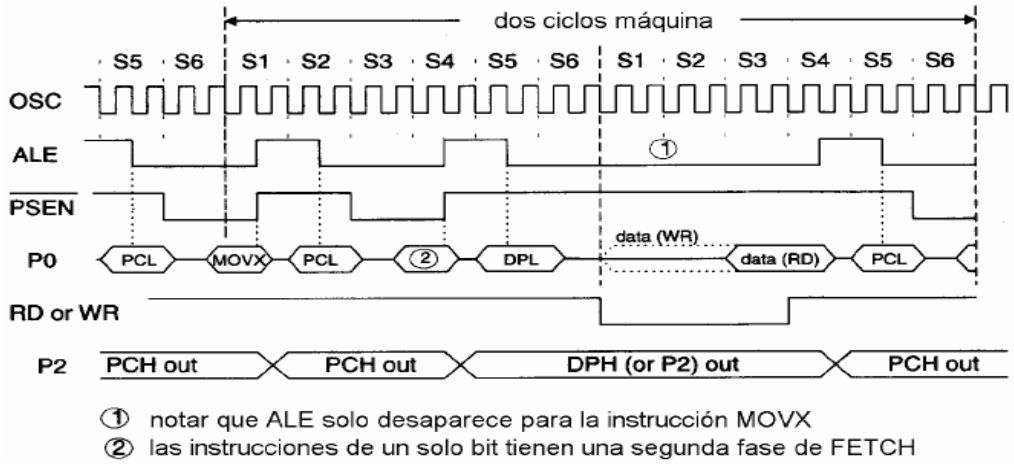


Figura 4.9 cronograma de Lectura/escritura externa

El oscilador

Todos los miembros de la familia µc51 usan un cristal externo para el oscilador. Podemos escoger, según el miembro de la familia, una frecuencia del cristal que varia desde 500 kHz a 33 o 40MHz. Habrá que hojear el manual del fabricante de cada dispositivo en concreto. En el caso del AT89 c51 se utilizará un cristal de 11.030 MHZ.

Cuando observamos como el procesador lleva a cabo realmente las instrucciones podemos constatar que el ciclo de máquina no es un periodo de reloj (cada instrucción no ocurre en 83 ns si tenemos un cristal de 12 MHz). Para un 80C51cada instrucción toma unos 12 ciclos de reloj para completarse.

Expansión de la memoria para el µC AT89C51.

La mayoría de las aplicaciones de los microcontroladores, incluyendo al µC AT89C51, hoy día usan chips adicionales para guardar el código y datos cuando se trata de sistemas como el desarrollado en este proyecto. La expansión de memoria, tanto de programa como de datos, es directa, siempre se hace de la misma manera para todos los miembros de la familia µC-51. El esquema de la figura 4.10 muestra una expansión de memoria del espacio de código que es obligada para cualquier aplicación que use el 8031, que no tiene ROM o EPROM en el chip. El mayor problema de la expansión de memoria es la perdida de al menos dos puertos de 8 bits en funciones de bus externo.

Código externo

Al expandir externamente el microcontrolador, estamos limitados por la información multiplexada direcciones/datos en los ocho bits bajos (P0). Para mantener el byte bajo de direcciones hasta que la CPU haya trasferido los datos, debemos almacenarlo en el flanco de bajada del pulso ALE. El latch 74LS373 es la elección común. El byte alto de direcciones permanece sin cambios en P2 durante el ciclo completo y podemos usarlo directamente para decodificación de direcciones.

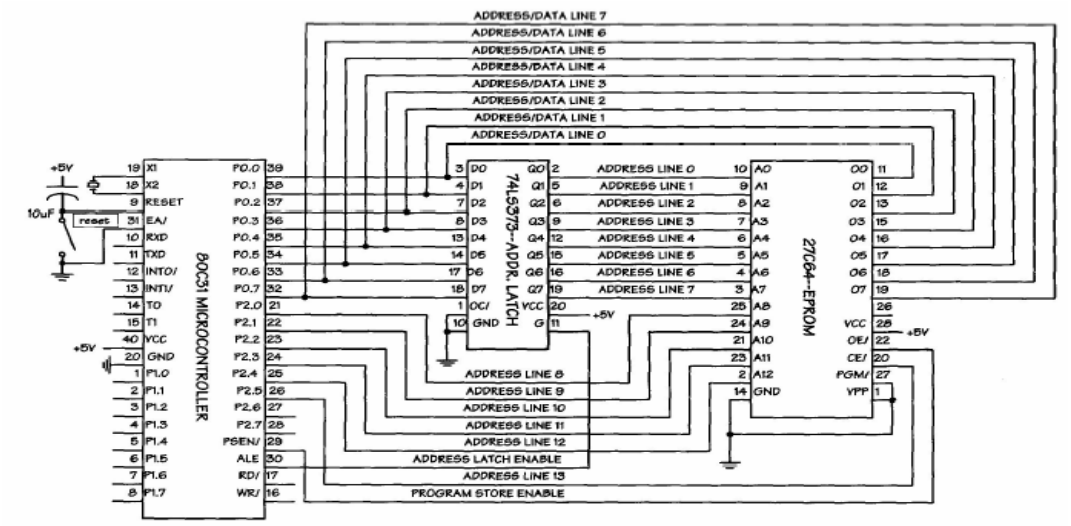


Figura 4.10 Diagrama de conexión para expandir la memoria de programa en el µC 8031

La señal PSEN proporciona la señal de lectura o habilitación de salida (OE) a la EPROM cuando la CPU dirige los datos en el bus. Con sólo una EPROM, la CPU nunca manda la señal PSEN excepto para acceder al código en este chip. El uso de A_{13} unido con la señal CS es un tipo de decodificación puesto que el rango de dirección donde A_{13} se encuentra a nivel lógico alto no direcciona la EPROM (o cualquier otra cosa). Así el rango de direcciones de código se va desde 0000h hasta 1FFFh

Código Externo, Memoria RAM Externa y Puertos Externos

Una vez que hemos perdido los dos puertos, resulta sencillo añadir una RAM externa, dando dos bits para RD y WR. La figura 4.11 muestra este tipo de sistema con RAM extra y un dispositivo controlador de puertos 8255.

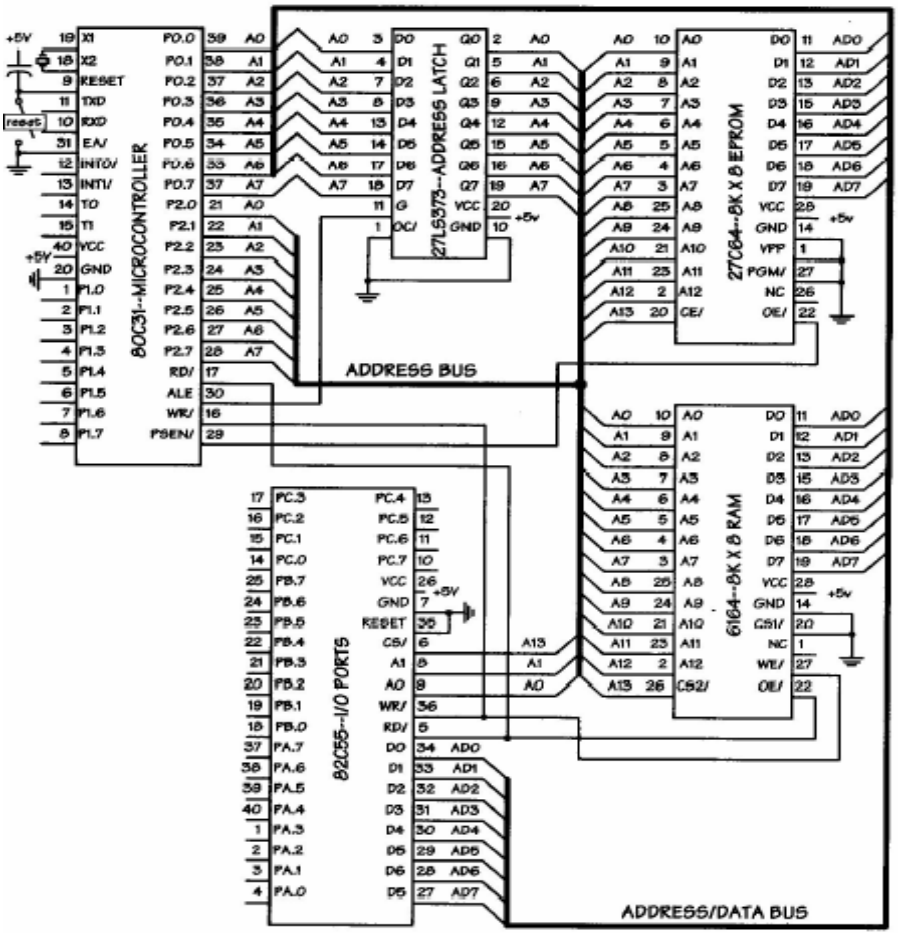


Figura 4.11 Conexión de memoria de programa EPROM y datos RAM externos a un µc 8031, utilizando un controlador de puertos 82C55.

Primero la RAM se mapea en la dirección base (0000H) y el 8255 se coloca en lo alto del espacio de direcciones. Observando el CI 6164 RAM, CS_1 se ha colocado a masa, por lo que se encuentra siempre habilitada. CS_2 se ha conectado a A_{13}. Puesto que esto es una entrada activa a nivel alto, A_{13} debe estar a nivel alto para seleccionar este chip.

En binario, esto significa que el sistema tiene RAM desde 00100000 00000000 hasta 00111111 11111111 (2000H- 3FFFH). La patilla de selección del 8255 es activa a nivel bajo (CS), por lo que A_{13} debe estar bajo para seleccionar el chip. Para cualquier dirección donde algunos bits de direcciones no están conectados (dan igual) los otros bits se consideran a nivel bajo normalmente, por lo que el CI 8255, usando solo los bits A_0 y A_1 internamente, tendrán direcciones desde (2000H a 3FFFH).

La ***expansión de la RAM*** emplea 2 bits del puerto 3 para las señales RD y WR. Los controles para la RAM externa (RD y WR) son diferentes de los controles para la EPROM (PSEN) por lo que el código y los datos externos se pueden encontrar en las mismas direcciones numéricas.

Control de un convertidor A/D.

Mediante un decodificador de direcciones latch realizado mediante un 74LS138 podemos direccionar los dispositivos de un sistema completo con RAM y ROM externas, además de un conversor A/D de 10 bits realizado mediante un AD573. Como el conversor dispone de una interfaz estándar de 8 bits es necesario transferir la información en dos partes, byte bajo y byte alto de la conversión. Esto se realiza mediante las patillas LBE (Low Byte Enable) y HBE (High Byte Enable). La conversión se inicia con la patilla CNV al escribir en un puerto determinado, realizando la lectura de este mismo puerto se leen los dos bits más significativos, al activar la señal HBE y después se lee de otra dirección el byte menos significativo activando la señal LBE. Un procedimiento parecido será utilizado en el capítulo 6, para implantar el sistema de adquisición de datos propuesto en este trabajo. La figura 4.12 muestra la circuitería y conexiones de un microcontrolador (µc 8031) que se conecta a una memoria de datos y de programa externos, un controlador de puertos 82C55, así como con un convertidor analógico/digital AD 573.

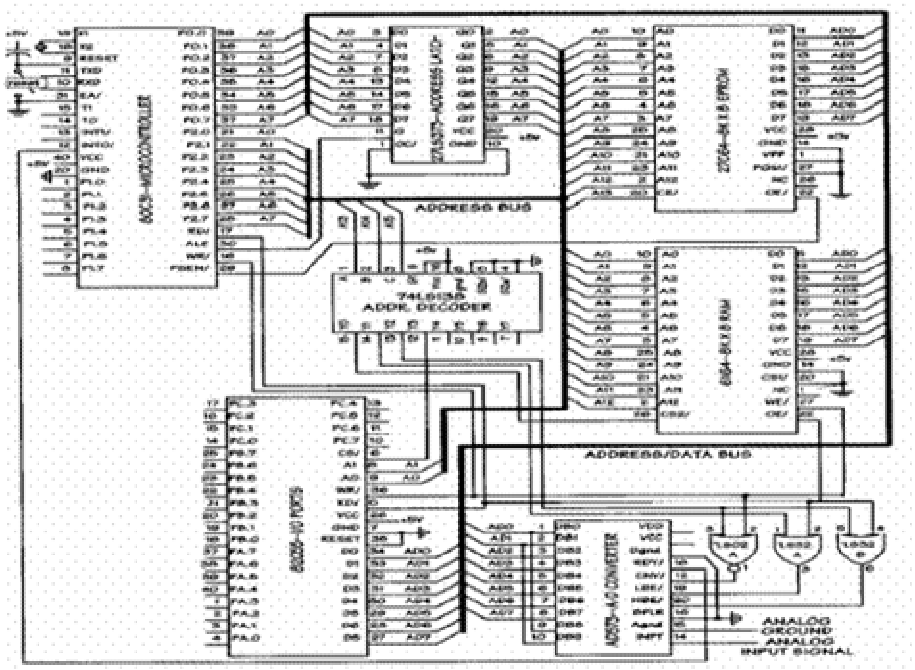


Figura 4.12 sistema de memorias RAM y ROM Externos al microcontrolador µc 80c31.

5. Diseño del circuito de medición de temperatura.

5.1 Amplificador de instrumentación.

El concepto original del AO (amplificador operacional) procede del campo de los computadores analógicos, en los que comenzaron a usarse técnicas operacionales en una época tan temprana como en los años 40. El nombre de amplificador operacional deriva del concepto de un amplificador dc (amplificador acoplado en continua) con una entrada diferencial y ganancia extremadamente alta, cuyas características de operación están determinadas por los elementos de realimentación utilizados. Cambiando los tipos y disposición de los elementos de realimentación, pueden implementarse diferentes operaciones analógicas; en gran medida, las características globales del circuito están determinadas solo por estos elementos de realimentación. De esta forma, el mismo amplificador es capaz de realizar diversas operaciones. Además con el amplificador operacional, podemos amplificar señales, atenuarlas, filtrarlas, etc.

Un amplificador operacional es un dispositivo electrónico activo capaz de ofrecer una tensión de salida en función de una tensión de entrada. Considerando de manera previa, única y exclusivamente el amplificador operacional ideal, como una aproximación muy precisa y perfectamente válida para el análisis de sistemas reales. Podemos identificar al amplificador operacional por cinco patillas. Dos de ellas representan las entradas del dispositivo; la primera de ellas llamada entrada inversora se halla indicada en los esquemas con un signo menos, la otra denominada entrada no inversora se indica mediante un signo más. Otro de las patillas del amplificador operacional corresponde a la salida del dispositivo mientras que las dos restantes corresponden a la alimentación requerida por el dispositivo (±Vcc).

Las características teóricas de un amplificador operacional, son muy similares a las reales:

- Ancho de banda infinito (podemos trabajar con señales de cualquier frecuencia).
- Tiempo de conmutación nulo
- Ganancia de tensión infinita.
- Impedancia de entrada infinita.
- Impedancia de salida nula.
- Corrientes de polarización nulas.
- Tensión de desplazamiento nula (si bien no es estrictamente cierto, diremos que la diferencia de potencial entre las entradas inversora y no inversora nula).
- Margen dinámico ±Vcc (la tensión de salida puede a nivel teórico alcanzar el valor de la tensión de alimentación, en la práctica se aproxima pero no puede ser igual ya que se producen saturaciones en el dispositivo).

Muchas veces existe en equipos de la industria, de la electromedicina, y en otras muchas aplicaciones, la necesidad de medir señales muy pequeñas del orden de micro voltios o pocos milivoltios en la presencia de comparativamente grandes señales de ruido

provenientes de distintas fuentes, como pueden ser motores, tubos de iluminación de descarga gaseosa, y la siempre presente inducción de la frecuencia de línea de alimentación, en nuestro caso 60Hz. Para realizar las mencionadas mediciones estos deberán utilizar en su entrada Amplificadores de Instrumentación con un adecuada Relación Rechazo de Modo Común (CMRR).

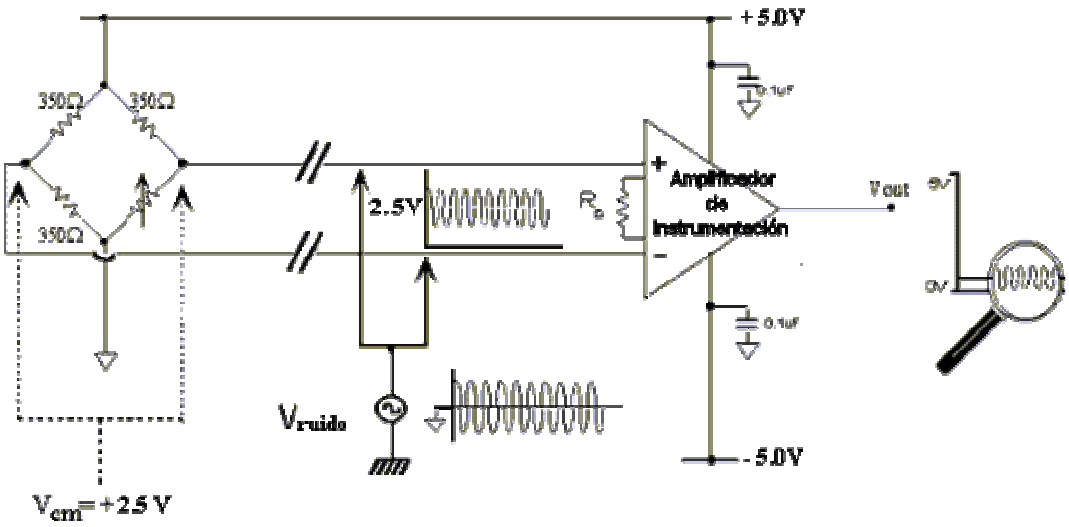


Figura 5.1 En la figura se ilustra un esquema básico de medición. Al Amplificador de Instrumentación ingresan dos señales de modo común: una de c.c. de +2.5V provenientes del puentes de resistencias y otra de c.a. o ruido, inducida sobre los cables de entrada al amplificador.

5.1.2 Rechazo de Modo Común

Los amplificadores de Instrumentación amplifican la diferencia entre dos señales. Esas señales diferenciales en la práctica provienen de sensores como termopares, foto sensores, puentes de medición resistivos, etc. En la figura anterior se ve que de un puente resistivo, en estado de equilibrio sin señal, en la mitad de las ramas del puente existe una señal de 2.5V respecto a masa. Esta señal de corriente continua es *común* a ambas entradas por lo cual es llamada Voltaje de Modo Común de la señal diferencial. Se puede ver que estas señales no contienen información útil en lo que se quiere medir y como el amplificador amplificará la *diferencia* de ambas, al ser iguales, se restan y a la salida el resultado será cero e idealmente no hay señal de salida. También se ve que se inducen señales de corriente alterna en ambas entradas a la vez y que serán rechazadas como en el caso de continua. Pero al producirse un desbalance del equilibrio del puente por la variación de una de sus resistencias se producirá una señal que será aplicada entre ambas entradas y será amplificada. Por lo expuesto, es que se justifica la utilización de amplificadores de instrumentación para rechazar señales que entran en modo común, en las dos entradas en que se presenta la misma señal.

En la práctica, las señales de modo común nunca serán rechazadas completamente, de manera que alguna pequeña parte de la señal indeseada contribuirá a la salida.

Para cuantificar la calidad del Amplificador de Instrumentación, se especifica la llamada Relación de Rechazo de Modo Común (CMRR) que matemáticamente se expresa como:

$$CMRR[db] = 20 \log\left(\frac{A_D}{A_{CM}}\right)$$

Siendo:

- A_D= Amplificación Diferencial
- A_D = Vout / Vin diferencial
- A_{CM}= Amplificación Modo Común
- V_{CM}= Voltage de modo común en la entrada
- A_{CM} = Vout / V_{CM}
- V_{out}= Voltage de salida

De la última fórmula podemos obtener la Vout como:

$$V_{out} = \frac{A_D}{\log^{-1}\left(\frac{CMRR}{20}\right)} \cdot V_{CM}$$

De las hojas de datos de los Amplificadores de Instrumentación, que proporciona el fabricante podemos obtener por ejemplo el factor de rechazo en modo común, la ganancia del amplificador, y el voltaje V_{CM} de modo común. La especificación de CMRR en función de la frecuencia se obtiene de las hojas de datos. En la figura siguiente se puede apreciar como varía el CMRR, este disminuye a medida que aumenta la frecuencia.

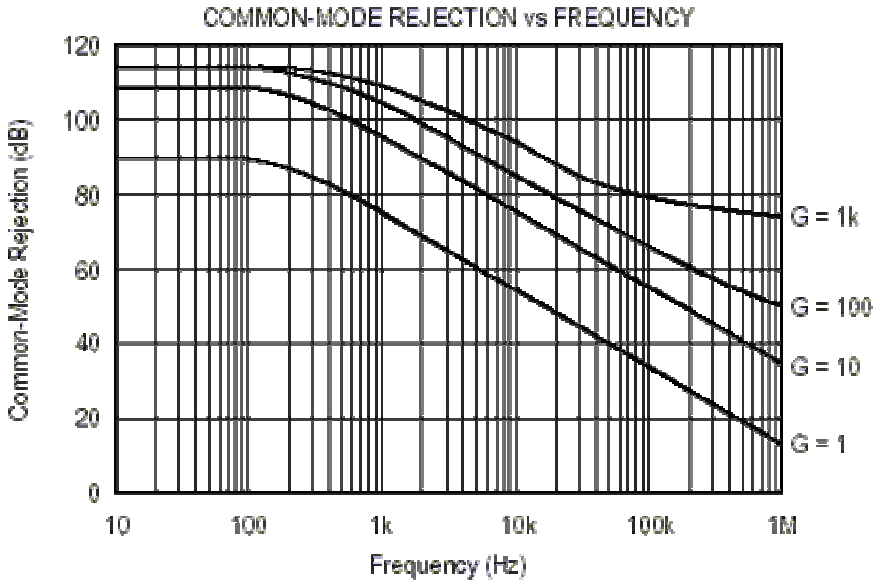


Figura 5.2 Relación del rechazo en modo común con la frecuencia

Una configuración típica de amplificador operacional se muestra en el siguiente circuito, La ganancia de estos dispositivos se consigue modificando una resistencia Rg que se coloca entre las terminales inversoras de los dos amplificadores de entrada. En el mercado existen dispositivos integrados con esta función, y el fabricante los presenta como amplificadores de precisión. Sin embargo estos son a veces relativamente caros, para aplicaciones que no requieren una precisión demasiado sofisticada

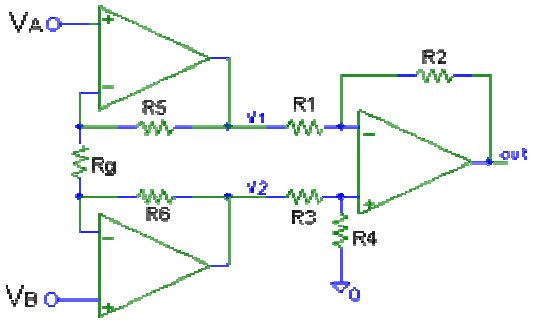


Figura 5.3. Configuración de tres Amplificadores de alta impedancia, como amplificador de instrumentación, con ganancia variable, mediante Rg.

En este amplificador diferencial, se acostumbra a hacer R5=R6=R; R1=R3 y R2=R4 en donde la amplificación diferencial será:

$$A_D := \frac{V_o}{(V_b - V_a)} = \frac{R2}{R1} \cdot \left(1 + \frac{2 \cdot R}{Rg}\right)$$

Considerando $V_A = V_B = V_{CM}$ y como los amplificadores de entrada están en una configuración simétrica, la misma tensión aparece en V1 y V2, de manera que:

$$\frac{V_{out}}{V_{CM}} = \frac{R4}{R3 + R4} \cdot \frac{R1 + R2}{R1} - \frac{R2}{R1}$$

Y el CMRR es:

$$CMRR = 20 \cdot \log\left[\frac{A_D}{\left(\frac{R4}{R3 + R4} \cdot \frac{R1 + R2}{R1} - \frac{R2}{R1}\right)}\right]$$

Nuevamente el CMRR depende de la ganancia A_D y del cuidado en seleccionar los valores de las resistencias, ya sea para que sean lo más iguales posibles o sus relaciones de unas a otras sean lo más exactas posibles. En este trabajo se utiliza esta configuración para ilustrar el acondicionamiento de la señal de salida del sensor AD590.

5.2 Sensor de Temperatura con Salida en Corriente (AD590)

La figura siguiente muestra el circuito para medir de -50 a +100 ºC con una salida de 10mV/ºC. La salida de la referencia de tensión del CI MC1403u, de 2.500V, es dividida por una red de resistencias para fijar un offset de 273.2mV para obtener la tensión de 0V a 0ºC (273ºK); el LF347 es un CI de 4 amplificadores de entrada bi FET con alta impedancia, mediante el cual se realiza un arreglo de amplificador de instrumentación con una ganancia de 10, con lo que el rango de salida corresponde con una tensión de -0.5V a 1V (para temperaturas desde -50ºC a +100ºC con una resolución de 10mV/ºC), pero para un rango de entrada de 10mV/ºC. Esta tensión, considerablemente pequeña, puede verse afectada por el ruido, cuanto más alejado se encuentre el sensor del circuito de amplificación.

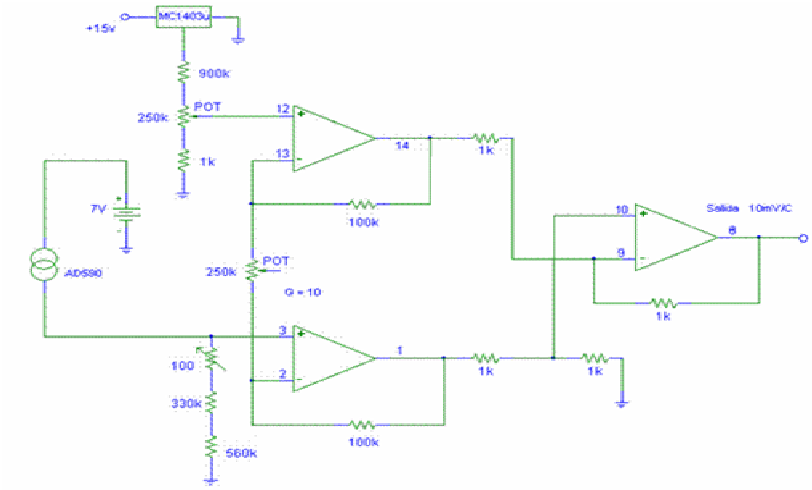


Figura 5.4. Esquema del sensor de temperatura

Cálculo de la función de transferencia para el circuito amplificador de instrumentación:

$$V_X = V_{R2} + V_{Rg} + V_{r2} + V_y \text{.....(1)} \qquad V_X = 2V_{R2} + V_{Rg} + V_Y = 2R_2\left(\frac{V_1 - V_2}{R_g}\right) + (V_1 - V_2) + V_Y$$

$$I_E = \frac{V_1 - V_2}{R_g} \text{...(2)}$$

$$V_X - V_Y = 2\left(\frac{2R_2}{R_g} + 1\right)(V_1 - V_2) \text{.....(3)}$$

$$\frac{V_Y - V_a}{R_1} - \frac{V_a}{R_1} = 0 \qquad \Rightarrow V_a = \frac{1}{2}V_Y$$

$$\frac{V_X}{R_1} - \frac{V_Y}{2R_1} + \frac{V_0}{R_1} - \frac{V_Y}{2R_1} = \frac{V_X}{R_1} - \frac{2V_Y}{2R_1} + \frac{V_0}{R_1} = 0 \qquad \Rightarrow -\frac{V_X - V_g}{R_1} = \frac{V_0}{R_1}$$

$$V_0 = -(V_X - V_y) = -\left(\frac{2R_2}{R_g} + 1\right)(V_1 - V_2)$$

$$\therefore \quad V_0 = \left(1 + \frac{2R_2}{R_g}\right)(V_2 - V_1)$$

Para que el amplificador de instrumentación tenga una ganancia de 10 se hacen los siguientes cálculos:

$$G = \left(\frac{2R_2}{R_g} + 1 \right) = 10 \qquad Si \quad hacemos \quad que \quad R_2 = 100k\Omega$$

$$1 + \frac{2(100k)}{R_g} = 10$$

$$\therefore \quad R_g = \frac{2(100k\Omega)}{9} = 22.22222k\Omega$$

Con G = 10

5.3 Configuración básica del circuito sensor de temperatura AD590.

Los ruidos pueden afectar la medida y pueden proceder del propio circuito como los transitorios en la alimentación o ruidos digitales. Se deberá proteger cuidadosamente el diseño del circuito impreso (masas analógicas y digitales separadas, desacoplos en las alimentaciones, etc.). Pero las fuentes de ruido también Sensores – Acondicionadores – Procesadores pueden ser externas y acoplarse o inducirse sobre el propio sensor o sobre el cable. Estas fuentes de ruido pueden ser ruidos industriales, la propia red eléctrica o incluso emisiones de radio (y ser "rectificadas" por el propio sensor que se comporta como una fuente de corriente de alta impedancia).

Para reducir todas estas influencias externas, una buena solución puede ser el empleo de cables trenzados, con lo que se eliminan los ruidos en modo diferencial o el uso de cable apantallado. Pero el apantallamiento, para que sea efectivo, debe hacerse de tal manera para no añadir efectos indeseados, debidos a las tensiones generadas en los bucles de masa. Por regla general, las pantallas solo deben conectarse a masa en uno de los extremos.

5.3.1 Reducción de interferencias en la medida de temperatura con el circuito integrado AD590.

El AD-590 es un circuito integrado de dos terminales que alimentado entre +4 V y +30 V funciona como sensor de temperatura. Las fuentes de señal en tensión típicas tratadas hasta ahora se caracterizan por una impedancia de salida baja, no así este dispositivo, que se comporta como una fuente de corriente de alta impedancia proporcionando una corriente de 1 μA/K. Se calibra tallando por láser las resistencias de película delgada de forma que proporcione una salida de 298.2 μA a 298.2 K (25ºC). Con un rango de medición de –55ºC a +150ºC con una precisión de ±1ºC.

El MC 1403u es una referencia de tensión que proporciona una salida estable de 2.5 V. Mediante la red de resistencias, genera una tensión de offset de 273.2 mV que es restada de la tensión de la resistencia de precisión de 1 kΩ. El LF347, es un CI de 4 amplificadores mediante el cual se ha hecho un arreglo de amplificador de

instrumentación. Para nuestro estudio es suficiente saber que su función de transferencia es en este caso $G = \left(\frac{2R_2}{R_g} + 1 \right) = 10$

La energía RF (Radio Frecuencia) entra y se refleja en un sistema dondequiera que exista un desacoplo de impedancias o discontinuidad. Esto incluye las discontinuidades que existen en los extremos del trayecto de la señal, es decir el arreglo de amplificadores en el LF 347 y el AD590. Para eliminar el efecto de esas discontinuidades, sería necesario un apantallamiento RF del circuito completo, encerrándolo por ejemplo en una caja metálica.

6. Diseño del Dispositivo de adquisición y registro de información

En este capítulo se presenta el diseño del dispositivo de registro de información. Presentamos el protocolo de comunicación (RS-232) de la tarjeta de adquisición de datos, con el puerto serial de una computadora personal. Se describe la arquitectura básica de dicha tarjeta, consistente en una serie de circuitos integrados dedicados a codificar información analógica y digital. Se explica la utilización de dos memorias SRAM externas a un microcontrolador AT89C51, así como la utilidad del circuito integrado 74LS573 latch, que sirve para direccionar la información de las memorias SRAM. También se muestra el desarrollo del código de programación del microcontrolador AT89C51, así como el empleado, en C ++ builder para mostrar de manera grafica los datos obtenidos en una PC. Entre los objetivos del proyecto, destacamos como más importantes; el bajo consumo de potencia para el funcionamiento autónomo de la tarjeta de adquisición de datos, junto con el sensor acondicionador, el bajo costo, gran capacidad de almacenamiento y funcionamiento en tiempo real. En la Figura 6.1 se puede apreciar un esquema general del sistema de adquisición de datos desarrollado en este trabajo. Donde se muestra al sensor con la circuitería necesaria para acondicionar su señal, y en la figura central se observa la tarjeta de adquisición de datos. En forma general se indica el orden que debe recorrer la señal para su posterior despliegue en una computadora.

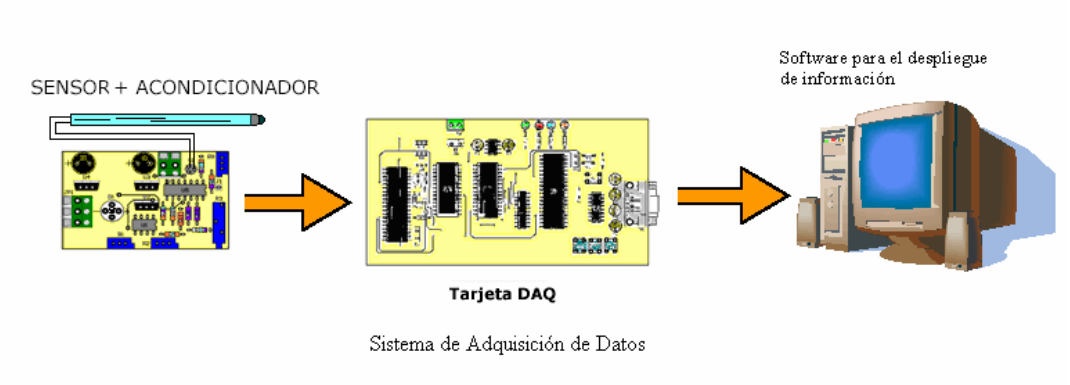


Figura 6.1 Esquema general del sistema de adquisición de datos.

Cabe resaltar que este trabajo comprende sólo el diseño de un sistema de adquisición de datos de propósito general, por lo que no se contempló el diseño de una fuente conmutada que fije un valor de tensión constante en el tiempo, para el funcionamiento adecuado del sistema de adquisición de datos expuesto en este trabajo. El diseño del sistema de adquisición de datos, se complementa utilizando un sensor de temperatura, el cual fue explicado junto con su acondicionamiento de señal en el capítulo 5. Para hacer las pruebas de funcionamiento del proyecto, presentado aquí, se emplea una fuente simétrica de $\pm 15V$ y +5V, utilizando un transformador de 120V/24V y reguladores de tensión LM 7815, LM 7915 y LM 7805.

El sistema de adquisición de datos expuesto (SAD) se basa en un sistema informático (un microcontrolador conectado a 2 memorias externas y circuitería para el

direccionamiento de información) que gobierna una cadena de adquisición analógica, cuya señal proviene de un sensor y acondicionador. En la figura 6.2 se puede ver la tarjeta de adquisición de datos, implementada, a la que entra una señal analógica de 10mV/1°C. La explicación del funcionamiento de la tarjeta de adquisición de datos, inicia con la salida de la señal analógica del amplificador de instrumentación, descrito en el capítulo 5, y como se describe de manera grafica en la figura 6.2.

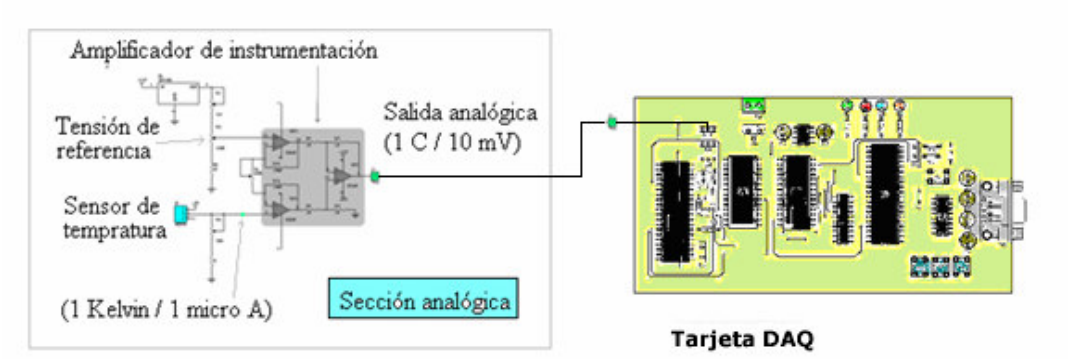


Figura 6.2 Salida de la señal analógica de 10mV/1°C conectada a la tarjeta de adquisición de datos.

La tarjeta de adquisición de datos de este proyecto, tiene como componentes básicos, un convertidor analógico/digital (TC 109A) de 12 bits más el signo, 2 memorias SRAM estáticas (HL 62256) de 32Kbytes, un latch 74LS573 para direccionar la información de las memorias, y un microcontrolador de 8 bits AT 89C51.

6.1 Convertidor Analógico/digital

La conversión analógica a digital es el proceso por el cual una magnitud analógica se convierte a formato digital. En este proyecto la conversión analógica/digital es necesaria, porque deben ser expresadas, en forma digital, una serie de magnitudes analógicas, para poder procesarlas en una computadora o bien almacenarlas, tal y como se explica en este apartado.

La señal de entrada en nuestra tarjeta DAQ es una señal analógica que es proporcional a una magnitud física, la temperatura (10 mV por cada 1 ^{0}C). Dicha magnitud se convierte a una señal digital, para poder manipularla con los sistemas del microcontrolador AT89C51, y para ello se ha utilizado el CI TC 109A, que es un convertidor analógico digital (ADC) CMOS de baja potencia, con una resolución de 12 bits más el signo.

Se decidió utilizar el CI TC 109A debido a la gran resolución que ofrece y por sus características típicas, las que incluyen una entrada de corriente que tiende a 1 pA, drift menor a 1µV/^{o}C, y ruido típico de entrada de 15µVp-p.

El TC109A proporciona una interfaze digital versátil, en el Modo directo, con las funciones chip select y HIGH/LOW byte (enable), que son parte de las funciones que habilitan la transmisión de datos, en el bus de la interfaze al puerto cero del microcontrolador, cuando se esta trabajando en línea. En el modo Handshake, el TC109A estándar opera controlando transmisiones de datos en UARTs serial. En la figura 6.3 podemos observar la arquitectura interna del convertidor analógico/digital TC

109A, así como sus componentes externos, necesarios para su funcionamiento y cuyos valores vienen especificados por el fabricante.

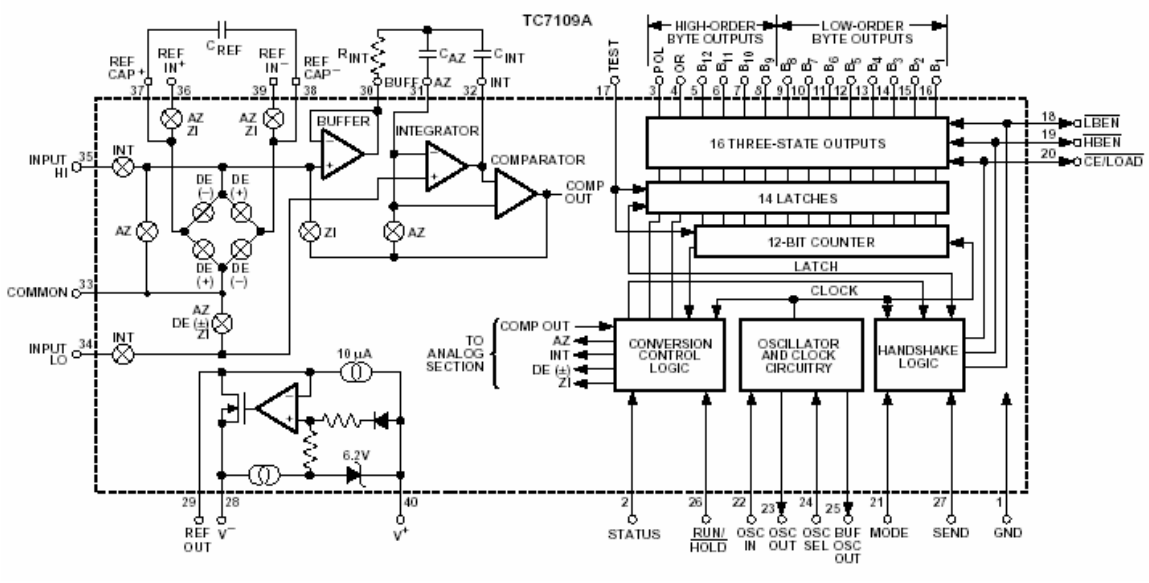


Figura 6.3 Arquitectura interna del Convertidor analógico/digital TC 109A.

El circuito de la figura 6.3 realiza las conversiones de forma digital/analógica en un porcentaje determinado por la frecuencia de reloj, 8192 periodos por cada ciclo. Por las especificaciones del fabricante, el pin 26 del TC 109A debe ser conectado a +5V, y entonces cada ciclo de medida se divide en cuatro fases, como se muestra en la figura 6.4; (1) Auto-Cero **(AZ)** (2) Señal Integrante **(INT)** (3) Referencia de Integración **(DE)** y (4) Cero integrador **(ZI)**, las cuales determinan el funcionamiento interno del TC 109A.

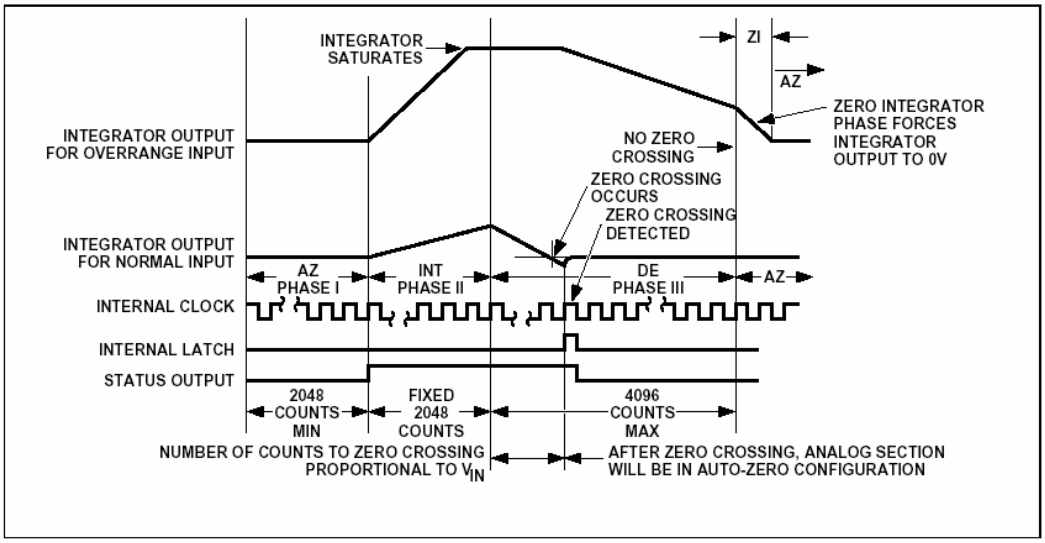


Figura 6.4 En la figura se muestra el cronograma de conversión analógico/digital de las cuatro fases de integración de la señal analógica a digital, para el CI TC109A.
Fase de auto-cero (AZ)

Se utiliza un generador de rampa integrador, en la fase de Auto-cero, el buffer y las entradas del circuito integrador (generador de rampa), figura 6.3, se desconectan internamente de la entrada alta y baja para conectarse a tierra, entonces la salida del integrador es cero, figura 6.4. El capacitor C_{REF} de referencia es cargado a la tensión de la referencia. Suponiendo que se aplica una tensión V_{IN} (pin 35) constante, durante un cierto periodo de tiempo, circulara una corriente constante a través de la resistencia R_{INT}, y por lo tanto a través del capacitor C_{AZ} (capacitor de auto-cero). El capacitor C_{AZ} se cargara linealmente, ya que la corriente es constante, se compensará la tensión de desviación en el amplificador buffer, en el integrador y en el comparador. La fase de auto-cero es limitada sólo por el ruido del sistema y la desviación referente a la tensión de entrada es menor de 10 μV.

Señal Integrante de Fase (INT).

Cuando el contador alcance un valor de cuenta especificado por la señal de reloj, figura 6.4, la entrada del buffer y del integrador se desconectan internamente de GND, por la lógica de control interna, y se conectan a la entrada alta (Input high pin 35) y baja (Input low pin 34) respectivamente. Entonces el capacitor de auto cero C_{AZ} se conecta internamente en serie a la entrada del comparador, para proporcionar una igual, pero opuesta compensación de tensión o voltaje de offset, proporcional a la tensión analógica de entrada (pin 35). El voltaje diferencial de la entrada alta y el de la entrada baja se integran para un tiempo fijo de 2048 periodos de reloj. Al final de esta fase la polaridad de la señal integrada se determina para una entrada en Input high.

Fase de integración (DE)

En esta fase, la entrada en alto (Input high) es conectada a través del capacitor de referencia C_{REF}, previamente cargado, y la entrada baja es conectada internamente a GND. La red de circuitos dentro del chip, asegura que el capacitor C_{REF} se conectara con polaridad correcta para causar que la salida del integrador vuelva a cruzar por cero, con una pendiente fija, figura 6.4. El tiempo representado por el número de periodos de reloj en el contador para que la salida regrese a cero, es proporcional a la señal de entrada, en cuanto a la velocidad de carga y descarga del capacitor de referencia C_{REF}. Cuando la tensión de salida del integrador alcanza el valor cero, el comparador conmuta al estado bajo e inhabilita la señal de reloj aplicado al contador, la cuenta binaria se almacena en los latches, completando un ciclo de conversión. La cuenta binaria es proporcional a Vin (tensión de entrada Input high ó input low) y el contador registra este intervalo de tiempo.

Integrador de Fase cero (ZI).

La fase de ZI sólo ocurre cuando existe una entrada en condición de sobre rango, esto es, que la lectura del sensor en su salida analógica, sobrepasa el rango de Vmax a la entrada del TC 109A. El pin 4 del convertidor analógico/digital indica el estado de sobre rango con un 1 lógico a su salida. La función de la fase ZI deberá eliminar la carga residual en el capacitor del integrador C_{INT} después del sobre rango medido, a menos que sea eliminada la carga residual será transferida al capacitor de auto cero C_{AZ} causando un error en la conversión posterior. Durante ZI el capacitor C_{REF}, es cargado a la tensión de referencia, las señales de entrada se desconectarán del buffer y del

integrador. Entonces la salida del comparador es conectada a la entrada del buffer causando que la salida del integrador pueda ser manejada a cero volts.

Durante un ciclo de conversión la señal de salida esta en alto (high) al principio de la señal integrante y esta abajo a la mitad del periodo del reloj, después de que los nuevos datos de conversión se han almacenado en los latches de salida, la señal se puede utilizar con los datos validos.

6.2 Resolución y velocidad del Convertidor ADC (TC 109A)

La velocidad de muestreo determina la precisión con la que la secuencia de códigos digitales representa la entrada analógica del TC 109A. Cuantas más muestras se toman en una determinada unidad de tiempo, más precisa es la señal digital que representa a la señal analógica. La figura 6.5 muestra de manera grafica la frecuencia de muestreo de una señal analógica. El ADC TC 109A realiza conversiones en un porcentaje determinado por la frecuencia de reloj, de 8192 periodos por ciclo.

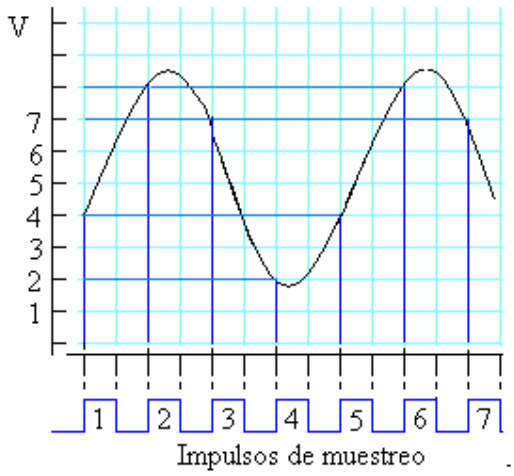


Figura 6.5. Representación grafica, de la velocidad de muestreo de una señal analógica, para convertirla en una señal digital.

Para generar la salida en la escala total de 4096 cuantas o combinaciones que proporciona el TC 109A, en los rangos positivo y negativo de las tensiones analógicas de entrada, cuando el circuito integrado realiza conversiones, a una frecuencia determinada por 8192 periodos por ciclo de reloj, según la hoja de especificaciones del TC 109A, la entrada analógica requerida es:

$$Vin = 2V_{REF} \quad \ldots\ldots 6.1$$

En nuestro sistema, el convertidor se implementa con la escala total de 4096 cuentas, para los rangos de tensión del sensor de temperatura AD 590 (-50 ^{0}C a 100 ^{0}C), el cual proporciona a su salida tensiones de -0.5V a 1V. De las especificaciones del fabricante, la resolución máxima que se puede obtener en el TC 109A es de 0.250 mV, es decir, el valor mínimo de tensión, a la salida de nuestro sensor de temperatura, que podemos cuantizar es de 0.250 mV.

La tensión máxima aplicada a la entrada del TC 109A esta definida por la siguiente ecuación:

$$\text{Re}\,solución_{TC109A} = \frac{V_{INman}}{2(2^n)}, \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.2$$

- V_{INmax} = Tensión de entrada máxima.
- n = numero de bits del TC 109A. n = 12
- $\text{Resolución}_{TC109A}$ = 0.25 mV

Despejando la ecuación de la resolución (ecuación. 6.2) del TC 109A, encontramos la tensión de entrada máxima.

$$\Rightarrow \quad V_{IN\max} = \text{Re}\,solución_{TC109A}\left[2(2^n)\right] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.3$$

$$V_{IN\max} = 8192(0.25mV) = 2.048 \quad V \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.4$$

La tensión de entrada máxima V_{Inmax} = 2.048 V, representa el rango completo de valores negativos y positivos de la tensión analógica, que entrega el sistema del sensor de temperatura, tal como se representa en la figura 6.6. Lo anterior implica una tensión de referencia V_{REF} = $\pm$1.024V, para una entrada V_{INmax} = $2V_{REF}$, definida por el fabricante.

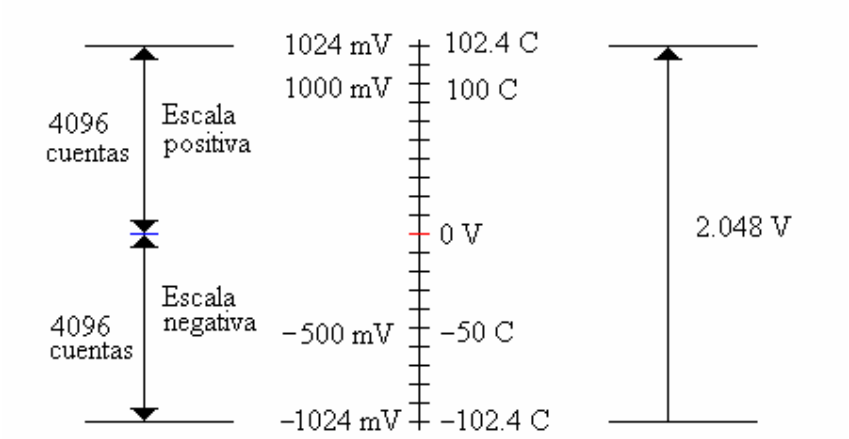


Figura 6.6 La figura representa el rango completo de cuantización de la señal de entrada para el proyecto de sistema de adquisición de datos.

A manera de ejemplo, si tenemos una señal de entrada, proveniente de nuestro sensor, en el pin 35 del TC 109A, cuyo valor sea V = 0.245 V que representa a 24.5 ^{0}C, el numero de cuentas que el convertidor ADC realizará, para convertir esta señal analógica a digital, será de 980 cuentas, como lo indican los datos de la ecuación 6.5 y la representación grafica de la figura 6.7.

$$\#núm \quad de \quad cuentas = \frac{V_{IN}}{V_{\text{Re}sol(ADC)}} = \frac{0.245 \quad V}{0.25 \quad mV} = 980 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.5$$

Donde:

- V_{IN} es la tensión analógica de entrada.
- $V_{Resol(ADC)}$ es la resolución mínima que ofrece el CI TC 109A.

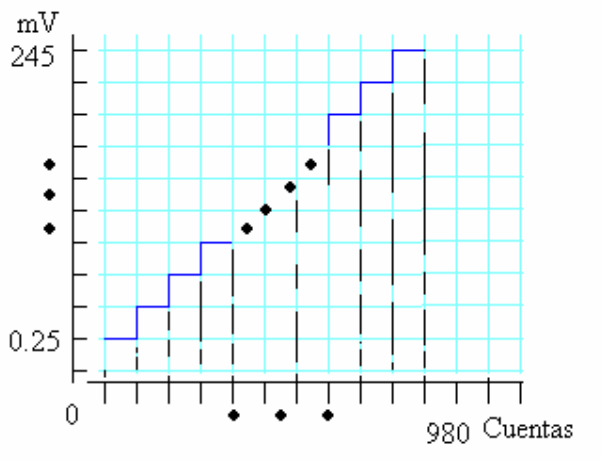


Figura 6.7 Representación grafica de la cuantización, de una señal analógica de entrada de 0.245 V equivalente a 24.5 ^{0}C. En la figura se representan la resolución mínima que ofrece el TC 109A de 0.25 mV, y las 980 cuentas en que es dividida la tensión de entrada.

El convertidor Analógico/digital divide la tensión de entrada (Vin = 0.245 V), en 980 cuentas, y a cada cuenta le asigna un valor binario que representa el valor en bits o bytes de la señal analógica de entrada. Estas señales en bits o bytes, las utilizará el microprocesador para controlar sus elementos periféricos; puertos, buses de direcciones y datos, comunicación con otros elementos de lógica digital, así como transmisión y recepción de información. Las salidas de los pines 5 a 16 del TC 109A representa en numero binario el numero de cuentas en que fue dividida la señal de entrada, como se puede observar en la figura 6.8.

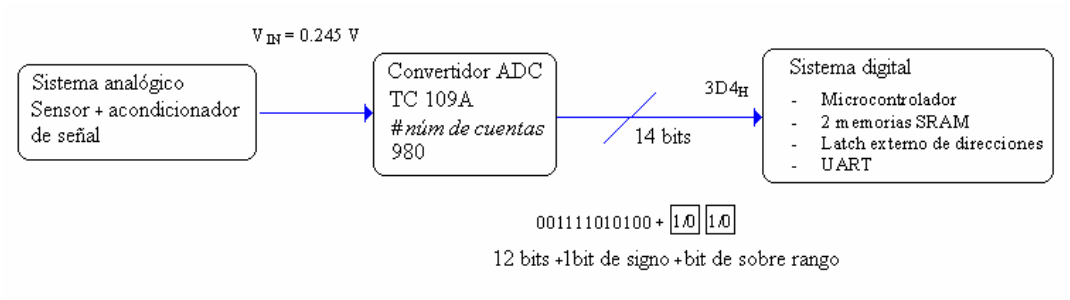


Figura 6.8 La señal de entrada V_{IN} = 0.245 v es cuantizada por el ADC en 980 cuentas, con una salida digital de 12 bits más dos bits de signo y sobre rango.

Cuando el pin 21 del TC 109A es conectado a Low (estado bajo), los datos de salida (pines 3 a 16) son directamente accesibles. En el ADC utilizado en este proyecto, el modo de salida es controlado por el modo de entrada (pin 21 Input low) el cual viene especificado en la tabla 6.1 para el pin 21.

Por medio de Software, el microcontrolador AT89C51 habilita al convertidor analógico digital enviando una señal baja (0 lógico) al pin 20 (CE/LOAD, chip enable/load), para que el TC 109A pueda enviar información al microcontrolador o a las memorias. Para trabajar en modo autónomo, es decir, sin conexión con la computadora, el microcontrolador también habilita una de las 2 memorias SRAM para que pueda recibir

la información que envía el convertidor analógico/digital, esto sucede cuando el microcontrolador manda desde el pin 6 del puerto 3 (Pin 3.6) un cero lógico, para habilitar en modo de escritura, una de las dos memorias SRAM en WE (pin 27 de la memoria). El convertidor analógico digital registra los datos, del sensor de temperatura, en cada una de las memorias SRAM externas, conforme cada una vaya agotando sus registros para almacenar información. En este proyecto se ha diseñado un prototipo de sistema de adquisición de datos que sólo direcciona 64K de memoria externa, por lo que la información almacenada del convertidor analógico/digital se ha ajustado a la capacidad de bytes de direccionamiento.

Cuando el convertidor analógico/digital entrega datos directamente al microcontrolador, también se hace una selección del CI TC 109A, a través del pin 0 del puerto 1, en el microcontrolador (Pin 1.0). El microcontrolador envía un cero lógico al pin 20 de TC 109A. Como todo el sistema esta trabajando en línea, conexión directa con la PC, el microcontrolador deshabilita las memorias SRAM por medio de software, y se utiliza el puerto 0 para leer en forma directa los datos que entrega el convertidor analógico/digital. Como la salida del TC 109A es de 12 bits más 2 bits de signo y otro de sobre rango, y el puerto P0 del CI AT89C51 es de 8 bits, la lectura del convertidor analógico/digital se hace por partes. El microcontrolador primero habilita la parte alta de los datos que entrega el convertidor (POL, OR, B12 a B9), tal como se describe en la tabla 6.2. Para habilitar la parte alta de datos que entrega el convertidor, se envía desde el microcontrolador P1.4 (pin 4 del puerto 1) un cero lógico al pin 19 (HBEN High byte enable) del TC 109A Para leer la parte baja de los datos que entrega el TC 109A se deshabilita el envió de la parte alta de los datos, escribiendo un nivel alto (1 lógico) en HBEN, como se observa en la tabla 6.2. De igual manera que se habilitó la parte alta de datos, podemos habilitar la parte baja de datos, enviando un 0 lógico desde el microcontrolador P1.3 (pin 3 del puerto 1) al pin 18 (LBEN Low byte enable) del TC 109A, para así, habilitar la parte baja del convertidor (B1 a B8). Una vez que el microcontrolador almaceno de forma temporal los datos, tanto de la parte alta como de la parte baja en su memoria interna RAM, despliega esa información en un ambiente grafico de programación, en una computadora personal.

6.3 Parámetros específicos del circuito CI TC 109A

Voltaje de referencia.

La tensión del capacitor referencia C_{REF} interactúa en cada una de las fases que se llevan acabo en la conversión de una magnitud analógica a digital dentro del CI TC 109A, por lo que, para generar una salida de la escala total de 4096 cuentas, tal como se esta manejando en este proyecto, la entrada analógica requerida es $V_{IN} = 2V_{REF}$, en este caso $V_{IN} = 2.048$ V (ecuación 6.3) y el $V_{REF} = 1.024$ V. la tensión de referencia determina un factor de escala (la resolución del convertidor) entre la tensión de entrada y la lectura digital, ver ecuación 6.5. Sin embargo como el sensor de temperatura AD 590 registra en la salida, de su acondicionamiento de señal, valores positivos y negativos correspondientes a los valores de temperaturas, sobre cero y bajo cero, es necesario tomar en cuenta que las 4096 cuentas de la escala total sólo representan a la parte de las temperaturas ligadas a las tensiones positivas de entrada (temperaturas positivas), por lo que es necesario tener una tensión de referencia, tanto para las tensiones positivas como para las tensiones negativas, que entran al convertidor analógico/digital. En la figura 6.9 se observa la configuración de dos potenciómetros de precisión trimpot, de una vuelta

(pin 39 y 36), los cuales nos sirven para fijar una tensión de referencia positiva y otra negativa (± 1.024 V).

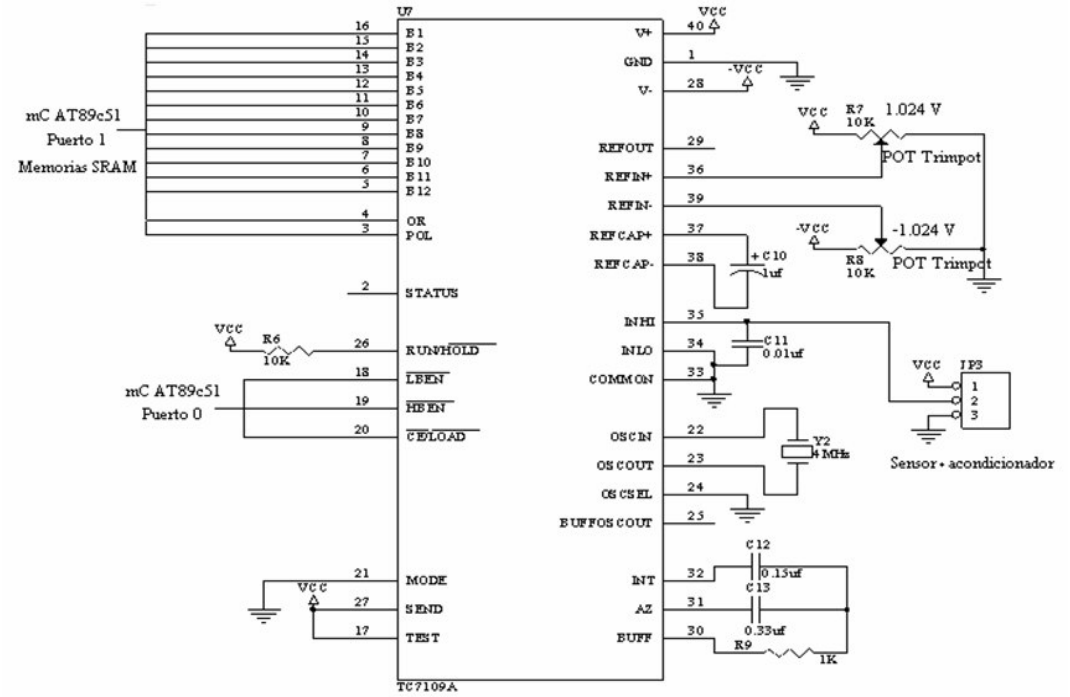


Figura 6.9. Configuración de los pines del CI TC 109A para la conversión de tensiones en rangos positivos y negativos. En los pines 36 y 39 se han fijado tensiones de referencia con potenciómetros trimpot de 10KΩ.

Cada una de las tensiones de referencia son introducidas conectando a los pines 36 y 39 (Ref in+, Ref in-) un potenciómetro de precisión de 10kΩ, como en la figura 6.9, para regular los tensiones de referencia, necesaria y para determinar la conversión de los valores analógicos del sensor de temperatura.

Oscilador para el TC 109A.

El ADC TC 109A necesita un oscilador para generar 8192 cuentas por igual numero de periodos de reloj. El oscilador puede ser operado con un circuito RC o un cristal oscilador. Cuando la entrada del pin OSC SEL es alta (V+), el oscilador puede ser configurado en RC para su operación. En la figura 6.10 se observa la configuración de un circuito RC que determina una frecuencia de oscilación dada por el fabricante.

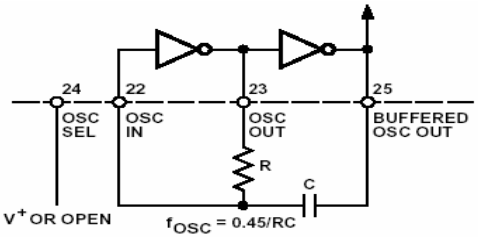


Figura 6.10. Configuración RC para generar una señal de reloj

Conectando un capacitor y una resistencia como se muestra en la figura 6.10, el circuito oscilara a una frecuencia f_{OSC} = 0.45/RC. Una resistencia de 100KΩ es recomendada por el fabricante, el valor del capacitor se debe escoger para cada 2048 periodos de reloj y cercano a un múltiplo integrante de 60 Hz.

Para este proyecto se ha decidido utilizar un cristal oscilador comercial de 4 MHz, para tener mayor precisión en la generación de la señal de reloj, para ello se ha conectado la entrada del pin 24 (OSC SEL) a tierra. En esta configuración el oscilador operará en el rango de 1 MHz a 5 MHz sin componentes externos, como se puede observar en la figura 6.9 (pin 22 y 23). El cristal oscilador utilizado proporciona un tiempo de integración de t = 29.69 ms (ecuación 6.6). Cuando el pin 24 es conectado a tierra, circuitos internos del TC 109 fijan 58 divisiones a la frecuencia de oscilación entre la salida del buffer oscilador y el reloj interno.

$$t = (2048 \text{ periodos de reloj})\frac{58}{4MHz} = 29.69ms \quad \ldots\ldots 6.6$$

El convertidor TC 109A opera con seguridad en porcentajes de conversión de 30 veces por segundo, correspondiendo a una frecuencia de reloj de 245.8 kHz Como el pin 24 (OSC SEL) esta a nivel bajo, el periodo del reloj es 1/58 con la frecuencia de entrada del oscilador de cristal. En la figura 6.10 se puede observar la configuración interna del TC 109A, para una conexión con un cristal oscilador en los pines 22 y 23, un cuadro indica la división, por 1/58, que ejecuta el reloj interno, a la frecuencia de entrada del cristal de 4 MHz.

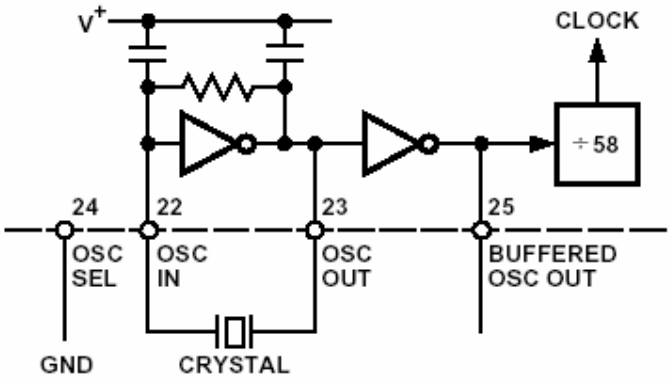


Figura 6.11 Configuración para un cristal oscilador de 4MHz. En la figura la salida del pin 24 esta conectada a un nivel bajo, caso contrario de la figura 6.10 donde se utiliza un circuito RC.

El capacitor de integración C_{INT}.

El capacitor de integración, C_{INT} se debe escoger para obtener en el integrador una máxima tensión de salida que no lo sature, más allá de 0.3V. Con ± 5 V suministrados como máximo en la tensión de referencia (V_{REF}). El valor nominal dado por el fabricante para C_{INT} es de 0.15 µF. Ver figura 6.9.

$$C_{INT} = 0.15\ \mu F \quad \ldots\ldots 6.7$$

Resistencia de integración R_{INT}.

La resistencia de integración debe ser lo suficientemente grande para poder actuar en una región lineal, del integrador interno sobre el rango de la tensión de entrada. Para 2.048 V en la escala completa, se recomienda una resistencia de 100 $K\Omega$. El fabricante proporciona una ecuación para determinar la resistencia de integración, para valores diferentes de 2.048 V de escala completa. Para este proyecto la resistencia de integración utilizada, es la especificada por el fabricante $R_{INT} = 100\ K\Omega$.

Para valores diferentes de la escala completa:

$$R_{INT} = \frac{Full\ \ scale\ \ voltage}{20\mu A} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.8$$

Capacitor de auto cero C_{AZ}.

Cuando el valor del capacitor de auto cero se hace grande, el ruido del sistema se reduce, entonces según las especificaciones del fabricante, el valor optimo del capacitor de auto cero esta entre 2 y 4 veces C_{INT}. El valor nominal dado por el fabricante es:

$$C_{AZ} = 0.33\ \mu F \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.9$$

Capacitor de referencia C_{REF}.

El fabricante recomienda utilizar un capacitor de 1µF para la mayoría de los arreglos de circuitos y aplicaciones donde sea utilizado el TC 109A. Pero cuando existe un gran voltaje en modo común se requiere un valor más grande, para prevenir este tipo de problemas.

$$C_{REF} = 1\ \mu F \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots 6.10$$

En la tabla 6.1 se describe el funcionamiento de cada uno de los pines del TC 109A, esta es información necesaria, al momento de programar el microcontrolador, ya que aquí se describe las funciones de cada uno de los pines del TC 109A.

Tabla 6.1 Descripción de los pines del convertidor Analógico/Digital.

Numero de Pin (40-Pin Dip)	Símbolo	descripción
1	GND	Tierra digital 0V GND para toda lógica digital
2	STATUS	Salida alta durante la integración y de integración hasta que los datos son detenidos en latched. Salida baja cuando la sección analógica es un auto-cero o configuración de cero integrador.
3	POL	Polaridad- alta para la entrada positiva
4	OR	Sobre el Recorrido- High (Alto) sobre el recorrido

		(Three-State Data bit)
5	B_{12}	B_{12} (Bit más signification) (Three-State Data bit)
6	B_{11}	B_{11} (Three-State Data bit)
7	B_{10}	B_{10} (Three-State Data bit)
8	B_9	B_9 (Three-State Data bit)
9	B_8	B_8 (Three-State Data bit)
10	B_7	B_7 (Three-State Data bit)
11	B_6	B_6 (Three-State Data bit)
12	B_5	B_5 (Three-State Data bit)
13	B_4	B_4 (Three-State Data bit)
14	B_3	B_3 (Three-State Data bit)
15	B_2	B_2 (Three-State Data bit)
16	B_1	B_1 (Three-State Data bit)
17	TEST	Entrada Alta – Operación normal. Entrada Baja. Pone a todos los bits de las salidas en Alto. Nota: esta entrada es usada solo para propósitos de prueba.
18	LBEN	Byte bajo habilitado-con el MODO LOW (Pin 21), y CE/LOAD (Pin 20) LOW Bajo, Tomando este Pin LOW activa las salidas del Byte de bajo orden B_1 –B_8. Con modo Alto MODE HIGH (Pin 21) este Pin sirve mientras que la bandera del byte bajo en la salida sea usada en el modo Handshake.
19	HBEN	Habilita el Byte Alto- Con el MODE LOW (Pin 21) y CE/LOAD (Pin 20) LOW tomando este Pin LOW se activan las salidas del Byte de alto orden B_9-B_{12}, POL, OR. Con MODE HIGH (modo alto), este Pin sirve mientras que la bandera del Byte alto es usada en modo Handshake.
20	CE/LOAD	Chip Enable/Load, con Mode LOW (Pin 21) CE/LOAD
21	MODE	Entrada Low- Modo directo de salida donde CE/LOAD (pin 20) HBEN (pin 19) y LBEN (pin 18) actúan como entradas controlando directamente la salida del byte. Entrada Pulsed HIGH – causa la entrada inmediata en Handshake mode y la salida de datos.
22	OSC IN	Entrada del oscilador
23	OSC OUT	Salida del oscilador
24	OSC SEL	Oscillator Select (Selección del oscilador) – Entrada HIGH configura OSC IN, OSC OUT y BUFF OSC OUT como oscilador RC – El reloj puede ser de la misma fase y debe ser del mismo ciclo como BUFF OSC OUT. Entrada LOW configura OSC IN, OSC OUT para el cristal del oscilador – la frecuencia de reloj será 1/58 de la frecuencia en BUFF OSC OUT.
25	BUFF OSC OUT	Salida del oscilador Buffered
26	RUN/HOLD	Entrada HIGH – Las conversiones son realizadas continuamente cada 8192 pulsos de reloj.
27	SEND	Utilizada en el modo Handshake para indicar la aceptabilidad de un dispositivo externo a aceptar datos.

		Se debe conectar a V+ si no es utilizada
28	V-	Suministro negativo analógico. Normalmente – 5V con respecto a GND, pin 1.
29	REF OUT	Referencia de voltaje de salida. Nominalmente 2.28V hacia debajo de V+ (pin 40)
30	BUFF	Salida del amplificador de buffer.
31	AZ	Nodo de Auto-Cero. Foil interior de C_{AZ}
32	INT	Integrador de salida (Integrator Output). Foil exterior de C_{INT}
33	COMMON	Analógico común. El sistema se pone automáticamente en cero.
34	IN LO	Entrada diferencial LOW.
35	IN HI	Entrada diferencial HIGH.
36	REF IN+	Referencia diferencial de entrada positiva
37	REF CAP+	Referencia del capacitor positiva
38	REF CAP-	Referencia del capacitor negativa
39	REF IN-	Referencia diferencial de entrada negativa
40	V+	Tensión de alimentación positiva. Nominalmente + 5V con respecto a GND (Pin 1)

En muchas ocasiones requerimos monitorear variables físicas en situaciones poco favorables, donde es poco probable, tener todo un equipo completo de supervisión que capture el valor de dichas variables, para visualizarlas en tiempo real. Pero también en muchas situaciones, no importa mucho, conocer el estado actual (en tiempo real) de la variable monitoreada. El sistema de adquisición de datos, presentado en este trabajo, fue diseñado para poder trabajar en forma autónoma y en línea. Para este proyecto, cuando decimos que se puede trabajar de manera autónoma, nos estamos refiriendo al proceso de capturar datos con un sistema remoto, para después poder visualizarlos de forma grafica o numérica utilizando una computadora personal.

Una de las prestaciones del prototipo de sistema de adquisición de datos de este proyecto, es poder programarlo desde la PC, para que trabaje de forma autónoma, es decir, sin estar necesariamente conectado por el puerto serial de una computadora personal, esto se logra mediante software de programación tanto en el microcontrolador, que gobierna la tarjeta de adquisición de datos, como creando una interfase grafica, mediante programación de alto nivel en una computadora personal, para que el usuario se comunique con la tarjeta de adquisición de datos, y a su vez le indique al microcontrolador en que forma quiere que el sistema trabaje, ya sea en forma autónoma, o en línea siempre muestreando valores de temperatura en tiempo real.

6.4 Almacenamiento de información en memoria SRAM externa

Para que el sistema de adquisición de datos trabaje en forma autónoma, como se ha definido, es necesario que el sistema cuente con una unidad o unidades de memoria, que le permitan al sistema almacenar los datos muestreados. De la capacidad de almacenamiento y el tiempo de trabajo, dependerán la cantidad de bytes con que cuentan las unidades de memoria, así como de la fuente de energía que estemos

ocupando. El microcontrolador AT 89C51 cuenta con una unidad de memoria interna RAM (128 bytes de memoria interna de datos SRAM), pero su capacidad es limitada por el almacenamiento de sus propias variables de programación y su tamaño de 128 bytes. En este proyecto se tiene contemplado que el tiempo mínimo de muestreo sea de 1 muestra por segundo, cada muestra implica la utilización de 2 bytes de registro o almacenamiento, esto es debido a la salida del CI TC 109A de 12 bits mas 2 bits de sobre rango y de signo que sumados son 14 bits, en caso de que quisiéramos utilizar los 128 bytes de la memoria interna del AT 89C51, sólo se podrían almacenar 64 datos, poco más de un minuto de muestreo, en ese lapso se acabaría la utilidad del sistema, pues no habría lugar para almacenar más datos, sin tomar en cuenta que el propio microcontrolador necesita memoria RAM para poder trabajar con sus propias variables de programación.

El prototipo de sistema de adquisición de datos presentado en este trabajo, se ha desarrollado de acuerdo a las características del microcontrolador AT89C51, éste sólo puede direccionar hasta 64 kbytes de memoria externa, es decir que necesitamos de dos puertos del microcontrolador, para acceder a la parte alta y baja del bus de direcciones en que se encuentra la información de la memoria, con ayuda de un CI latch 74LS573, podemos multiplexar, en un solo bus, direcciones y datos de la parte menos significativa de la información. Como cada puerto del AT89C51 es de 8 bits, y es necesario ocupar dos puertos, entonces la cantidad de bits ocupados será de $2^{16} = 65536$ que representan a 64K bytes.

Para almacenar la información procedente del Convertidor analógico/digital, necesitamos implementar unidades de memoria externa de datos. Una parte importantísima, en la mayoría de los sistemas digitales, es la dedicada a contener la información que está tratando dicho sistema. Los datos o instrucciones del programa de un sistema de adquisición de datos son almacenados en memorias externas o internas. Independientemente de ello cada "celda" de la memoria puede almacenar un bit, estando las memorias constituidas por varios miles de estas celdas. El conjunto de celdas en las que se almacena una palabra se llama "Posición de memoria" o dirección. Se han desarrollado numerosos sistemas capaces de almacenar o memorizar una información digital.

Memoria SRAM

Una memoria de tipo RAM puede ser borrada y grabada las veces que deseemos. La única desventaja, es que la información grabada en ella, solo puede ser utilizada mientras tenga energía. En cuanto se corte la alimentación, los datos que se grabaron se borrarán instantáneamente. Por lo que este tipo de memorias se usan solo como almacenamiento temporal. Existen dos variantes entre las memorias RAM: SRAM y DRAM. La SRAM es conocida como la RAM estática, en ella los valores binarios o información se almacenan utilizando configuraciones de compuertas flip flop, por lo que, mientras tenga la alimentación, los datos guardados en ella se mantendrán intactos. LA DRAM es conocida como RAM dinámica y esta hecha con celdas, que almacenan los datos como cargas en capacitores. La presencia o ausencia de carga en el capacitor se interpreta como el 1 o 0 binarios. Ya que los capacitores tienen una tendencia natural a descargarse, las RAM dinámicas requieren refrescos periódicos para mantener memorizados los datos. Aunque el costo de la RAM dinámica no es muy elevado, precisan de complejos circuitos de refrescamiento. En comparación, una memoria

SRAM requiere ser refrescada menos veces que la DRAM, y tienen un tiempo de acceso del orden de 10 a 30 nano segundos, por lo que las memorias SRAM son muy rápidas, tenido como única desventaja que usan mucha energía y espacio.

En el mercado existen memorias comerciales SRAM de 2 kbytes, 8k bytes o 32 kbytes. Para el desarrollo del registro de información, de la tarjeta de adquisición de datos, se utilizarán dos memorias comerciales SRAM de 32 K bytes, capaces de almacenar información durante nueve horas continuas, por supuesto, para ello se deberá contar con una fuente de energía que agote el suministro al sistema con suficiente tiempo para poder leer el dispositivo, una vez que este sea recogido por el usuario.

Si se precisa de mayor tiempo de muestreo, para trabajar por mas de nueve horas, se tendrán que implementar más memorias SRAM, por lo que dependiendo del tiempo de trabajo se requerirá de mas memoria, esto conlleva a diseñar una fuente de tensión capaz de suministrar corriente de forma regular al dispositivo durante el tiempo de trabajo, el diseño de dicha fuente no fue contemplado en este proyecto, por lo que, este punto se tendrá que retomar más adelante.

Para poder trabajar, en forma autónoma, durante cierto periodo de tiempo, y saber que cantidad de memoria necesitamos en ese lapso, es necesario definir un tiempo máximo, en el que el dispositivo trabajará en forma autónoma. En la implementación de nuestro prototipo de adquisición de datos, se ha definido que el dispositivo pueda trabajar durante nueve horas, como tiempo máximo, al utilizar 64K bytes de memoria SRAM externa. Para determinar el tamaño de la memoria a implementar se hacen los siguientes cálculos:

Tiempo mínimo de muestreo t_{MIN} = 1 segundo
En un minuto se tendrán: 60 muestras / minuto
En una hora: (60 muestras) (60 minutos) = 3600 muestras / hora.
En nueve horas: (3600 muestras/hora) (9 horas) = 32400 muestras

Si se ha diseñado el dispositivo para que este trabaje, de forma autónoma, durante nueve horas continúas, entonces se tendrán 32400 muestras por cada nueve horas de trabajo para un tiempo de muestreo de 1 segundo, las memorias comerciales de 32 K bytes son de 8 bits, y una sola muestra del sistema de adquisición de datos ocupa 12 bits más dos bits de signo y sobre rango, que sumados dan un total de 14 bits, entonces una sola muestra ocupara dos localidades de memoria SRAM. Por lo tanto la cantidad de memoria requerida será:

Memoria SRAM = (32400 muestras) (2 Localidades de memoria) = 64800 bytes 6.11

Podemos concluir que necesitaremos implementar dos memorias SRAM externas de 32K bytes cada una para tener 64 k bytes o bien 65536 localidades de memoria, mas que suficientes para albergar 64800 bytes

Para explicar cómo se guardan los datos en la memoria, supongamos, de manera imaginaria, que tenemos un librero con 65535 líneas para poner libros, como se muestra en la figura 6.12 a). En cada línea caben hasta 8 libros. Las líneas, de este librero, representan las direcciones que tiene una memoria SRAM de 64 kbytes. Las direcciones son los espacios disponibles para guardar información. Cada libro representa un bit de

información, por lo que nuestro librero representa una memoria con 65536 (2^{16}) direcciones, de ocho bits de capacidad. Cada dirección se representa por un número binario. Ahora si cambiamos nuestro librero por una memoria verdadera y los libros por bits tendremos que la memoria sólo puede almacenar 8 bits de información, como se muestra en la figura 6.12 b). Para nuestro proyecto, los primeros ocho bits menos significativos (B_1 – B_8) se almacenarán en un solo registro, mientras que los 6 bits restantes (POR, OR B_9 - B_{12}), mas significativos, ocuparan el registro siguiente de la memoria, esto es debido a la forma en que el convertidor analógico/digital entrega su información (12 bits mas dos bits de signo y sobre rango) al microcontrolador y a las memorias SRAM externas.

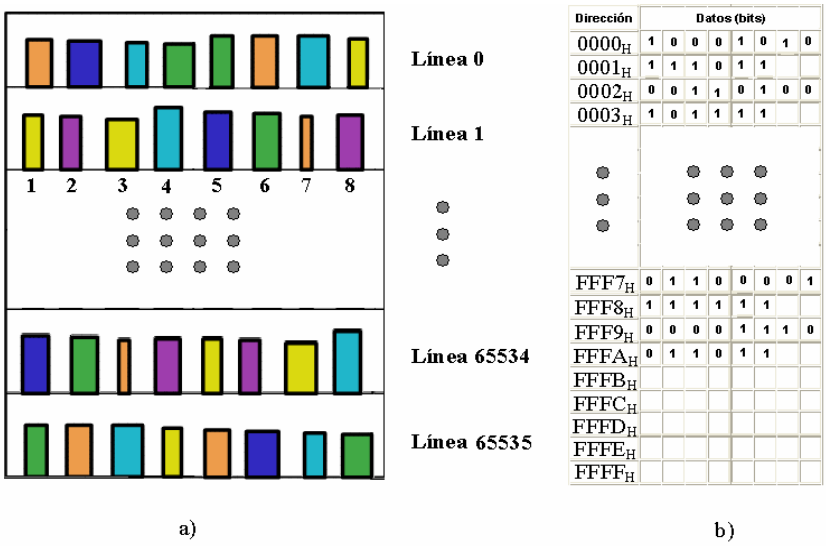


Figura. 6.12 La figura 6.12 a) es la representación de la memoria RAM externa, para almacenar datos, como un librero de 65536 líneas, con 8 libros por cada línea. La figura 6.12 b) muestra el almacenamiento binario de la información del sistema de adquisición de datos, junto con la dirección de cada localidad.

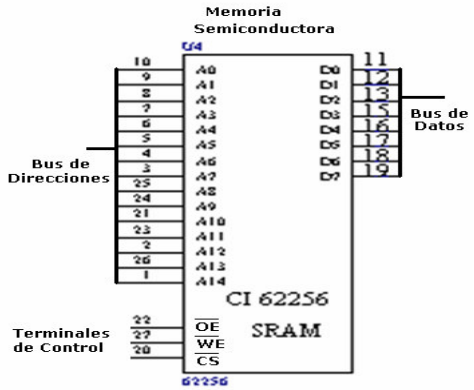


Figura 6.13. Estructura de la Memoria SRAM CI 62256. En la figura se pueden observar las terminales de control del circuito (OE, WE y CS), así como, el bus de datos y direcciones.

En la figura 6.13 se muestra la configuración básica de una memoria CI 62256, con capacidad para almacenar hasta 32 K bytes de información. El bus de datos (Do a D_7), como su nombre lo indica, es la entrada de los datos que entrega el convertidor analógico digital (TC109A). Por el bus de direcciones el microcontrolador realiza el direccionamiento del dato almacenado en la memoria. La señal Ale (adress lacht enable) es un pulso que permite multiplexar por el puerto 0 el bus de datos y los 8 bits menos significativos del bus de direcciones de la memoria externa, esto significa que el microcontrolador, al activar mediante software la señal ALE envía la dirección, que contiene el dato en la memoria externa, la memoria externa codifica esa información, y por el mismo bus de direccionamiento envía el dato, pero para ello, es necesario utilizar un dispositivo externo lacht 74LS 573.

Cuando el microcontrolador envía un cero lógico, desde el puerto P3.7, a la terminal de control OE (Output Enable) en el CI SRAM, se habilita la memoria externa para que, por el bus de datos, se muestre la información requerida. De igual manera sucede con la terminal de la memoria, WE (Write Enable), la cual se habilita desde el microcontrolador cuando este envía un cero lógico a dicha terminal en el CI SRAM. La terminal WE (Write Enable) nos va permitir escribir en la memoria los datos que envía el convertidor analógico digital. CS (Chip Select) es la terminal que permite la activación o desactivación de la memoria. En la figura 6.14. Se muestra la configuración de la circuitería electrónica para acceder a una memoria de datos. La terminal de escritura del microcontrolador (WR Pin 6 del puerto 3) se conecta a la terminal de la memoria (WE de la SRAM), y la terminal de lectura del microcontrolador (Rd pin 7 del puerto 3) se conecta a la terminal de la memoria SRAM OE. La terminal del microcontrolador que se encarga de la habilitación, que puede ser cualquiera según el tipo de direccionamiento dado en la programación del micro, en este caso la terminal 3 del puerto 1, va conectada a la terminal de selección de la memoria CS (Chip select).

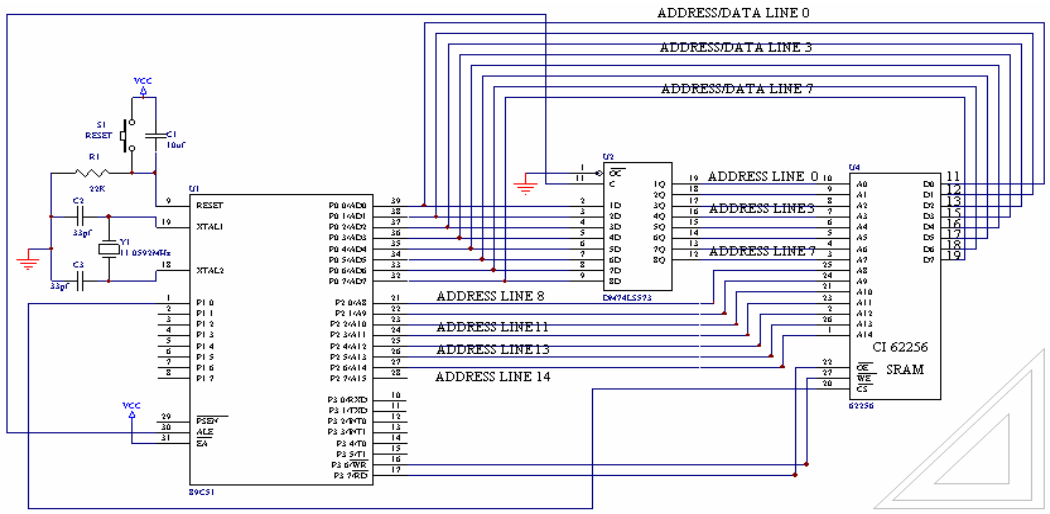


Figura 6.14. Configuración de acceso a una memoria de datos de 32 Kbytes CI 62 256 SRAM. Proyecto del sistema de adquisición de datos.

6.5 El microcontrolador AT 89C51 como microprocesador

Los cuatro puertos del AT89C51 son bidireccionales, es decir, permiten la lectura y escritura en el periférico correspondiente. Las salidas están "latcheadas", lo que permite mantener el dato indefinidamente hasta que se sobre-escriba la información original. Otra característica importante es que los puertos pueden ser utilizados como buses de direcciones, de datos y control, debido a estas características el microcontrolador puede trabajar como microprocesador, y adaptarse fácilmente al diseño del sistema de adquisición de datos de nuestro proyecto.

Los puertos P0 y P2 se pueden utilizar para acceder a la memoria externa del sistema. En estas condiciones, el puerto P0 se configura como salida del byte bajo del bus de direcciones (A0 – A7), de un bus que consta de 16 bytes que como ya se ha mencionado equivalente a una capacidad de direccionamiento de 64 K bytes, y multiplexado, el puerto P0, en el tiempo como bus de datos bidireccional (D0 a D7). El puerto P2 se configura como salida del byte alto del bus de direcciones (A_8 – A_{15}). De esta manera el microcontrolador se configura a modo de CPU de un sistema externo con unas capacidades de expansión definidas mediante software. En estas condiciones, sólo nos quedan dos puertos, el puerto 1 y el puerto 3, para el control de los periféricos y transmisión de información. Ver la figura 6.14.

La mayoría de las aplicaciones del μC AT89C51, usan chips adicionales para guardar código de programación, y datos. La expansión es directa, siempre se hace de la misma manera, tal como se muestra en la figura 6.14. El mayor problema de la expansión de memoria es la perdida de al menos dos puertos de 8 bits en función de bus externo. Al expandir externamente el microcontrolador, estamos limitados por la información multiplexada direcciones/datos en los ocho bits bajos (P0). Para mantener el byte bajo de direcciones hasta que la CPU haya trasferido los datos, debemos almacenarlo en el flanco de bajada del pulso ALE, el cual graba en el CI lacht 74LS573 la información de dirección con la finalidad de liberar esas líneas para que posteriormente puedan circular datos por estos mismos cables. El byte alto de direcciones permanece sin cambios en P2 durante el ciclo completo y podemos usarlo directamente para decodificación de direcciones.

Para leer la memoria externa de datos se utiliza la señal de control RD, función alternativa del bit 7 del puerto 3. El ciclo de lectura del bite de entrada es aceptado en el puerto P0, justo antes de que la señal de control RD que autoriza la lectura sea desactivada. Para la escritura en la memoria externa de datos, se utiliza la señal de control WR, función alternativa del bit 6 del puerto 3. En el ciclo de escritura, el byte de dato debe permanecer sobre el puerto P0, y después que la señal de control de escritura WR sea desactivada. Las líneas de control de lectura y escritura, RD y WR (pines 16 y 17), sólo son utilizadas cuando se ejecutan instrucciones de tipo MOVX. En general tanto para la lectura como para la escritura, cuando una dirección de 16 bits es utilizada, el byte alto de dirección sale por el puerto P2, donde permanecerá mientras dura el ciclo de lectura o escritura. En el caso concreto de nuestro proyecto sólo se manejara la señal de lectura RD a cada una de las memorias SRAM externas.

Cada uno de los dispositivos, (CI latch 74LS 573, CI 62256 y CI TC7109), cuentan con señales de habilitación, esto quiere decir, que si el microcontrolador no habilita al dispositivo, aunque este se encuentre conectado, no entregara ningún nivel sobre el bus

de datos, esto desde luego debe controlarse vía software, para de esta manera evitar que mas de un dispositivo se habilite en un momento determinado. Así varios dispositivos pueden compartir las mismas líneas y de este modo se optimiza el uso de las terminales del microcontrolador.

De acuerdo con lo anterior se implementa la siguiente tabla de verdad (Tabla 6.2), considerando el diagrama electrónico de la figura 6.15.

Tabla 6.2 Habilitación/des-habilitación de los periféricos del microcontrolador AT89C51. RD y WR, se generan automáticamente al ejecutarse la instrucción MOVX.

P1.0 CS RAM1	P1.1 CS RAM2	P1.2 CE TC7109	P1.3 LBEN TC7109	P1.4 HBEN TC7109	P3.6 WR	P3.7 RD	Función	Bus de datos
1	1	1	X	X	X	X	Ningún dispositivo seleccionado	HZ (Alta impedancia)
0	1	1	X	X	1	1	Ningún dispositivo seleccionado	HZ (Alta impedancia)
0	1	1	X	X	1	0	Lectura de la RAM #1	D7-D0
0	1	1	X	X	0	1	Escritura en la RAM #1	D7-D0
1	0	1	X	X	1	1	Ningún dispositivo seleccionado	HZ (Alta impedancia)
1	0	1	X	X	1	0	Lectura de la RAM #2	D7-D0
1	0	1	X	X	0	1	Escritura en la RAM #2	D7-D0
1	1	0	1	1	X	X	Ningún dispositivo seleccionado	HZ (Alta impedancia)
1	1	0	1	0	X	X	Lectura de la parte alta del ADC	POL,OR,B12-B9
1	1	0	0	1	X	X	Lectura de la parte baja del ADC	B8-B1

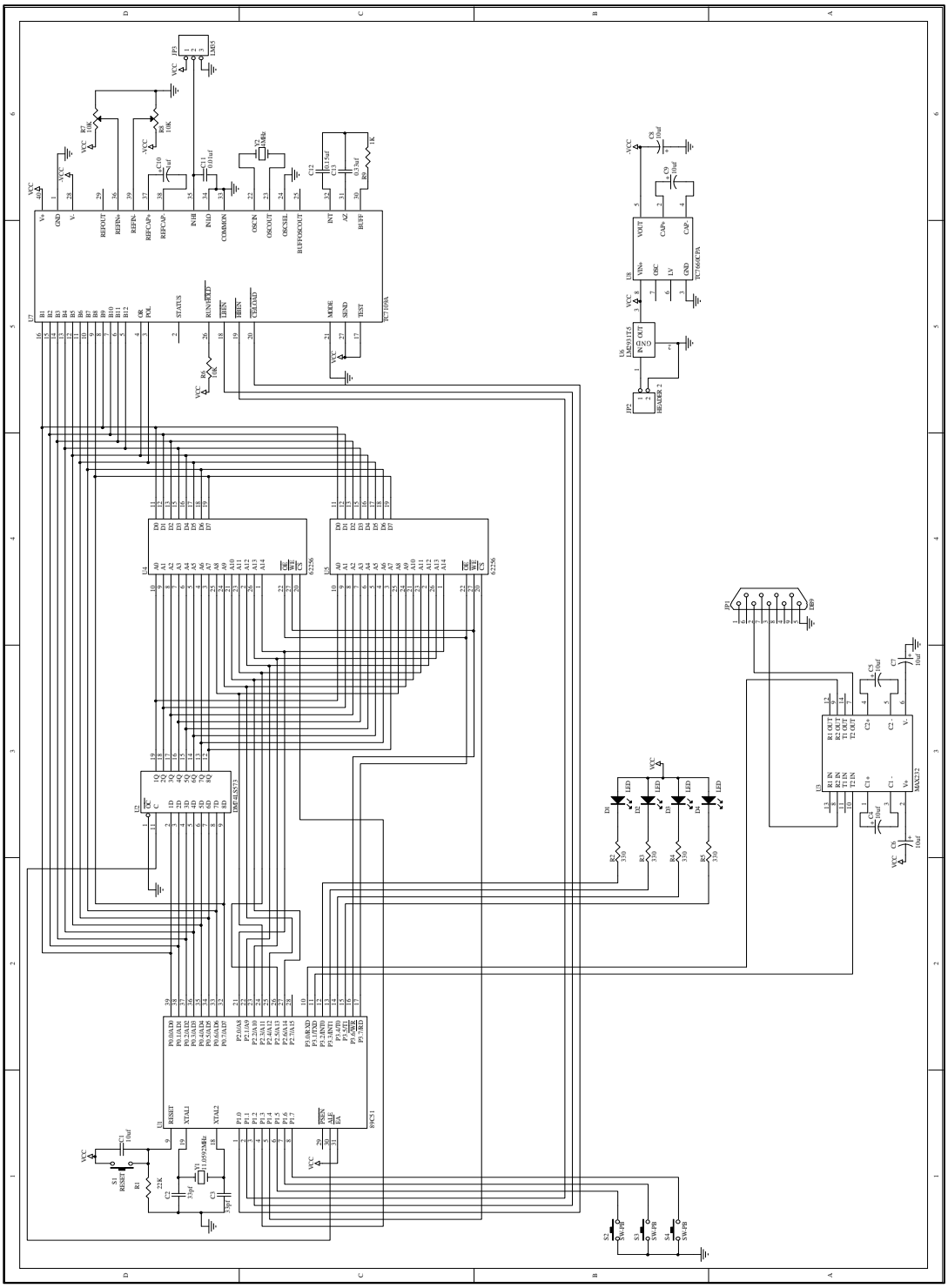


Figura 6.15 diagrama principal del dispositivo de registro de información para el sistema de adquisición de datos.

6.6 Sección de comunicaciones

En este proyecto se plantea la transmisión de información al puerto serial de una computadora personal, con información proveniente de la memoria SRAM externa que controla el AT 89C51 en el sistema de adquisición de datos, o bien directamente del sensor de temperatura cuando el sistema adquiere valores de temperatura en tiempo real, y para hacer todo esto necesitamos hacer uso del puerto serie del microcontrolador y su interconexión a la interfase del tipo RS-232C en una computadora personal.

En los pines P3.0 y P3.1 del microcontrolador AT 89c51, se encuentran como segunda función las señales de comunicación serial asíncrona (UART), esto implica que si se desea efectuar una comunicación serie con una computadora personal, este tipo de interfase es compatible, y solo hace falta adaptar los niveles lógicos TTL a niveles RS-232 (+10 y -10 V), para esto se agrego al proyecto presentado el circuito integrado MAX-232, el cual contiene los convertidores de TTL a RS-232 y viceversa, además de incluir un conversor de DC a DC, por lo que para su alimentación de tensión solo requiere de +5V.

El circuito MAX-232

El circuito integrado MAX232 (figura 6.16) cambia los niveles TTL a los del estándar de la PC RS-232 (cerca de +10 y -10 V) cuando se hace una transmisión desde la PC al microcontrolador, y cambia los niveles RS-232 a TTL (0V a +5 V) cuando se tiene una recepción. La configuración típica de este circuito se muestra en la figura 6.16.

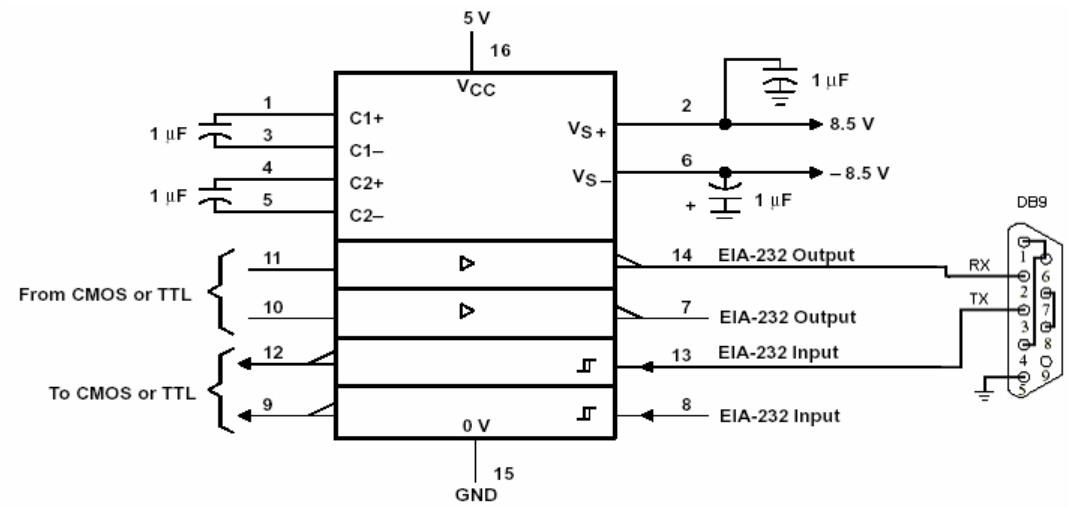


Figura 6.16.Configuración típica del circuito básico MAX-232.

El circuito integrado MAX-232 contiene dos drivers TTL a RS-232 y dos RS-232 a TTL. Este circuito integrado sólo necesita cuatro condensadores externos de 1uF para generar el voltaje RS-232 internamente, para el funcionamiento de este circuito se requiere nada mas que una fuente de +5 V.

Transmisor-Receptor-Asíncrono Universal UART

Las computadoras personales (PC) controlan el puerto serie mediante un circuito integrado específico, llamado UART (Transmisor-Receptor-Asíncrono Universal). El circuito integrado que convierte los datos de paralelo a serie y viceversa se llama UART ò Universal Asynchronous Receiver-Transmitter por sus siglas en inglés. La UART típica para una PC es el circuito integrado Intel 8251A, este circuito puede ser programado para realizar comunicaciones serie síncronas o asíncronas. Para controlar al puerto serie, la CPU emplea direcciones de puertos de E/S y líneas de interrupción (IRQ Interrupt Request line).

El UART incluye cuatro registros internos:

THR: Registro temporal de salida.

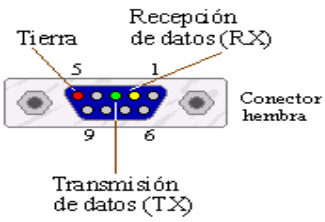
TSR: Registro de salida.

RDR: Registro de entrada.

RSR: Registro temporal de entrada.

Cada carácter a transmitir es almacenado en el registro THR. La UART añade los bits de start y stop. Luego copia todos los bits (datos, start y stop) al registro TSR. Para acabar el proceso los bits son enviados a la línea a través de la señal TxD. En la figura 6.17 se muestra la configuración de un conector estándar DB – 9 para una PC, así como también el tipo de señal en cada uno de sus pines.

Cada carácter recibido de de la línea RxD (línea de recepción) es almacenado en el registro RSR. Los bits de start y stop son eliminados y la UART escribe el carácter en el registro RDR. Para acabar el proceso el carácter es leído por la PC desde la unidad central de proceso.



a)

Configuración del conector serial DB-9	
Número de pin DB-9	**Nombre de la señal.**
1	Detección de portadora (DCD)
5	Tierra
3	Transmisión de datos (TX)
2	Recepción de datos (RX)
7	Solicitud de envío (RTS)
8	Autorización de envío (CTS)
6	Datos listos para enviar (DSR)
4	Terminal de datos lista (DTR)
9	Detector de llamada

b)

Figura 6.17. La figura 6.17 a) Muestra un conector DB-9 hembra de una computadora personal. En este conector al puerto serial solo ocuparemos las señales de los pines 2,3 y 5. En la figura 6.17 b) se muestra una tabla con el significado de las señales que maneja el puerto serial de la PC.

Las comunicaciones serie.

Los equipos de comunicaciones serie se pueden dividir entre simplex, half-duplex y full-duplex. Una comunicación serie simplex envía información en una sola dirección (por ejemplo una emisora de radio comercial). Half-duplex significa que los datos pueden ser enviados en ambas direcciones entre dos sistemas, pero en una sola dirección al mismo tiempo. En una transmisión full-duplex cada sistema puede enviar y recibir datos al mismo tiempo. El tipo de comunicación que se maneja en la transmisión y recepción de información entre nuestro sistema de adquisición de datos y la computadora es la comunicación serie half duplex, transmisión y recepción de datos en una sola dirección a la vez.

El puerto serie de la PC es un dispositivo asíncrono. En una transmisión asíncrona, un bit identifica su bit de comienzo y 1 o 2 bits identifican su final. Los bits de datos son enviados al receptor después del bit de start. Las palabras de 8 bits son utilizadas para corresponder a un byte. Por convención, el bit menos significativo de la palabra es enviado primero y el más significativo al final.

Nuestro sistema transmite datos en grupos de 8 bits a través del CI RS MAX 232 a unas velocidades de 9600 bits por segundo (9600 bauds). Después que los datos son transmitidos, le puede seguir un bit opcional de paridad como verificación de integridad de los datos, en la configuración de nuestro sistema este no se ha tomado en cuenta, ya que la distancia de transmisión y recepción es menor a 1 metro. Al finalizar la transferencia del byte se transmite un bit de stop.

Una vez que ha comenzado la transmisión de un dato, los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes de tiempo. Por eso se dice que el RS-232 es asíncrono por carácter y síncrono por bit.

El protocolo de transmisión de datos en serie, contempla lo siguiente:

El primer bit que se transmite es el bit de inicio (Start bit), el cual tiene la característica de ser siempre un “0” lógico. Posteriormente los bits que le siguen son los 8 bits de datos, comenzando con el menos significativo “D0”. La transmisión termina con el envió de un último bit, denominado bit de paro Stop que siempre será un “1” lógico. La figura 6.18 muestra la secuencia de transmisión de un byte sincronizada con la frecuencia de reloj de quien envía la información.

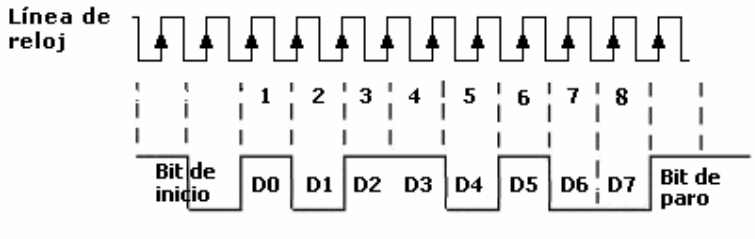


Figura 6.18 Transmisión en forma serial de un byte sincronizado con la secuencia de reloj del dispositivo que envía la información. Adicionales al byte son enviados dos bits de inicio y de paro

El microcontrolador AT 89C51, cuenta con 4 modos de transmisión de datos en serie, de los cuales utilizaremos, el modo 1, es decir, con el "Baud Rate" variable dependiendo del valor almacenado en TH1, con un cristal de reloj igual a 11.0592 MHz.

El Timer 1 es usado para generar la frecuencia de transmisión /Recepción de datos en serie, cuando el puerto es programado para trabajar en el modo 1 ó 3. La frecuencia de transmisión es obtenida a partir del valor almacenado en TH1 y el valor de SMOD mediante la siguiente ecuación:

Baud Rate = 2SMOD x Frec. Del Oscilador
32 12 x [256 – (TH1)]

El valor 2SMOD / 32, es debido a los circuitos divisores de frecuencia que se encuentran en la etapa de control del Puerto Serie, los cuales dividen entre 16 ó32 dependiendo del bit 7 (SMOD) del registro de control PCON y la frecuencia que nos proporciona la salida del Timer 1 (overflow).
El valor 12 que divide a la frecuencia del oscilador proviene del divisor, que se encuentra en la etapa de control del Timer 1, cuando éste es utilizado como temporizador.

NOTA: Cabe recalcar que el valor que se almacena en TH1 es el valor negativo de la cuenta que se desea, debido a que, el contador se incrementa cada vez que un pulso es detectado, de ahí que en la ecuación se representa como 256-(TH1).

La interrupción del Timer 1 en éste caso no tendría mucha aplicación por lo que se podría deshabilitar.

El Timer 1 actúa en modo 2, es decir en modo recargable, el valor de conteo se encuentra fijo en el registro TH1, el cual se recarga cada vez que existe un overflow.
La figura 6.7 muestra una tabla de valores de TH1, para generar el Baud Rate, tomando en cuenta la frecuencia del oscilador.

6.7 Despliegue de la información.

En muchos sistemas, se efectúa la recopilación de datos a intervalos regulares de tiempo o en una secuencia particular. En este caso el sistema de adquisición de datos puede llevar a cabo esta función mediante su programación desde una computadora personal. El despliegue de la información se muestra en un menú amigable, mediante el cual se puede configurar a que intervalo de muestreo queremos que el Sistema de Adquisición de Datos trabaje. Con base en la explicación dada en la figura. 6.7 para desplegar la información recopilada el microcontrolador entrega datos en forma binaria en los registros temporales del puerto serie de la PC, luego esta información en hexadecimal se puede leer desde la programación en C++ builder, para ello desde este software dividimos el valor leído en el puerto serie entre $FA0_{hex}$ que representa, dada la ecuación 6.5 la mínima resolución del convertidor analógico digital. Ecuación 6.12.

$$Factor\ \ de\ \ escala = V_{\mathrm{Re}sol(ADC)} = 0.25mV \Rightarrow \frac{1}{V_{\mathrm{Re}sol(ADC)}} = 4000_{DEC} = FA0_{HEX} \ldots\ldots\ldots 6.12$$

Lo que implica que si dividimos de la ecuación 6.5 el valor de 980 ($3D4_{HEX}$) cuentas por 4000 ($FA0_{HEX}$), obtenemos el valor de entrada de 0.245 V que multiplicado por 100 equivale a 24.5 °C

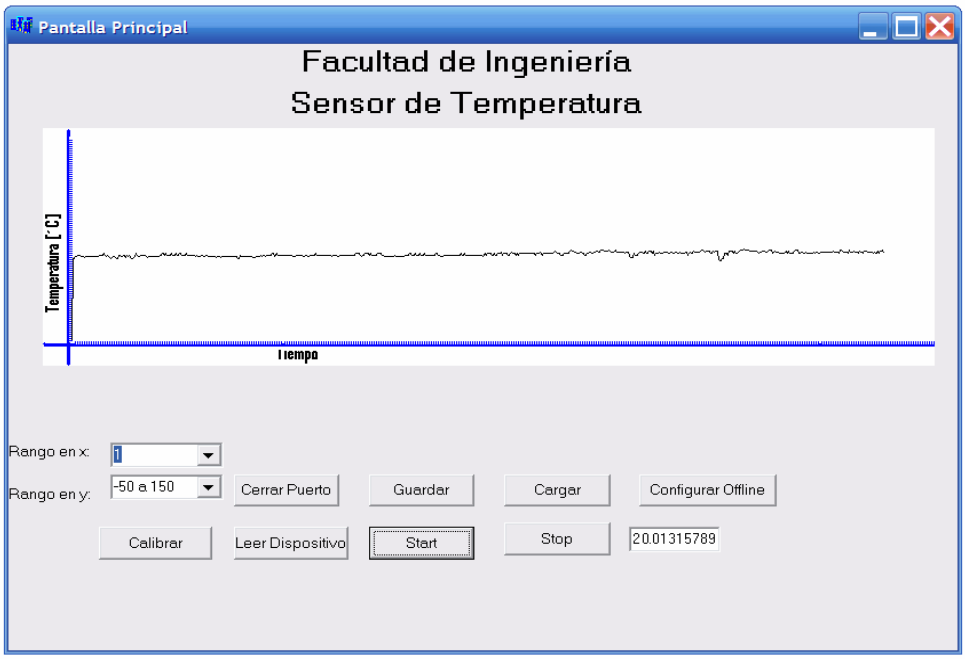


Figura 6.19 Pantalla principal del Sistema de Adquisición de Datos.

En la figura 6.19 se muestra la pantalla principal del sistema de adquisición de datos, desarrollado en este trabajo. Una de las características importantes de este recuadro, es poder configurar el rango de tiempo en X y el rango de temperatura en Y que se esta muestreando, también podemos guardar la información en un archivo de block de notas o leer información guardada en la memoria RAM del dispositivo. La figura 6.20 muestra el submenú de configuración del sistema.

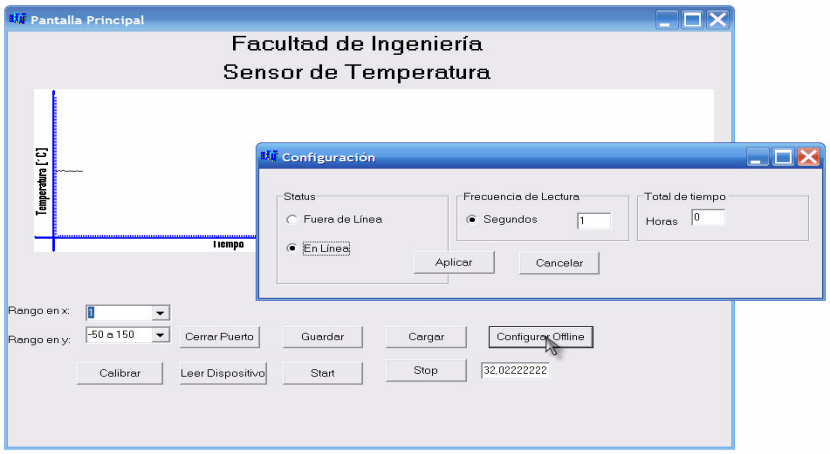


Figura 6.20 Submenú del sistema para su configuración en línea o fuera de ella

Software de programación C++ builder

Los lenguajes visuales orientados a eventos y con manejo de componentes dan al usuario que no cuenta con mucha experiencia en desarrollo, la posibilidad de construir sus propias aplicaciones utilizando interfaces gráficas sobre la base de ocurrencia de eventos.

Para soportar este tipo de desarrollo interactúan dos tipos de herramientas, una que permite realizar diseños gráficos y un lenguaje de alto nivel que permite codificar los eventos. Con dichas herramientas es posible desarrollar cualquier tipo de aplicaciones basadas en el entorno.

Basados en estas características de planeo el desarrollo de una interfase grafica para poder desplegar la información del sistema de adquisición de datos tratado en este proyecto, en la siguiente tabla se muestran las características más importantes que el programa ejecuta para la comunicación y lectura de la información registrada a través del puerto serial de la computadora.

Tabla 6.3 Comandos utilizados para la comunicación de la PC con el dispositivo.

Solicitud	Estado	Descripción	Respuesta
L	En Línea	Establece el estado del dispositivo en línea y envía una lectura por segundo.	L + 2 bytes A + 2bytes
O	En Línea	Establece el dispositivo fuera de línea	O
A + 2 bytes (HHMM)	Fuera de Línea	Configura el dispositivo para la cantidad de Horas y minutos que debe grabar en memoria estando fuera de línea	A + 2 bytes (HHMM)
R	No importa	Solicita una lectura del dispositivo.	V – vacío E – Error R + 2bytes (longitud total del paquete) + 2 bytes (dato) … datos

En el apéndice A se muestra el diagrama de flujo del programa en c++ builder, junto con el diagrama de flujo de la programación de microcontrolador

7. Resultados

En este capítulo, como parte de los objetivos y contenidos del presente trabajo, además de su desarrollo, se hace uso del método lineal de mínimos cuadrados, para obtener un análisis que represente el comportamiento del sistema de adquisición de datos de temperatura, de acuerdo con los parámetros de comparación de otros instrumentos de medición. Para lograr este fin, durante 5 horas, se realizaron pruebas en el sensado de la temperatura en un recipiente con agua, al que se le indujo un cambio gradual de temperatura, lo cual permitió obtener graficas del comportamiento de cada uno de los instrumentos de medición utilizados. El análisis de esta información, servirá como base, en la interpretación de los resultados de calibración, y para realizar una exposición adecuada en este trabajo.

7.1 El método de los mínimos cuadrados

El trabajo de laboratorio tiene como fruto una serie de datos experimentales representados, normalmente, por un conjunto discreto de N pares de datos $\{x_i, y_i\}$, siendo i un índice natural que varía de 1 a N. La manipulación de los datos tiende normalmente a uno de estos objetivos:

- bien a determinar una ley experimental
- bien a comprobar una ley experimental previamente supuesta

En cualquier caso esa ley se expresa en forma de una aplicación (función) $y = f(x)$ que exprese de forma fidedigna la correlación entre x e y. Entonces, debemos proveernos de un método que sea capaz de determinar la función que mejor describe a una serie de puntos experimentales, además de tener un criterio para conocer lo parecido o diferentes que son esos puntos respecto de la ley $f(x)$ propuesta.

7.2 Experimento de calibración.

En un recipiente con 1 litro de agua a temperatura ambiente, se sumerge el transductor de temperatura AD590 (sistema de adquisición de datos), un termómetro de mercurio, y la sonda de un termómetro digital, en seguida se baja la temperatura con 500gr de hielo, y se comienza durante cada minuto a tomar la lectura de cada uno de los tres instrumentos. Con los datos adquiridos, se realizarán graficas para después comparar los resultados, así como para observar, el comportamiento de cada instrumento con el tiempo, utilizando estas graficas se determinara la sensibilidad de nuestro transductor, para proceder a su ajuste final.

Observaciones

Antes de iniciar el experimento de caracterización, los tres sensores de temperatura (Termómetro de mercurio, Termómetro digital y el AD590) deben medir el mismo valor en grados centígrados, cuando están en equilibrio térmico a una temperatura constate, pero al hacer variar esta temperatura, también varía la respuesta de un sensor con respecto de otro.

Durante los primeros diez minutos:

Los hielos se comienzan a derretir en el agua, por lo tanto la temperatura sigue aumentando durante todo este tiempo transcurrido, lo cual es una constante en cada uno de los experimentos realizados. En el termómetro de mercurio la respuesta ha sido lenta en todas la pruebas durante la mayor parte del tiempo, ya que este tiene una baja sensibilidad a los cambios en la temperatura.

Al principio el AD590 registra variaciones en la lectura de la temperatura, debido a la no homogeneidad en el interior del recipiente, lo que es una característica que se repite en las pruebas por las razones ya mencionadas, ya que al inicio aun no se alcanza el equilibrio térmico en todo el sistema experimental, formado por el recipiente con agua y hielo. El termómetro digital también registra variaciones en la temperatura que arroja durante cada minuto, aunque al igual que el termómetro de mercurio y el AD590, registra el aumento de la temperatura de nuestro sistema conforme pasa el tiempo, esto se explica por la ley cero de la termodinámica, la cual dice que si dos cuerpos se encuentran en equilibrio térmico con un tercer cuerpo, están en equilibrio térmico entre si. Al sustituir el tercer cuerpo por un termómetro, la ley cero se pronuncia de la siguiente manera: "dos cuerpos están en equilibrio térmico si indican la misma lectura de temperatura, incluso si no se encuentran en contacto" [7], mientras que de la segunda ley de la termodinámica, podemos decir que la energía tiene cantidad y calidad, esto quiere decir que la energía tiende a degradarse desde un estado alto asta alcanzar el equilibrio con su entorno. En este experimento, lo primero que hemos hecho es disminuir la temperatura del agua que contiene el recipiente, sin embargo la temperatura alrededor de nuestro sistema (recipiente), es la que indica el estado de equilibrio de los alrededores, que en este caso es el laboratorio o lugar donde se realiza el experimento, por lo tanto, la temperatura de nuestro sistema tiene que aumentar después de cierto lapso de tiempo, para poder alcanzar el equilibrio térmico de los alrededores.

Cuando el hielo se ha descongelado totalmente, la temperatura del agua comienza a homogenizarse y aumenta más rápido. En este tiempo, aproximadamente en casi todas las pruebas, parece que la sensibilidad del termómetro de mercurio tiende a mejorar arriba de los 7 ^{0}C. Sin embargo la respuesta al cambio de temperatura conforme pasa el tiempo, sigue siendo menor en el Termómetro de mercurio que en el AD590 y el termómetro digital.

Después de una hora:

La temperatura del sistema aumenta considerablemente, junto con la sensibilidad del termómetro de mercurio, aunque la respuesta a las variaciones de temperatura, en el transductor AD590 y el termómetro digital se vuelve más lenta. En teoría con el tiempo el sistema ha de alcanzar el equilibrio térmico con respecto de los alrededores, por lo que los tres sensores de temperatura deben de tener la misma lectura.

7.2.2 Pruebas de caracterización para el transductor AD590, usado en el Sistema de adquisición de datos.

Experimento para obtener las curvas de respuesta de un sensor de temperatura con un CI AD590.

Descripción del experimento.

Las curvas de respuesta, son las curvas obtenidas de la temperatura vs el tiempo Figura 7.2. Este experimento consta de dos partes graficas. En la primera parte se induce un cambio en la temperatura del recipiente con agua., haciendo disminuir, mediante hielo, la temperatura del sistema experimental, y registrando su valor térmico cada minuto. Al pasar el tiempo la temperatura tiende aumentar para alcanzar el equilibrio térmico de los alrededores. Entonces se registran los cambios de temperatura en el sistema, y desde el principio la temperatura ambiente del lugar donde se realiza el experimento. Se dispone de un recipiente con agua a temperatura ambiente, un termómetro de mercurio (TM marca Pirex, con rango de -20 ^{o}C a 50 ^{o}C), un termómetro digital (de un multímetro de la marca Mastech) y el sensor con el CI AD590. Los tres termómetros deben de registrar teóricamente la misma temperatura. Una vez que se ha hecho disminuir la temperatura del agua, agregando hielo al recipiente, se registra la temperatura cada minuto hasta aproximadamente dos horas y media para una prueba, tiempo suficiente para que el sistema comience a alcanzar el equilibrio térmico. Así registramos el aumento de la temperatura y la respuesta de cada uno de nuestros sensores Figura 7.1.

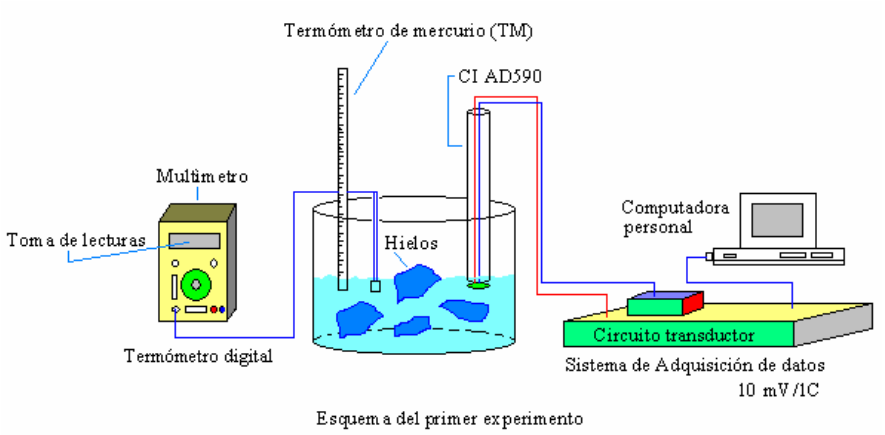


Figura 7.1 Esquema del experimento de caracterización. del sistema de adquisición de datos

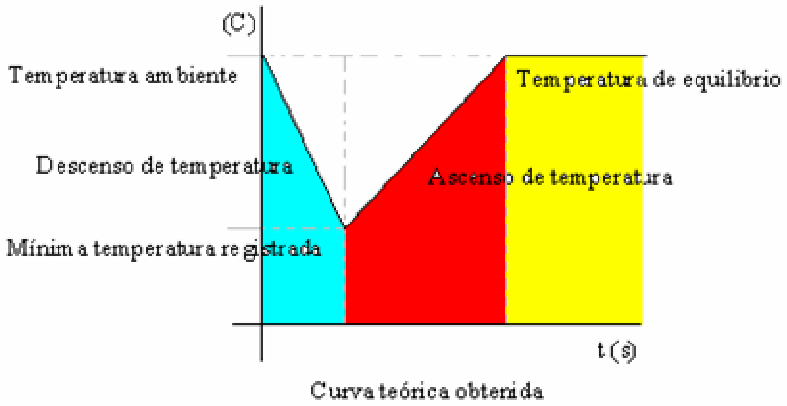


Figura 7.2 Curva de respuesta del experimento de caracterización (T[^{0}C] vs t [min])

Tabla 7.1 Datos para las temperaturas registradas en el experimento 1

Prueba 01. 1er. Experimento Gráfica 1

Lecturas Registradas cada minuto	Termómetro de mercurio T.M. (C)	Circuito sensor AD 590 (C)	Termómetro digital	Lecturas Registradas cada minuto	Termómetro de mercurio T.M. (C)	Circuito sensor AD 590 (C)	Termómetro digital
Temperatura ambiente	21	21.4	21.31				
0	21	21.4	21.31	76	11	8.08	7.985
1	11	11.3	11.551	77	11.5	8.32	8.021
2	10	9.2	10.225	78	11.5	8.5	8.34
3	9	7.31	9.56	79	12	8.72	8.356
4	8	4.27	4.468	80	12	8.93	8.72
5	7	2.19	2.001	81	12	9.1	8.952
6	7	0.9	0.6	82	12	9.34	9.01
7	6	0	0.5	83	12	9.54	9.32
8	6	-0.71	-0.689	84	12.5	9.73	9.561
9	6	-0.86	-0.7	85	13	9.92	9.861
10	6	-0.69	-0.69	86	13	10.1	9.921
11	6	-0.44	-0.678	87	13	10.3	10.11
12	6	-0.28	-0.682	88	13	10.46	10.321
13	6	-0.92	-0.782	89	13	10.6	10.561
14	6	-1.36	-0.756	90	13	10.8	10.781
15	6	-0.45	-0.75	91	13	11	10.951
16	6	-0.35	-0.689	92	13.5	11.16	11.02
17	6.5	0.08	-0.6	93	13.5	11.38	11.121
18	7	0.57	-0.325	94	14	11.55	11.231
19	7	0.85	0.56	95	14	11.7	11.461
20	7	0.98	0.899	96	14	11.92	11.741
21	7	0.97	0.923	97	14	12.04	11.984
22	7	0.9	0.911	98	14	12.19	12.03
23	7	0.83	0.978	99	14	12.35	12.21
24	7	0.85	0.991	100	14.5	12.5	12.345
25	7	0.9	1	101	14.5	12.66	12.41
26	7	1.02	1.005	102	14.5	12.84	12.56
27	7	1.08	1	103	15	13	12.78
28	7	1.14	1.11	104	15	13.24	12.894
29	7	1.22	1.18	105	15	13.34	13.02
30	7	1.21	1.22	106	15	13.5	13.2
31	7	1.23	1.235	107	15	13.65	13.452
32	7	1.26	1.24	108	15	13.78	13.785
33	7	1.33	1.287	109	15	13.93	13.891
34	7	1.39	1.302	110	15.5	14.05	13.945
35	7	1.39	1.301	111	16	14.02	14
36	7	1.26	1.321	112	16	14.37	14.01
37	7	1.2	1.325	113	16	14.51	14.301
38	7	1.27	1.33	114	16	14.68	14.378
39	7	1.32	1.334	115	16	14.82	14.412
40	7	1.4	1.341	116	16	14.93	14.874
41	7	1.48	1.342	117	16	15.1	14.952
42	7	1.6	1.345	118	16	15.24	14.99
43	7	1.6	1.35	119	16	15.35	15
44	7	1.34	1.352	120	16.5	15.4	15.321
45	7	0.99	1.356	121	16.5	15.56	15.368

46	7	0.96	1.468	122	16.5	15.52	15.456
47	7	1.02	1.476	123	16.5	15.69	15.52
48	7	1.32	1.521	124	17	15.75	15.623
49	7	1.78	1.578	125	17	15.83	15.7
50	7	2.05	1.621	126	17	15.94	15.79
51	7.5	2.32	1.657	127	17.5	16.12	15.891
52	8	2.64	1.75	128	17.5	16.25	15.963
53	8	2.73	1.83	129	17.5	16.32	16.21
54	8	2.86	2	130	17.5	16.55	16.312
55	8	2.9	2.42	131	17.5	16.68	16.325
56	8	2.12	2.512	132	18	16.76	16.458
57	8.5	3.3	2.831	133	18	16.89	16.765
58	8.5	3.44	2.952	134	18	16.92	16.89
59	8.5	3.58	3.215	135	18	17	16.911
60	9	3.7	3.301	136	18.5	17.21	17.3
61	9	3.84	3.33	137	19	17.37	17.31
62	9	4	3.561	138	19	17.44	17.41
63	9	4.18	3.781	139	19	17.65	17.56
64	9	4.5	3.981	140	19	17.77	17.6
65	9	4.88	4.651	141	19	17.89	17.78
66	9	5.1	4.781	142	19	17.96	17.89
67	10	5.6	5.23	143	19	18.02	17.985
68	10	5.9	5.647	144	19	18.26	18.23
69	10	6.23	5.94	145	19.5	18.38	18.32
70	10	6.41	6.32	146	20	18.52	18.46
71	10.5	6.92	6.598	147	20	18.67	18.589
72	11	7.17	6.981	148	20	18.78	18.631
73	11	7.4	7.32	149	20	18.82	18.721
74	11	7.64	7.456	150	20	18.97	18.869
75	11	7.87	7.689				

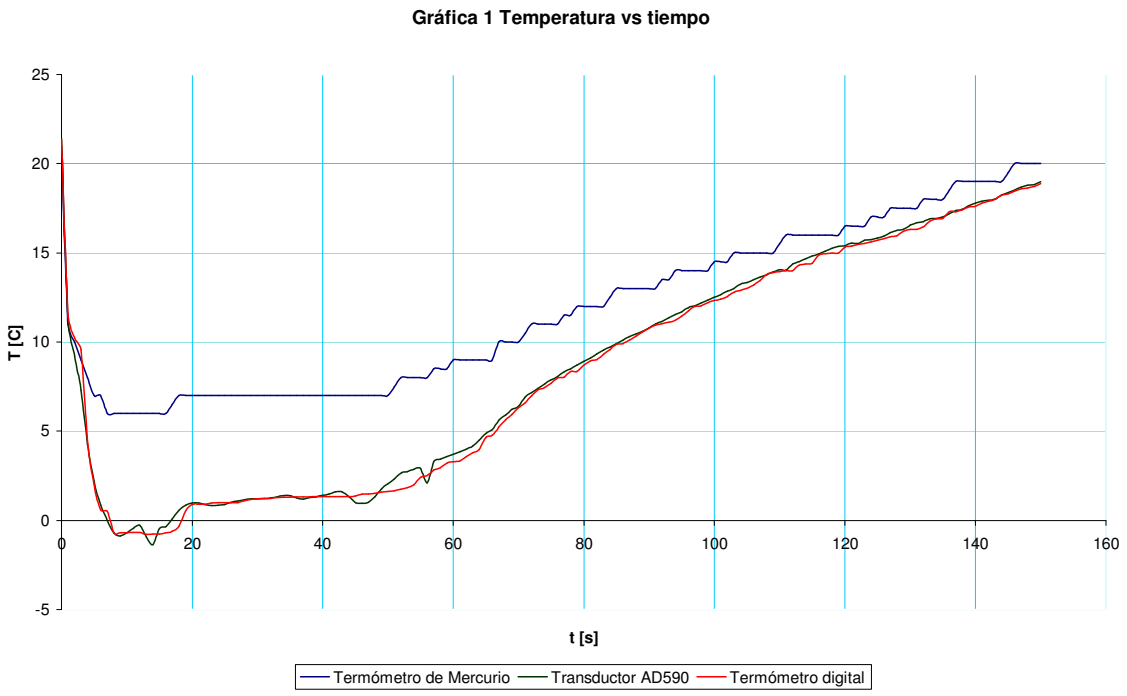


Figura 7.3 Comportamiento de la Temperatura vs tiempo para el termómetro de mercurio, el Sistema de Adquisición de datos (CI AD590) y el Termómetro digital (Tabla 7.1)

Tabla 7.2 Datos para las temperaturas registradas en el experimento 2

Prueba 02. 2o. Experimento Gráfica 2

Lecturas Registradas cada minuto	Termómetro de mercurio T.M. (C)	Circuito sensor AD 590 (C)	Termómetro digital	Lecturas Registradas cada minuto	Termómetro de mercurio T.M. (C)	Circuito sensor AD 590 (C)	Termómetro digital
Temperatura ambiente	21	21.4	21.31				
0	22	22.5	22.466	76	12	10.56	10.578
1	15	15.39	15.421	77	12	10.66	10.677
2	15	14.28	14.325	78	12.5	11.4	11.402
3	14.5	13.15	13.22	79	13	11.41	11.419
4	14	14	13.125	80	13	11.42	11.428
5	14	12.19	12.831	81	13	11.66	11.694
6	13.5	12.42	12.642	82	13	11.96	11.991
7	12	12.08	12.324	83	13.5	12.14	12.181
8	13	12	12.256	84	14	12.43	12.438
9	12.5	12	12.142	85	14	12.63	12.661
10	13	11.89	12.056	86	14	13	13.025
11	13	11.78	11.923	87	14	13.17	13.201
12	13	11.63	11.812	88	14.5	13.44	13.472
13	13	11.52	11.723	89	15	13.6	13.645
14	13	11.314	11.632	90	15	13.8	13.842
15	13	11.24	11.331	91	15	14	14.12
16	12	10.78	10.928	92	15	14.21	14.281
17	12	10.49	10.687	93	15.5	14.43	14.453
18	11	10.22	10.256	94	16	14.76	14.768
19	9.5	9.86	10.012	95	16	14.77	14.78
20	10	9.62	9.701	96	16	15.26	15.321
21	10	9.23	9.291	97	17	15.56	15.623
22	10	8.82	8.856	98	17	15.86	15.904
23	10	8.44	8.501	99	17	16.03	16.083
24	10.5	8.39	8.435	100	17	16.04	16.125
25	10.5	8.49	8.521	101	17	16.33	16.391
26	10.5	8.55	8.573	102	17	16.5	16.591
27	11	8.76	8.782	103	17.5	16.7	16.785
28	11	8.77	8.786	104	18	17.03	17.12
29	11	8.77	8.788	105	18	17.6	17.689
30	11	8.78	8.795	106	18	16.65	17.71
31	11	8.79	8.796	107	18	17.7	17.803
32	11	8.79	8.798	108	18.5	17.9	17.985
33	11	8.79	8.801	109	19	18	18.11
34	11	8.86	8.868	110	19	18.5	18.589
35	11	8.88	8.904	111	19.5	19.03	19.142
36	11	8.92	8.927	112	19.5	19.25	19.301
37	11	8.94	8.951	113	19.5	19.34	19.361
38	11	8.99	9.054	114	19.5	19.34	19.368
39	11	9.04	9.101	115	19.5	19.42	19.431
40	11	9.07	9.108	116	19.5	19.48	19.486
41	11	9.13	9.16	117	19.5	19.49	19.501
42	11	9.14	9.17	118	19.5	19.54	19.548
43	11	9.17	9.206	119	19.5	19.58	19.592
44	11	9.18	9.208	120	19.5	19.6	19.61

45	11	9.18	9.211
46	11	9.21	9.215
47	11	9.24	9.246
48	11	9.36	9.371
49	11	9.38	9.388
50	11	9.4	9.41
51	11	9.41	9.418
52	11	9.42	9.423
53	11	9.42	9.425
54	11	9.43	9.432
55	11	9.45	9.458
56	11	9.5	9.559
57	11	9.57	9.601
58	11	9.6	9.608
59	11	9.62	9.626
60	11	9.63	9.632
61	11	9.68	9.723
62	11	9.8	9.803
63	11.5	9.99	10.023
64	11.5	9.95	10.046
65	11.5	9.91	10.087
66	11.5	9.95	10.092
67	11.5	10	10.105
68	12	10.28	10.305
69	12	10.31	10.308
70	12	10.31	10.315
71	12	10.31	10.318
72	12	10.35	10.358
73	12	10.36	10.362
74	12	10.41	10.421
75	12	10.44	10.458
121	19.5	19.6	19.619
122	19.5	19.62	19.308
123	19.5	19.65	19.654
124	19.5	19.65	19.656
125	19.5	19.65	19.658
126	19.5	19.66	19.691
127	19.5	19.7	19.712
128	19.5	19.73	19.738
129	19.5	19.75	19.758
130	19.5	19.76	19.762
131	19.5	19.76	19.768
132	19.5	19.76	19.77
133	19.5	19.77	19.81
134	19.5	19.82	19.829
135	19.5	19.84	19.85
136	19.5	19.86	19.862
137	20	19.86	19.864
138	20	19.86	19.868
139	20	19.87	19.875
140	20	19.9	20.045
141	20	19.88	20.048
142	20	19.91	20.065
143	20	19.95	20.07
144	20	19.9	20.073
145	20	19.99	20.086
146	20	20	20.095
147	20	19.98	20.12
148	20	20.01	20.125
149	20	20.1	20.162
150	20	20.18	20.2

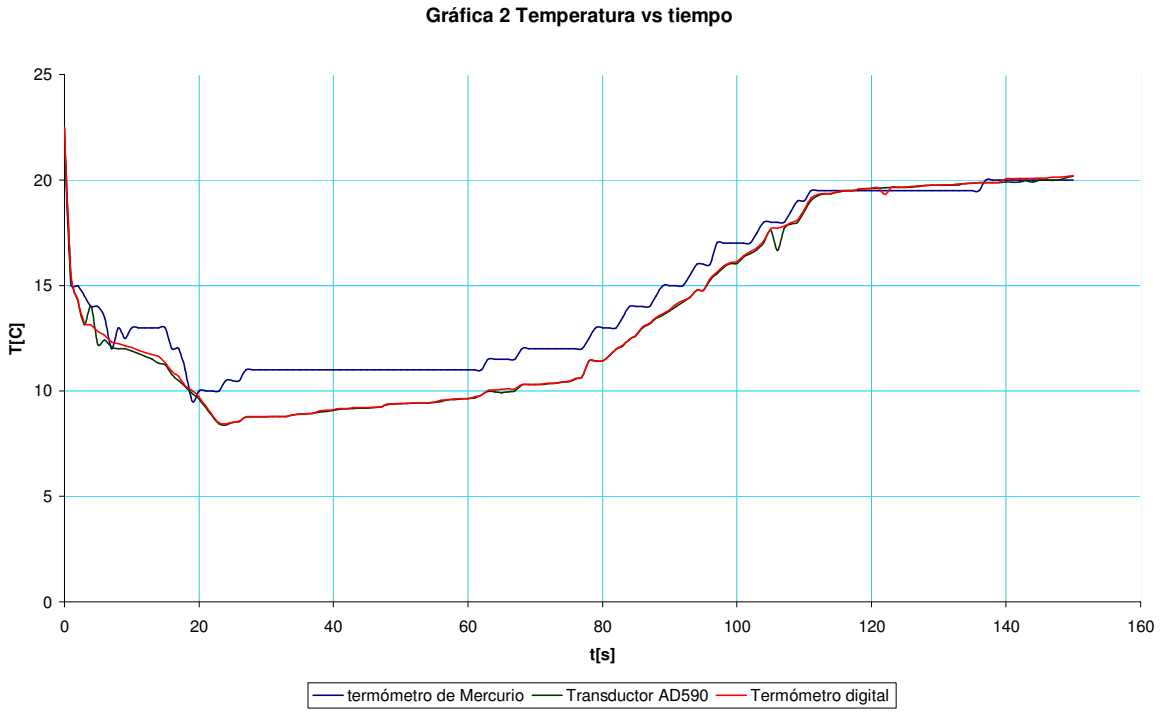


Figura 7.4 Comportamiento de la Temperatura vs tiempo para el termómetro de mercurio, el Sistema de Adquisición de datos (CI AD590) y el Termómetro digital (Tabla 7.2)

7.3 Análisis de regresión lineal.

Debido a los errores indeterminados del proceso de medida, no todos los datos experimentales se sitúan exactamente sobre la función de calibración (supondremos que es una línea recta), por tanto debe tratarse de obtener la "recta" que, a partir de los valores experimentales "represente mejor a la función de calibración".

Con tal finalidad, se utiliza una técnica estadística denominada análisis de regresión. En nuestro caso sólo se considerará el proceso de regresión más simple: el método de mínimos cuadrados.

La ecuación es de la forma: $y = mx + b$

- Pendiente de la línea: $$m = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - \left(\sum x\right)^2}$$

- Abscisa en el origen: $$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{n\sum x^2 - \left(\sum x\right)^2}$$

Análisis de resultados de los experimentos de caracterización mostrados en la tabla 7.1

Experimento 1

Para el primer experimento y de acuerdo a sus condiciones, hacemos el análisis de regresión lineal comparando el funcionamiento del CI AD590 contra el termómetro de mercurio, así como del CI AD590 contra el Termómetro digital de un multímetro de la marca MASTECH.

Para hacer la regresión lineal, solo tomamos en cuenta los valores más significativos que arroja el experimento cuando el termómetro de mercurio ha alcanzado la temperatura más baja. Como se ha descrito anteriormente en este capítulo.

Datos de regresión lineal del CI AD590 vs el Termómetro de mercurio
Evaluación de datos a partir de $t_{minutos}$ = 16 de la tabla 7.1

AD590 (X) TM (Y1)

$\sum x$	$\sum y_1$	$\sum xy_1$	$\sum x^2$	$m_{Pendiente}$	$b_{abscisa\ origen}$
1215,15	1651	18672,01	16326,4391	0.5972	5.9982

Ecuación de la recta $y = mx + b$: $y = 0.5972x + 5.9982$

Desviación estándar

$$s = \sqrt{\frac{\sum\left(x_i - \bar{x}\right)^2}{n-1}}$$

$$s = \sqrt{30.2132} = 5.4967$$

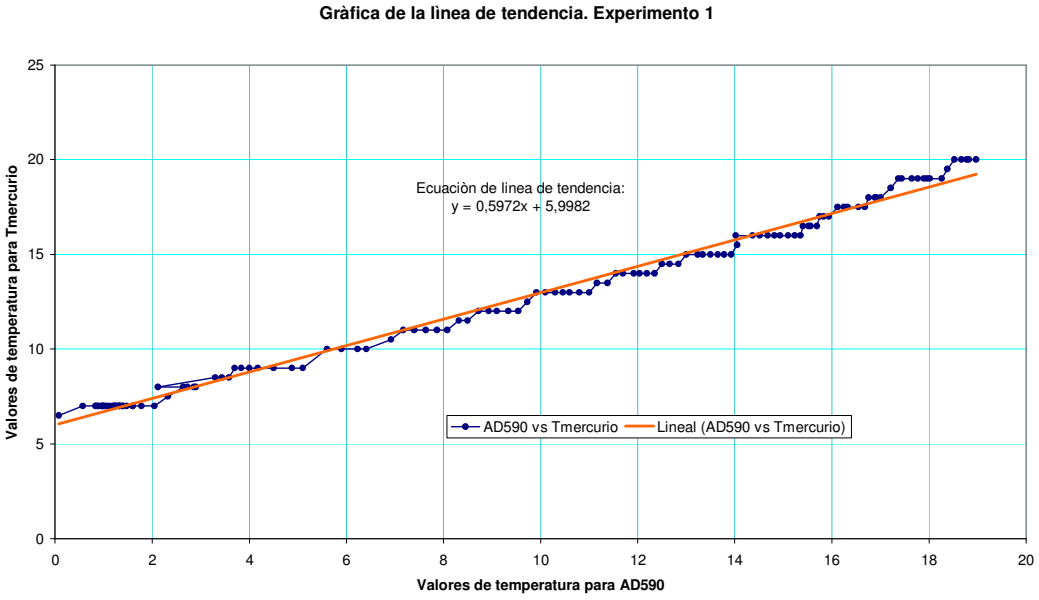


Figura 7.5 Grafica de regresión lineal para el primer experimento AD590 vs Termómetro de mercurio.

Datos de regresión lineal del CI AD590 vs el Termómetro digital de un multímetro Marca MASTECH
Evaluación de datos a partir de $t_{minutos}$ = 16 de la tabla 7.1

AD590 (X) TDigital (Y_2)

$\sum x$	$\sum y_2$	$\sum xy_2$	$\sum x^2$	$m_{Pendiente}$	$b_{abscisa\ \ origen}$
1215,15	1191,937	16120,9285	16326,4391	0.9650	0,05356

Ecuación de la recta $y = mx + b$: $y = 0.9650x + 0.05256$

Desviación estándar

$$s = \sqrt{\frac{\sum\left(x_i - \bar{x}\right)^2}{n-1}}$$

$$s = \sqrt{40.0556} = 6.3289$$

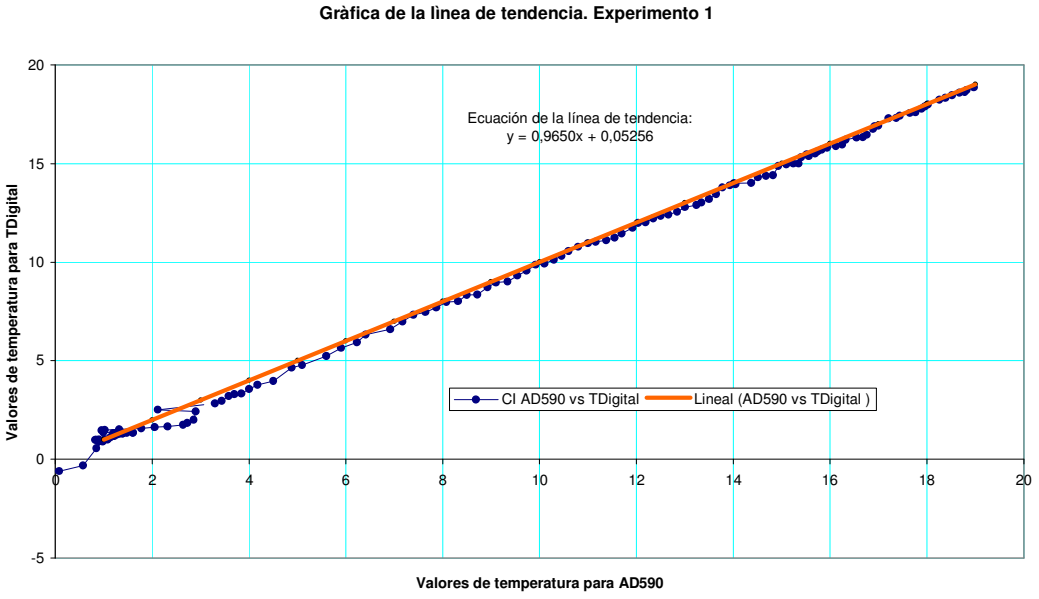


Figura 7.6 Grafica de regresión lineal para el primer experimento AD590 vs Termómetro Digital.

Análisis de resultados de los experimentos de caracterización mostrados en la tabla 7.2

Experimento 2.

Datos de regresión lineal del CI AD590 vs el Termómetro de mercurio
Evaluación de datos a partir de $t_{minutos}$ = 24 de la tabla 7.2

AD590 (X)	TM (Y1)				
$\sum x$	$\sum y_1$	$\sum xy_1$	$\sum x^2$	$m_{Pendiente}$	$b_{abscisa\ \ origen}$
1785.22	1912	29048.115	27715.1456	0.8455	3.1456

Ecuación de la recta $y = mx + b$: $y = 0.8455x + 3.1456$

Desviación estándar

$$s = \sqrt{\frac{\sum\left(x_i - \bar{x}\right)^2}{n-1}}$$

$$s = \sqrt{15.2218} = 3.9015$$

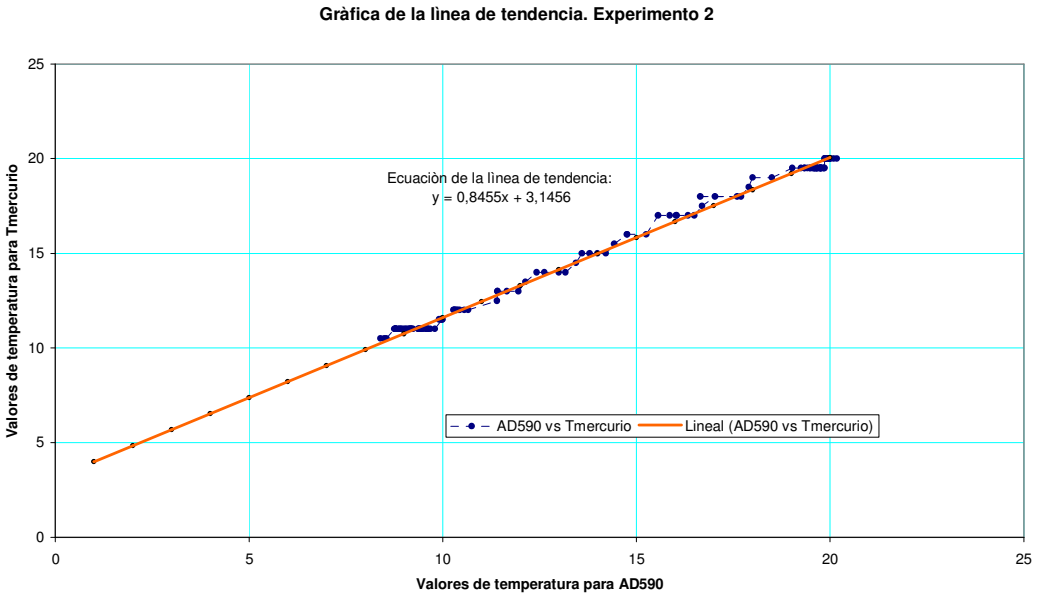


Figura 7.7 Grafica de regresión lineal para el segundo experimento AD590 vs Termómetro de mercurio.

Datos de regresión lineal del CI AD590 vs el Termómetro digital
Evaluación de datos a partir de $t_{minutos}$ = 24 de la tabla 7.2

AD590 (X) Digital (Y_2)

$\sum x$	$\sum y_2$	$\sum xy_2$	$\sum x^2$	$m_{Pendiente}$	$b_{abscisa\ \ origen}$
1785.22	1790.845	27801.5734	27715.1456	0.9625	0.0372

Ecuación de la recta $y = mx + b$: $y = 0.9625x + 0.0372$

Desviación estándar

$$s = \sqrt{\frac{\sum\left(x_i - \bar{x}\right)^2}{n-1}}$$

$$s = \sqrt{20.7632} = 4.5567$$

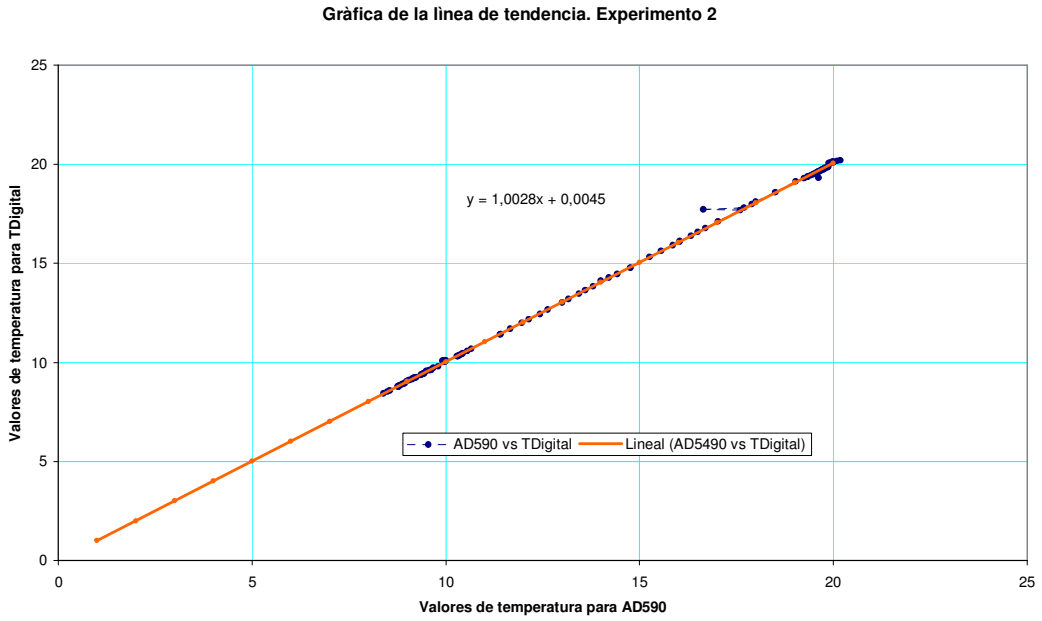


Figura 7.8 Grafica de regresión lineal para el segundo experimento AD590 vs Termómetro Digital.

8. Conclusiones.

El diseño de la instrumentación, es un trabajo que requiere suficiente tiempo, dedicación e investigación exhaustiva, para poder alcanzar estándares de calidad adecuados, estándares que estamos acostumbrados a observar en equipos de medición e instrumentación que actualmente compiten en el mercado, diseñados por compañías que cuentan con la experiencia de años de investigación y desarrollo. Sin embargo, esto no significa que sea imposible, para todo aquel que quiera adentrarse en este campo, presentar soluciones a determinados problemas específicos, como podrían ser aminorar costos. En este trabajo, se ha presentado un sistema de adquisición de datos para un sensor de temperatura, el cual es capaz de almacenar datos y desplegarlos de manera gráfica en una computadora personal, además de ser barato y confiable.

Este trabajo es una propuesta a una infinidad de sistemas de adquisición de datos que actualmente hay en el mercado, y si bien se ha podido aquí, alcanzar un nivel de calidad aceptable, comparable al de un equipo diseñado por un grupo de profesionales, con el trabajo de una sola persona, entonces es más posible que un grupo de ingenieros sea capaz de adentrarse poco a poco al campo de diseño electrónico, a niveles competitivos, pero para ello, alguien tiene que atreverse a invertir de manera seria al campo de la investigación y desarrollo de equipo electrónico.

La idea de este trabajo, como se mencionó anteriormente, era diseñar y construir un sistema de adquisición de datos, para ser usado, en un principio, con un sensor de temperatura, pero que también este sistema de adquisición de datos, fuera, así mismo funcional, y pudiera ser usado para medir otras variables como presión, densidad, salinidad, etc., para situaciones in situ o de contacto, en el que el sensor ha de estar directamente relacionado con la variable a medir, aunque no necesariamente, ya que mediante la implementación de la electrónica adecuada, también se podría con algún tipo de sensor remoto.

Este diseño puede servir de base para la construcción sistemas de adquisición de datos aun mejores, siempre y cuando se le dedique el suficiente tiempo para refinarlo, incluyendo la posibilidad de ser usado para su uso comercial, o en algunos laboratorios dentro de la UNAM.

Como es evidente en el diseño de un prototipo, como el presentado en este trabajo, en algunos parámetros como ruido y tiempo de respuesta a niveles de milisegundos, no fue posible acercarse a los más altos niveles de estándares de calidad de algunos otros dispositivos, Sin embargo, dentro de su revisión y perfeccionamiento, algunas posibles soluciones para mejorar el desempeño del diseño planteado en este trabajo, sería la implementación de filtros activos para atenuar los ruidos, y usar transductores de temperatura de mayor tiempo de respuesta. Cabe aclarar que, aunque, esto elevaría el costo del sistema, este no sería significativo, por lo que la propuesta presentada en este trabajo seguiría y sigue siendo una buena opción, en cuanto a calidad y costo.

Con el diseño presentado en este trabajo, tenemos la ventaja de que todos los componentes utilizados, se pueden encontrar en México, en el caso de alguna falla.

Además, debido la topología utilizada, es posible que este sistema sea de fácil reparación y durabilidad, de acuerdo a las especificaciones deseadas.

Algo muy importante que se aprende el desarrollar este tipo de proyectos, es que un diseño no necesariamente tiene que ser novedoso, y por más simple que sea una propuesta a la solución de un problema de ingeniería, conforme uno se adentra al diseño y desarrollo, en este caso del Sistema de Adquisición de Datos, resulta ser un verdadero reto al descubrir que pequeños detalles no previstos, tienen que ser solucionados para poder alcanzar nuestro objetivo de calidad y funcionalidad.

Durante la etapa del diseño se hizo claro, que al conocer los fundamentos teóricos de un instrumento electrónico e incluso haber construido uno como proyecto en alguna materia de la carrera, no nos prepara para poder realizar un verdadero diseño de un instrumento comercial. Incluso detalles que seguramente son conocidos por los diseñadores de sistemas de sensores comerciales, se vuelven un verdadero desafío cuando se esta diseñando por primera vez un producto.

Otro punto muy importante que hay que tomar en cuenta, para el desarrollo de este proyecto, es el manejo de microcontroladores diferentes al 68HC11, y del software de diseño electrónico como PROTEL, que no se ve en la licenciatura y que facilita mucho la simulación de los circuitos diseñados y más adelante el diseño de su circuito impreso. Ya que el manejo de este software para circuitos pequeños no es indispensable, pero es de gran ayuda conforme el volumen y complejidad del circuito aumenta.

Posteriormente se tiene contemplado el diseño de un sistema de adquisición de datos, para sensores remotos, el cual tendría como base el diseño propuesto en este trabajo, dando lugar a las mejoras necesarias, e incluso alcanzar un nivel de competitividad con algunos otros sistemas de adquisición de datos, que actualmente existen en el mercado.

Bibliografía

- [1] Holman J. P. Tr. Ing. José de Jesús Vega J. *Métodos experimentales para Ingenieros*. MC Graw Hill México 447pp.

- [2]. Norton Harry N. *Sensores y Analizadores*. Editorial Gustavo Gil S.A. 2ª edición. México, 588pp.

- [3] Mano Morris M. *Lógica Digital y Diseño de computadores*. Prentice Hall 1ª edición México 1982 636pp

- [4] Pallas Areny Ramón. *Sensores y acondicionadores de señal*. Editorial Marcombo BOIXAREU editores. Barcelona España 1994.

- [5] Pertence Junior Antonio *Amplificadores Operacionales y Filtros Activos* Editorial Mc Graw Hill México 295 pp.

- [6] Meter Aiken et al. Tr. Ing. Ruiz Faudon Sergio *Programación en C*. Prentice Hall 1ª edición México 1944.

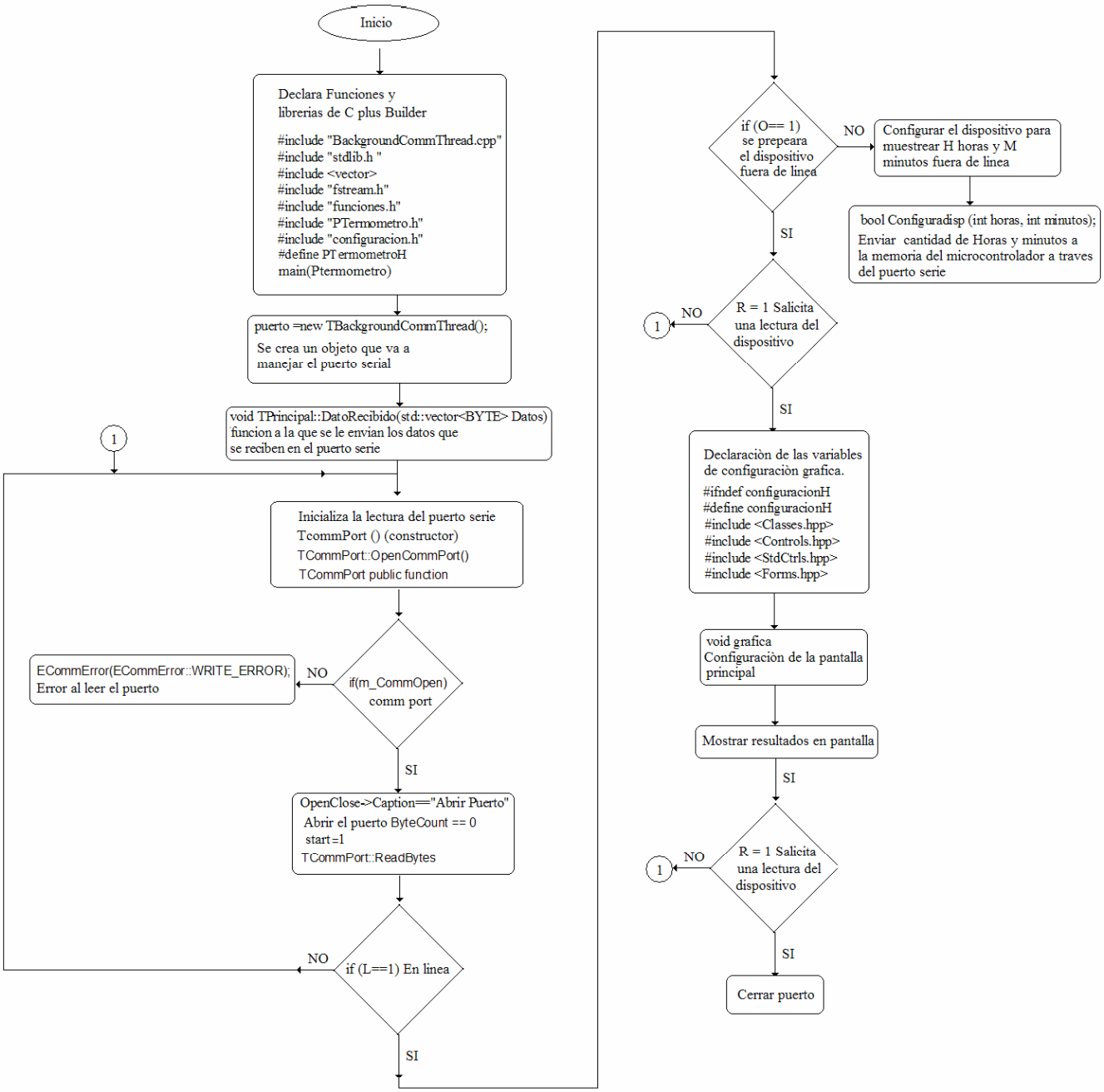
- [7] Yanus A Cengel, Michael A. Boles *Termodinámica Tomo I* Mc Graw Hill, México. 448 pp.

- [8] Sánchez Raya Manuel. *Introducción a los sistemas basados en microprocesador* Universidad de Huelva, España. 2000

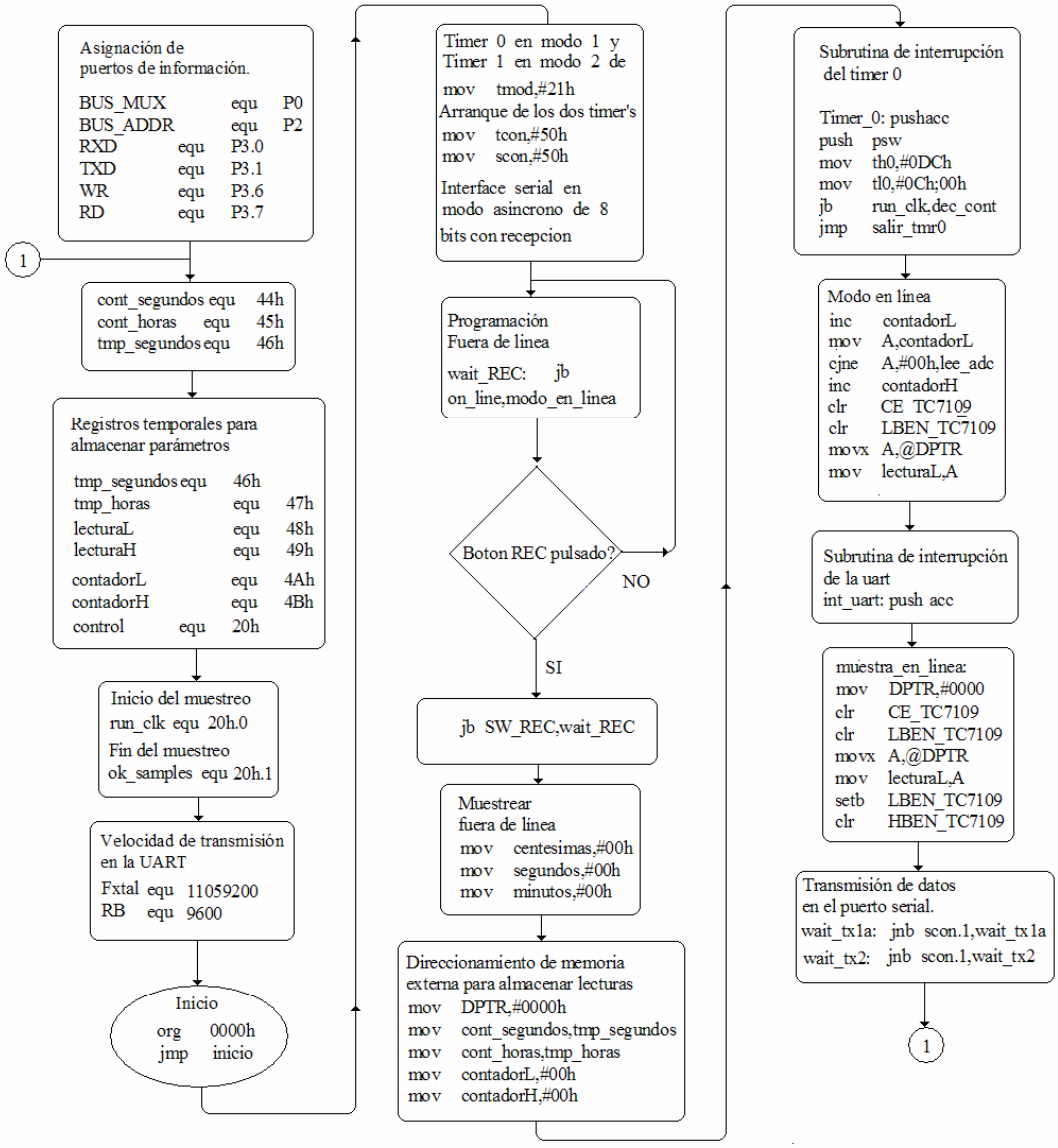
ANEXO A

DIAGRAMA DE FLUJO DE LOS PROGRAMAS DEL MICROCONTROLADOR Y DE C++ BUILDER PARA LA PC (PROGRAMAS FUENTE DE PROGRAMACIÓN ANEXOS EN CD)

El siguiente diagrama de flujo describe el programa C++ builder, del sistema de adquisición de datos. Código de programación anexo en CD



El siguiente diagrama de flujo describe la programación del microcontrolador, para el sistema de adquisición de datos. Código de programación anexo en CD



ANEXO B

HOJAS DE ESPECIFICACIONES

ANALOG DEVICES

Two-Terminal IC Temperature Transducer

AD590

FEATURES
Linear Current Output: 1 μA/K
Wide Range: –55°C to +150°C
Probe Compatible Ceramic Sensor Package
Two Terminal Device: Voltage In/Current Out
Laser Trimmed to ±0.5°C Calibration Accuracy (AD590M)

PIN DESIGNATIONS

AD590–SPECIFICATIONS (@ +25°C and V_S = +5 V unless otherwise noted)

Model	AD590J Min	AD590J Typ	AD590J Max	AD590K Min	AD590K Typ	AD590K Max	Units
ABSOLUTE MAXIMUM RATINGS							
Forward Voltage (E+ or E–)			+44			+44	Volts
Reverse Voltage (E+ to E–)			–20			–20	Volts
Breakdown Voltage (Case E+ or E–)			±200			±200	Volts
Rated Performance Temperature Range[1]	–55		+150	–55		+150	°C
Storage Temperature Range[1]	–65		+155	–65		+155	°C
Lead Temperature (Soldering, 10 sec)			+300			+300	°C
POWER SUPPLY							
Operating Voltage Range	+4		+30	+4		+30	Volts
OUTPUT							
Nominal Current Output @ +25°C (298.2K)		298.2			298.2		μA
Nominal Temperature Coefficient		1			1		μA/K
Calibration Error @ +25°C			**±5.0**			**±2.5**	°C
Absolute Error (Over Rated Performance Temperature Range)							
Without External Calibration Adjustment			**±10**			**±5.5**	°C
With +25°C Calibration Error Set to Zero			**±3.0**			**±2.0**	°C
Nonlinearity			**±1.5**			**±0.8**	°C
Repeatability[2]			±0.1			±0.1	°C
Long-Term Drift[3]			±0.1			±0.1	°C
Current Noise		40			40		pA/√Hz
Power Supply Rejection							
+4 V ≤ V_S ≤ +5 V		0.5			0.5		μA/V
+5 V ≤ V_S ≤ +15 V		0.2			0.2		μV/V
+15 V ≤ V_S ≤ +30 V		0.1			0.1		μA/V
Case Isolation to Either Lead		10^{10}			10^{10}		Ω
Effective Shunt Capacitance		100			100		pF
Electrical Turn-On Time		20			20		μs
Reverse Bias Leakage Current[4] (Reverse Voltage = 10 V)		10			10		pA
PACKAGE OPTIONS							
TO-52 (H-03A)		AD590JH			AD590KH		
Flatpack (F-2A)		AD590JF			AD590KF		

NOTES
[1]The AD590 has been used at –100°C and +200°C for short periods of measurement with no physical damage to the device. However, the absolute errors specified apply to only the rated performance temperature range.
[2]Maximum deviation between +25°C readings after temperature cycling between –55°C and +150°C; guaranteed not tested.
[3]Conditions: constant +5 V, constant +125°C; guaranteed, not tested.
[4]Leakage current doubles every 10°C.

Specifications subject to change without notice.

Specifications shown in boldface are tested on all production units at final electrical test. Results from those tests are used to calculate outgoing quality levels. All min and max specifications are guaranteed, although only those shown in boldface are tested on all production units.

12-BIT μP-Compatible Analog-To-Digital Converters

FEATURES

- **Zero-Integrator Cycle for Fast Recovery From Input Overloads**
- **Eliminates Cross -Talk in Multiplexed Systems**
- **12-Bit Plus Sign Integrating A/D Converter With Overrange Indication**
- **Sign Magnitude Coding Format**
- **True Differential Signal Input and Differential Reference Input**
- **Low Noise ..... 15μV_{P-P} Typ.**
- **Input Current ..... 1pA Typ.**
- **No Zero Adjustment Needed**
- **TTL-Compatible, Byte-Organized Tri-State Outputs**
- **UART Handshake Mode for Simple Serial Data Transmission**

ORDERING INFORMATION

PART CODE **TC7109X**

A or blank* ——

Package Code	Package	Temperature Range
CKW	44-PQFP	0°C to +70°C
CLW	44-Pin PLCC	0°C to +70°C
CPL	40-Pin Plastic DIP	0°C to +70°C
IJL	40-Pin CerDIP	– 25°C to +85°C

* The "A" version has a higher I_{OUT} on the digital lines.

GENERAL DESCRIPTION

The TC7109A is a 12-bit plus sign, CMOS low-power analog-to-digital converter (ADC). Only eight passive components and a crystal are required to form a complete dual-slope integrating ADC.

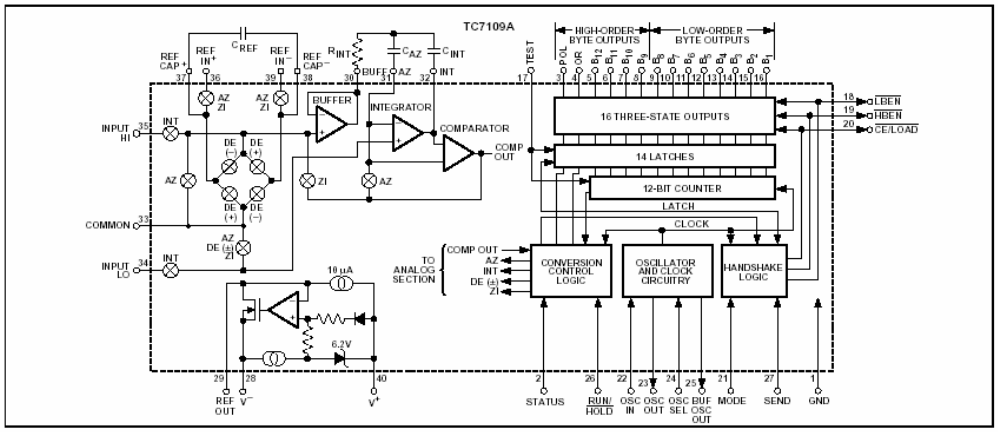
The improved V_{OH} source current TC7109A has features that make it an attractive per-channel alternative to analog multiplexing for many data acquisition applications. These features include typical input bias current of 1pA, drift of less than 1μV/°C, input noise typically 15μV_{P-P}, and auto-zero. True differential input and reference allow measurement of bridge-type transducers such as load cells, strain gauges, and temperature transducers.

The TC7109A provides a versatile digital interface. In the direct mode, chip select and HIGH/LOW byte enables control parallel bus interface. In the handshake mode, the TC7109A will operate with industry-standard UARTs in controlling serial data transmission — ideal for remote data logging. Control and monitoring of conversion timing is provided by the RUN/HOLD input and STATUS output.

For applications requiring more resolution, see the TC500, 15-bit plus sign ADC data sheet.

The TC7109A has improved overrange recovery performance and higher output drive capability than the original TC7109. All new (or existing) designs should specify the TC7109A wherever possible.

FUNCTIONAL BLOCK DIAGRAM



© 2001 Microchip Technology Inc.

ABSOLUTE MAXIMUM RATINGS*

Positive Supply Voltage (GND to V+) ..... +6.2V
Negative Supply voltage (GND to V–) ..... – 9V
Analog Input Voltage (Low to High) (Note 1)..... V+ to V–
Reference Input Voltage (Low to High (Note 1) .. V+ to V–
Digital Input Voltage (Pins 2–27) (Note 2).... GND – 0.3V
Power Dissipation, $T_A \leq 70°C$, (Note 3)
- CerDIP 2.29W
- Plastic DIP 1.23W
- PLCC 1.23W
- PQFP 1.00W

Operating Temperature Range
- Plastic Package (C).....0°C to +70°C
- Ceramic Package (I) ..... – 25°C to +85°C
- (M) ..... – 55°C to +125°C

Storage Temperature Range ..... – 65°C to +150°C
Lead Temperature (Soldering, 10 sec) ..... +300°C

*Static-sensitive device. Unused devices must be stored in conductive material. Protect devices from static discharge and static fields. Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specifications is not implied. Exposure to Absolute Maximum Rating Conditions for extended periods may affect device reliability.

NOTES:
1. Input voltages may exceed supply voltages if input current is limited to ±100μA.
2. Connecting any digital inputs or outputs to voltages greater than V+ or less than GND may cause destructive device latch-up. Therefore, it is recommended that inputs from sources other than the same power supply should not be applied to the TC7109A before its power supply is established. In multiple supply systems, the supply to the device should be activated first.
3. This limit refers to that of the package and will not occur during normal operation.

ELECTRICAL CHARACTERISTICS: All parameters with V+ = +5V, V– = –5V, GND = 0V, T_A = +25°C, unless otherwise indicated.

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
Analog						
	Overload Recovery Time (TC7109A)		—	0	1	Measurement Cycle
	Zero Input Reading	V_{IN} = 0V Full Scale = 409.6mV	-0000_8	$\pm0000_8$	$+0000_8$	Octal Reading
	Ratio Metric Reading	$V_{IN} = V_{REF}$ V_{REF} = 204.8mV	3777_8	3777_8 4000_8	4000_8	Octal Reading
NL	Nonlinearity (Max Deviation From Best Straight Line Fit)	Full Scale = 409.6mV to 2.048V Over Full Operating Temperature Range	–1	±0.2	+1	Count
	Roll-Over Error (Difference in Reading for Equal Positive and Negative Inputs Near (Full Scale)	Full Scale = 409.6mV to 2.048V Over Full Operating Temperature Range	–1	±0.02	+1	Count
CMRR	Input Common-Mode Rejection Ratio	V_{CM} ±1V, V_{IN} = 0V Full Scale = 409.6mV	—	50	—	μV/V
V_{CMR}	Common-Mode Voltage Range	Input High, Input Low, and Common Pins	V–+1.5	—	V+–1	V
e_N	Noise (P-P Value Not Exceeded 95% of Time)	V_{IN} = 0V Full Scale = 409.6mV	—	15	—	μV
I_{IN}	Leakage Current at Input	V_{IN}, All Packages: +25°C C Device: 0°C ≤ T_A ≤ +70°C I Device: – 25°C ≤ T_A ≤ +85°C M Device: – 55°C ≤ T_A ≤ +125°C	—	1 20 100 2	10 100 250 5	pA pA pA nA
TC_{ZS}	Zero Reading Drift	V_{IN} = 0V	—	0.2	1	μV/°C
TC_{FS}	Scale-Factor Temperature Coefficient	V_{IN} = 408.9mV = >7770_8 Reading, Ext Ref = 0ppm/°C	—	1	5	μV/°C
I^+	Supply Current (V+ to GND)	V_{IN} = 0V, Crystal Oscillator 3.58MHz Test Circuit	—	700	1500	μA
I_S	Supply Current (V+ to V–)	Pins 2–21, 25, 26, 27, 29 Open	—	700	1500	μA

 © 2001 Microchip Technology Inc.

ELECTRICAL CHARACTERISTICS (Cont.)

Symbol	Parameter	Test Conditions	Min	Typ	Max	Unit
V_{REF}	Ref Out Voltage	Referenced to V+, 25kΩ Between V+ and Ref Out	– 2.4	– 2.8	– 3.2	V
TC_{REF}	Ref Out Temperature Coefficient	25kΩ Between V+ and Ref Out 0°C ≤ T_A ≤ +70°C	—	80	—	ppm/°C
Digital						
V_{OH}	Output High Voltage TC7109A: I_{OUT} = 700µA	TC7109: I_{OUT} = 100µA Pins 3 – 16, 18, 19, 20	3.5	4.3	—	V
V_{OL}	Output Low Voltage	I_{OUT} = 1.6mA	—	0.2	0.4	V
	Output Leakage Current	Pins 3 – 16 High Impedance	—	±0.01	±1	µA
	Control I/O Pull-Up Current	Pins 18, 19, 20 V_{OUT} = V+– 3V Mode Input at GND	—	5	—	µA
	Control I/O Loading	$\overline{HBEN}$, Pin 19; $\overline{LBEN}$, Pin 18	—	—	50	pF
V_{IH}	Input High Voltage	Pins 18 – 21, 26, 27 Referenced to GND	2.5	—	—	V
V_{IL}	Input Low Voltage	Pins 18 – 21, 26, 27 Referenced to GND	—	—	1	V
	Input Pull-Up Current	Pins 26, 27; V_{OUT} = V+– 3V Pins 17, 24; V_{OUT} = V+– 3V	—	5 25	—	µA µA
	Input Pull-Down Current	Pin 21; V_{OUT} = GND = +3V	—	1	—	µA
	Oscillator Output Current, High	V_{OUT} = 2.5V	—	1	—	mA
	Oscillator Output Current, Low	V_{OUT} = 2.5V	—	1.5	—	mA
	Buffered Oscillator Output Current, High	V_{OUT} = 2.5V	—	2	—	mA
	Buffered Oscillator Output Current, Low	V_{OUT} = 2.5V	—	5	—	mA
t_W	Mode Input Pulse Width		60	—	—	nsec

HANDLING PRECAUTIONS: These devices are CMOS and must be handled correctly to prevent damage. Package and store only in conductive foam, antistatic tubes, or other conducting material. Use proper antistatic handling procedures. Do not connect in circuits under "power-on" conditions, as high transients may cause permanent damage.

National Semiconductor

August 2000

LF147/LF347
Wide Bandwidth Quad JFET Input Operational Amplifiers

General Description

The LF147 is a low cost, high speed quad JFET input operational amplifier with an internally trimmed input offset voltage (BI-FET II™ technology). The device requires a low supply current and yet maintains a large gain bandwidth product and a fast slew rate. In addition, well matched high voltage JFET input devices provide very low input bias and offset currents. The LF147 is pin compatible with the standard LM148. This feature allows designers to immediately upgrade the overall performance of existing LF148 and LM124 designs.

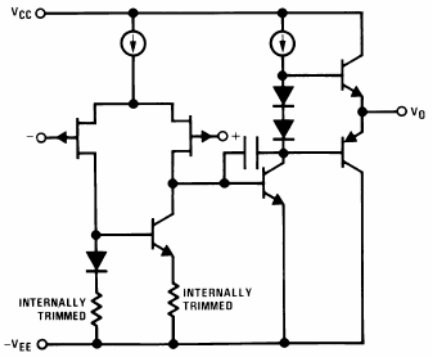
The LF147 may be used in applications such as high speed integrators, fast D/A converters, sample-and-hold circuits and many other circuits requiring low input offset voltage, low input bias current, high input impedance, high slew rate and wide bandwidth. The device has low noise and offset voltage drift.

Features

■ Internally trimmed offset voltage:	5 mV max
■ Low input bias current:	50 pA
■ Low input noise current:	0.01 pA/√Hz
■ Wide gain bandwidth:	4 MHz
■ High slew rate:	13 V/µs
■ Low supply current:	7.2 mA
■ High input impedance:	$10^{12}\Omega$
■ Low total harmonic distortion:	≤0.02%
■ Low 1/f noise corner:	50 Hz
■ Fast settling time to 0.01%:	2 µs

Simplified Schematic

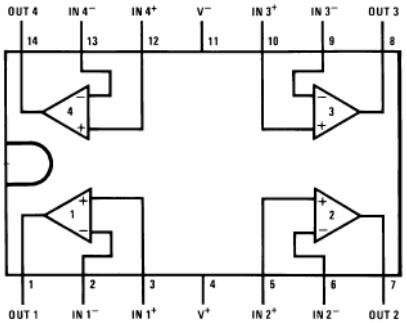
¼ Quad



DS005647-13

Connection Diagram

Dual-In-Line Package



DS005647-1

Note 1: LF147 available as per JM38510/11906.

Top View
Order Number LF147J, LF147J-SMD, LF347M, LF347BN, LF347N, LF147J/883, or JL147 BCA (Note 1)
See NS Package Number J14A, M14A or N14A

BI-FET II™ is a trademark of National Semiconductor Corporation.

© 2000 National Semiconductor Corporation DS005647

Absolute Maximum Ratings (Note 2)

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

	LF147	LF347B/LF347
Supply Voltage	±22V	±18V
Differential Input Voltage	±38V	±30V
Input Voltage Range (Note 3)	±19V	±15V
Output Short Circuit Duration (Note 4)	Continuous	Continuous
Power Dissipation (Notes 5, 11)	900 mW	1000 mW
T_j max	150˚C	150˚C
θ_{jA}		
Ceramic DIP (J) Package		70˚C/W
Plastic DIP (N) Package		75˚C/W
Surface Mount Narrow (M)		100˚C/W
Surface Mount Wide (WM)		85˚C/W
Operating Temperature Range	(Note 6)	(Note 6)
Storage Temperature Range	$-65˚C \le T_A \le 150˚C$	
Lead Temperature (Soldering, 10 sec.)	260˚C	260˚C
Soldering Information		
Dual-In-Line Package		
Soldering (10 seconds)		260˚C
Small Outline Package		
Vapor Phase (60 seconds)		215˚C
Infrared (15 seconds)		220˚C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

ESD Tolerance (Note 12)	900V

DC Electrical Characteristics (Note 7)

Symbol	Parameter	Conditions	LF147			LF347B			LF347			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
V_{OS}	Input Offset Voltage	R_S=10 kΩ, T_A=25˚C		1	5		3	5		5	10	mV
		Over Temperature			8			7			13	mV
$\Delta V_{OS}/\Delta T$	Average TC of Input Offset Voltage	R_S=10 kΩ		10			10			10		μV/˚C
I_{OS}	Input Offset Current	T_j=25˚C, (Notes 7, 8)		25	100		25	100		25	100	pA
		Over Temperature			25			4			4	nA
I_B	Input Bias Current	T_j=25˚C, (Notes 7, 8)		50	200		50	200		50	200	pA
		Over Temperature			50			8			8	nA
R_{IN}	Input Resistance	T_j=25˚C		10^{12}			10^{12}			10^{12}		Ω
A_{VOL}	Large Signal Voltage Gain	V_S=±15V, T_A=25˚C	50	100		50	100		25	100		V/mV
		V_O=±10V, R_L=2 kΩ										
		Over Temperature	25			25			15			V/mV
V_O	Output Voltage Swing	V_S=±15V, R_L=10 kΩ	±12	±13.5		±12	±13.5		±12	±13.5		V
V_{CM}	Input Common-Mode Voltage Range	V_S=±15V	±11	+15		±11	+15		±11	+15		V
				−12			−12			−12		V
CMRR	Common-Mode Rejection Ratio	R_S≤10 kΩ	80	100		80	100		70	100		dB
PSRR	Supply Voltage Rejection Ratio	(Note 9)	80	100		80	100		70	100		dB
I_S	Supply Current			7.2	11		7.2	11		7.2	11	mA

AC Electrical Characteristics (Note 7)

Symbol	Parameter	Conditions	LF147			LF347B			LF347			Units
			Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
	Amplifier to Amplifier Coupling	T_A=25˚C, f=1 Hz−20 kHz (Input Referred)		−120			−120			−120		dB
SR	Slew Rate	V_S=±15V, T_A=25˚C	8	13		8	13		8	13		V/µs
GBW	Gain-Bandwidth Product	V_S=±15V, T_A=25˚C	2.2	4		2.2	4		2.2	4		MHz
e_n	Equivalent Input Noise Voltage	T_A=25˚C, R_S=100Ω, f=1000 Hz		20			20			20		$nV/\sqrt{Hz}$
i_n	Equivalent Input Noise Current	T_j=25˚C, f=1000 Hz		0.01			0.01			0.01		$pA/\sqrt{Hz}$
THD	Total Harmonic Distortion	A_V=+10, R_L=10k, V_O=20 V_{p-p}, BW=20 Hz−20 kHz		<0.02			<0.02			<0.02		%

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. Operating Ratings indicate conditions for which the device is functional, but do not guarantee specific performance limits.

Note 3: Unless otherwise specified the absolute maximum negative input voltage is equal to the negative power supply voltage.

Note 4: Any of the amplifier outputs can be shorted to ground indefinitely, however, more than one should not be simultaneously shorted as the maximum junction temperature will be exceeded.

Note 5: For operating at elevated temperature, these devices must be derated based on a thermal resistance of θ_{jA}.

Note 6: The LF147 is available in the military temperature range −55˚C≤T_A≤125˚C, while the LF347B and the LF347 are available in the commercial temperature range 0˚C≤T_A≤70˚C. Junction temperature can rise to T_j max = 150˚C.

Note 7: Unless otherwise specified the specifications apply over the full temperature range and for V_S=±20V for the LF147 and for V_S=±15V for the LF347B/LF347. V_{OS}, I_B, and I_{OS} are measured at V_{CM}=0.

Note 8: The input bias currents are junction leakage currents which approximately double for every 10˚C increase in the junction temperature, T_j. Due to limited production test time, the input bias currents measured are correlated to junction temperature. In normal operation the junction temperature rises above the ambient temperature as a result of internal power dissipation, P_D. $T_j=T_A+\theta_{jA} P_D$ where θ_{jA} is the thermal resistance from junction to ambient. Use of a heat sink is recommended if input bias current is to be kept to a minimum.

Note 9: Supply voltage rejection ratio is measured for both supply magnitudes increasing or decreasing simultaneously in accordance with common practice from V_S = ± 5V to ±15V for the LF347 and LF347B and from V_S = ±20V to ±5V for the LF147.

Note 10: Refer to RETS147X for LF147D and LF147J military specifications.

Note 11: Max. Power Dissipation is defined by the package characteristics. Operating the part near the Max. Power Dissipation may cause the part to operate outside guaranteed limits.

Note 12: Human body model, 1.5 kΩ in series with 100 pF.

TC7660

Charge Pump DC-to-DC Voltage Converter

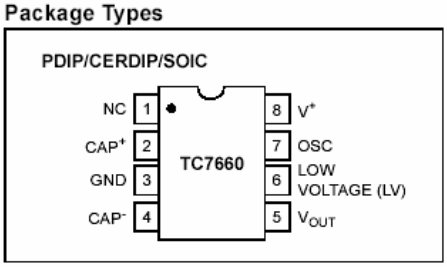
Features

- Wide Input Voltage Range: +1.5V to +10V
- Efficient Voltage Conversion (99.9%, typ)
- Excellent Power Efficiency (98%, typ)
- Low Power Consumption: 80 µA (typ) @ V_{IN} = 5V
- Low Cost and Easy to Use
  - Only Two External Capacitors Required
- Available in 8-Pin Small Outline (SOIC), 8-Pin PDIP and 8-Pin CERDIP Packages
- Improved ESD Protection (3 kV HBM)
- No External Diode Required for High-Voltage Operation

Applications

- RS-232 Negative Power Supply
- Simple Conversion of +5V to ±5V Supplies
- Voltage Multiplication $V_{OUT} = \pm n V^+$
- Negative Supplies for Data Acquisition Systems and Instrumentation

Package Types



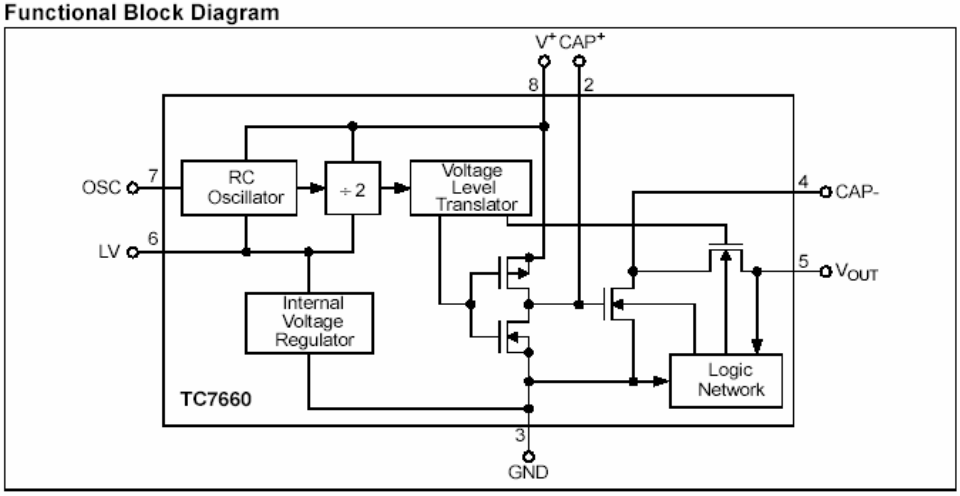
General Description

The TC7660 is a pin-compatible replacement for the industry standard 7660 charge pump voltage converter. It converts a +1.5V to +10V input to a corresponding -1.5V to -10V output using only two low cost capacitors, eliminating inductors and their associated cost, size and electromagnetic interference (EMI).

The on-board oscillator operates at a nominal frequency of 10 kHz. Operation below 10 kHz (for lower supply current applications) is possible by connecting an external capacitor from OSC to ground.

The TC7660 is available in 8-Pin PDIP, 8-Pin Small Outline (SOIC) and 8-Pin CERDIP packages in commercial and extended temperature ranges.

Functional Block Diagram



1.0 ELECTRICAL CHARACTERISTICS

Absolute Maximum Ratings*

Supply Voltage .....+10.5V
LV and OSC Inputs Voltage: **(Note 1)**
..... -0.3V to V_{SS} for V^+ < 5.5V
.....(V^+ – 5.5V) to (V^+) for V^+ > 5.5V
Current into LV .....20 µA for V^+ > 3.5V
Output Short Duration (V_{SUPPLY} ≤ 5.5V).....Continuous
Package Power Dissipation: (T_A ≤ 70°C)
8-Pin CERDIP .....800 mW
8-Pin PDIP .....730 mW
8-Pin SOIC .....470 mW
Operating Temperature Range:
C Suffix.....0°C to +70°C
I Suffix .....-25°C to +85°C
E Suffix.....-40°C to +85°C
M Suffix .....-55°C to +125°C
Storage Temperature Range.....-65°C to +160°C
ESD protection on all pins (HBM) .....≥ 3 kV
Maximum Junction Temperature..... ..... 150°C

*** Notice:** Stresses above those listed under "Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at those or any other conditions above those indicated in the operational sections of this specification is not intended. Exposure to maximum rating conditions for extended periods may affect device reliability.

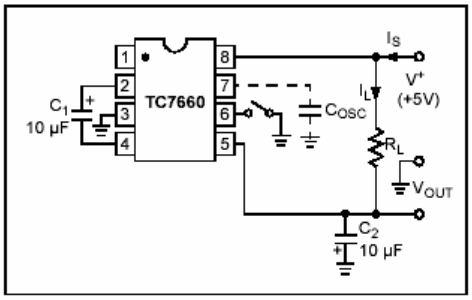


FIGURE 1-1: *TC7660 Test Circuit.*

ELECTRICAL SPECIFICATIONS

Electrical Characteristics: Unless otherwise noted, specifications measured over operating temperature range with V^+ = 5V, C_{OSC} = 0, refer to test circuit in Figure 1-1.

Parameters	Sym	Min	Typ	Max	Units	Conditions
Supply Current	I^+	—	80	180	µA	$R_L = \infty$
Supply Voltage Range, High	V^+_H	3.0	—	10	V	Min ≤ T_A ≤ Max, R_L = 10 kΩ, LV Open
Supply Voltage Range, Low	V^+_L	1.5	—	3.5	V	Min ≤ T_A ≤ Max, R_L = 10 kΩ, LV to GND
Output Source Resistance	R_{OUT}	—	70	100	Ω	I_{OUT}=20 mA, T_A = +25°C
		—	—	120		I_{OUT}=20 mA, T_A ≤ +70°C (C Device)
		—	—	130		I_{OUT}=20 mA, T_A ≤ +85°C (E and I Device)
		—	104	150		I_{OUT}=20 mA, T_A ≤ +125°C (M Device)
		—	150	300		V^+ = 2V, I_{OUT} = 3 mA, LV to GND 0°C ≤ T_A ≤ +70°C
		—	160	600		V^+ = 2V, I_{OUT} = 3 mA, LV to GND -55°C ≤ T_A ≤ +125°C (M Device)
Oscillator Frequency	f_{OSC}	—	10	—	kHz	Pin 7 open
Power Efficiency	P_{EFF}	95	98	—	%	R_L = 5 kΩ
Voltage Conversion Efficiency	V_{OUTEFF}	97	99.9	—	%	$R_L = \infty$
Oscillator Impedance	Z_{OSC}	—	1.0	—	MΩ	V^+ = 2V
		—	100	—	kΩ	V^+ = 5V

Note 1: Destructive latch-up may occur if voltages greater than V^+ or less than GND are supplied to any input pin.

INTEGRATED CIRCUITS

DATA SHEET

89C51/89C52/89C54/89C58

80C51 8-bit microcontroller family
4K/8K/16K/32K Flash

Product data

2001 Jul 09

Supersedes data of 1999 Oct 27

Philips
Semiconductors

PHILIPS

DESCRIPTION

The 89C51/89C52/89C54/89C58 contain a non-volatile FLASH program memory that is parallel programmable. For devices that are serial programmable (In-System Programmable (ISP) and In-Application Programmable (IAP) with a boot loader), see the 89C51Rx2 or 89C66x datasheets.

All three families are Single-Chip 8-bit Microcontrollers manufactured in advanced CMOS process and are derivatives of the 80C51 microcontroller family. All the devices have the same instruction set as the 80C51.

SELECTION TABLE FOR FLASH DEVICES

	MTP devices (this data sheet)		ISP/IAP devices (see separate data sheets)	
	89C51	**89C52/54/58**	**89C51Rx2**	**89C66x**
ROM/EPROM memory size	4K	8K/16K/32K	16K–32K	16K–64K
RAM size (byte)	128	256	512–1K	512–8K
Parallel programming	yes	yes	yes	yes
In-System Programming (ISP)	no	no	yes	yes
In-Application Programming (IAP)	no	no	yes	yes
PWM	no	no	yes	yes
Programmable Timer/Counter (PCA)	no	no	yes	yes
Hardware Watchdog Timer	no	no	yes	yes
Serial Channels	UART	UART	UART	UART + I^2C

MTP = Multi-Time Programming (via parallel programmer)

ISP = In-System Programming (via serial interface)

IAP = In-Application Programming

Please note that the FLASH programming algorithm of these parts has been modified. Please see the Device Comparison table for details.

DEVICE COMPARISON TABLE

Item	Old devices	New devices	Reason for change
Type description	P89C5xUBxx / P89C5xUFxx	P89C5xBxx / P89C5xFxx	Letter U dropped for shorter type descriptions (formerly designated speed (0–33 MHz))
Programming algorithm	When using parallel programmer, be sure to select P89C5xUxxx devices	When using a parallel programmer, be sure to select P89C5xxxx devices (no more letter U). IF DEVICES ARE NOT YET SELECTABLE, ASK YOUR VENDOR FOR A SOFTWARE UPDATE.	Programming algorithm modification required by process change!
Quad Flat Package type	PQFP package (P89C5xUxBB)	PQFP package replaced by LQFP package (P89C5xxBD). SEE NEW DIMENSIONS AT THE END OF THIS DATA SHEET.	Reduction in package height
Package identifiers	PLCC = AA PQFP = BB PDIP = PN	PLCC = A LQFP = BD PDIP = P	Shorter type descriptions
Flash memory program and erase cycles	100 program and erase cycles	10,000 program and erase cycles	Process change allows more program and erase cycles
Power consumption	Active mode: $I_{CC(MAX)}$= (0.9 × FREQ. + 20)mA Idle mode: $I_{CC(MAX)}$ = (0.37 × FREQ. + 1.0)mA	Active mode: $I_{CC(MAX)}$= (0.55 × FREQ. + 8.0)mA Idle mode: $I_{CC(MAX)}$ = (0.3 × FREQ. + 2.0)mA	Process change allows lower power consumption

80C51 8-bit microcontroller family 4K/8K/16K/32K Flash

89C51/89C52/89C54/89C58

FEATURES

- 80C51 Central Processing Unit
- On-chip FLASH Program Memory
- Speed up to 33 MHz
- Fully static operation
- RAM expandable externally to 64 k bytes
- 4 interrupt priority levels
- 6 interrupt sources
- Four 8-bit I/O ports
- Full-duplex enhanced UART
  - Framing error detection
  - Automatic address recognition
- Power control modes
  - Clock can be stopped and resumed
  - Idle mode
  - Power down mode
- Programmable clock out
- Second DPTR register
- Asynchronous port reset
- Low EMI (inhibit ALE)
- 3 16-bit timers
- Wake up from power down by an external interrupt

ORDERING INFORMATION

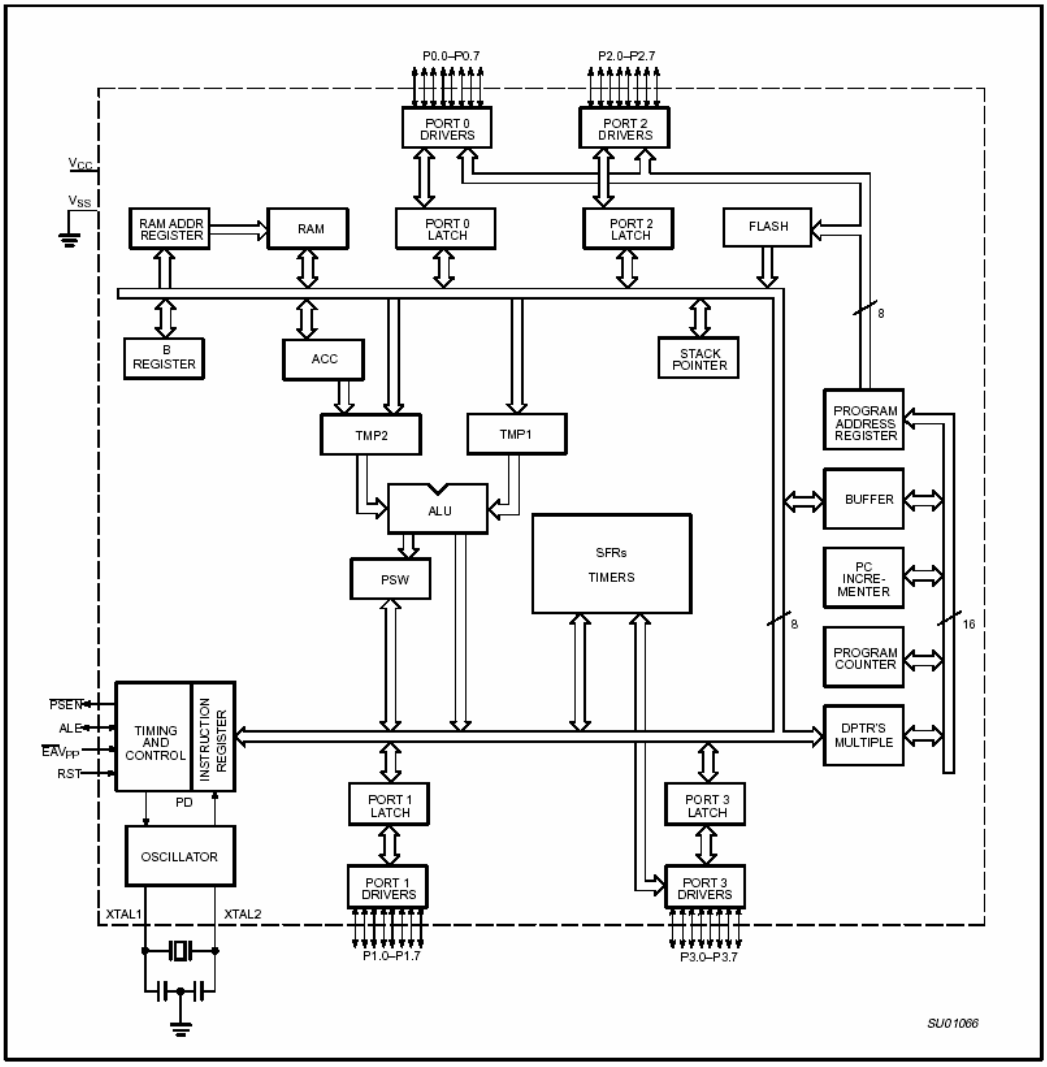
Type number				Package			Temperature Range[3] (°C)	Voltage Range (V)	Frequency (MHz)
4K Flash version	8K Flash version	16K Flash version	32K Flash version	Name	Description	Version			
P89C51BA	P89C52BA	P89C54BA	P89C58BA	PLCC44	plastic leaded chip carrier; 44 leads	SOT187-2	0 to +70	5	0 to 33
P89C51BP[1] P89C51BN[2]	P89C52BP[1] P89C52BN[2]	P89C54BP[1] P89C54BN[2]	P89C58BP[1] P89C58BN[2]	DIP40	plastic dual in-line package; 40 leads (600 mil)	SOT129-1	0 to +70	5	0 to 33
P89C51BBD	P89C52BBD	P89C54BBD	P89C58BBD	LQFP44	plastic low profile quad flat package; 44 leads; body 10 x 10 x 1.4 mm	SOT389-1	0 to +70	5	0 to 33

NOTES:
1. Philips (except North America) Part Order Number
2. Philips North America Part Order Number. Note that parts will be marked "P89C5xBP" or P89C5xFP", respectively (x = 1, 2, 4, 8)
3. Industrial temperatures will be released with P89C5xX2.

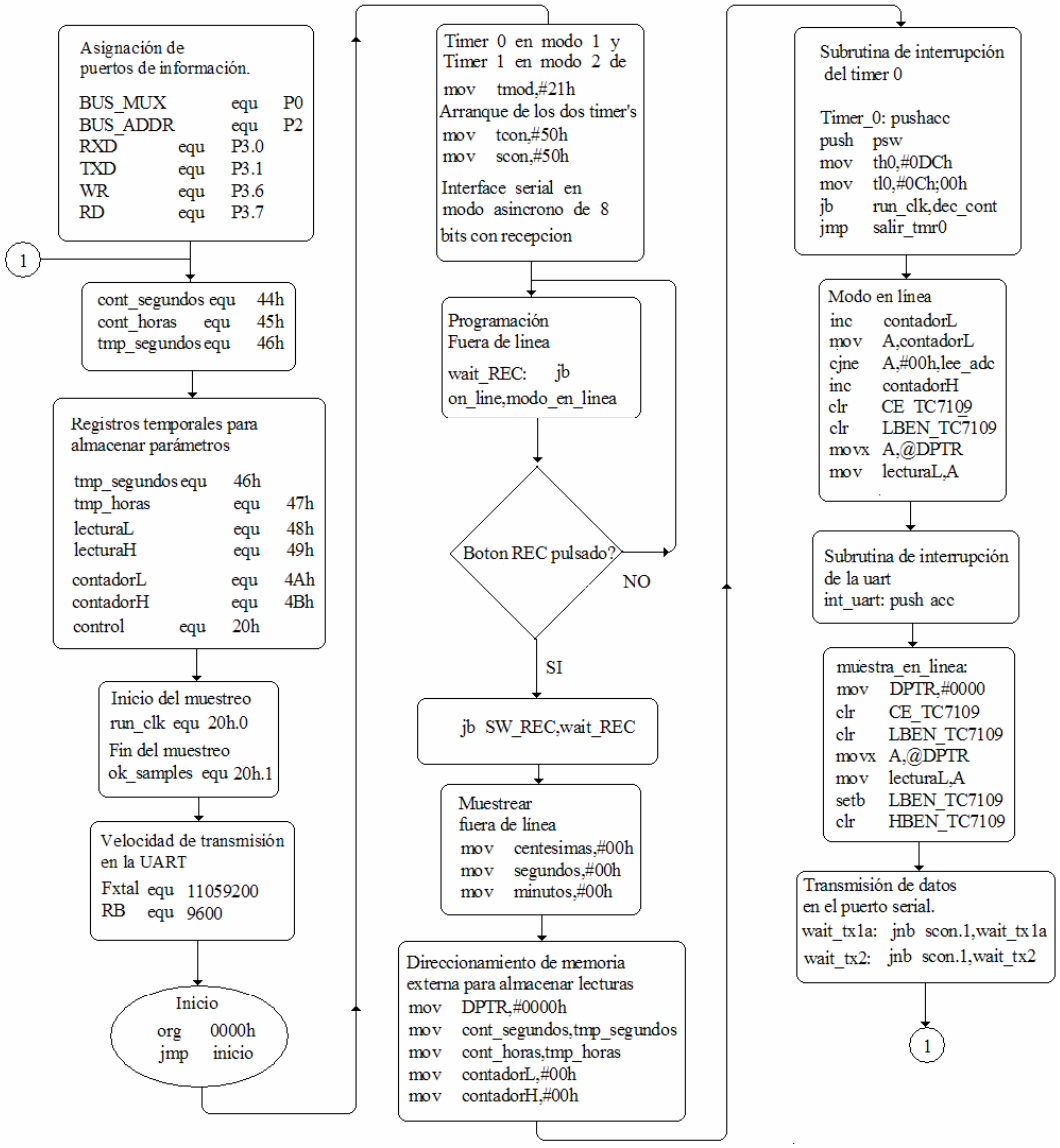
PART NUMBER DERIVATION

Device number (P89C5x)	Temperature range	Package
P89C51 P89C52 P89C54 P89C58	B = 0 °C to 70 °C F = –40 °C to 85 °C	BD = LQFP A = PLCC P = PDIP

BLOCK DIAGRAM



El siguiente diagrama de flujo describe la programación del microcontrolador AT89C51 para el proyecto del sistema de adquisición de datos de propósito general. Código de programación anexo en CD.



Código de programación para el microcontrolador AT 89C51

```
;*****************************************************************

;La velocidad de la comunicación es de 9.6KB's/Seg

                    ;BUS_MUX        equ     P0
                    ;BUS_ADDR       equ     P2
                    ;RXD        equ     P3.0
                    ;TXD        equ     P3.1
                    ;WR         equ     P3.6
                    ;RD         equ     P3.7

CE_TC7109           equ     P1.0
HBEN_TC7109         equ     P1.1
LBEN_TC7109         equ     P1.2
CS_RAM0             equ     P1.3
CS_RAM1             equ     P1.4
SW_REC              equ     P1.5
SW_PAUSA            equ     P1.6
SW_STOP             equ     P1.7
LED_ON              equ     P3.2
LED_REC             equ     P3.3
LED_COM_PC          equ     P3.4
LED_PAUSA           equ     P3.5

cont_led            equ     40h     ;Contador para determinar el tiempo para el
                                     LED_REC
centesimas     equ  41h     ;Registros para el conteo de tiempo
segundos            equ     42h
minutos             equ     43h

cont_segundos       equ     44h     ;Contador de tiempo entre cada muestra
cont_horas     equ  45h             ;Contador de tiempo para el número de horas que
                                    ;durara la prueba

tmp_segundos equ    46h     ;Registros temporales para almacenar los parametros
tmp_horas           equ     47h     ;programados

lecturaL            equ     48h
lecturaH            equ     49h

contadorL           equ     4Ah
contadorH           equ     4Bh

control        equ  20h
run_clk             equ     20h.0   ;Inicio del muestreo
ok_samples     equ  20h.1   ;Fin del muestreo
parametros     equ  20h.2   ;Indicador de parametros asignados
```

```
on_line             equ     20h.3   ;Indicador del modo en linea
ram0_full           equ     20h.4

Fxtal       equ     11059200
RB          equ     9600
;*****************************************************************

;*****************************************************************
            org     0000h
            jmp     inicio

            org     000Bh
            jmp     Timer_0

            org     0023h
            jmp     int_uart

            DB      "Sensor de Temperatura V1.0"

inicio:     mov     sp,#70h
            mov     p1,#0FFh
            mov     p3,#0FFh
            mov     th1,#256-((2*Fxtal)/(384*RB))
            mov     tl1,#256-((2*Fxtal)/(384*RB))
            mov     th0,#0DCh
            mov     tl0,#12h;00h
            mov     tmod,#21h           ;Timer 0 en modo 1 y Timer 1 en modo 2
de operación
            mov     pcon,#80h
            mov     tcon,#50h           ;Arranque de los dos timer's
            mov     scon,#50h           ;Interface serial en modo asincrono de 8
bits con recepcion
                                ; habilitada (velocidad determinada por el timer 1)
            mov     ie,#92h
            clr     LED_ON
            setb    LED_REC
            mov     control,#00h

wait_REC:           jb      on_line,modo_en_linea
            jb      SW_REC,wait_REC  ;Boton REC pulsado?
            jnb     parametros,wait_REC

            mov     centesimas,#00h
            mov     segundos,#00h
            mov     minutos,#00h
            mov     DPTR,#0000h;Direccion de la memoria externa para
                                ; almacenar las lecturas
            mov     cont_segundos,tmp_segundos
            mov     cont_horas,tmp_horas
            mov     contadorL,#00h
```

```
                mov     contadorH,#00h


                setb    run_clk
                mov     cont_led,#50

ciclo:          jb      ok_samples,wait_REC
                jnb     SW_STOP,inicio
                jb      SW_PAUSA,ciclo

                clr     run_clk
                setb    LED_REC
                clr     LED_PAUSA

pausa:          jb      SW_REC,pausa
                setb    run_clk
                setb    LED_PAUSA
                jmp     ciclo

modo_en_linea:
wait_O:                 jb      on_line,wait_O
                jmp     inicio
;*****************************************************************

;*****************************************************************
;Subrutina de interrupcion del timer 0
Timer_0:                push    acc
                push    psw
                mov     th0,#0DCh
                mov     tl0,#0Ch;00h
                jb      run_clk,dec_cont
                jmp     salir_tmr0

dec_cont:               djnz    cont_led,reloj
                mov     cont_led,#50
                cpl     LED_REC

reloj:          mov     A,centesimas
                add     A,#01h
                da      A
                mov     centesimas,A
                cjne    A,#00h,salir_tmr0

                djnz    cont_segundos,verifica_total
                mov     cont_segundos,tmp_segundos
muestra:                inc     contadorL
                mov     A,contadorL
                cjne    A,#00h,lee_adc
                inc     contadorH
lee_adc:                clr     CE_TC7109
```

```
                clr     LBEN_TC7109
                movx    A,@DPTR
                mov     lecturaL,A
                setb    LBEN_TC7109
                clr     HBEN_TC7109
                movx    A,@DPTR
                mov     lecturaH,A
                setb    HBEN_TC7109
                setb    CE_TC7109
                anl     lecturaH,#3Fh

                jb      ram0_full,sel_ram1

                mov     A,lecturaH
                clr     CS_RAM0
                movx    @DPTR,A
                setb    CS_RAM0
                inc     DPTR
                mov     A,lecturaL
                clr     CS_RAM0
                movx    @DPTR,A
                setb    CS_RAM0
                inc     DPTR
                mov     A,DPL
                cjne    A,#00h,verifica_total
                mov     A,DPH
                cjne    A,#80h,verifica_total
                setb    ram0_full
                jmp     verifica_total
sel_ram1:       mov     A,lecturaH
                clr     CS_RAM1
                movx    @DPTR,A
                setb    CS_RAM1
                inc     DPTR
                mov     A,lecturaL
                clr     CS_RAM1
                movx    @DPTR,A
                setb    CS_RAM1
                inc     DPTR
                jmp     verifica_total

verifica_total: mov     A,segundos
                add     A,#01h
                da      A
                mov     segundos,A
                cjne    A,#60h,salir_tmr0
                mov     segundos,#00h

                mov     A,minutos
                add     A,#01h
```

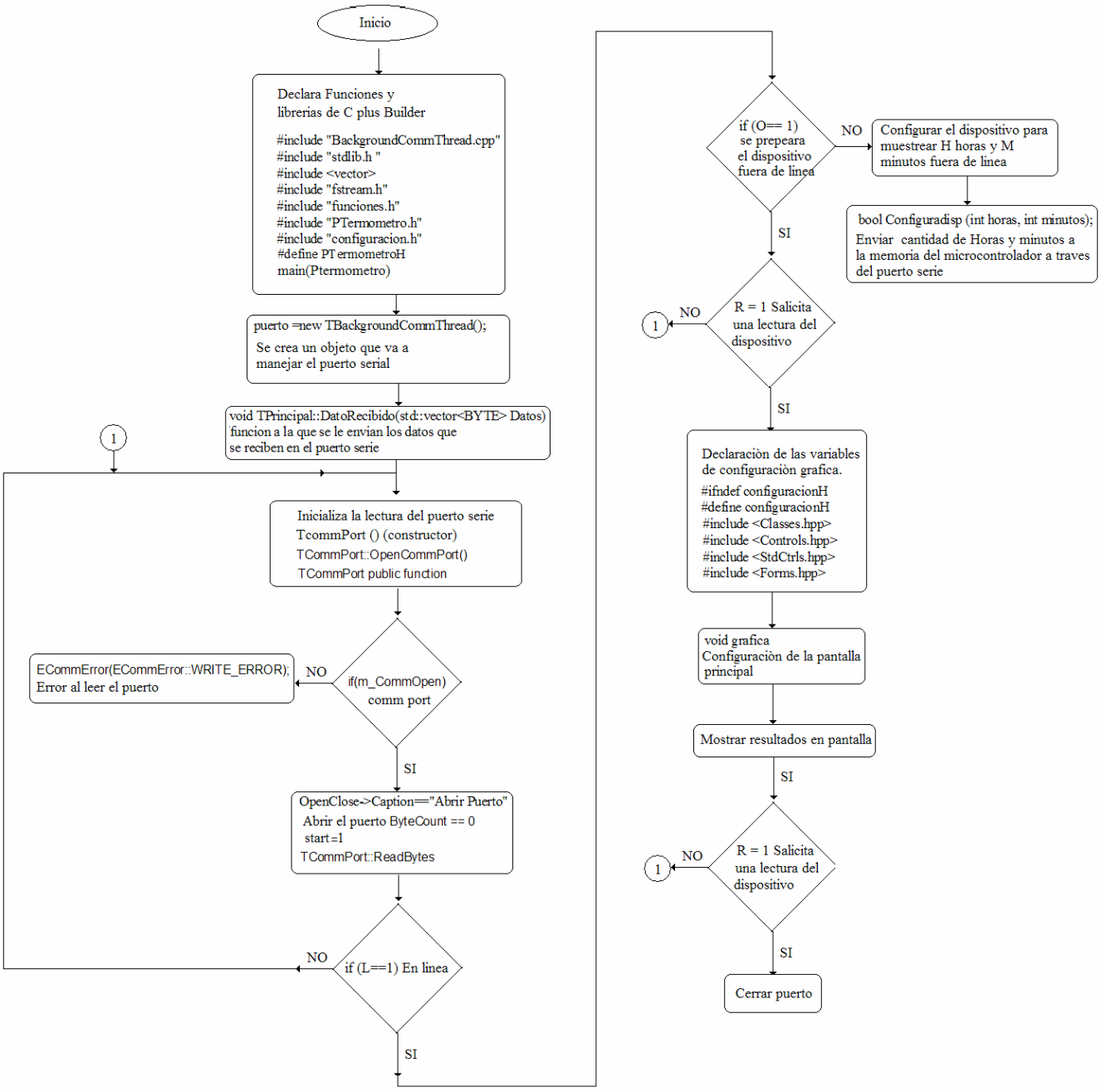
```
                da      A
                mov     minutos,A
                cjne    A,#60h,salir_tmr0
                mov     minutos,#00h

                djnz    cont_horas,salir_tmr0
                clr     run_clk
                setb    ok_samples

salir_tmr0:     pop     psw
                pop     acc
                reti
;*********************************************************************

;*********************************************************************
;Subrutina de interrupcion de la uart
int_uart:               push    acc
                push    psw
                jnb     scon.1,int_rx   ;TI=0?
                clr     scon.1
                jmp     salir_uart

int_rx:         clr     scon.0
                mov     A,sbuf

                cjne    A,#'L',next_cp1
                mov     sbuf,A
wait_tx1:               jnb     scon.1,wait_tx1
                clr     scon.1
                setb    on_line
muestra_en_linea:       mov     DPTR,#0000
                clr     CE_TC7109
                clr     LBEN_TC7109
                movx    A,@DPTR
                mov     lecturaL,A
                setb    LBEN_TC7109
                clr     HBEN_TC7109
                movx    A,@DPTR
                mov     lecturaH,A
                setb    HBEN_TC7109
                setb    CE_TC7109
                anl     lecturaH,#3Fh
                mov     sbuf,lecturaH
wait_tx1a:      jnb     scon.1,wait_tx1a
                clr     scon.1
                mov     sbuf,lecturaL
wait_tx1b:      jnb     scon.1,wait_tx1b
                clr     scon.1
                jmp     salir_uart
```

```
next_cp1:           cjne    A,#'O',next_cp2
                mov     sbuf,A
wait_tx2:           jnb     scon.1,wait_tx2
                clr     scon.1
                clr     on_line
                jmp     salir_uart

next_cp2:           cjne    A,#'A',next_cp3
                mov     sbuf,A
wait_tx3:           jnb     scon.1,wait_tx3
                clr     scon.1
wait_horas:     jnb     scon.0,wait_horas
                clr     scon.0
                mov     tmp_horas,sbuf
                mov     sbuf,tmp_horas
wait_tx4:           jnb     scon.1,wait_tx4
                clr     scon.1
wait_segundos:      jnb     scon.0,wait_segundos
                clr     scon.0
                mov     tmp_segundos,sbuf
                mov     sbuf,tmp_segundos
wait_tx5:           jnb     scon.1,wait_tx5
                clr     scon.1
                setb    parametros
                jmp     salir_uart


msg_error:      mov     sbuf,#'E'
wait_tx6:           jnb     scon.1,wait_tx6
                clr     scon.1
                jmp     salir_uart

msg_vacio:      mov     sbuf,#'V'
wait_tx7:           jnb     scon.1,wait_tx7
                clr     scon.1
                jmp     salir_uart


next_cp3:           cjne    A,#'R',msg_error
                jb      run_clk,msg_error
                jnb     ok_samples,msg_vacio
                clr     LED_COM_PC
                mov     sbuf,A
wait_tx5a:      jnb     scon.1,wait_tx5a
                clr     scon.1
                mov     sbuf,contadorH
wait_tx5b:      jnb     scon.1,wait_tx5b
                clr     scon.1
                mov     sbuf,contadorL
wait_tx5c:      jnb     scon.1,wait_tx5c
```

```
                clr     scon.1
                mov     DPTR,#0000h;Apunta hacia la direccion de la primer muestra
                clr     ram0_full
ciclo_tx:               mov     sbuf,#'r'
wait_tx_r:      jnb     scon.0,wait_tx_r
                clr     scon.0
                jb      ram0_full,tx_ram1
                clr     CS_RAM0
                movx    A,@DPTR
                setb    CS_RAM0
                mov     sbuf,A
wait_tx_H:      jnb     scon.0,wait_tx_H
                clr     scon.0
                inc     DPTR
                clr     CS_RAM0
                movx    A,@DPTR
                setb    CS_RAM0
                mov     sbuf,A
wait_tx_L:      jnb     scon.0,wait_tx_L
                clr     scon.0
                inc     DPTR
                mov     A,DPL
                cjne    A,#00h,dec_contador
                mov     A,DPH
                cjne    A,#80h,dec_contador
                setb    ram0_full
                jmp     dec_contador
tx_ram1:                clr     CS_RAM1
                movx    A,@DPTR
                setb    CS_RAM1
                mov     sbuf,A
wait_tx_H1:     jnb     scon.0,wait_tx_H1
                clr     scon.0
                inc     DPTR
                clr     CS_RAM1
                movx    A,@DPTR
                setb    CS_RAM1
                mov     sbuf,A
wait_tx_L1:     jnb     scon.0,wait_tx_L1
                clr     scon.0
                inc     DPTR
dec_contador:   dec     contadorL
                mov     A,contadorL
                cjne    A,#00h,es_ff?
                mov     A,contadorH
                cjne    A,#00h,es_ff?
                mov     control,#00
                setb    LED_COM_PC
                jmp     inicio          ;Verificar este salto
es_ff?:         mov     A,contadorL
```

```
            cjne    A,#0FFh,ciclo_tx
            dec     contadorH
            jmp     ciclo_tx

salir_uart: pop     psw
            pop     acc
            reti
```

El siguiente diagrama de flujo describe de manera general el programa C++ builder, del sistema de adquisición de datos de propósito general. Código de programación anexo en CD



ptermometro.h

****

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
#include "BackgroundCommThread.cpp"
#include "stdlib.h "
#include <vector>
#include "fstream.h"
#include "funciones.h"
#include "PTermometro.h"
#include "configuracion.h"
#include "calibracion.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TPrincipal *Principal;
extern TConfig *Config;
extern Tfrmcalibracion *frmcalibracion;
int vuelta;
//---------------------------------------------------------------------------
__fastcall TPrincipal::TPrincipal(TComponent* Owner)
        : TForm(Owner)
{
        puerto =new TBackgroundCommThread();//se crea un objeto que
        //va a manejar el puerto serial
        Ndatos=0;
        datos=new double[graficacion->Picture->Width*4];
        actual=0;
        vuelta=0;
        AnsiString Ruta=ExtractFilePath(Application->ExeName);
        Ruta+="cnf.cnf";
        LeeConfig(Ruta.c_str());
}
//---------------------------------------------------------------------------
void TPrincipal::DatoRecibido(std::vector<BYTE> Datos)
{//funcion a la que se le envian los datos que se reciben en el puerto serie
          int min,max;
  switch(rangoy->ItemIndex )
  {
   case 0:
           min=-30;max=30;
           break;
```

```
  case 1:
        min=0;max=50;
        break;
  }

      int pos=0;
      int dato,dato2;
      int longitud=Datos.size();
      if(Datos[pos]=='L')
      {
        pos++;

        dato=(char)Datos[pos++];
        dato2=(char)Datos[pos++];
        dato&=0x000F;
        dato2&=0x00FF;
        d1->Text=dato;
        d2->Text=dato2;
        dato<<=8;
        datol=dato2;
        res->Text=dato;
        double decimal;
        decimal=(double)1 /(double)((rand()%100)+1);
        double temperatura=(double)dato-offsetfact+decimal;
        dato/=factor;
        dato-=restafact;
        TEMP->Text=temperatura;
        graficaonline(temperatura,graficacion->Picture,false);
        datos[(vuelta*graficacion->Picture->Width)+actual]=temperatura;
        actual++;
        Ndatos++;
        if(actual==(graficacion->Picture->Width -100))
        {
            actual=0;
            vuelta++;
            if(vuelta==4)
            {
              int auxi=graficacion->Picture->Width;
              while(auxi<(vuelta*graficacion->Picture->Width))
              {
                  datos[auxi]=datos[auxi+graficacion->Picture->Width];
                  Ndatos=((Ndatos/4)*3);
                  auxi++;
              }
              vuelta=3;
//              Ndatos=0;
            }
        }
        return;
      }
```

```
        if(Datos[pos]=='E')
        {
                Application->MessageBoxA("Ocurrio un error en el
dispositivo","Información");
                return;
        }
        if(Datos[pos]=='V')
        {
                Application->MessageBoxA("No contiene datos el
dispositivo","Información");
                return;
        }
        if(Datos[pos]=='R')
        {

           dato=(char)Datos[pos++];
           dato<<8;
           dato|=(char)Datos[pos++];
           recibiendo=dato;
//////////////////////////
int min,max;
  switch(rangoy->ItemIndex )
  {
   case 0:
           min=-30;max=30;
           break;
   case 1:
          min=-50;max=150;
          break;
   case 2:
             min=0;max=150;
          break;
  }
  graficaonline( 0,graficacion->Picture,grafica,rangox->Text.ToInt(),min,max);
///////////////

           if(datos)
                delete []datos;
           datos=new double [recibiendo];
           Ndatos=recibiendo;
        }
        while(Datos.size()<pos+1)
        {
          pos++;
          dato=(char)Datos[pos++];
           dato2=(char)Datos[pos++];
           dato&=0x000F;
           dato2&=0x00FF;
           d1->Text=dato;
           d2->Text=dato2;
```

```
            dato<<=8;
            dato|=dato2;
            res->Text=dato;
            double decimal;
            decimal=(double)1 /(double)(rand()%100);
            double temperatura=(double)dato-offsetfact+decimal;
            dato/=factor;
            dato-=restafact;
            TEMP->Text=temperatura;

            graficaonline(temperatura,graficacion->Picture,false);
            datos[actual++]=dato;
            Ndatos++;
            if(actual==graficacion->Picture->Width)
            {
                actual=0;
                Ndatos=0;
            }

        }
}
//---------------------------------------------------------------------------
void __fastcall TPrincipal::OpenCloseClick(TObject *Sender)
{
        //si el puerto esta cerrado se da el comando de abrir puerto, de lo contrario
        //se cierra.
        if(OpenClose->Caption=="Abrir Puerto")
        {
          OpenClose->Caption="Cerrar Puerto";
          puerto->Connect();
        }
        else
        {
          OpenClose->Caption="Abrir Puerto";
          puerto->Disconnect();
          stopClick(Sender);
        }
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::GuardarClick(TObject *Sender)
{
        //se permite que se vean los archivos con extensión .tof que es la que le asignamos
        //al programa
        save->Filter="Archivos de Termometro Offline|*.tof";
        AnsiString Narch;
        if(save->Execute())//si se ejecuta el
        {
          Narch=save->FileName;
          if(Narch.SubString(Narch.Length()-3,1)!=".")
```

```
            Narch+=".tof";
        ofstream Archivo(Narch.c_str());
        int conta=Ndatos;
        Archivo<<Ndatos<<endl;
        double dato;
        while(conta--)
        {
            dato=datos[conta];
            Archivo<<dato<<endl;
        }
      }
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::CargarClick(TObject *Sender)
{
      open->Filter="Archivos de Termometro Offline|*.tof";
      AnsiString Ruta=ExtractFilePath(Application->ExeName);
      open->InitialDir=Ruta;
      AnsiString Narch;
      if(open->Execute())
      {
        Narch=open->FileName;
        if(Narch.SubString(Narch.Length()-3,1)!=".")
            Narch+=".tof";
        datos=*LeeArchivo(Narch.c_str(),Ndatos);
      }
      if(Ndatos==0)
      {
        Application->MessageBoxA("No contiene datos","Aviso");
        return;
      }
      grafica(datos,Ndatos,graficacion->Picture);
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::ConfigurarClick(TObject *Sender)
{
      int cantidad;
      Config= new TConfig(this,puerto,&cantidad);
      Config->ShowModal();
      intervalo=cantidad;


}
//---------------------------------------------------------------------------


void __fastcall TPrincipal::Button1Click(TObject *Sender)
{
```

```
OpenClose->Caption="Abrir Puerto";
        puerto->Disconnect();
        stopClick(Sender);
 OpenClose->Caption="Cerrar Puerto";
        puerto->Connect();

  int min,max;
  if(intervalo==0)
    {
       Application->MessageBoxA("Configura primero el intervalo","Error");
       return;
  }
  if(puerto->GetConnected()==false)
  {
       Application->MessageBoxA("El puerto no esta abierto","Error");
       return;
  }
  switch(rangoy->ItemIndex )
  {
   case 0:
          min=-30;max=30;
          break;
   case 1:
        min=-50;max=150;
        break;
   case 2:
           min=0;max=100;
        break;
   case 3:
              min=0;max=150;
        break;
  }
  graficaonline( 0,graficacion->Picture,grafica,rangox->Text.ToInt(),min,max);
  transc=0;
  this->Online->Enabled =true;
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::OnlineTimer(TObject *Sender)
{
  if(transc==intervalo)
  {
   char trama[100];
   int pos=0;
   trama[pos++]='L';
    // trama[pos++]=ConvierteHex(frecuencia->Text);
   trama[pos++]='\0';
   puerto->WriteString(trama);
   transc=0;
   return;
```

```
  }
  transc++;
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::rangoxKeyPress(TObject *Sender, char &Key)
{
        Key=0;
}
//---------------------------------------------------------------------------


void __fastcall TPrincipal::stopClick(TObject *Sender)
{
        this->Online->Enabled=false;
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::LeerClick(TObject *Sender)
{
   char trama[100];
   int pos=0;
   trama[pos++]='R';
    // trama[pos++]=ConvierteHex(frecuencia->Text);
   trama[pos++]='\0';
   puerto->WriteString(trama);
   transc=0;

}
//---------------------------------------------------------------------------




void __fastcall TPrincipal::rangoyChange(TObject *Sender)
{
 int min,max;
  switch(rangoy->ItemIndex )
  {
   case 0:
           min=-30;max=30;
           break;
   case 1:
          min=-50;max=150;
          break;
   case 2:
             min=0;max=150;
          break;
  }
```

```
  graficaonline( 0,graficacion->Picture,grafica,rangox->Text.ToInt(),min,max);
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::rangoxChange(TObject *Sender)
{
 int min,max;
  switch(rangoy->ItemIndex )
  {
   case 0:
          min=-30;max=30;
          break;
   case 1:
         min=-50;max=150;
         break;
   case 2:
            min=0;max=150;
         break;
  }
  graficaonline( 0,graficacion->Picture,grafica,rangox->Text.ToInt(),min,max);
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::FormCreate(TObject *Sender)
{
  randomize();
}
//---------------------------------------------------------------------------

void __fastcall TPrincipal::Button2Click(TObject *Sender)
{
       frmcalibracion=new Tfrmcalibracion(this);
       frmcalibracion->ShowModal();
}
//---------------------------------------------------------------------------
```

ptermometro.h

```
#ifndef PTermometroH
#define PTermometroH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <ExtCtrls.hpp>
#include <Graphics.hpp>
#include <Dialogs.hpp>
```

```
#include "fstream.h"
//---------------------------------------------------------------------------
class TPrincipal : public TForm
{
__published:   // IDE-managed Components
        TImage *graficacion;
        TButton *OpenClose;
        TButton *Guardar;
        TButton *Cargar;
        TButton *Configurar;
        TSaveDialog *save;
        TOpenDialog *open;
        TLabel *Label1;
        TLabel *Label2;
        TTimer *Online;
        TButton *Button1;
        TButton *stop;
        TComboBox *rangox;
        TComboBox *rangoy;
        TButton *Leer;
        TEdit *d1;
        TEdit *d2;
        TEdit *res;
        TEdit *TEMP;
        TLabel *Label3;
        TLabel *Label4;
        TButton *Button2;
        void __fastcall OpenCloseClick(TObject *Sender);
        void __fastcall GuardarClick(TObject *Sender);
        void __fastcall CargarClick(TObject *Sender);
        void __fastcall ConfigurarClick(TObject *Sender);
        void __fastcall Button1Click(TObject *Sender);
        void __fastcall OnlineTimer(TObject *Sender);
        void __fastcall rangoxKeyPress(TObject *Sender, char &Key);
        void __fastcall stopClick(TObject *Sender);
        void __fastcall LeerClick(TObject *Sender);
        void __fastcall rangoyChange(TObject *Sender);
        void __fastcall rangoxChange(TObject *Sender);
        void __fastcall FormCreate(TObject *Sender);
        void __fastcall Button2Click(TObject *Sender);
private:        // User declarations
        TBackgroundCommThread *puerto;
        int Ndatos;
        double *datos;
        int actual,recibiendo;

public:         // User declarations
        __fastcall TPrincipal(TComponent* Owner);
        double intervalo,transc;
        void DatoRecibido(std::vector<BYTE> m_ReceivedChars);
```

```
};
//---------------------------------------------------------------------------
extern PACKAGE TPrincipal *Principal;
//---------------------------------------------------------------------------
#endif
```

BackgroundCommThread.h

****

```
#ifndef BackgroundCommThreadH
#define BackgroundCommThreadH
//---------------------------------------------------------------------------
#include <Classes.hpp>

#include "comm.h"
#include <vector>

#include <string>

//---------------------------------------------------------------------------
class TBackgroundCommThread : public TThread
{
protected:
    std::vector<BYTE> m_ReceivedChars;
    TCommPort m_CommPort;
    void __fastcall Execute();

    // Synchronized methods
    void __fastcall SynchronizeNewBytes(void);
public:
    __fastcall TBackgroundCommThread();
    virtual __fastcall ~TBackgroundCommThread();

    void Connect();
    void Disconnect();
    void SetParameters(unsigned int baud, const std::string &port);
    bool GetConnected()
    {
        return m_CommPort.GetConnected();
    }

    void WriteString(const AnsiString &str);
//    void WriteStringAndWaitForAck(const AnsiString &str);



    TCommPort& GetCommPort()
```

```
    {
        return m_CommPort;
    }

};
//---------------------------------------------------------------------------
#endif
```

Comm..cpp maneja el puerto serie

****

```
#ifdef __BORLANDC__
#include <vcl.h>
#pragma hdrstop
#endif

#include "comm.h"

ECommError::ECommError( ErrorType error)
   : Error(error),
     Errno(GetLastError())
{
}
TCommPort::TCommPort()
  :m_CommOpen(false),
   m_CommPort("COM1"),
   m_hCom(0)
{
    // Inicializa el Puerto comm a N81 9600 baud communications.  Estos valores
    // serán utilizados para inicializar el Puerto si OpenCommPort es llamado antes
    m_dcb.DCBlength = sizeof(DCB);
    m_dcb.BaudRate  =9600;
    m_dcb.ByteSize  =8;
    m_dcb.Parity    =NOPARITY;    //NOPARITY
    m_dcb.StopBits  =ONESTOPBIT;  //ONESTOPBIT
}
///// end of TCommPort::TCommPort (constructor)
//////////////////////////////////////////////////////////////////////////////

TCommPort::~TCommPort()
{
    if(m_CommOpen)
      CloseCommPort();
}

    DCB tempDCB;
    tempDCB.BaudRate  =  m_dcb.BaudRate;
```

```
   tempDCB.ByteSize  =  m_dcb.ByteSize;
   tempDCB.Parity    =  m_dcb.Parity;
   tempDCB.StopBits  =  m_dcb.StopBits;

   m_hCom = CreateFile(m_CommPort.c_str(),
                    GENERIC_READ | GENERIC_WRITE,

void TCommPort::CloseCommPort(void)
{
   if(!m_CommOpen)       // if already closed, return
     return;

   if(CloseHandle(m_hCom) != 0) // CloseHandle is non-zero on success
   {
      m_CommOpen = false;
   }
   else
      throw ECommError(ECommError::CLOSE_ERROR);
}

void TCommPort::SetParity(BYTE newParity)
{
   BYTE oldParity = m_dcb.Parity;     // make a backup of the old parity
   m_dcb.Parity = newParity;                 // assign new parity

   if(m_CommOpen)                    // check for open comm port
   {
      if(!SetCommState(m_hCom,&m_dcb))     // try to set the new comm settings
      {                                 // if failure
         m_dcb.Parity = oldParity;              // restore old parity
         throw ECommError(ECommError::BAD_PARITY);  // bomb out
      }
   }
}

void TCommPort::SetStopBits(BYTE newStopBits)
{
   BYTE oldStopBits = m_dcb.StopBits; // make a backup of old #of stop bits
   m_dcb.StopBits = newStopBits;             // assign new # of stop bits

   if(m_CommOpen)                    // check for open comm port
   {
      if(!SetCommState(m_hCom,&m_dcb))     // try to set the new comm settings
      {                                 // if failure
         m_dcb.StopBits = oldStopBits;         // restore old # of stop bits
         throw ECommError(ECommError::BAD_STOP_BITS);                    // bomb
out
      }
   }
}
```

```
///// end of TCommPort::SetParity()
//////////////////////////////////////////////////////////////////////////////

unsigned int TCommPort::GetBaudRate(void)
{
  return m_dcb.BaudRate;
}

BYTE TCommPort::GetByteSize(void)
{
  return m_dcb.ByteSize;
}

BYTE TCommPort::GetParity(void)
{
  return m_dcb.Parity;
}

BYTE TCommPort::GetStopBits(void)
{
  return m_dcb.StopBits;
}

void TCommPort::WriteBuffer(BYTE *buffer, unsigned int ByteCount)
{
    VerifyOpen();
    DWORD dummy;
    if( (ByteCount == 0) || (buffer == NULL))
        return;

    if(!WriteFile(m_hCom,buffer,ByteCount,&dummy,NULL))
        throw ECommError(ECommError::WRITE_ERROR);
}

void TCommPort::WriteBufferSlowly(BYTE *buffer, unsigned int ByteCount)
{
    VerifyOpen();
    DWORD dummy;
    BYTE *ptr = buffer;

    for (unsigned int j=0; j<ByteCount; j++)
    {
        if(!WriteFile(m_hCom,ptr,1,&dummy,NULL))
            throw ECommError(ECommError::WRITE_ERROR);

        // Use FlushCommPort to wait until the character has been sent.
        FlushCommPort();
        ++ptr;
    }
}
```

```
void TCommPort::WriteString(const char *outString)
{
   VerifyOpen();

   DWORD dummy;
   if(!WriteFile(m_hCom,outString, strlen(outString),&dummy,NULL))
      throw ECommError(ECommError::WRITE_ERROR);
}

int TCommPort::ReadBytes(BYTE *buffer, unsigned int MaxBytes)
{
   VerifyOpen();
   DWORD bytes_read;

   if(!ReadFile(m_hCom,buffer,MaxBytes,&bytes_read,NULL))
      throw ECommError(ECommError::READ_ERROR);

   // add a null terminate if bytes_read < byteCount
   // if the two are equal, there is no space to put a null terminator
   if(bytes_read < MaxBytes)
      buffer[bytes_read]='\0';

   return bytes_read;
}

int TCommPort::ReadString(char *str, unsigned int MaxBytes)
{
   VerifyOpen();

   if(MaxBytes == 0u)
      return 0;
   str[0]='\0';
   if(BytesAvailable() ==0)
      return 0;

   BYTE NewChar;
   unsigned int Index=0;
   while(Index < MaxBytes)
   {
      NewChar = GetByte();

      // if the byte is a \r or \n, don't add it to the string
      if( (NewChar != '\r') && (NewChar != '\n'))
      {
         str[Index] = (char) NewChar;
         Index++;
      }

      // when /r is received, we are done reading the string, so return
```

```
        // don't forget to terminate the string with a \0.
        if(NewChar == '\r')
        {
            str[Index] = '\0';
            return Index +1;
        }
    }

    // if the while loop false through, then MaxBytes were received without
    // receiveing a \n. Add null terminator to the string and return the number
    str[MaxBytes-1]='\0';
    return MaxBytes;
}

void TCommPort::DiscardBytes(unsigned int MaxBytes)
{
    VerifyOpen();
    if(MaxBytes == 0)
        return;

    BYTE *dummy= new BYTE[MaxBytes];
    ReadBytes(dummy, MaxBytes);
    delete []dummy;
}

void TCommPort::PurgeCommPort(void)
{
    VerifyOpen();
    if(!PurgeComm(m_hCom,PURGE_RXCLEAR))
        throw ECommError(ECommError::PURGECOMM);

}

void TCommPort::FlushCommPort(void)
{
    VerifyOpen();

    if(!FlushFileBuffers(m_hCom))
        throw ECommError(ECommError::FLUSHFILEBUFFERS);
}

void TCommPort::PutByte(BYTE value)
{
    VerifyOpen();

    DWORD dummy;
    if(!WriteFile(m_hCom,&value,1,&dummy,NULL))
        throw ECommError(ECommError::WRITE_ERROR);
}
```

```
BYTE TCommPort::GetByte()
{
   VerifyOpen();

   DWORD dummy;
   BYTE  value;
   if(!ReadFile(m_hCom,&value,1,&dummy,NULL))
      throw ECommError(ECommError::READ_ERROR);

   return value;
}


unsigned int TCommPort::BytesAvailable(void)
{
   VerifyOpen();

   COMSTAT comstat;
   DWORD dummy;

   if(!ClearCommError(m_hCom, &dummy, &comstat))
      throw ECommError(ECommError::CLEARCOMMERROR);
   return comstat.cbInQue;
}

void TCommPort::SetCommPort(const std::string & port)
{
   VerifyClosed();   // can't change comm port once comm is open
                     // could close and reopen, but don't want to

   m_CommPort = port;
}

std::string TCommPort::GetCommPort(void)
{
   return m_CommPort;
}

void TCommPort::GetCommProperties(COMMPROP &properties)
{
   VerifyOpen();

   COMMPROP prop;
   ZeroMemory(&prop, sizeof(COMMPROP));
   ::GetCommProperties(m_hCom, &prop);

   properties = prop;
}
```

funciones.h Funciones que grafican y decodifican los datos

```
#include "fstream.h"
int orx,ory;
double factor,restafact,offsetfact;
bool guarda_archivo (char * ruta, double *datos, int cantidad)
{
 //Esta función guarda en un archivo los datos contenidos en el apuntador, para lo cual debemos enviarle la cantidad de datos a guardar y la ruta donde se desea crear el archivo. Todos los archivos son guardados en forma binaria para lograr un almacenamiento de datos óptimo y minimizar el espacio ocupado.
 return true;
}

double **LeeArchivo (char *ruta, int &Ndatos)
{
	ifstream Archivo(ruta);
	  Archivo>>Ndatos;
	  int conta=Ndatos;
	  double **datos;
	  datos=new double*;
	  datos[0]=new double[Ndatos];
	  while(conta--)
		Archivo>>datos[0][conta];
	  return datos;
 //Esta función lee de la ruta especificada un archivo y devuelve los datos en un arreglo de enteros y guarda la cantidad de estos en la variable longitud.
}
void LeeConfig (char *ruta)
{
	ifstream Archivo(ruta);
	  Archivo>>orx;
	  Archivo>>ory;
	  Archivo>>factor;
	  Archivo>>restafact;
	  Archivo>>offsetfact;
}

bool Online (bool estado);
/*
Esta función se encarga de enviar el comando necesario a el dispositivo a fin de ponerlo en estado en línea o fuera de línea según indique la variable estado, sie esta es verdadera se establece en línea, de lo contrario se establece fuera de línea.
*/
bool Configuradisp (int horas, int minutos);
```

/*
Esta función se encarga de enviarle al dispositivo por el puerto serie la cantidad de horas y minutos que debe guardar en su memoria en estado fuera de línea y devuelve falso si no esta conectado el dispositivo o bien si se obtuvo un error en la comunicación.
*/
int * RequestLectura (int &longitud);
/*
Esta funcion solicita lectura de datos, en caso de que el dispositivo este en línea se obtendrán datos cada segundo, de lo contrario si estaba fuera de linea pero esta conectado, se enviaran las lecturas actuales mientras este grabando, si el dispositivo esta preparado para enviar todos los datos de su memoria (es decir tiene algo en memoria) se devolverán todos los datos en el arreglo de enteros y se indicara su longitud en la variable con el mismo nombre.
*/
int Decodifica (char parte1, char parte2);
/*
Esta funcion se encarga de decodificar un dato que recibe por puerto serie, ya que se recibe un dato entero de 2 bytes segmentado en parte baja y parte alta, por lo que se encarga de juntarlo en la variable entera y devolverlo.
*/
```
void Limpia(TPicture *grafica)
{
  AnsiString Ruta=ExtractFilePath(Application->ExeName);
  Ruta+="fondo.bmp";
  grafica->LoadFromFile(Ruta );
}

void mayormenor(double *datos,int cantidad,double &may,double &men)
{
   may=men=-1;
   if(cantidad<=0)
       return ;
   men=may=datos[0];
   while(--cantidad)
   {
     if(may<datos[cantidad])
        may=datos[cantidad];
     if(men>datos[cantidad])
        men=datos[cantidad];
   }

}
float Escala(double * datos, int altura,int cantidad)
{
```
//Esta función recibe un apuntador a enteros, del cual se obtiene el mayor y menor de ellos. Una vez que se tienen el mayor y menor se hace un calculo de la escala para lo cual se realiza una división de altura / |mayor-menor, con lo que obtenemos el factor por el que debemos multiplicar cada uno de los datos para que se grafiquen correctamente en el espacio asignado.
```
   double may,men;
```

```
  mayormenor(datos,cantidad,may,men);
  float esc=(float)altura/(may-men);
  return esc;
}
void grafica (double *datos, int cantidad,TPicture *grafica ,bool escala=true)
//Esta función es la encargada de graficar un arreglo de datos enteros en la pantalla, para
lo cual recibe la cantidad de datos contenidos en el apuntador y un dato booleano que
indica si se va a aplicar o no escala de acondicionamiento de los datos, es decir para que
se grafiquen todos los datos dentro del rango de la pantalla.
{
      double escy,escx,x=orx,y=grafica->Height;
      double may,men;
  mayormenor(datos,cantidad,may,men);
      if(escala==true)
      {
        escy=Escala(datos,grafica->Height-ory,cantidad);
        escx=(double)(grafica->Width-orx)/((double)cantidad+1.5);
      }
      else
        escx=escy=1;
      y=grafica->Height-ory;
      grafica->Bitmap->Canvas->MoveTo(x,y);
      grafica->Bitmap->Canvas->Pen->Color =clBlack;
      Limpia(grafica);
      int conta=0;
      while(cantidad>0&&x<grafica->Width)
      {
            cantidad--;
            y=grafica->Height-datos[conta++]*escy+men;
            grafica->Bitmap->Canvas->LineTo((int)x,(int)y);
            x+=escx;
      }
}
//////////////////////////////////
void graficaonline (int dato,TPicture *grafica ,bool reset=true,int cantidadx=0,int
min=0,int max=0)
//Esta función es la encargada de graficar un arreglo de datos enteros en la pantalla, para
lo cual recibe la cantidad de datos contenidos en el apuntador y un dato booleano que
indica si se va a aplicar o no escala de acondicionamiento de los datos, es decir para que
se grafiquen todos los datos dentro del rango de la pantalla.
{
      static double escy,escx,x,y;
      static int offset;
      if(reset==true)
      {
        escy=(double)grafica->Height/(double)(max-min);
        escx=cantidadx;
        x=orx;
        y=grafica->Height-ory;
        offset=min;
```

```
        Limpia(grafica);
        return;
      }

      grafica->Bitmap->Canvas->MoveTo(x,y);
      grafica->Bitmap->Canvas->Pen->Color =clBlack;

      y=grafica->Height-dato*escy+offset-ory;
      grafica->Bitmap->Canvas->LineTo((int)x+escx,(int)y);
      x+=escx;
      if(grafica->Width<=(x+100+orx))
       {
            Limpia(grafica);
            x=orx;
       }
}
```

****

Configuración.cpp Configuración de la pantalla principal

****

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop

#include "BackgroundCommThread.cpp"
#include <vector>
#include "PTermometro.h"
#include "configuracion.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
TConfig *Config;
extern TPrincipal *Principal;
//---------------------------------------------------------------------------
__fastcall TConfig::TConfig(TComponent* Owner,TBackgroundCommThread
*puerto,int *cantidad)
      : TForm(Owner)
{
      this->puerto =puerto;
      this->cantidad=cantidad;
}
//---------------------------------------------------------------------------
void __fastcall TConfig::BTNAplicarClick(TObject *Sender)
{
      if(horas->Text =""||frecuencia->Text="")
      {
```

```
            Application->MessageBoxA("Valor Invalido  de horas ,minutos o
frecuencia","Error");
            return;
        }
        Aplicar();
}
//---------------------------------------------------------------------------
void __fastcall TConfig::CancelarClick(TObject *Sender)
{
this->Close();
this->Release();
}
//---------------------------------------------------------------------------
void TConfig::Aplicar(bool reintento)
{
   if(puerto->GetConnected()==false)//Si el puerto está cerrado intentamos abrirlo
   {
        if(reintento==true)
        {
          Application->MessageBoxA("No se pudo reconectar al Puerto indicado","");
          this->Close();
          this->Release();
        }
        puerto->Connect();
        Aplicar(true);
        puerto->Disconnect();
        this->Close();
        this->Release();
        return;
   }
   char trama[100];
   int pos=0;

   if(online->Checked==true)//si esta en line solo actualizamos intervalo
   {
        double tiempo;

            tiempo=frecuencia->Text.ToInt();
            Principal->Online->Interval =tiempo;
            *cantidad=tiempo;
   }
   else
   {
     trama[pos++]='O';
     trama[pos++]='A';
    // trama[pos++]=ConvierteHex(frecuencia->Text);
    trama[pos++]=horas->Text.ToInt();
    trama[pos++]=frecuencia->Text.ToInt();
    trama[pos++]='\0';
     puerto->WriteString(trama);
```

```
    }
   Application->MessageBoxA("Configuración Aplicada","");
   if(reintento==false)
   {
     this->Close();
     this->Release();
   }
}
//---------------------------------------------------------------------------
void __fastcall TConfig::frecuenciaKeyPress(TObject *Sender, char &Key)
{
        if((Key<48||Key>57)&&Key!=8)
                Key=0;
}
//---------------------------------------------------------------------------
char TConfig::ConvierteHex(int dato)
{
        int i=2*sizeof(char) - 1;
        char resp=((dato >> i*4) & 0xF);
        i--;
        resp+=16;
        resp+=((dato >> i*4) & 0xF);
        return resp;
}
//---------------------------------------------------------------------------
void __fastcall TConfig::RadioButton2Click(TObject *Sender)
{
        frecuencia->Text="1";
        frecuencia->Enabled=false;
        frecsegundos->Checked=true;
}
//---------------------------------------------------------------------------

void __fastcall TConfig::onlineClick(TObject *Sender)
{
        frecuencia->Enabled=true;
}
//---------------------------------------------------------------------------
```

proytermometro.cpp Interface grafica de todos los eventos del proyecto

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
```

```
//---------------------------------------------------------------------------
USEFORM("PTermometro.cpp", Principal);
USEFORM("configuracion.cpp", Config);
USEFORM("calibracion.cpp", frmcalibracion);
//---------------------------------------------------------------------------
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)
{
	try
	{
		Application->Initialize();
		Application->CreateForm(__classid(TPrincipal), &Principal);
		Application->CreateForm(__classid(TConfig), &Config);
		Application->CreateForm(__classid(Tfrmcalibracion), &frmcalibracion);
		Application->Run();
	}
	catch (Exception &exception)
	{
		Application->ShowException(&exception);
	}
	catch (...)
	{
		try
		{
			throw Exception("");
		}
		catch (Exception &exception)
		{
			Application->ShowException(&exception);
		}
	}
	return 0;
}
//---------------------------------------------------------------------------
```

calibración.cpp

****

```
//---------------------------------------------------------------------------

#include <vcl.h>
#pragma hdrstop
#include "fstream.h"
#include "calibracion.h"
//---------------------------------------------------------------------------
#pragma package(smart_init)
#pragma resource "*.dfm"
```

```
Tfrmcalibracion *frmcalibracion;
//---------------------------------------------------------------------------
__fastcall Tfrmcalibracion::Tfrmcalibracion(TComponent* Owner)
	: TForm(Owner)
{
}
//---------------------------------------------------------------------------

void __fastcall Tfrmcalibracion::oxKeyPress(TObject *Sender, char &Key)
{
	if((Key<48||Key>57)&&Key!='.'&&Key!=8&&Key!='-')
		Key=0;
}
//---------------------------------------------------------------------------

void __fastcall Tfrmcalibracion::Button3Click(TObject *Sender)
{
	this->Close();
	this->Release();
}
//---------------------------------------------------------------------------
void __fastcall Tfrmcalibracion::Button1Click(TObject *Sender)
{
	AnsiString Ruta=ExtractFilePath(Application->ExeName);
	Ruta+="cnf.cnf";
	ofstream Archivo(Ruta.c_str());
	Archivo<<ox->Text.c_str()<<endl;
	Archivo<<oy->Text.c_str()<<endl;
	Archivo<<fc->Text.c_str()<<endl;
	Archivo<<os->Text.c_str()<<endl;
	Archivo<<ohex->Text.c_str()<<endl;
	Archivo.close();
	Button2Click(Sender);
	this->Close();
	this->Release();
}
//---------------------------------------------------------------------------
void __fastcall Tfrmcalibracion::Button2Click(TObject *Sender)
{
	AnsiString Ruta=ExtractFilePath(Application->ExeName);
	int orxc,oryc;
	double factorc,restafactc,offsetfactc;
	Ruta+="cnf.cnf";
	ifstream Archivo(Ruta.c_str());
	  Archivo>>orxc;
	  ox->Text=orxc;
	  Archivo>>oryc;
	  oy->Text=oryc;
	  Archivo>>factorc;
	  fc->Text=factorc;
```

```
        Archivo>>restafactc;
        os->Text=restafactc;
        Archivo>>offsetfactc;
        ohex->Text=offsetfactc;
}
//---------------------------------------------------------------------------
void __fastcall Tfrmcalibracion::FormActivate(TObject *Sender)
{
      Button2Click(Sender);
}
//---------------------------------------------------------------------------
```

Calibración.h

****

```
//---------------------------------------------------------------------------

#ifndef calibracionH
#define calibracionH
//---------------------------------------------------------------------------
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
//---------------------------------------------------------------------------
class Tfrmcalibracion : public TForm
{
__published:   // IDE-managed Components
      TLabel *Label1;
      TLabel *Label2;
      TEdit *ox;
      TEdit *oy;
      TLabel *Label3;
      TLabel *Label4;
      TEdit *ohex;
      TEdit *fc;
      TLabel *Label5;
      TEdit *os;
      TButton *Button1;
      TButton *Button2;
      TButton *Button3;
      void __fastcall oxKeyPress(TObject *Sender, char &Key);
      void __fastcall Button3Click(TObject *Sender);
      void __fastcall Button1Click(TObject *Sender);
      void __fastcall Button2Click(TObject *Sender);
      void __fastcall FormActivate(TObject *Sender);
```

```
private:        // User declarations
public:         // User declarations
        __fastcall Tfrmcalibracion(TComponent* Owner);
};
//---------------------------------------------------------------------------
extern PACKAGE Tfrmcalibracion *frmcalibracion;
//---------------------------------------------------------------------------
#endif
```