INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

No. 118 **Serie:** VERDE **Fecha:** Abril 2007

Diseño y Construcción de un Sistema Básico de Interacción y Acción con Procesos Reales Reconfigurable y de Bajo Costo HMI-CIC

Antonio Hernández Zavala[1] Oscar Camacho Nieto[1]
Osvaldo Espinosa Sosa[1] Leopoldo Galindo Soria[2]
Alfonso Gutiérrez Aldana[1]

RESUMEN

Con el objetivo de reducir la brecha tecnológica que se tiene ante países de primer mundo, en relación con el control de procesos industriales, en este documento se presenta el diseño y construcción de un sistema de control y adquisición de datos básico y reconfigurable, compuesto de tres partes principales: un software de Interfaz Hombre-Máquina HMI basado en PC, una tarjeta de adquisición de datos de propósito general que utiliza un microcontrolador para interactuar con procesos reales a través de 16 canales analógicos de 10 bits de resolución y 20 digitales; y finalmente la tercera parte corresponde a la variable física es decir, al proceso y acondicionamiento del mismo. Esta última parte es variable y está en dependencia del proceso estudiado. Como caso de ejemplo práctico se utiliza un tanque con agua donde se introducen seis sensores de temperatura colocados a distinta profundidad y un calentador para visualizar o modificar su estado. Finalmente se obtuvo un desarrollo tecnológico propio, que es versátil, realizado a bajo costo y puede ser utilizado ya sea como apoyo didáctico en los niveles medio superior y superior o por el sector industrial en lo general, para diversos casos de estudio con distintos procesos.

Palabras Clave: SCADA, Adquisición de Datos, Control. Variable, Microcontrolador uC, Interfaz Hombre-Máquina HMI.

1Centro de Investigación en computación-IPN
2Escuela Superior de Ingeniería Mecánica y Eléctrica, Unidad Zacatenco-IPN

Copyright © 2007
Instituto Politécnico Nacional
Centro de Investigación en Computación
Av. Juan de Dios Bátiz casi esq.
Miguel Othón de Mendizabal Ote.
México, 07738, D.F.
ISBN 10: 970-36-0406-4
ISBN 13: 978-970-36-0406-7
Impreso en México

ADVERTENCIA

Este reporte contiene información desarrollada por el Centro de Investigación en Computación del Instituto Politécnico Nacional a partir de datos y documentos con derechos de propiedad y por lo tanto su uso queda restringido a las aplicaciones que explícitamente se convenga.

La aplicación no convenida exime al Centro su responsabilidad técnica y da lugar a las consecuencias legales que para efecto se determinen.

Información adicional sobre este reporte podrá obtenerse acudiendo a la Unidad de Publicaciones y Reportes Técnicos del Centro de Investigación en Computación.
Av. Juan de Dios Batiz s/n en la unidad profesional Adolfo López Mateos.
Teléfono 5729-6000 ext. 56571

INDICE

Introducción		5
Capítulo I.	Interpretación Eléctrica de la Realidad	7
1.1	Sensores	7
1.2	Acondicionamiento de la señal	8
1.3	Acción sobre un proceso	8
Capítulo II.	Tarjeta de Adquisición de Datos	9
2.1	Conversión Analógico a Digital	10
2.2	Conversión Digital a Analógica	11
2.3	Transmisión Serial asíncrona RS-232	12
2.4	Pantalla LCD	13
2.5	Reset e Interrupción Externa	13
2.6	Unidad Central de Procesamiento	14
2.6.1	Arquitectura	14
2.6.2	Registros	15
Capítulo III.	Interfaz Hombre-Máquina	17
3.1	Algoritmo	17
3.2	Interpretación de la información	18
3.3	Manejo de Archivos	19
3.4	Inicio y Paro de Lecturas	21
Capítulo IV.	Resultados y Conclusiones	22
Lista de componentes		27
Bibliografía		28

INTRODUCCIÓN

Los procesos físicos en tiempo real de la industria, demandan sistemas de supervisión que proporcionen datos confiables a través de una interfaz de fácil uso y además a bajo costo. Actualmente es viable contar con una computadora, que al incorporarla dentro de los bloques de ejecución de un proceso de control, se puede interactuar con el ambiente de la variable, para ver sus cambios y actuar sobre ella según lo dicte el usuario. Un sistema que permite interacción directa con procesos reales por medio de una computadora personal es comúnmente llamado SCADA (System of Control And Data Acquisition).

Los desarrolladores de sistemas SCADA comerciales, proporcionan todas las herramientas necesarias para interactuar con procesos reales, desde controladores y tarjetas de adquisición de datos hasta software de HMI programable a través de su herramienta de programación, que es compatible con todos los demás elementos, ya que contiene las librerías específicas para cada uno. La desventaja de estos sistemas comerciales, es el alto costo que presentan y la dependencia en relación a la actualización de software, así como el mantenimiento preventivo y correctivo del hardware, lo cual los hace inaccesibles tanto a investigadores como a los estudiantes.

Por el lado de los estudiantes de ingeniería relacionados con el control de procesos, la mayor parte de ellos, carecen de los recursos apropiados para acceder a dichos sistemas, ya sea para poder manipularlos, modificarlos o simplemente conocer como trabajan comprobando de esta forma sus conocimientos que a futuro le serán útiles; por lo tanto, nos vemos en la necesidad de generar nuestros propios sistemas de adquisición de datos.

En el presente documento, se describe la realización de una plataforma básica para monitoreo y control de procesos en tiempo real SCADA que se caracteriza por utilizar herramientas y componentes genéricos para la creación de cada modulo.

El sistema en general consta de tres bloques principales:

1. Interfaz Hombre-Máquina HMI Gráfica basada en PC [1][2], creado en un lenguaje de programación de uso general de alto nivel orientado a objetos C++Builder 5.0 [3][4], desde donde se puede monitorear, analizar y afectar al proceso.
2. Hardware dedicado al envió de mediciones al software HMI a través de interfaz serial RS-232, digitalización de muestras provenientes de los sensores en hasta 16 canales de entrada analógica con 10 bits de resolución, generación de hasta 20 entradas-salidas digitales, reconstrucción de señal

DAC de 8 bits, pantalla LCD, todo esto a través de un procesador RISC de tipo harvard de propósito general a 20MHz.

3. Finalmente la etapa de acondicionamiento de señal correspondiente a cada sensor y a cada actuador involucrado en el proceso en estudio, cabe mencionar, que está etapa puede variar de acuerdo con cada elemento distinto, pero todos deben ser acondicionados a operar en el rango de [0-5] Vcc.

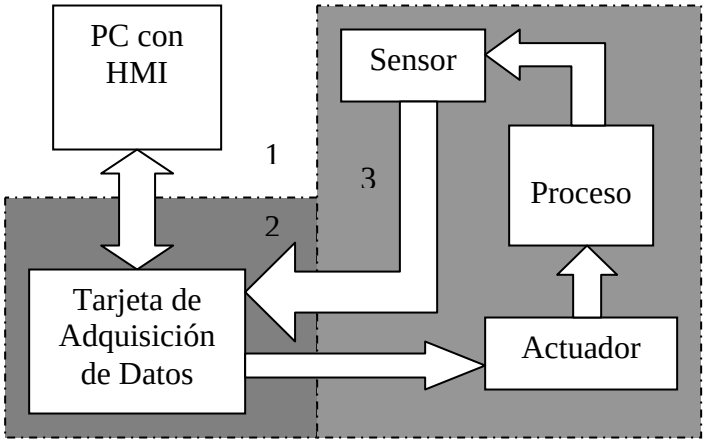

Figura 1. Bloques del Sistema.

Como ejemplo práctico y de pruebas, se mide la distribución de temperatura en un tanque de agua, se utilizan sensores de temperatura colocados en distintos puntos dentro del tanque, son medidos a través del microcontrolador y enviados a la PC, donde se monitorea y manipula la temperatura del tanque a través de la interfaz Hombre-Maquina.

El documento está dividido en cuatro capítulos principales: en el primero se trata al proceso (**Fig.1 sección 3**), es decir, los aspectos de captación y acondicionamiento de señal con la circuitería necesaria para obtener un equivalente en voltaje de la temperatura del agua en distintos puntos, hasta obtener un valor digital para cada sensor, la interpretación del dato obtenido y el envío del mismo a través de una interfaz serial RS-232, el capítulo se llama *Interpretación eléctrica de la realidad*; el segundo capítulo corresponde al diseño de la tarjeta de adquisición de datos (**Fig. 1 sección 2**). El tercer capítulo corresponde a la interfaz HMI en PC (**Fig.1 sección 1**), que se encarga de recolectar las mediciones y presentarlas al usuario para que éste pueda analizar los datos y de acuerdo a ello tomar una decisión de control. Finalmente, en el cuarto capítulo se presentan los resultados y las conclusiones del presente trabajo.

Capítulo I. Interpretación eléctrica de la realidad.

En esta parte, se trata de la interpretación de una variable física real, mediante dispositivos electrónicos, tomando como un ejemplo de variable física real a la temperatura: existen variaciones que podemos sentir y en ocasiones medir haciendo uso de un termómetro, para poder dar a entender algo físico a una computadora, es necesario contar con una serie de etapas con dispositivos electrónicos que transformen el valor real a un equivalente eléctrico variable de acuerdo a los cambios de la misma variable física, en la siguiente sección se presenta la forma de trabajo de dichos dispositivos, comúnmente llamados sensores.

1.1 Sensores.

Dado que se pretende utilizar temperatura como variable física a medir, dadas su simplicidad y características lineales de operación [5], se utilizan sensores de temperatura LM335 en configuración con calibrador según el fabricante [6].

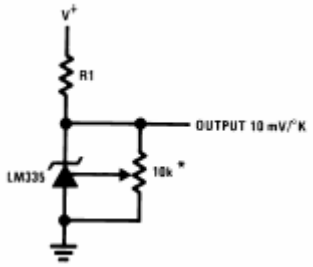

Figura 2. Circuito para sensor de temperatura [6]

Este sensor entrega una señal en voltaje de 10mV por cada grado Kelvin, proporcionando una salida con comportamiento lineal; el rango de temperatura que soporta va de -40ºC a 100ºC. Para efectos de calibración, el sensor entrega 2.982V a 25ºC, estos valores se utilizan como referencia para determinar el resto de valores, haciendo uso de la ecuación 1 donde T es la temperatura desconocida en grados centígrados, T_0 es la temperatura de referencia [6] y *Vsalida* es el nivel de voltaje obtenido para la temperatura T.

$$Vsalida_T = Vsalida_{T_0} \times \frac{T}{T_0} \qquad (1)$$

De la ecuación anterior, se requiere saber que temperatura tiene el ambiente de la variable cuando se tiene el voltaje $Vsalida_T$ en la salida del sensor, entonces se replantea la ecuación 1, obteniendo:

$$T = T_0 \left(\frac{Vsalida_T}{Vsalida_{T_0}} \right) \qquad (2)$$

Cabe mencionar que las unidades de medición para la temperatura entregadas por el sensor son grados kelvin, por lo tanto, se utiliza la siguiente formula:

$$T^{o}K = T^{o}C + 273 \qquad (3)$$

1.2 Acondicionamiento de Señal

Con la finalidad de no demandar al sensor una corriente tal que pueda alterar su operación, se agrega un amplificador operacional en configuración de seguidor [7], este último proporciona al Convertidor Analógico Digital (por sus siglas en inglés ADC) una señal confiable, también se incorpora un filtro antialias para la frecuencia de 60Hz. El circuito resultante para cada sensor se presenta en la **figura 3**, donde *o1* es la temperatura en voltaje; en la **figura 4** puede apreciar la tarjeta que contiene los circuitos de acondicionamiento para los seis sensores de temperatura utilizados.

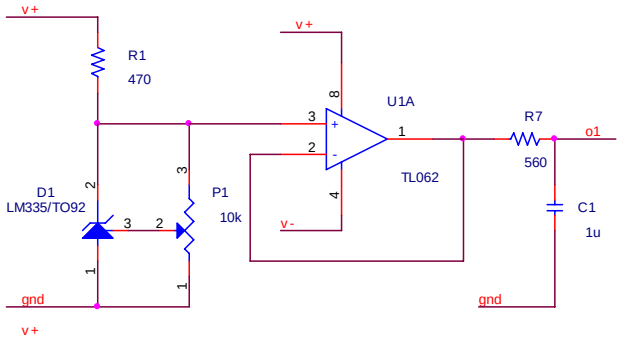

Figura 3. Acondicionamiento de un sensor.

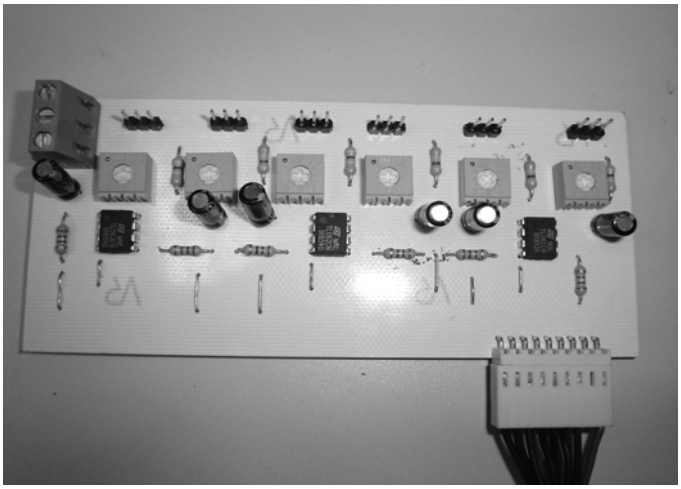

Figura 4. Tarjeta de acondicionamiento de señal.

Al no realizar correctamente el acondicionamiento del sensor correspondiente, se producen errores de inconsistencia en el dato obtenido, ya sea por una mala calibración o por ruido en el medio, que afecta directamente la medición.

1.3 Acción sobre un proceso.

Para actuar sobre el proceso existen diversas formas de hacerlo desde una computadora, pero todas ellas requieren de una circuitería extra, para esto, la tarjeta desarrollada, cuenta con salidas digitales donde se puede activar o desactivar un relevador, además de esto, se cuenta con un Convertidos Digital a Analógico (por sus siglas en inglés

DAC) de 8 bits, 3 salidas con modulación de ancho de pulso (por sus siglas en ingles PWM) y contador de eventos externos [7], que serán detallados posteriormente.

Después de haber comprendido la forma de captación y acondicionamiento de la señal, el siguiente capitulo se enfoca en el diseño y construcción de la tarjeta usada en la construcción del prototipo.

Capítulo II. Tarjeta de Adquisición de Datos

En esta parte, se tratara acerca de la realización de la tarjeta de adquisición de datos, encargada de muestrear cada canal analógico utilizado, a través de un ADC, posteriormente al haber capturado las muestras necesarias, las manda vía RS-232, a través de una USART; también se incluyen salidas tanto analógicas como digitales de distintos tipos. En la **figura 5**, se muestra el diagrama a bloques de la tarjeta y los dispositivos involucrados, se utiliza un microcontrolador de propósito general COP8 como parte central de esta tarjeta.

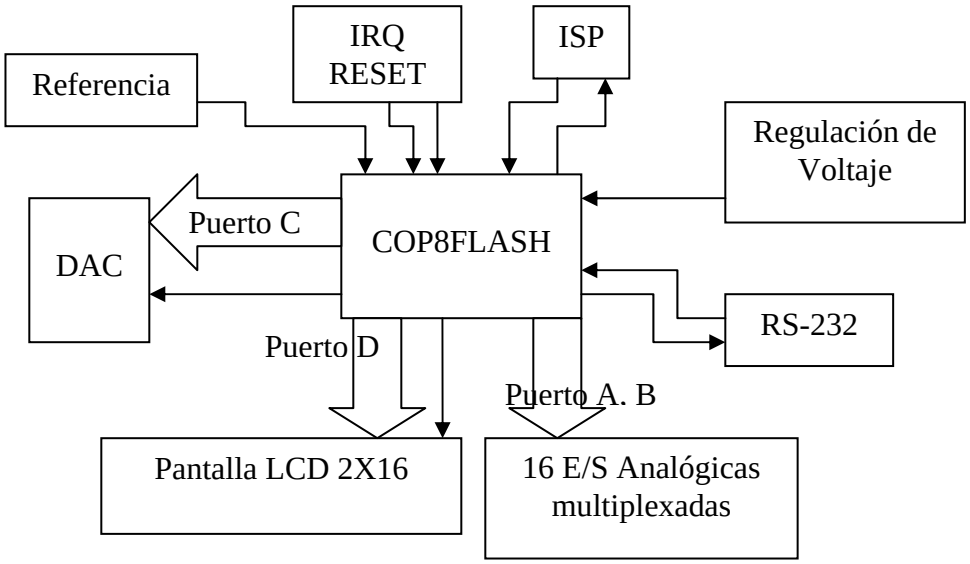

Figura 5. Bloques de la tarjeta de adquisición de datos.

En la **Tabla 1** se muestra la conexión entre el pin del microcontrolador uC y el dispositivo externo correspondiente, mencionando todas las entradas y salidas, posteriormente se hablará de cada dispositivo por separado.

Tabla 1. Conexión de microcontrolador y los dispositivos externos.

Puerto en uC	Dispositivo Externo	Descripción
	LCD	Pantalla de cristal liquido 2x16
L5	RS	Selección de Registros
L6	R/W	Lectura o escritura de datos
L7	E	Habilitación de chip
D0 – D7	D0 – D7	Bus para salida de datos
	MAX232	Transmisor - receptor para interfaz serial RS232
L2	Transmisor	Transmisión de datos
L3	Receptor	Recepción de datos
	TLC7524	Convertidor Digital a Analógico
C0-C7	DB0-DB7	Dato de entrada a conversión
L0	CS	Selección de chip
L1	WR	Petición de Conversión
	Entrada/Salida	Canales disponibles para comunicación con el exterior
A0-A7	Conector 1	E Analógica o E/S digital
B0-B7	Conector 2	E Analógica o E/S digital
E0-E5	Teclado	Dato de entrada
F0-F3	Conector 3	Drive de motor a pasos externo
G0	Botón 1	Señal de interrupción externa
G1	Conector 4	Salida WatchDog
G2	Conector 4	Timer 1 A
G3	Conector 4	Timer 1 B
G4	Pin 4 RJ-11	ISP
G5	Pin 2 RJ-11	ISP
G6	Pin 3 RJ-11	ISP
G7	Cristal	Reloj del sistema
CKI	Cristal	Reloj del sistema
RESET	Botón 2	Reinicio del sistema

2.1 Conversión Analógico a Digital.

Una vez que esta lista la etapa de captura y acondicionamiento de señal, se procede a obtener un valor digital a partir del voltaje obtenido, para lo cual, se utiliza un ADC de 16 canales de 10 bits, que opera con el método de aproximaciones sucesivas, disponible a través de los puertos A y B, del *uC*, este convertidor realiza la conversión de un dato en un tiempo fijo de 15 ciclos de reloj, transcurrido este tiempo, el dato puede ser leído por la unidad de control accediendo al registro de datos correspondiente.

El convertidor se configura a través del registro ENAD según los siguientes bits [8], ver **Figura 6**:

ADCH3 - ADCH0	Bits de selección de canal.
ADMOD	Modo simple o Diferencial.
MUX	Habilita la salida multiplexada.
PSC	Reloj directo o dividido por 16.
ADBSY	Ocupado en conversión. Inicio de conversión.

Bit 7							Bit 0
Channel Select				Mode Select	Mux/Out	Prescale	Busy
ADCH3	ADCH2	ADCH1	ADCH0	ADMOD	MUX	PSC	ADBSY

Figura 6. Registro ENAD [2].

El resultado de la conversión queda disponible en los registros ADRSTH y ADRSTL, cuando ADBSY es borrado por hardware al completar una conversión. Los registros ADRSTH y ADRSTL, son de 8 bits, y contienen un dato de 10 bits según la **figura 7**, estos datos son almacenados en memoria para cuando se haya completado una secuencia de una muestra de cada canal que serán enviados como paquete hacia la PC donde serán decodificados.

Bit 7							Bit 0
Bit 1	Bit 0	0	0	0	0	0	0

a) ADRSTL

Bit 7							Bit 0
Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2

b) ADRSTH

Figura 7. Registros ADRSTH y ADRSTL [2]

Se utilizan 6 canales en modo simple, el reloj que se utiliza es igual al del microcontrolador que es de 20MHz, el ciclo de instrucción es de 0.5ms y completa una muestra en 7.5us. Esta velocidad puede modificarse mediante los registros de configuración. Note que mientras más canales se utilicen más tiempo pasara en volver a tomar una lectura del mismo canal por lo que el muestreo es más espaciado.

El voltaje de referencia para la parte analógica del circuito, es decir AVCC en uC y el DAC, esta dada por el circuito REF02 calibrado a 5Vcc.

2.2 Conversión Digital a Analógica

Para transformar un dato binario de 8 bits proveniente de la unidad de control a su equivalente en voltaje de 0 a 5 volts, se utiliza un convertidor digital a analógico (DAC) de 8 bits [9], de red tipo R–2R [10], configurado en modo voltaje como el de la **Figura 8**.

Cada incremento binario en la entrada se presenta como una variación de 0.02v a la salida, esta función es dada por:

$$Vo = 5\frac{dato}{256} \qquad \textbf{(4)}$$

donde *Vo* es el voltaje a la salida y *dato* es el valor binario en la entrada.

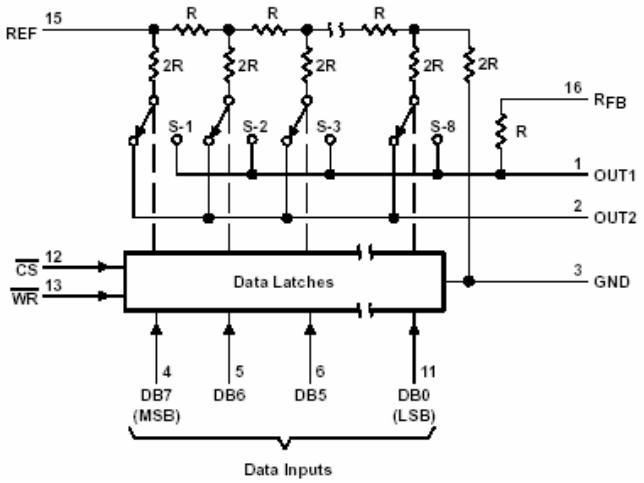

Figura 8. Convertidor Digital a Analógico en modo voltaje.

La secuencia de control para tener un nuevo valor en la salida según la entrada, esta dada por la siguiente gráfica, **Figura 9** esta será realizada por la unidad de control [9].

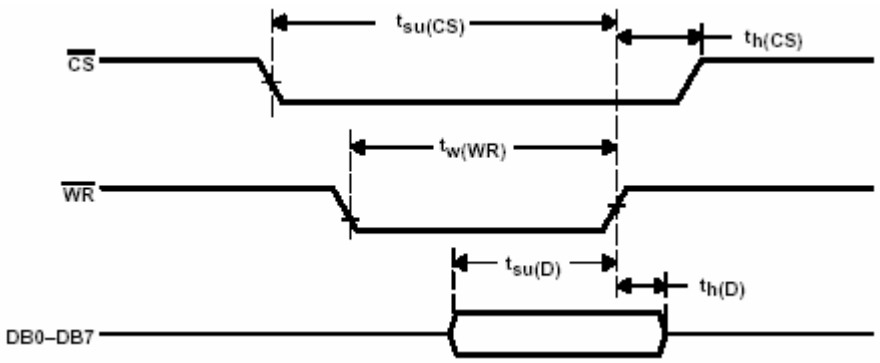

Figura 9. Secuencia de control del DAC.

2.3 Transmisión serial asíncrona RS-232

Cuando queremos visualizar la temperatura de una manera más agradable es necesario enviar las lecturas hacia una computadora; una forma de comunicación de lo más simple y económica es la transmisión serial RS-232 asíncrona, en donde el receptor espera un dato para generar una interrupción, entregándolo a la rutina encargada de decidir que acción hay que tomar, ver **Figura 10**.

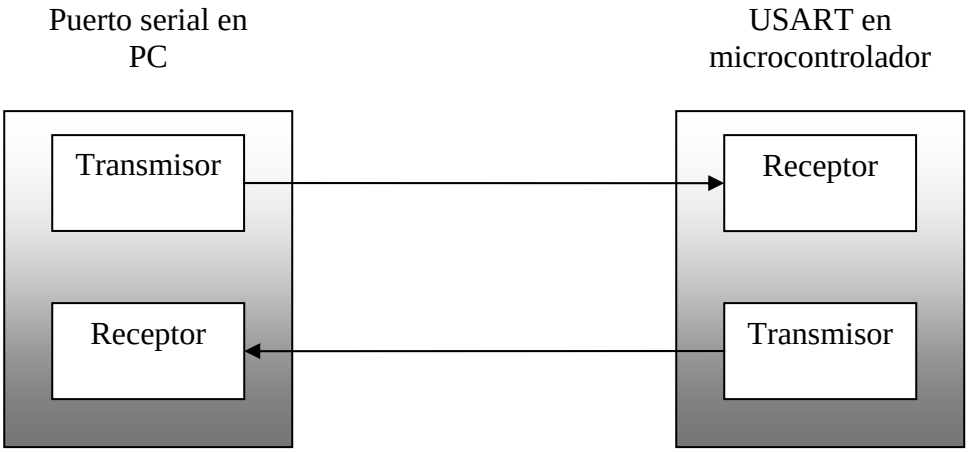

Figura 10. Transmisión Serial

La USART en el microcontrolador MCU, es configurada para trabajar a 9600 *Bps*, con un *bit* de paridad par, un *bit* de paro, en palabras de 8 *bits*. Configurando los registros de MCU con los siguientes valores [11]:

ENU	10100000	Establece una palabra de ocho bits con paridad par
ENUI	00100000	Un bit de paro, transmisión Asíncrona, pin L2 como transmisor.
PSR	11001000	Velocidad 9600 [P=13, (N-1)=9]
BAUD	00001001	

2.4 Pantalla LCD

Con el propósito de visualizar los datos que están fluyendo en el interior del controlador se utiliza una pantalla de cristal líquido LCD de 2 líneas de 16 caracteres cada una, modelo TM162-AAA6-2 [12], su diagrama a bloques se muestra en la **Figura 11.**

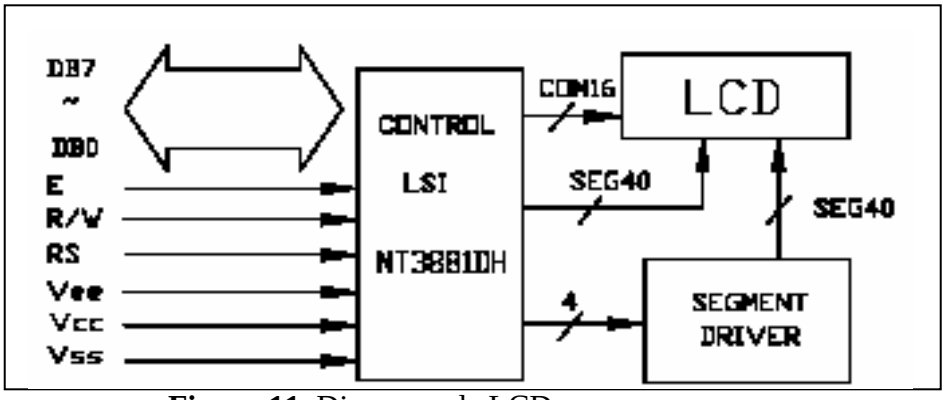

Figura 11. Diagrama de LCD

La pantalla LCD solo se utiliza con propósitos de visualización de datos.

2.5 Reset e Interrupción Externa

Los circuitos realizados para los botones de reinicio (reset) e interrupción Externa (EXT_IRQ) son idénticos, al presionar el botón, se genera una señal que ocasiona el resultado deseado al conectarse al *pin* indicado. El circuito se muestra en la **Figura 12**.

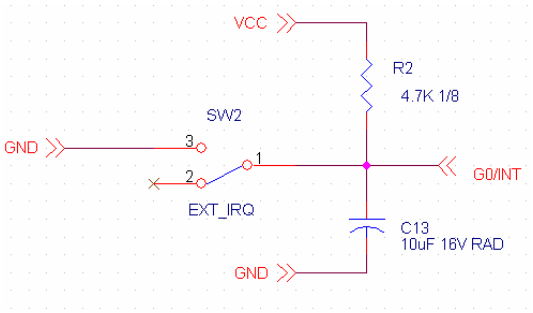

Figura 12. Circuito para reset e interrupción externa.

2.6 Unidad Central de Procesamiento

Un microcontrolador es un circuito integrado de alta escala de integración que incorpora la mayor parte de los elementos que configuran un controlador [13].

2.6.1 Arquitectura

En este caso, para implementar la unidad de control, se utiliza un microcontrolador COP8 [7], ver **Figura 13** que se encarga del manejo de la entrada/salida entre los sensores y la PC, haciendo uso de los recursos disponibles en el microcontrolador como se menciono anteriormente.

Las características principales del microcontrolador que se utiliza son el contar con arquitectura tipo Harvard, y un conjunto de instrucciones RISC. 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.

En esta arquitectura, la carga de instrucción y las transferencias de datos, se traslapan mediante una segmentación de dos etapas, lo cual permite que la siguiente instrucción sea cargada de memoria mientras la instrucción actual esta siendo ejecutada usando la memoria de datos. Lo anterior no es posible en una arquitectura Von Newman, debido a que tiene un bus único para datos e instrucciones. La CPU puede realizar las operaciones suma, resta, lógicas o corrimientos en un ciclo de instrucción.

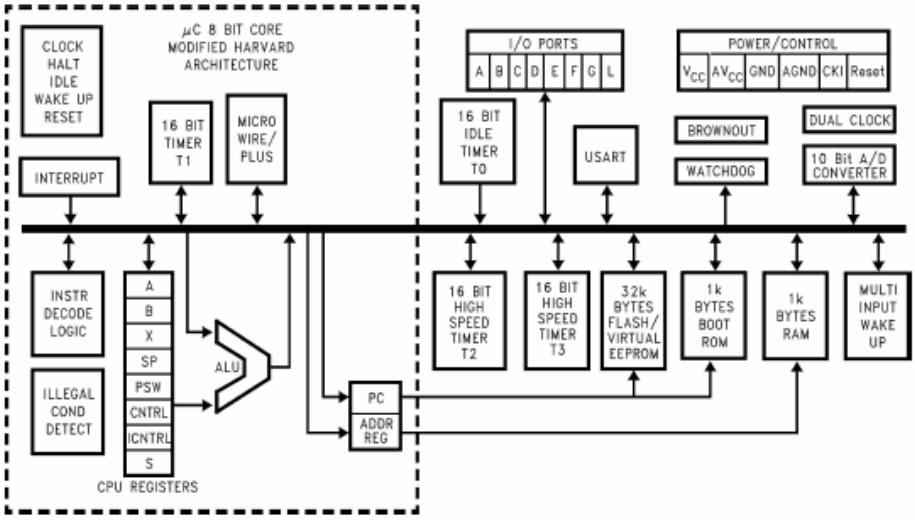

Figura 13. Arquitectura de COP8CDR

2.6.2 Registros

A continuación se mencionan los registros de control y de configuración del microcontrolador que serán utilizados.

Existen seis registros de CPU disponibles para el usuario:

- A es el registro acumulador de 8 bits.
- PC es el contador de programa de 15 bits.
  - PU son los 7 bits más significativos de PC.
  - PL son los 8 bits menos significativos de PC.
- B es un apuntador de direcciones RAM de 8-bit, que puede opcionalmente ser auto incremento o decremento.
- X es un apuntador alternativo de direcciones RAM de 8-bit, que puede opcionalmente ser auto incremento o decremento.
- S es el registro de direcciones del segmento de datos de 8-bit usado para extender la mitad menor del rango (00 a 7F) en 256 segmentos de datos de 128 bytes cada uno
- SP es el apuntador de pila de 8-bits que apunta al lugar de interrupción de la subrutina en RAM.

Registros de Configuración.

PSW Este registro se utiliza para configurar las interrupciones.

HC	C	T1PNDA	T1ENA	EXPND	BUSY	EXEN	GIE
Bit 7							bit 0

- HC Bandera de Medio Acarreo
- C Bandera de acarreo
- T1PNDA Bandera de interrupción pendiente de Timer T1
- T1ENA Habilita interrupción en Timer T1
- EXPND Interrupción externa pendiente.
- BUSY Bandera de Ocupado para MICROWIRE/PLUS
- EXEN Habilita interrupción externa.
- GIE Habilita interrupción global

ENAD Este registro permite hacer uso del convertidor AD integrado al microcontrolador.

ADCH3	ADCH2	ADCH1	ADCH0	ADMOD	MUX	PSC	ADBSY
Selección de canal				Modo	Salida de Mux	Preescala	Ocupado
Bit 7							Bit 0

- ADCH3 bit de selección de canal de ADC.
- ADCH2 bit de selección de canal de ADC.
- ADCH1 bit de selección de canal de ADC.
- ADCH0 bit de selección de canal de ADC.
- ADMOD coloca al ADC en modo simple o diferencial.
- MUX habilita la salida del multiplexor del ADC.
- PSC Cambia el reloj del ADC entre dividir por uno o por dieciséis MCLK.
- ADBSY significa que el ADC esta ocupado ejecutando una conversión. Cuando es establecido por el usuario, se inicia la conversión.

Una vez que se explicó la forma de trabajo de la tarjeta de adquisición de datos, en el siguiente capítulo de presenta la interfaz hombre – maquina.

Capítulo III. Interfaz Hombre-Máquina

El software de la interfaz hombre – maquina, en inglés Human-Machine Interface (HMI) se desarrollo en el lenguaje de programación C++ Builder 5.0, que es un lenguaje de programación de uso general orientado a objetos, donde la aplicación se adapta a las necesidades que surgen para el propósito de visualizar la temperatura correspondiente a cada sensor en una forma grafica ver **figura 14**. En este lenguaje, cuando ocurre algo, se genera un evento que se encarga de llamar a la subrutina correspondiente para esa tarea en específico.

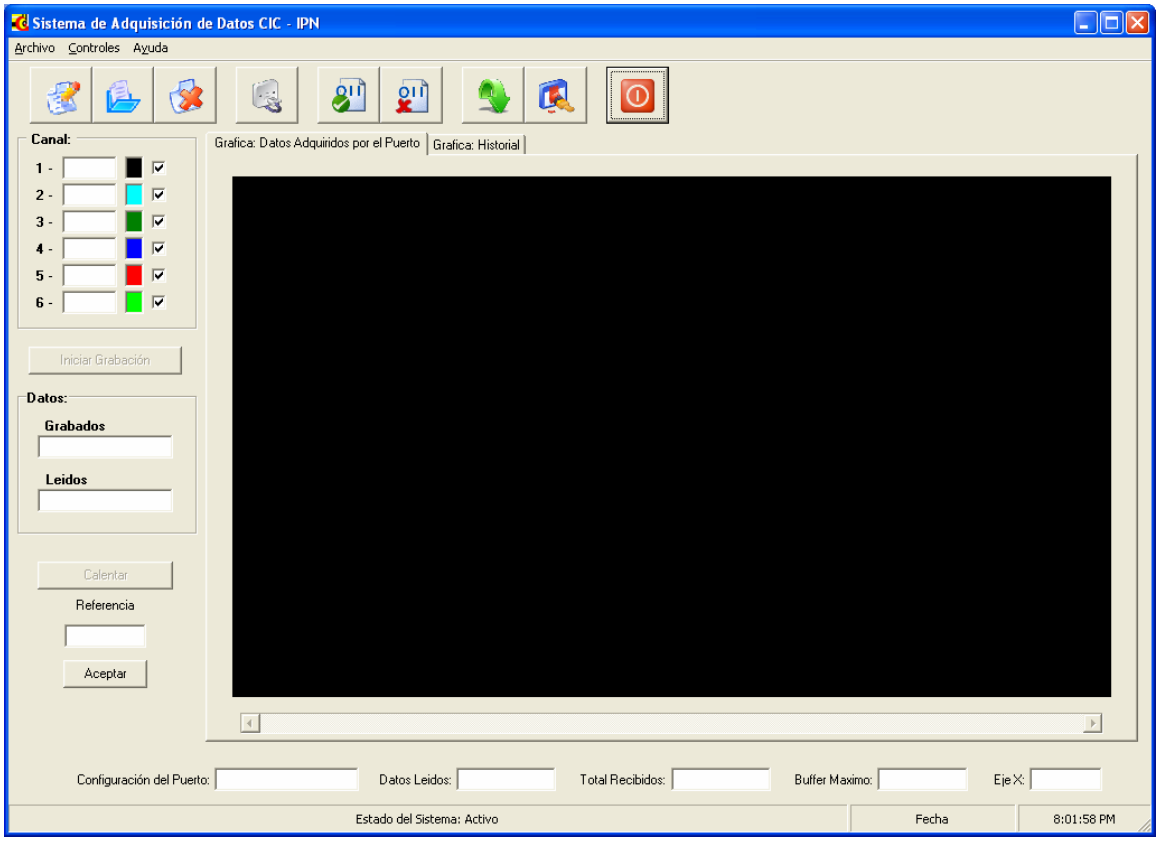

Figura 14. Pantalla principal del software HMI.

3.1 Algoritmo

El programa trabaja de la siguiente manera: primero se ejecuta una configuración e inicialización de parámetros útiles para el programa, posteriormente, el usuario debe establecer los parámetros de comunicación a través del botón correspondiente. En este momento la transmisión puede iniciarse y las curvas correspondientes al estado actual del ambiente medido serán graficadas en pantalla. Esto iterará hasta que el usuario lo desee.

En la **figura 15** se muestra el diagrama de flujo del programa dividido en dos partes, la parte *a* corresponde a la etapa de manejo de errores al establecer la comunicación, posteriormente a esto, la parte *b* realiza una iteración donde decodifica y grafica la

información recibida, esto se detendrá hasta que no haya información que procesar o el usuario lo desee.

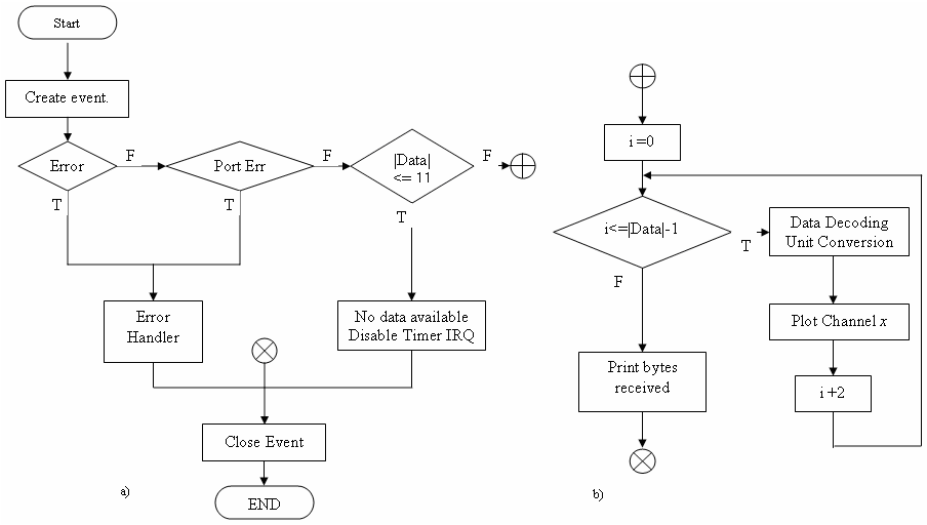

Figura 15. a) Manejo de errores, b) Decodificación de datos.

3.2 Interpretación de la Información.

Esta parte es fundamental, dado que si existen errores en la decodificación, se tendrán gráficos inconsistentes como se muestra en la siguiente figura, en la parte *a* los datos están desordenados, en la parte *b* hay ruido que altera las mediciones. Estos se corrigen al revisar la forma de decodificar, y al agregar un filtro antialías para la frecuencia de 60Hz, ver **figura 16**.

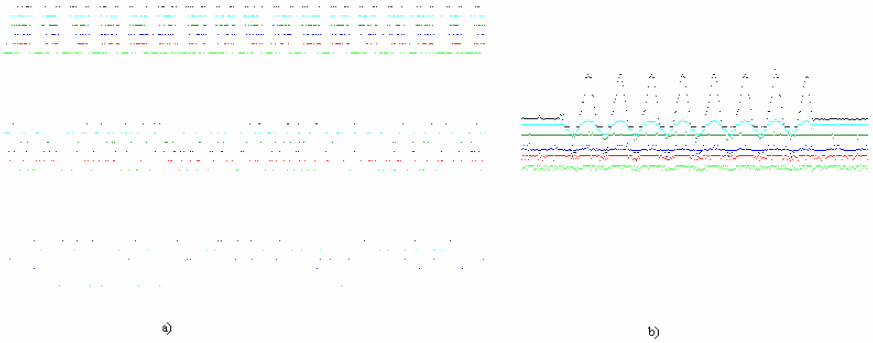

Figura 16. Errores en decodificación y ruido respectivamente.

De acuerdo con los registros de la sección 2.1, una muestra se encuentra en 10 bits, los 8 bits más significativos se encuentran en un solo registro y los dos menos significativos, son los dos más significativos de otro registro. Para lograr un dato de 10 bits es necesario hacer dos corrimientos hacia la izquierda [14] o lo que seria equivalente multiplicar por cuatro, el registro más significativo y dividir por 64 al registro menos significativo y sumarlos.

9999 9999	Dato de ocho bits MSB
0000 0099 9999 9900	Multiplicado por cuatro

9900 0000	Dato de ocho bits LSB
0000 0000 0000 0099	Dividido entre 64
0000 0099 9999 9999	Dato de 10 bits representado en dos bytes.

Que puede ser interpretado como:

$$Dato_{10b} = MSB * 4 + LSB / 64 \qquad (5)$$

Entonces, cada muestra de cada sensor estará representada por un dato de 2 bytes, por lo que se tienen entonces 12 bytes que contienen una muestra de cada sensor. A este conjunto de 12 bytes, le llamaremos "*paquete*", y será la unidad de envió de muestras hacia la terminal.

Para obtener una temperatura en grados centígrados se utiliza

$$T^{o}C = \frac{swDATA * 298}{608} - 273 \qquad (6)$$

Dado que se cuenta con un convertidor de 10 bits, se tiene un total de 1024 valores para interpretar el voltaje en [0, 5] Vcc. El rango de operación dado por el fabricante para el sensor, esta determinado de acuerdo a la **tabla 2**:

Tabla 2. Equivalencias

Voltaje	**Temperatura ºK**	**Temperatura ºC**	**Valor decimal equivalente**
2.331	233	-40	
2.341	234	-39	
2.972	297	24	
2.982	298	25	608
2.992	299	26	
3.722	372	99	
3.732	373	100	

3.3 Manejo de Archivos

Cuando la transmisión esta siendo ejecutada, el usuario puede almacenar en un archivo previamente creado las medidas obtenidas en un lapso de tiempo dado por el mismo. Las funciones de archivo que se utilizan son *Abrir*, *Guardar*, *Nuevo* y *Eliminar*, y están disponibles en el menú principal en la pestaña *Archivo*.

Para grabar una secuencia de datos obtenidos, primero se crea el archivo, una vez hecho esto, solamente habrá que presionar el botón *grabar* para iniciar y nuevamente presionarlo para detener la grabación. Si se vuelve a presionar grabar, se sobrescribirán los datos del archivo.

```
//Rutina para nuevo archivo
SaveDialog1->Title = "Nuevo Archivo";
 if (SaveDialog1->Execute())
  {
   ruta = SaveDialog1->FileName.c_str();
   if ((archivo = fopen(ruta, "w+b")) == NULL)
      {
       fprintf(stderr, "Cannot open output file.\n");
       return;
      }
   ShowMessage("Archivo Creado");
   banderaarchivo=TRUE;
   fclose(archivo);
  }
```

Después de haber grabado un dato en archivo, este se puede abrir y visualizar gráficamente, lo cual permite hacer un análisis detallado en donde se aplique un algoritmo de procesamiento de señal al dato sin afectar realmente al proceso y sin utilizar la tarjeta de adquisición.

```
//Rutina para Abrir y Graficar archivo
SaveDialog1->Title = "Abrir Archivo";
 if (SaveDialog1->Execute())
  {
   rutaabrir=SaveDialog1->FileName.c_str();
   if ((archivo = fopen(rutaabrir, "rb")) == NULL)
     {
      fprintf(stderr, "Cannot open output file.\n");
      return;
     }
   //limpiar datos de pantalla
   imagen->Canvas->Brush->Color = clWhite;
   imagen->Canvas->FillRect(Rect(0,0,ancho,alto));
   ejex=0;
   fseek(archivo,sizeof(struct datos1),SEEK_SET); //Lee datos desde el inicio
   while(!feof(archivo))
     {
      fread(&lector,sizeof(datos1) ,1,archivo);//obtiene una estructura
      //decodificacion
      for(j=0;j<=5;j++)
       {
        punto = lector.canal[j]/2;
        punto =  imagen->Height - punto;
        switch(j)
          {   //visualizacion de los datos de cada canal
           case 0: //canal 1
              imagen->Canvas->Pixels [ejex][punto]=clBlack;
              break;
               case 1: //canal 2
              imagen->Canvas->Pixels [ejex][punto+10]=clAqua;
              break;
           case 2: //canal 3
              imagen->Canvas->Pixels [ejex][punto+20]=clGreen;
              break;
```

```
      case 3: //canal 4
        imagen->Canvas->Pixels [ejex][punto+30]=clBlue;
        break;
      case 4: //canal 5
        imagen->Canvas->Pixels [ejex][punto+40]=clRed;
        break;
      case 5: //canal 6
        imagen->Canvas->Pixels [ejex][punto+50]=clLime;
        break;
     }
   }
  txtleidos->Text = z;
```

3.4 Inicio y paro de lecturas.

Cuando se desea iniciar una transmisión y se presiona el objeto correspondiente, se envía un comando a la tarjeta de adquisición que le indica la activación del TIMER, que ejecutará una rutina donde se toma una muestra de cada canal y se envía a la PC como un paquete de datos, esto se ejecutara cada vez que el TIMER genere la interrupción correspondiente. Para desactivar el TIMER y por consecuencia detener la transmisión se envía otro comando desde la PC que indica el deshabilitarlo.

```
//Control de inicio - paro
  if(portopen == TRUE)   //puerto listo
    {
     OVERLAPPED lector={0};
     char datoini[1];
     lector.hEvent =CreateEvent(NULL,TRUE,FALSE,NULL);
     if(lector.hEvent == NULL)
           ShowMessage("Error en Escritura");
     if(banderaIO == TRUE)
      {
        datoini[0]=20;     // Activa microcontrolador
        if(!WriteFile(puerto,datoini,1,&datosleidos,&lector))
          {
           if(GetLastError() != ERROR_IO_PENDING)
           ShowMessage("Error en Escritura");
          }

       Timer1->Enabled = TRUE;  //Activa timer
       cmdstartstop->Caption = "Detener Lectura";
       banderaIO = FALSE;
       tbinicio->Enabled = FALSE;
       tbparo->Enabled = TRUE;
      }
     else
      {
       Timer1->Enabled = FALSE;
       datoini[0]=21;   //Desactiva microcontrolador
       if(!WriteFile(puerto,datoini,1,&datosleidos,&lector))
          {
           if(GetLastError() != ERROR_IO_PENDING)
```

```
            ShowMessage("Error en lectura ");
           }
        cmdstartstop->Caption = "Iniciar Lectura";
        banderaIO = TRUE;
        tbinicio->Enabled = TRUE;
        tbparo->Enabled = FALSE;
       }
     CloseHandle(lector.hEvent);
     cmdcalentar->Enabled = TRUE;
   }
else
ShowMessage("El puerto esta cerrado");
```

Capítulo IV. Resultados y Conclusiones

El sistema resultante tiene las siguientes características en sus dos módulos principales:

Software HMI

- Ambiente gráfico de fácil uso.
- Área de graficado amplia.
- Manejo de Archivos.
- Protocolo de comunicación de acuerdo a la tarjeta de adquisición.
- Facilidad de extensión a distintos procesos.
- Facilidad de modificar el código fuente y el entorno.

Tarjeta de Adquisición de datos

- Microprocesador central de arquitectura harvard a una velocidad de reloj de 20MHz.
- Convertidor Analógico a Digital ADC de 16 canales de 10 bits referenciados de 0 a 5Vcc.
- Pantalla LCD de 2 líneas de 16 caracteres para mensajes en sitio.
- Expansión para teclado matricial.
- Hasta 20 entradas/salidas digitales.
- Convertidor Digital a Analógico DAC de 8 bits.
- Conexiones para Timers de eventos externos.
- Comunicación Serial RS-232
- Programación en sitio ISP

Con el diseño y construcción del sistema presentado, se generó un desarrollo tecnológico a bajo costo, lo cual nos ayudará a tener herramientas industriales y educativas de primer mundo. Con esto, tenemos una herramienta de monitoreo que puede ser utilizada con distintos procesos industriales previamente acondicionados para operar de 0 a 5 Vcc reduciendo considerablemente el costo de adquisición.

Como ejemplo de aplicación se utiliza un tanque con agua donde se colocan sensores de temperatura en distintos puntos y un calentador, al monitorear la distribución de la temperatura, se encuentra que el sistema efectivamente visualiza los cambios en el medio.

Este sistema puede ser utilizado como controlador multipropósito, dado que cuenta con estradas y salidas tanto analógicas como digitales, por lo que es capaz de interactuar con procesos reales previamente acondicionados a operar en el rango de 0 a 5 Volts CC, que es el voltaje de referencia que utiliza la tarjeta tanto para entradas como para salidas.

Puede ser utilizado como controlador de lazo cerrado con múltiples parámetros a valorar y accionar, entonces puede ser configurado para trabajar como sistema SISO, MISO o MIMO. Debido a que solo tiene un procesador, mientras mas entradas y salidas sean utilizadas, el tiempo de procesamiento decrece, alcanzando su máxima velocidad cuando se trata de un sistema de una sola entrada y una sola salida SISO.

La etapa de software para el controlador, puede ser implementada ya sea en lenguaje de programación para la PC o en lenguaje ensamblador del microcontrolador, siendo esta última más rápida, dado el número de instrucciones que realiza para lograr el objetivo de control.

Todos los gráficos presentados fueron captados por este software, note que algunos de ellos están reproducidos desde archivo y las medidas tienen un espaciado que se coloca para permitir la visibilidad y el análisis, pero no sucede en gráficos en tiempo real, donde si tienen la misma medida, entonces estarán traslapados.

En la **figura 17**, se muestra un error producido por el consumo de corriente del calentador al encenderse, esto se corrige al independizar la fuente del calentador.

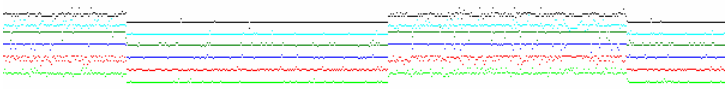

Figura 17. Calentador alterando la medición.

En la **figura 18** se muestran lecturas correctas a una temperatura uniforme, la parte *a* esta reproducida por lo que tiene espaciado, en la parte *b* la misma medida pero en tiempo real, note que uno de estos tiene una medida distinta debido a que no se encentra dentro del agua.

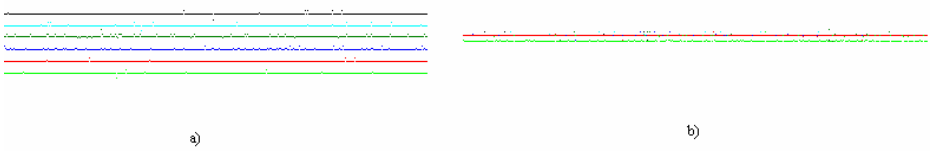

Figura 18. Temperatura uniforme. a) Reproducida, b) Tiempo real.

Cuando el calentador está encendido la temperatura del agua incrementa debido al efecto del calentador fluyendo desde el origen, como se muestra en la **figura 19.**

Figura 19. Distribución de la temperatura.

Con respecto al costo del software, es de fácil acceso, muy versátil y cualquier programa realizado puede ser ampliamente escalado. Además de esto el protocolo de comunicación puede ser tan simple como lo demande la aplicación. Por el lado del hardware, se redujeron considerablemente los costos debido a que todo el sistema fue diseñado con componentes económicos, lo cual aun puede modificarse tomando en cuenta que a grandes volúmenes se reduce el costo.

Finalmente se hace hincapié en que este trabajo tiene la proyección de continuar creciendo, por lo que, se busca la adición de nuevas herramientas que permitan un mejor uso y entendimiento de los procesos en cuestión. Además de mejorar las funciones existentes tanto en el lado de la PC como del microcontrolador.

Las **figuras 20 y 21** respectivamente, presentan el diagrama esquemático del circuito correspondiente a la tarjeta de adquisición.

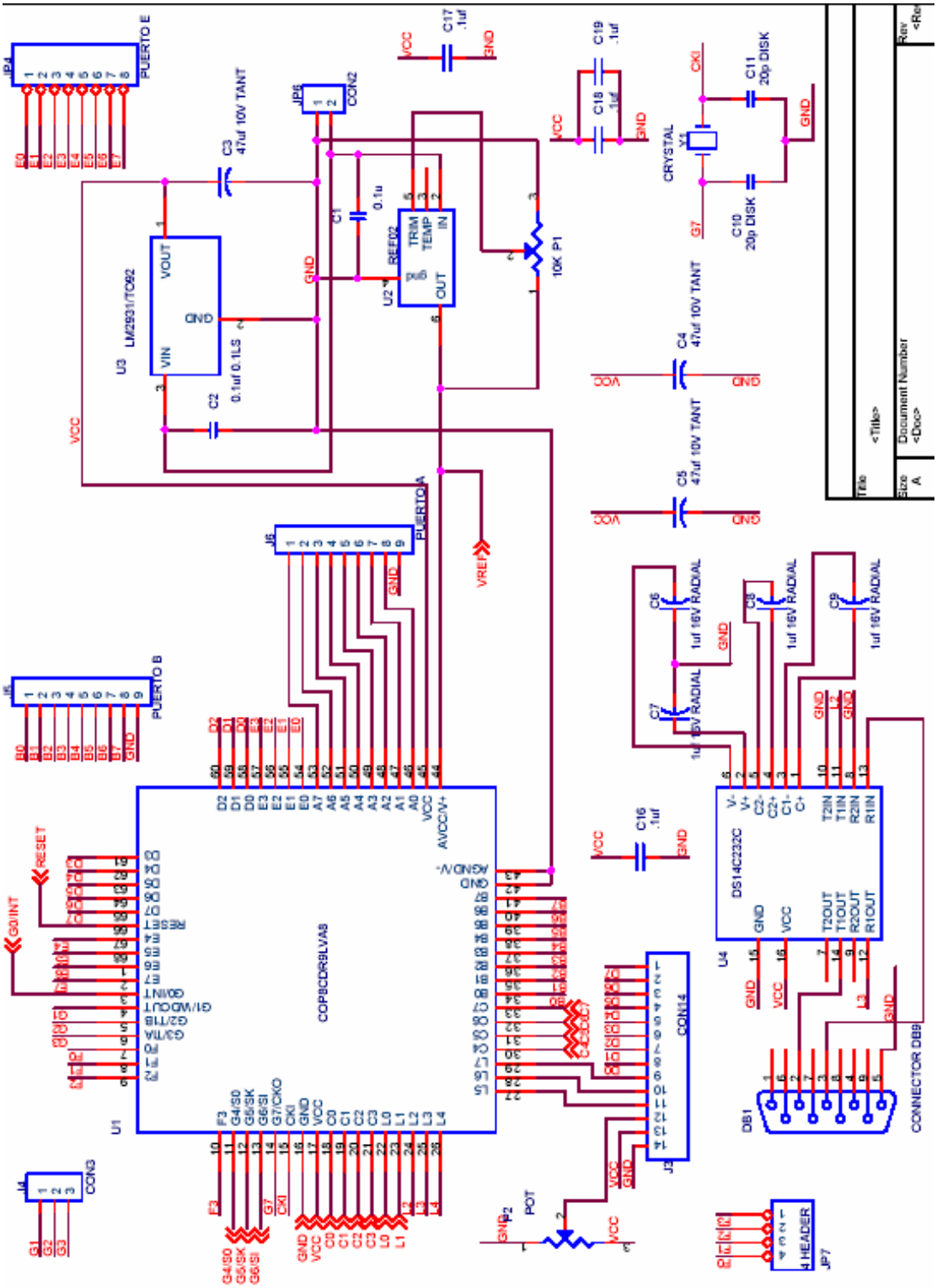

Figura 20. Diagrama esquemático parte 1.

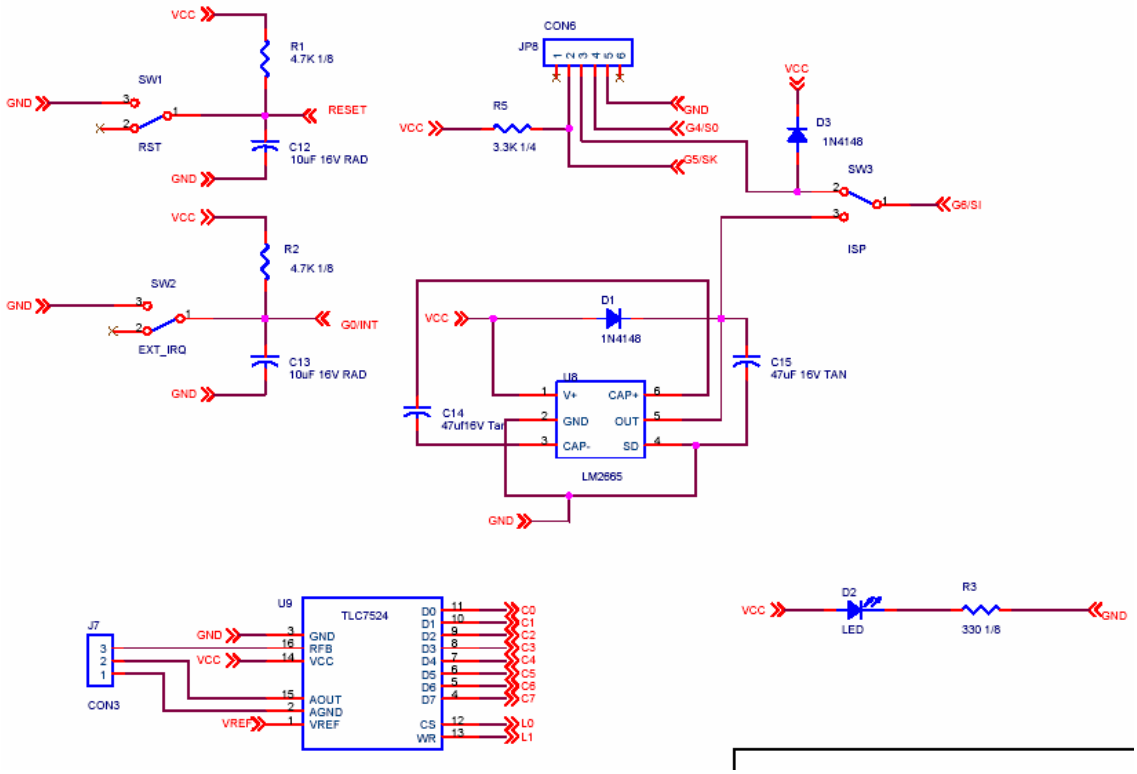

Figura 21. Diagrama esquemático parte 2.

LISTA DE COMPONENTES

Tarjeta de Acondicionamiento de seis canales de 0 a 5 Vcc

No.	Cantidad	Número de Parte
1	6	CAP 1u
2	6	LM335/TO92
3	1	BORNE 3
4	1	CONECTOR 9
5	6	TRIMPOT 10k
6	6	R 470k
7	6	R 560k
8	3	TL-062

Tarjeta de Adquisición de datos ADCDA

9	1	D1 1N4148
10	1	D2 LED
11	1	BORNE2
12	1	4 HEADER
13	1	RJ11 6 PCB
14	2	BORNE 3
15	2	9 HEADER
16	2	TRIMPOT 10k
17	2	R 4.7K 1/8
18	1	R 330K 1/8
19	1	R 1K
20	1	R 3.3K 1/4
21	3	PUSH BUTTON
22	1	COP8CDR9LVA8
23	1	BASE PLCC68
24	1	LM2931/TO92
25	1	DS14C232C
26	1	LM2665
27	1	TLC7524
28	1	CRYSTAL 10MHz
29	2	CAP 0.1u
30	3	CAP 47uf 10V TANT
31	4	CAP 1uf 16V RADIAL
32	2	CAP 20p DISK
33	2	CAP 10uF 16V RAD
34	2	CAP 47uF 16V TAN
35	1	CAP 470pF DISC
36	1	CON DB9 HEMBRA PCB

BIBLIOGRAFÍA

[1] Antonio Hernández Z., Oscar Camacho N., Cornelio Yánez M. Osvaldo Espinosa S.; Low Cost Multi-channel Data Acquisition System with Graphical HMI. En Control, Virtual Instrumentation and Digital Systems. L Sánchez, O. Espinosa (Eds.) Research in Computing Science 24, ,pp. 85-93, 2006.

[2] Antonio Hernández Z., Oscar Camacho N., Cornelio Yánez M. Osvaldo Espinosa S.; Multipurpose Interface Card for Data Acquisition and Control With Low Technological Hardware Costs. CP-21. IEEE ROC&C 2006.

[3] J. Hollingworth, D. Butterfield, B. Swart, J. Alsop, Borland C++ Builder 5.0 Developers guide. SAMS Publishing, 2001.

[4] H. Schildt, Borland C++ Builder: The Complete Reference, Mc Graw Hill, 2001.

[5] T. J. Maloney, Electrónica Industrial Moderna, Prentice-Hall, 5ª Edicion, 2006.

[6] "LM135/LM235/LM335, LM135A/LM235A/LM335A Precision Temperature sensors"; Hoja de datos de National Semiconductor, DS005698. 2000.

[7] "COP8CBR9/COP8CCR9/COP8CDR9 8-Bit CMOS Flash Microcontroller with 32k Memory, Virtual EEPROM, 10-Bit A/D and Brownout"; Manual de usuario National Semiconductor, DS101374. 2002.

[8] "Algoritmo de Lógica Difusa para Correcion de Error en Aplicaciones de Tiempo Real"; Antonio Hernández Zavala Tesis Instituto Politecnico Nacional – Centro de Investigaciones en Computación. 2004.

[9] TLC7524, Texas Instruments Datasheet, 1998.

[10] Boylestad, Nashelsky, Electrónica: Teoría de Circuitos y Dispositivos Electrónicos, Prentice-Hall, 8th. Edition, 2003.

[11] A. Denver, "Serial Communications in Win32", Microsoft Windows Developer Support, 1995.

[12] TM162-ABA6, Tianma microelectronics Co.Datasheet, 2000.

[13] Gene H. Miller, Microcomputer Engineering; Prentice Hall, 1999.

[14] R. J. Tocci, Sistemas Digitales, principios y aplicaciones, Prentice-Hall, 6th Edition, 1996.