

A. Introducción	2
A.1 – Resumen.....	2
A.2 – Requerimientos establecidos por la cátedra.....	3
A.3 – Características Generales	4
A.4 - Aplicaciones Prácticas de los Sistemas de Adquisición de Datos	5
B. Desarrollo	6
B.1 – Armado del Circuito Electrónico	6
B.1.1 - Componentes Utilizados	10
B.1.1.1 - Microcontrolador	10
B.1.1.2 - Regulador de Tensión	10
B.1.1.3 - Cristal de Cuarzo.....	10
B.1.1.4 - Convertidor de Tensión.....	11
B.1.1.5 - Interfaz RS232	11
B.1.1.6 - Sensores	11
B.1.1.6.1 - Sensor de luminosidad.....	11
B.1.1.6.1 - Sensor de temperatura.....	11
B.1.1.7.1 - Detector de presencia.....	11
B.2 – Programación del Microcontrolador	12
B.2.1 - Características de las herramientas de desarrollo.....	12
B.2.2 - Herramienta MPLAB	13
B.2.3 - Compilador CC5X - Características.....	14
B.2.4 - Instalación y Requerimientos del Sistema.....	14
B.2.5 - Dispositivo PICmicro.....	16
B.2.6 - Criterio de Desarrollo para testeo del circuito:	16
B.2.7 - Descripción del programa del microcontrolador:.....	17
B.3 – Matlab.....	21
B.3.1 - ¿Qué es la caja de herramientas de adquisición de datos?	22
B.4 - Aplicación de adquisición de datos en Visual Basic.....	26
C. Conclusiones	28
Algunas de las aplicaciones de los sistemas de adquisición de datos.....	33
Area de Energía.....	33
Area de Industria Frigorífica	33
Area de Industria Pesquera.....	34
Area Industrial.....	34
Area Hospitalaria.....	35
Area Servicios Sanitarios	35
Monitoreo y Análisis de una Central Termoeléctrica.....	37

A. Introducción

A.1 – Resumen

Se desarrolló un módulo de Adquisición de datos que maneja dos entradas analógicas, a una de las cuales está conectado un sensor de temperatura ambiente y a la otra se conectó un sensor de luminosidad.

Además maneja dos entradas de señales discretas de las cuales solo se utilizó una, y la misma está conectada a un detector de presencia.

Las señales de entrada son procesadas por un microcontrolador que está programado utilizando el lenguaje embebido de C y el software integrador “MPLAB” con el compilador “CC5X”.

Las señales procesadas se envían información vía puerto serie “RS232” a la computadora. La información enviada es interpretada y graficada por el programa “Matlab”.

Este módulo es un diseño propio no comercial.

A.2 – Requerimientos establecidos por la cátedra

Se solicitó diseñar e implementar un Sistema de Adquisición de datos que cumpliera con las siguientes características:

- Manipular entradas de señales analógicas y entradas de señales discretas.
- En los canales de entrada se deben conectar sensores de magnitudes analógicas y de magnitudes discretas para demostrar la adquisición de datos
- Posibilidad de convertir las señales analógicas de entrada, en señales discretas
- Poder enviar la información adquirida desde el circuito electrónico vía puerto serie RS232 a la Computadora
- Utilizar “MatLab” para recolectar los datos enviados por el circuito electrónico, interpretar y graficar la información recibida
- El módulo debe ser de diseño propio no comercial.

A.3 – Características Generales

La función principal del Sistema de Adquisición de Datos es transmitir la información obtenida a través de sus canales de entrada, de diferentes señales analógicas y/o discretas las cuales son previamente captadas mediante el uso de diferentes sensores.

El núcleo del circuito está basado en la utilización de un microcontrolador “PIC16C73B” que, el cual, está dotado de varias funciones que lo hacen muy versátil. Mediante una adecuada codificación se pueden convertir señales analógicas a señales discretas, darles tratamiento mediante los puertos de entrada / salida y conectarlo con una computadora a través de su interfase RS 232.

La magnitudes captadas por los sensores ingresan al microcontrolador con una frecuencia que se puede modificar en tiempo de ejecución.

“MatLab” se encarga de la recolección de los datos enviados por el circuito electrónico, los cuales son almacenados en una matriz y luego graficados. Además se graban en un archivo histórico para su posterior lectura.

Paralelamente se ha desarrollado una aplicación en “Visual Basic” que cumple la misma función del programa realizado en “Matlab” con fines netamente comparativos.

El costo del material ascendió a 300 pesos.

A.4 - Aplicaciones Prácticas de los Sistemas de Adquisición de Datos

Estos son utilizados para control de:

- edificios
- equipos de calefacción y aire acondicionado
- dosificaciones
- procesos de señalización
- proceso de secado y cocción
- criaderos
- invernaderos
- riego
- procesos de embalajes
- incubadoras

ello se procedió de la siguiente manera, lo primero fue dibujar sobre papel el diseño original del circuito impreso tal como quedó terminado, para esta tarea se utilizó un programa de diseño de circuitos impresos cuyo nombre comercial es “STAMP EDITOR”, teniendo también a los componentes electrónicos a montar en el circuito para así poder ver el espacio físico que requerían y la distancia entre cada uno de sus terminales. Una vez obtenido el circuito en papel se procedió al corte de la placa de “Pertinax” como se muestra en la figura 3.

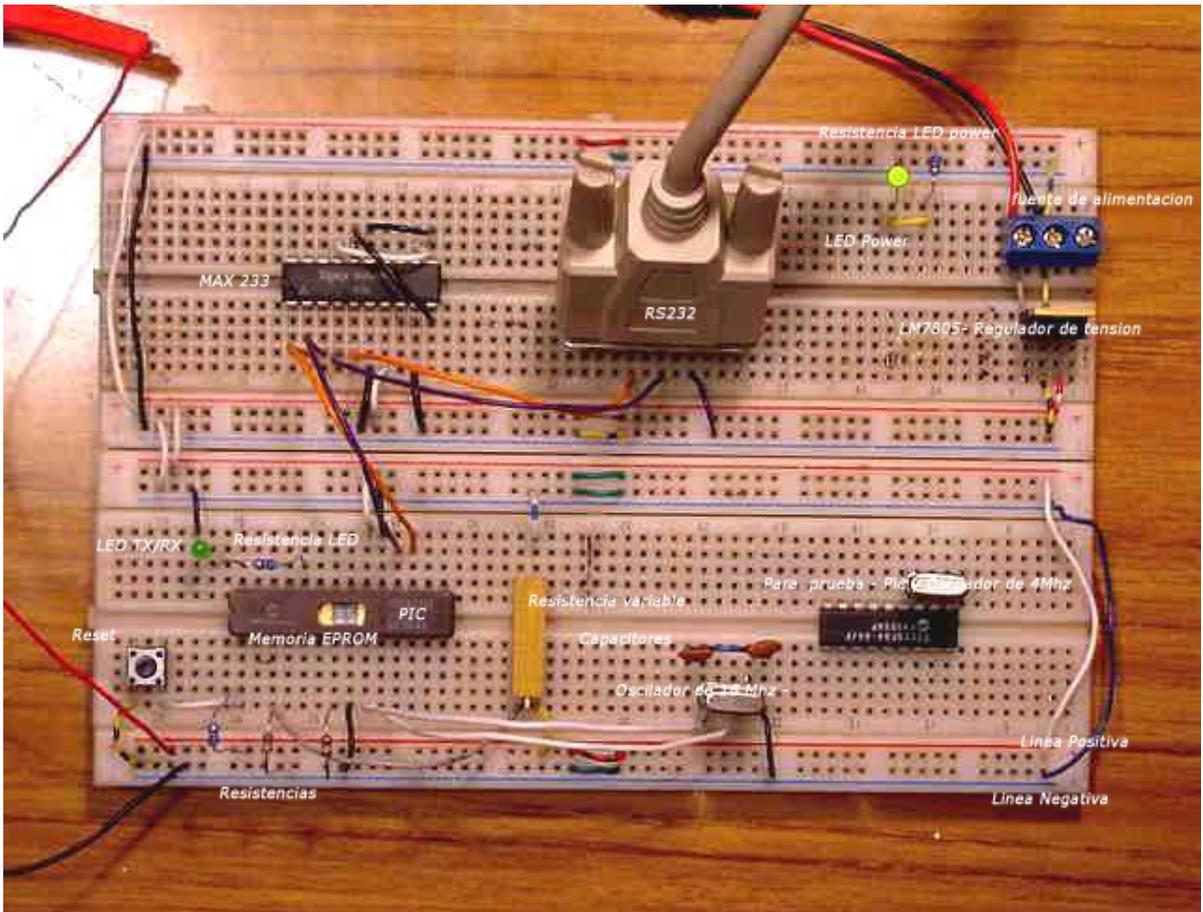


Figura 2.



Figura 3.

El siguiente paso consistió en hacer que el dibujo del impreso en papel quedara sobre la cara de cobre de la placa, de alguna forma indeleble; para esto fueron necesarios un marcador y varias plantillas “LOGOTYP” de islas para aplicar los dibujos de la plantilla, se colocaron las mismas sobre la lamina de cobre frotando cada una hasta que quedaran estampadas sobre el circuito impreso.

Una vez concluido el dibujo sobre la placa de cobre fue necesario atacarla químicamente para lograr que el cobre sólo quedara depositado en las zonas que habían sido dibujadas.

Para esto se sumergió la placa de “PERTINAX” en un recipiente conteniendo “Percloruro de hierro” por un lapso aproximado de 35 minutos. Al cabo de este tiempo ya se tuvo el circuito impreso prácticamente terminado. Con un probador de continuidad se verificó que todas las pistas llegaran “enteras” de una isla a otra, sólo faltaba realizar los agujeros correspondientes para poder efectuar el montaje correspondiente.

Una vez concluido, se procedió al ensamblado definitivo del impreso y al armado final en un gabinete como se muestra en la siguiente figura.

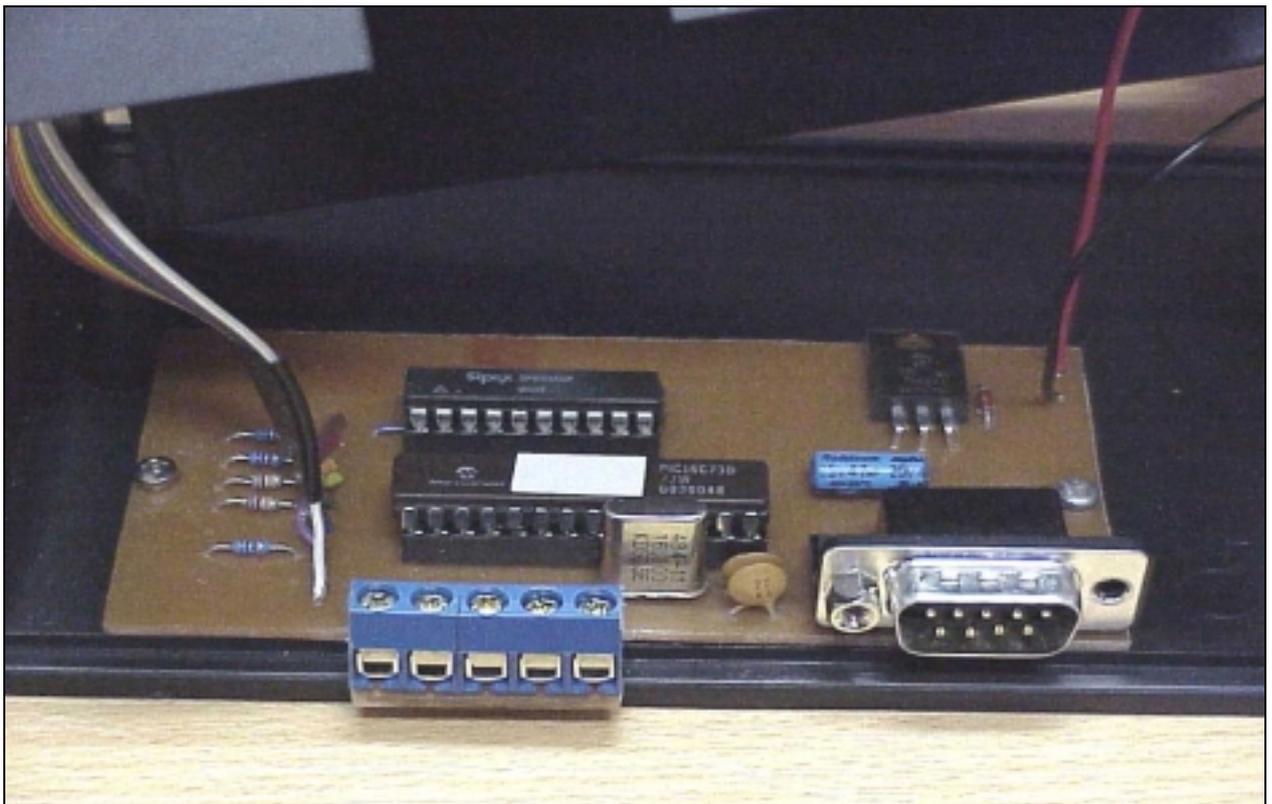


Figura 4.



Figura 5.

También se realizó el armado de un cable RS232 (sólo con 3 líneas: transmisión, recepción y tierra).

A lo largo de las pruebas realizadas se fueron encontrando algunas dificultades en lo referente a la estabilidad de respuesta del microcontrolador. En principio se pensó que esto se debía a errores en la codificación del mismo (o al modo de codificación), por lo que se fueron probando diferentes variantes dentro del espectro ofrecido por el compilador.

En la medida que el código fue evolucionando y por ende se hacía más complejo la inestabilidad se incrementaba. Motivo por el cual se comenzaron a realizar pruebas para modificar el circuito electrónico.

Dentro de los intentos realizados se procedió al cambio del cristal de 16 Mhz por uno de 4 Mhz con lo que el problema se resolvía, pero implicaba tener que bajar la velocidad de comunicación con la computadora de 9600 a 2400 baudios, lo cual resultaba inconveniente. No obstante a través de este cambio en el comportamiento presentado ante la eventual modificación de frecuencia en el reloj, fue posible obtener la pista para resolver el problema de fondo.

Mediante la utilización de un osciloscopio, se detectó un alto componente de ruido eléctrico, en la tensión de alimentación, problema que fue resuelto mediante la incorporación de un filtro capacitivo en el circuito.

A partir de este momento quedaron definitivamente resueltos los problemas de inestabilidad.

B.1.1 - Componentes Utilizados

B.1.1.1 - Microcontrolador

Todo el funcionamiento del circuito electrónico tiene como eje principal al microcontrolador “PIC16C73B” de Microchip. Cuyas características principales se muestran en la Tabla 1.

CARACTERÍSTICAS PRINCIPALES
MEMORIA DE PROGRAMA: 4 K x 14
MEMORIA DE DATOS: 192 bytes (PIC16C73B)x 8
MEMORIA DE DATOS EPROM: 14 bits
PILA (Stack): De 8 niveles
INTERRUPCIONES: 11 tipos diferentes
JUEGO DE INSTRUCCIONES: 35
ENCAPSULADO: Plástico DIP de 28 pines
FRECUENCIA DE TRABAJO: 20 MHz máxima
TEMPORIZADORES: 3 TIMER. Tambien tiene Perro Guardián (WDT)
LINEAS DE E/S DISCRETAS: 22 (6 en el Puerto A, 8 en el Puerto B y 8 en el C)
VOLTAJE DE ALIMENTACION (VDD): De 2.5 a 6 V DC
VOLTAJE DE GRABACION (VPP): De 12 a 14 V DC
CONVERTIDOR A/D con 5 canales de entrada
DOS MÓDULOS DE CAPTURA Y COMPARACIÓN Y PWM (CCP)
PUERTO SERIE SINCRONICO (SSP)
PUERTO SERIE ASINCRONICO (ASP)
INTERFAZ DE COMUNICACIÓN SERIE (SCI)

Tabla 1.

B.1.1.2 - Regulador de Tensión

Como todo dispositivo electrónico el circuito debe poseer una fuente de alimentación para su funcionamiento. En este caso se utilizó un circuito integrado “LM7805”, el cual provee una tensión fija de 5 Vcc. Este dispositivo posee sólo tres terminales: Una para tensión no regulada, otra correspondiente para tensión regulada y la tercera a la masa común de ambas. Una de las características fundamentales del componente es que posee una protección térmica y contra corriente excesiva, la cual hace disminuir la tensión de salida a 0 volt cuando la corriente o la temperatura es demasiada elevada.

B.1.1.3 - Cristal de Cuarzo

Se utilizó un cristal de cuarzo de 16 Mhz, como elemento de control de frecuencia para generar el pulso de reloj necesario para el funcionamiento del microcontrolador.

B.1.1.4 - Convertidor de Tensión

Debido a que el microcontrolador maneja tensiones entre 0 y 5 Vcc, y el voltaje necesario para la interface RS232 es de -12 Vcc y + 12 Vcc fue necesario intercalar entre estos dos elementos un componente que elevara los niveles de tensión entregada por el microcontrolador a los requeridos por la RS232 para lo cual se utilizó el circuito integrado “MAX233A”.

B.1.1.5 - Interfaz RS232

La comunicación entre el circuito electrónico y la computadora es realizada utilizando la interface estandar RS232.

La misma es implementada mediante el uso de un conector “DB9” del cual sólo se han utilizado el terminal “2” para recepción, el “3” para transmisión, el “5” para tierra de señal y el “1” para tierra de protección.

B.1.1.6 - Sensores

En base a que los requerimientos iniciales establecian el empleo de como mínimo un sensor analógico y un sensor de señal discreta, y teniendo en cuenta los costos en el mercado y la posibilidad de obtener algunos de ellos en calidad de prestamo se optó por los siguientes dispositivos.

B.1.1.6.1 - Sensor de luminosidad

En función a la disponibilidad de dispositivos sensibles a las variaciones lumínicas, se seleccionó un semiconductor que cumple con estas características denominado “fotoresistor”. Este elemento tiene la particularidad de variar su resistencia al pasaje de la corriente eléctrica en forma proporcional a la intensidad de luz que la afecta. Este sensor fue conectada a una de las entradas analógicas.

B.1.1.6.1 - Sensor de temperatura

A la otra entrada analógica se conectó un componente similar al anterior, pero con la diferencia que su resistencia al pasaje de la corriente eléctrica varía en función de la temperatura que le afecta, denominado “termistor”.

B.1.1.7.1 - Detector de presencia

Como sensor de señal discreta se eligió utilizar un detector de presencia. Este dispositivo se comporta basicamente como una llave normalmente cerrada, la cual se abre al detectar movimientos.

B.2 – Programación del Microcontrolador

Se analizaron las variables tomadas para este proyecto. Recolectando la información con respecto a los elementos necesarios para programar el microcontrolador se determinó:

- El microcontrolador “PIC16C73B” de MicroChip Technology Inc, que cumplió con los requisitos necesarios en cuanto a la cantidad de entradas de señales discretas y analógicas, así como también el de disponer de una interfaz USART (Universal Synchronous Asynchronous Receiver Transmitter) óptima para implementar una comunicación RS232 con una PC.

- El lenguaje necesario para el desarrollo del software embebido en C. El compilador empleado es el CC5X versión 3.1 de B. Knudsen Data (Noruega) en su versión free, que esta limitado a 1000 palabras de programa. La elección del compilador se debió fundamentalmente a:
 - Distribución gratuita
 - Integración con el MPLAB (Software para programar PICs)
 - Se empleó lenguaje C en lugar de Assembler atendiendo a las facilidades de desarrollo que implica un lenguaje de alto nivel
 - La estructura de la programación será: Código fuente principal
 - Enviará esta información vía puerto serie RS232

B.2.1 - Características de las herramientas de desarrollo

EL MPLAB es un “Entorno de Desarrollo Integrado” (Integrated Development Environment, IDE) que corre en “Windows”, mediante el cual se puede desarrollar aplicaciones para los microcontroladores de las familias PIC 16/17.

EL MPLAB permite escribir, depurar y optimizar los programas (firmware) de sus diseños con PIC 16/17. EL MPLAB incluye un editor de texto, un simulador y un organizador de proyectos. Además soporta otras herramientas de desarrollo de Microchip como el PICSTART-Plus utilizado para la grabación del microcontrolador.

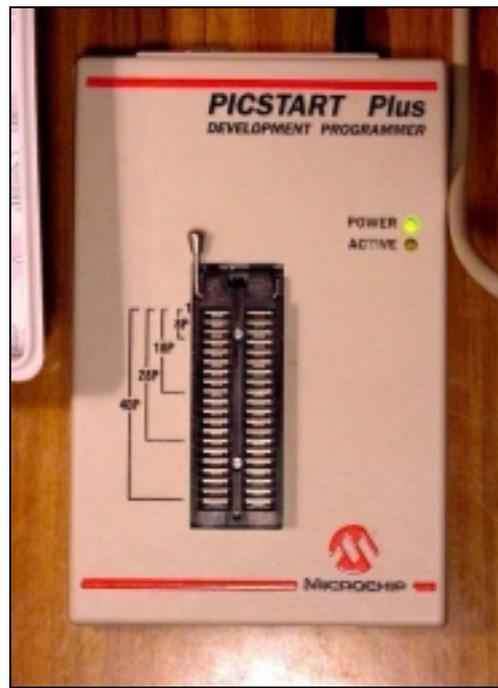


Figura 1.

PicStar – Plus

La grabadora de programas para microcontroladores, se conecta vía puerto serie con la computadora y mediante MPLab se transfiere el código ejecutable a la memoria del microcontrolador y se borra mediante lampara UV.

B.2.2 - Herramienta MPLAB

El organizador de proyectos (Project Manager) es parte fundamental de MPLAB. Sin crear un proyecto no se puede realizar una depuración simbólica. Con el Organizador de Proyectos (Project manager) se pueden realizar las siguientes operaciones:

- Crear un proyecto.
- Agregar un archivo de programa fuente de proyecto.
- Ensamblar o compilar programas fuente.
- Editar programas fuente.
- Reconstruir todos los archivos fuente, o compilar un solo archivo.
- Depurar su programa fuente.

El conjunto de instrucciones de los microcontroladores “PIC” es en esencia la base del lenguaje ensamblador soportado por este software.

B.2.3 - Compilador CC5X - Características

- Código fuente estándar
- Programa de rápido desarrollo
- Documentación de fácil entendimiento
- Mantenimiento simple
- Código Portátil
- Variables locales y globales de 1, 8, 16, 24 y 32 bits
- Reutilización eficiente del espacio ocupado por la variable local
- Genera binario, assembler, listas, funciones de línea, COD, error y logs.
- Actualización automática de selección de páginas de bits.
- Actualización automática de selección de bancos de bits
- Soporte para operaciones con bits incluyendo funciones binarias.
- Permite utilizar modelos de 8 y 16 bits en la misma aplicación.
- Puntero a la memoria RAM y/o ROM
- El tamaño puede ser automáticamente seleccionado por el compilador
- Nivel de llamada usando GOTO en vez de CALL
- Links para subrutinas ocultas
- Soporte integrado para interrupciones
- Información sobre la configuración del Chip para el código fuente

El CC5X es un lenguaje diseñado para generar códigos óptimos. El optimizador reduce el código automáticamente. Es posible escribir código utilizando instrucciones muy simples siguiendo la sintaxis del C.

Es fundamental en los casos en que se requiere uso limitado de recursos de memoria.

B.2.4 - Instalación y Requerimientos del Sistema

El CC5X usa 32 bits para procesamiento y corre en una Computadora compatible con modo MSDOS y Windows (NT/95/98/2000).

En la instalación, se creó un directorio en el disco rígido donde se ubicaron los archivos y una vez copiado el CC5X estuvo listo para compilar.

El código fuente y los archivos cabecera deben ser editados utilizando un MPLAB.

De esta manera, una vez diseñada la lógica de control, se compilo el programa y haciendo referencia a éste desde el software principal del proyecto.

CC5X se seleccionó como una herramienta de MPLAB con entorno de desarrollo integrado que incluye un editor, herramientas de soporte (compilador, assembler, simuladores, emuladores, dispositivos para programadores)

Los errores de compilación se manejaron fácilmente por la ayuda que ofrece MPLAB, que indica directamente en que línea se encuentra el error.

El MPLAB es gratuito, y se puede bajar desde la página de Microchip.

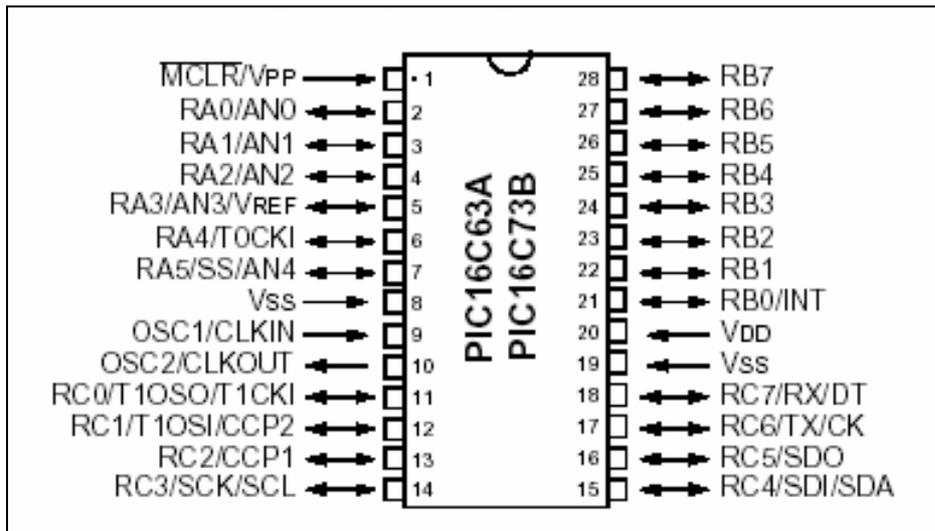


Figura 2.

Pines utilizados para la programación

- MCLR/Vpp: Reset cuando el “PIC” está operando. Si se le pone 12 volts se puede grabar
- RA0 y RA1: Son entradas analógicas
- RA2 y RA5: Son entradas de señales discretas
- RA3/VRef: Tensión de referencia (Cable a tierra)
- OSC1 y OSC2: Osciladores .Están conectados directamente al circuito para la generación de la frecuencia de reloj utilizada para temporizar todos los ciclos de funcionamiento interno. Generan ondas cuadradas, estan toman el “PIC” y disparan la ejecución de las instrucciones (diviendio los megaciclos por 4). Con esto logra un temporizador que le indica que hacer secuencialmente. De esta frecuencia depende casi la totalidad de las operaciones internas y en particular la velocidad con la cual el PIC ejecuta las instrucciones del programa. En el caso del PIC16C73B dicha frecuencia puede alcanzar un máximo de 16 Mhz de la cual se obtiene una velocidad de ejecución de las instrucciones igual a 1 millón de instrucciones por segundo. En este caso para la generación de reloj se utilizó un cuarzo externo de 4 Mhz y dos condensadores de 22pf.
- RB3: Led
- RC6 y RC7: Puerto serie RS232

B.2.5 - Dispositivo PICmicro

CC5X ofrece tres formas de seleccionar una aplicación para un PICmicro:

1) Por comando:

En MPLAB, la opción de comandos de línea se selecciona en el Project->Edit Project..(target)..Node Properties.
-p16C73

2) Definición del recurso en el código utilizando la palabra reservada pragma:

```
#pragma chip PIC16C73
```

3) Incluyendo el archivo

```
#include "16C73.h"
```

NOTA 1: Cuando se usa la declaración pragma o un archivo de cabecera, recordar usarlo al comienzo del código del programa C, porque es lo primero que compila.

NOTA 2: Cuando se usa el comando de línea o pragma, CC5X usará las definiciones internas para algunos dispositivos conocidos, entre ellos: 16C54,55,56,57,58, 61,64,65, 71,73,74, 84, 620,621,622.

Si el dispositivo no es reconocido, automáticamente incluye un archivo de cabecera al comienzo.

NOTA 3: Si el archivo de cabecera no reside por defecto en el directorio del proyecto, requerirá el path, que se puede incluir con un comando de línea
directorio/proyecto(-Ipath).

NOTA 4: Se puede definir un archivo de cabecera nuevo según el texto 'chip.txt'.

B.2.6 - Criterio de Desarrollo para testeo del circuito:

Para las pruebas iniciales se generó un código muy sencillo con el objeto de determinar si la electrónica funcionaba, para ello el código propuesto simplemente tenía la misión de cambiar el estado de una salida de señal discreta, en forma alternativa a periodos especificados de tiempo, se conectó un LED en esta salida con el propósito de visualizar si el prototipo funcionaba.

Fue necesario conseguir la grabadora de PIC para realizar la primer grabacion.

Pasos:

Se realiza la primer prueba prueba con resultados negativos

Se estudia y modifica el código, se borra la EPROM exponiéndola a luz ultravioleta 2 minutos y se reprograma en el PICSTAR.

Se hace la 2da. Prueba y se logra un resultado satisfactorio prendía y apagaba el LED

A continuación se escribe un código para enviar caracteres vía puerto serie desde el micro a la computadora, se prueba este código y se depura hasta que funciona correctamente.

Para esto se hace uso del hiperterminal de Windows configurado acorde a los parámetros especificados en la configuración del PIC (9600 baudios, 8bits de datos, sin paridad, 1 bit de parada ,sin control de flujo)

Mas tarde entonces se incorpora al código anterior la capacidad de recepción , creando un algoritmo que permitiera determinar si efectivamente el PIC recibía la información enviada desde la computadora.

Como se mencionó anteriormente a lo largo de estas pruebas se encontraron algunas dificultades en lo referente a la estabilidad de respuesta del microcontrolador; en principio se entendio que esto se debía a errores en la codificación (o al modo de codificación) por lo que se probó diferentes variantes dentro del espectro ofrecido por el compilador.

Puesto que en la medida que evolucionaba el código y este se iba haciendo mas complejo la inestabilidad aumentaba. Esto llevo a realizar diferentes pruebas electronicas. A partir de un cambio en el circuito electrónico quedaron definitivamente resueltos los problemas de inestabilidad.

Una vez depurado el código , referente al mecanismo de recepción (Rx) y envío de datos(Tx), se apunto a la mecánica de la conversión A/D, para lo cual se inyecto un valor de tensión de aproximadamente 2.5 volt en ambos canales analógicos de entrada y se fueron realizando pruebas hasta lograr resultados satisfactorios.

En este punto del desarrollo ya se habían resuelto los dos problemas mas relevantes, ahora solo faltaba integrar las partes del software.

Para lograr la funcionalidad deseada, atendiendo los resultados del estudio realizado sobre el manejo del puerto serie, por parte del Matlab se resolvió que la forma mas adecuada de comunicación de datos entre el microcontrolador y la computadora debía ser por medio de caracteres. Teniendo en cuenta que el objeto de trabajo era la representación grafica en tiempo real de los valores adquiridos, era menester que la frecuencia de adquisición de datos por parte del controlador se mantuviera dentro de un rango admisible para la interfaz grafica por lo que se seleccionaron como frecuencia de adquisición entre 1 y 8 Hz; y en función de tan baja frecuencia de adquisición, y con el objeto de generar una respuesta inmediata a la interfaz ante una eventual solicitud de la misma , se diseño la rutina de atención a través de un mecanismo de interrupción que brindo la posibilidad de poder responder en tiempo real a la solicitudes externas.

B.2.7 - Descripción del programa del microcontrolador:

El programa consta de un loop principal que invoca en forma secuencial las funciones encargadas de realizar la lectura y el envío de los valores de los dos canales analógicos y de los dos canales de señal discreta, en la secuencia del loop principal se intercalaron dos funciones de “espera” que son las responsables de la frecuencia de adquisición y además se encargan de generar la señalización exterior (por medio de un LED) del envío de datos por medio del puerto serie.

En el loop principal no existe ninguna llamada explícita a funciones que “lean” información proveniente de la computadora, por cuanto esta función fue implementada mediante un vector de interrupción, generada internamente por el microcontrolador cuando detecta un nuevo dato

en la entrada del puerto serie. Esto se hizo por cuanto las frecuencias de adquisición de datos son relativamente bajas, en función de lograr una buena respuesta de graficación en la interfaz de la computadora; en realidad esto significa que el microcontrolador pasa la mayor parte del tiempo “dentro” de las funciones de “espera” y por lo tanto, con el objeto de independizar los tiempos de respuesta a comandos enviados desde la computadora, era necesario poder interrumpir el loop principal.

Cuando arranca el microcontrolador (o se lo resetea manualmente) primero envía por el puerto serie un mensaje de presentación (en formato de texto) y luego queda dentro del loop principal, esperando alguna instrucción a realizar proveniente de la computadora.

El protocolo implementado entre el PIC y la computadora referente al mecanismo de recepción (Rx) y envío de datos(Tx) – manejo del puerto serie-está basado en caracteres, para esta aplicación en particular. El objetivo del mismo es solicitarle al PIC algunos cambios de comportamiento, como iniciar o detener la adquisición y modificar la velocidad de adquisición entre valores predefinidos. El único requerimiento que se le impuso fue devolver una confirmación de reconocimiento por cada carácter recibido; en el caso de que el carácter recibido no sea reconocido, ya sea por ser un carácter no válido o por estar dentro de una secuencia no válida, el protocolo devolverá un carácter de error (E), con esto queda resuelto el requerimiento de que para cada carácter recibido el protocolo si o si devolviera un carácter. Esta imposición surge como consecuencia del diseño de la aplicación que se ejecuta en la computadora la que verifica el retorno de cada carácter que envía como comando, con el objeto de determinar si el periférico comprende y realiza la tarea encomendada. Estamos utilizando al RS232 en una forma sencilla y por lo tanto cualquier tipo de validación que se requiera hacer debe ser implementada en el software ya que no dispone de bit de error ni de control de flujo pero dada las características del sistema y de acuerdo a las observaciones realizadas durante el desarrollo podemos decir que el diseño del protocolo, es eficiente y seguro.

A continuación se detalla la “interfaz” que presenta el programa a través del puerto serie:

Cuando la PC envía	El micro responde	Estado del microcontrolador
W	W	En espera del final del comando
S	K	Inicia la adquisición
W	W	Continúa con la adquisición
S	K	Detiene la adquisición
W	W	En espera del final del comando
T	K	Duplica el periodo de adquisición
...	M	Sin efecto (máximo periodo alcanzado)
W	W	En espera del final del comando
T	K	Divide por dos el periodo de adquisición
...	M	Sin efecto (mínimo periodo alcanzado)
W	W	En espera del final del comando
W	E	Error (doble prefijo de comando)
Cualquier char	E	Comando desconocido

Configuración de la USART como Asíncrono (Full duplex) para funcionar como UART (9600 baudios, 8N1).

Poniendo a "1" bit **SPEN** (**RCSTA** <7>), y los bits **TRISC** <7:6>, se configura **RC6/TX** y **RC7/RX** como USART.

Hay tres registros para configurar que son: TXSTA, RCSTA y SPBRG.

TXSTA: Registro de Estado y Control (Transmisión) configuramos el siguiente bit:

TXEN Bit de habilitación de la transmisión, el resto de los bits quedan con los valores predefinidos después de un reset.

Habilita transmisión = 1

Deshabilita transmisión = 0

RCSTA: Registro de estado y Control (Recepción)

SPEN: Bit de habilitación del puerto serie

Habilita puerto serie (configura patillas RC7/RX/DT y RC6/TX/CK para el puerto serie) = 1

Deshabilita puerto serie = 0

CREN: Bit de habilitación de recepción continua

Modo Asíncrono

Habilita recepción continua = 1

Deshabilita recepción continua = 0

SPBRG

Se cargó el valor 25 que para un BRGH = 0 (tal como quedó configurado) y un clock de 16 MHz. Tratándose de una comunicación asincrónica significan 9600 baudios.

BRGH: Bit de selección del rango de baudios alto

Modo Asíncrono

Alta velocidad = 1

Baja velocidad = 0

La interfaz es simple, cada respuesta del microcontrolador viene acompañada por un “\n (carácter 13 nueva línea) “

Las lecturas de los canales analógicos (convertidores A/D de 8 bits) son directamente el número de “cuentas” que devuelve el convertidor A/D cuando se le solicita una lectura, es decir un número entero entre 0 y 255.

Por otra parte las frecuencias de adquisición, se ajustaron en función de la velocidad de respuesta de la interfaz que corre en la computadora, y se asegura que dentro de los valores empleados, y atendiendo al mecanismo de respuesta implementado en el microcontrolador (que no se queda “esperando” a la computadora), la frecuencia de adquisición es altamente estable.

La codificación garantizó que el sistema estuviera bajo control, verificando el buen funcionamiento de la interfaz, dotándolo de parámetros indicadores que se disparen cuando se supere la frecuencia de adquisición, para garantizar así la máxima seguridad y estabilidad de los datos interpretados.

B.3 – Matlab

MATLAB es un lenguaje de alta performance para computación técnica. Está integrado por: cálculo, visualización y programación en un entorno fácil de usar donde los problemas y soluciones son expresados en una notación matemática familiar.

Los usos típicos incluyen:

- Matemática y computación
- Desarrollo de algoritmos
- Modelado, simulación y creación de prototipos
- Análisis de datos, exploración y visualización
- Desarrollo de aplicaciones, incluyendo construcción de interfaces gráficas de usuario

MATLAB es un sistema interactivo cuyo elemento básico es un array que no requiere dimensionamiento, esto permite la resolución de muchos problemas técnicos de cálculo, especialmente aquellos con formulaciones de matrices y vectores. En una fracción de tiempo permite escribir un programa en un lenguaje no interactivo como C o Fortran.

El nombre MATLAB proviene de “Laboratorio de Matriz”. MATLAB fue originalmente escrito por los proyectos LINKPACK y EISPACK, para proveer fácil acceso a desarrollo de software de matrices. Hoy, MATLAB utiliza software desarrollado por los proyectos LAPACK y ARPACK, que juntos representan el estado del arte (vanguardia) en la computación para matrices.

En entornos universitarios, es la herramienta standard de enseñanza para cursos introductorios y avanzados en matemática, ingeniería y ciencia. En la industria, MATLAB es la herramienta de elección para investigaciones de alta productividad, desarrollo y análisis.

MATLAB presenta una familia de soluciones a aplicaciones específicas llamadas “cajas de herramientas”. Es muy importante para la mayoría de los usuarios de MATLAB, las cajas de herramienta permiten *aprender y aplicar* tecnología especializada. Las cajas de herramientas son colecciones integrales de funciones MATLAB (Archivos .M) que extienden el entorno MATLAB para resolver clases particulares de problemas. Las áreas en las cuales las cajas de herramientas están disponibles incluyen procesamiento de señal, adquisición de datos, sistemas de control, redes neuronales, lógica difusa, pequeñas ondulaciones entre muchas otras.

El Sistema MATLAB consiste de 5 partes principales:

- **Entorno de Desarrollo.** Es un conjunto de instrumentos de trabajo y facilidades que ayudan el uso de las funciones y archivos MATLAB. Muchos de estas herramientas son interfaces gráficas de usuario. Esto incluye el Escritorio de MATLAB, la Ventana

de Comando, un Historial de Comandos, Navegadores para la Visualización de Ayuda, el Área de Trabajo y un Buscador de Archivos.

- **Biblioteca de funciones matemáticas.** Es una extensa colección de algoritmos computacionales ordenados desde funciones elementales tales como suma, seno, coseno y algoritmos complejos, a funciones más sofisticadas como inversas de matrices, funciones Bessel y las transformadas rápidas de Fourier.
- **El lenguaje MATLAB.** Es un lenguaje de alto nivel matriz/array con declaración de control de flujo, funciones, estructura de datos, entrada/salida, y características de programación orientada a objetos. Esto permite “programación en partes pequeñas” para crear programas cortos, y “programación en partes grandes” para crear programas de aplicaciones grandes y complejos.
- **Punteros gráficos.** Es un sistema gráfico MATLAB que incluye comandos de alto nivel para dos y tres dimensiones de visualización de datos, procesamiento de imágenes, animación y graficas de presentación. También incluyen comandos de bajo nivel que permiten la construcción a medida de interfaces gráficas de usuario en las aplicaciones MATLAB.
- **Interface de Aplicación de Programa en MATLAB (API).** Esta librería permite escribir programas en C y Fortran que interactúan con MATLAB. Incluyen facilidades de llamadas a rutinas desde MATLAB (ligadura dinámica).

B.3.1 - ¿Qué es la caja de herramientas de adquisición de datos?

La caja de herramientas de adquisición de datos es una colección de funciones de archivos .M y archivos MEX, DLLs construidas sobre la tecnología MATLAB®. La caja de herramientas le provee de las siguientes partes principales:

- Un área de trabajo para dar vida, a los datos dentro de MATLAB usando plug-in de adquisición de datos del hardware compatible con la PC
- Soporta subsistemas de entradas analógicas (AI), salidas analógicas (AO), y entrada / salida de señales discretas incluyendo conversiones entrada /salida de señales discretas.

La caja de herramientas de adquisición de datos consiste en tres componentes distintos: funciones de los archivos .M, la ingeniería de la adquisición de datos, y adaptadores. A continuación se muestran, estos componentes que permiten pasar información entre MATLAB y el hardware de adquisición de datos.

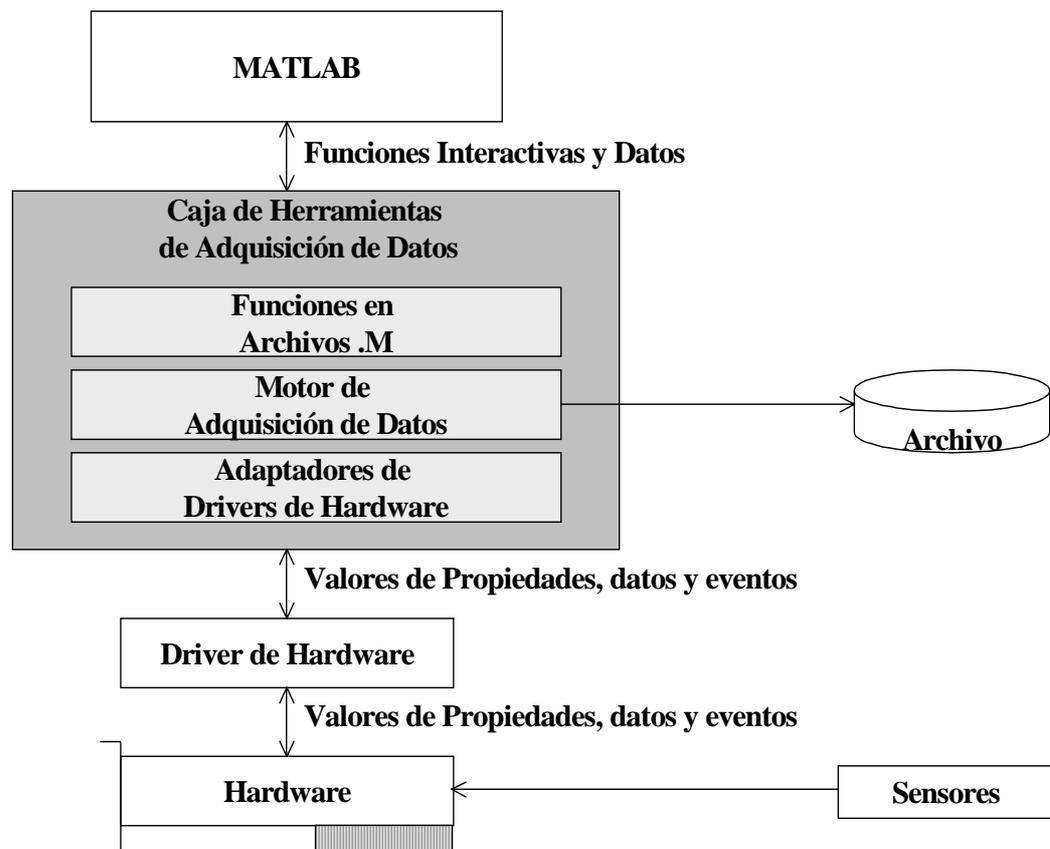


Figura 1.

El diagrama precedente ilustra como la información fluye desde un componente a otro. La información consiste en:

- **Valores de propiedad**

Se podrá tener control sobre el comportamiento de la aplicación de adquisición de datos configurando valores de propiedad, se puede pensar una propiedad como una característica de la caja de herramientas o el driver de hardware que se podrá manipular según la necesidad.

- **Datos**

Se pueden adquirir datos desde un sensor conectado a un subsistema de entrada analógica y almacenar esto en el MATLAB, o datos de salida desde MATLAB conectado a un sistema de salida de señales discretas. Adicionalmente se puede transferir valores (1 o 0) entre MATLAB y un subsistema de entrada / salida de señales discretas.

Se podrán adquirir datos desde un sensor conectado a un subsistema de entrada analógica y almacenarlos en MATLAB, o datos de salida de un actuador conectado a un subsistema de salida analógica. Adicionalmente se podrá transferir valores (1 o 0) entre MATLAB y un subsistema de entrada/ salida de señales discretas.

- **Eventos**

Un evento ocurre después de un tiempo particular, una condición es encontrada y puede resultar en una o más funciones de callback que se especificaron. Un evento es generado después de configurar las propiedades asociadas. Algunos de los caminos que se pueden usar los eventos incluyen inicialmente análisis después de una determinada cantidad de datos adquirida o mostrando un mensaje en el área de trabajo de MATLAB después que ocurre un error.

Para el desarrollo de la aplicación de Adquisición de Datos en Matlab se utilizó el objeto Serial, el cual permite la comunicación con cualquier puerto serial de la computadora. Mediante este objeto se pueden obtener los datos enviados utilizando el protocolo RS232.

Los datos son cargados en una matriz y luego graficados. Por cada lectura al puerto serie se graba en disco un archivo de log de datos; el formato del archivo de log consta de fecha y hora de la adquisición, los valores de los datos que son enviados a la placa y los que son recibidos desde ésta.

El programa en Matlab cuenta con una Interfaz de Usuario; dos archivos .M, uno con las funciones de callback y otro con la clase RS232 donde se crea el objeto Serial antes mencionado.

Desde la interfaz gráfica es posible iniciar o detener la adquisición de datos, aumentar o disminuir la frecuencia como así también ver un gráfico con los datos que se están leyendo.

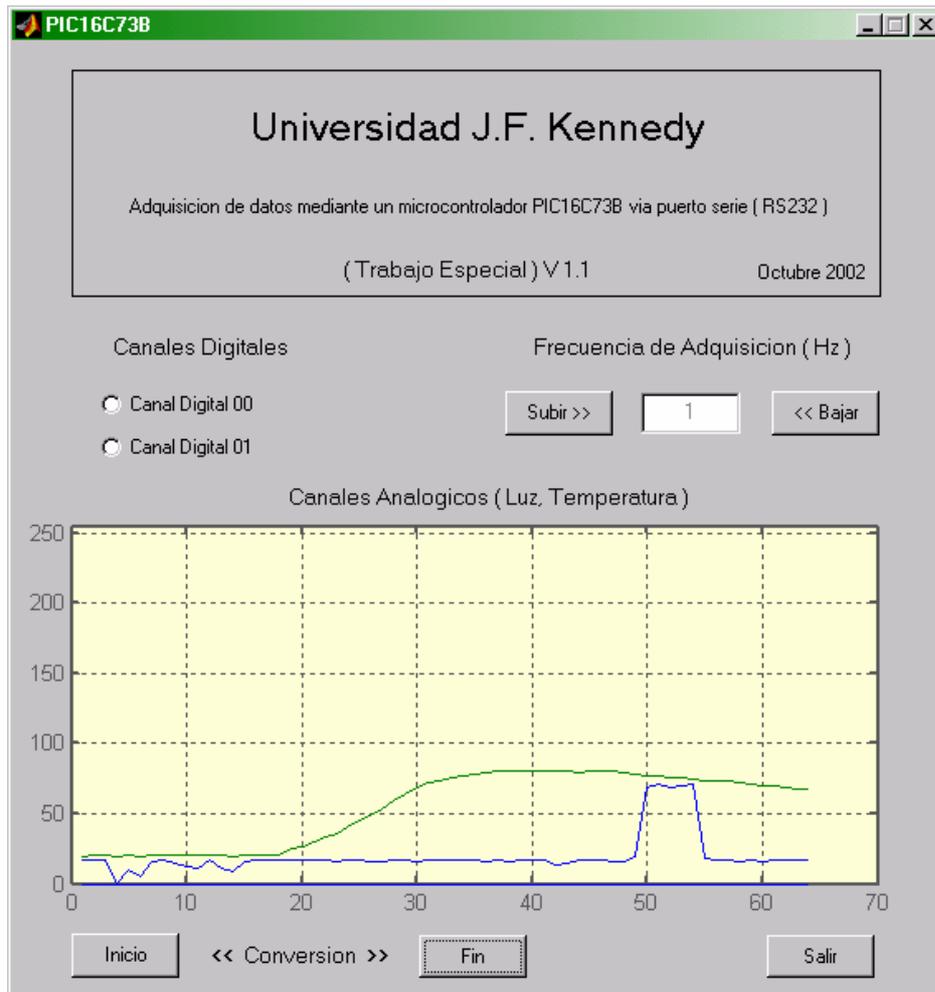


Figura 2.

B.4 - Aplicación de adquisición de datos en Visual Basic

Esta aplicación es la que se encarga de obtener los datos enviados a través del puerto serial utilizando el protocolo RS232 y graficarlos. Está desarrollada en Visual Basic y utiliza básicamente dos componentes estándares para realizar esto:

- **MSComm** que se encarga del manejo de la comunicación a través del puerto serial, permitiendo la comunicación desde y hacia el hardware.
- **MSChart** que permite representar en forma gráfica los datos recibidos desde el hardware.

Al iniciar la aplicación se le establece al objeto MSComm los parámetros de conexión, la conexión se realiza a 9600 bits por segundo, sin utilizar bit de paridad, con 8 bits de datos y 1 bits de parada. Una vez establecido estos parámetros se abre el puerto serial COM1 para la conexión con el hardware, además se abre un archivo de Log en el cual se registrará la hora, en formato “hh:mm:ss”, en la que se reciben los datos y el valor para cada uno de los canales en el formato “Digital 00 Analógico 00 Digital 01 Analógico01”.

Para comenzar la transmisión de datos con el hardware, se debe hacer click en el botón “Inicio”, en ese momento se le envía la señal de que el sistema esta listo para recibir la información. A medida que se van recibiendo los datos, se van procesando. Los datos que se reciben desde los canales analógicos se cargan en una matriz para que posteriormente puedan ser graficados utilizando el control MSChart, por el contrario la información recibida a través de los canales de señales discretas, se visualizan en pantalla utilizando para ello dos controles OptionButton.

Para finalizar la comunicación con el hardware se debe hacer click en el botón “Fin”

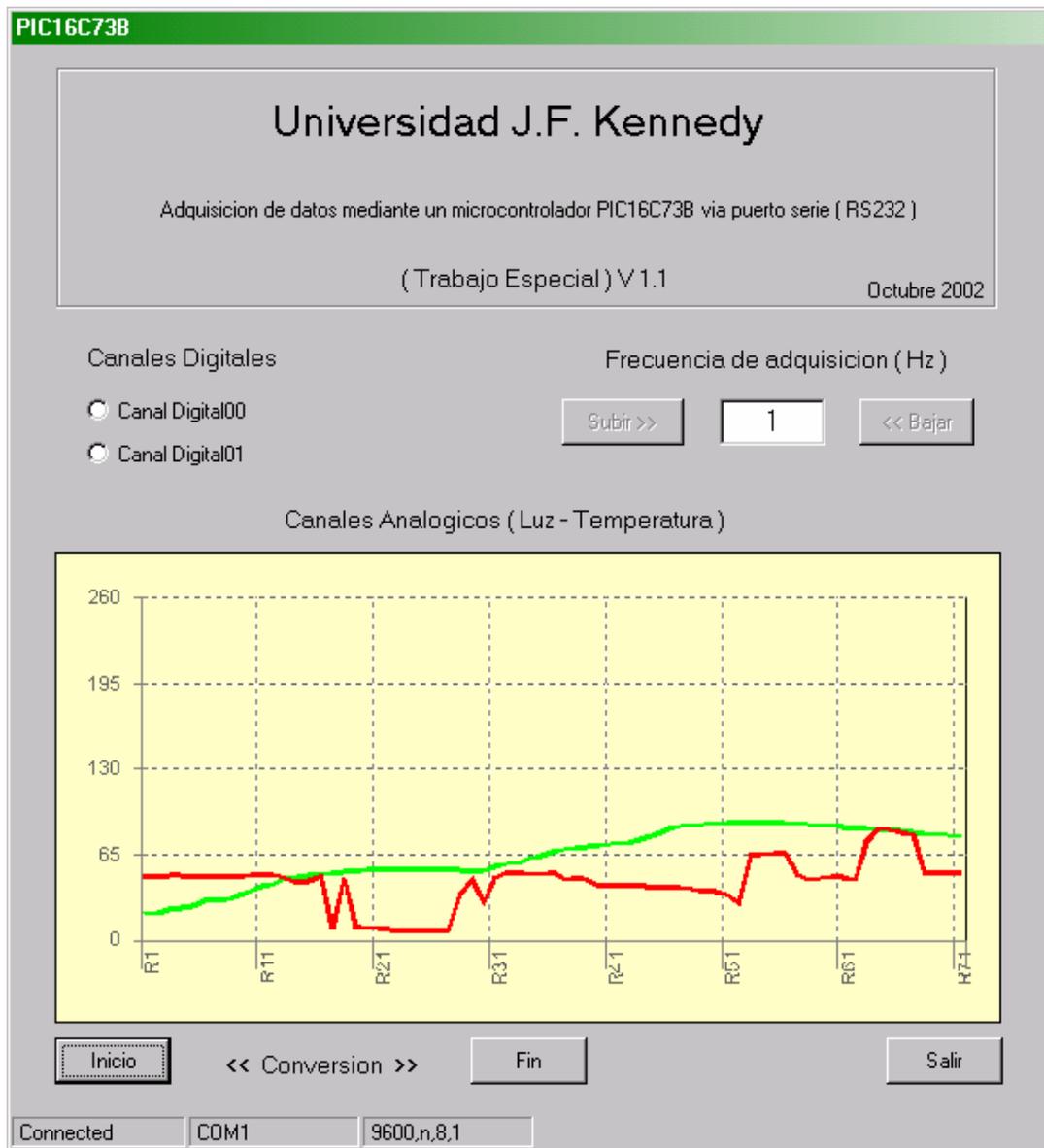


Figura 1.

C. Conclusiones

Para lograr el objetivo se recurrió:

A la comunicación con una computadora para la visualización y registro de datos.

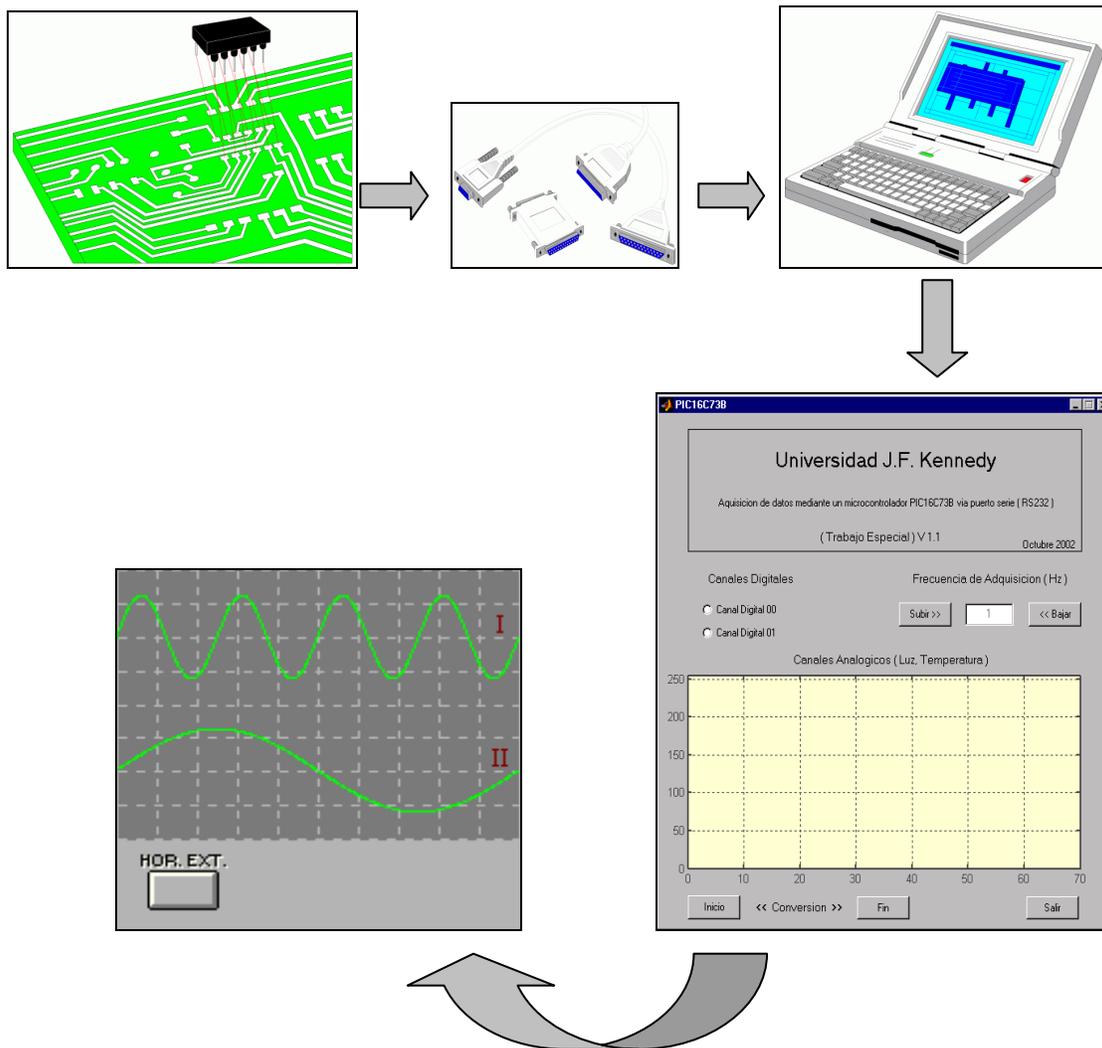


Figura 1.

El software utilizado para la comunicación y el monitoreo de los parámetros fue:

- **Una Interfaz de usuario en MatLab.**

Se pudo observar como el “PIC16C73B” se conectó al MatLab a través del puerto serie, para la adquisición de datos enviados por:



- **un Sensor de temperatura:**

La medición de temperatura se realizó utilizando un termistor NCT como sensor primario y en conjunto con el conversor Analógico Digital del PIC.

En algunos casos, la resistencia de un termistor a temperatura ambiente puede disminuir hasta un 6% por cada 1°C que se eleve la temperatura. En la característica voltaje-corriente, la caída de voltaje aumentó con el incremento de corriente hasta que alcanzó un valor pico, más allá del cual la caída de voltaje decrece con el incremento de corriente.

El termistor tiene una variación que va desde los 0.5 ohms a 75 ohm. Para obtener una tensión que varíe con la temperatura y que esté acorde con la resolución del Conversor Analógico Digital, se lo calentó el termistor con un encendedor.

Se pudo notar notar que cuando se aumentaba la temperatura, aumentaba la resistencia y viceversa.



- **un Detector de presencia:**

Informó al PIC la detección o no de presencia. La escala que se utilizó fue de dígitos binarios, ya que ha sido demostrado, que la detección de movimiento es una señal discreta que envía un 0 ó 1 como respuesta.



○ **y un Detector de Luminosidad**

Un Fotoresistor con características de ser sensible a la luz. Si se mide la variación de su resistencia acepta valores entre los 0 y 5 Vcc.

Este detector, es el encargado de proporcionar la información respecto de la magnitud de corriente que circula por la resistencia. Esta tensión ingresa al PIC, tendiente a que el sistema lo detecte. Cuando el fotoresistor se encuentra en completa oscuridad presenta una elevada resistencia. En cambio cuando es afectada por la luz, su resistencia disminuye. De lo anterior se desprende que el fotoresistor ofrece una resistencia que es inversamente proporcional a la intensidad de la luz que lo afecta.

Se comprobó el control de los sensores, mediante diferentes alternativas, registrando en un archivo histórico la evolución de las magnitudes medidas. Para ello se repitió un itinerario de eventos que a continuación se datallan:

```
"03:28:27"," -> "," 0 48 1 20 "  
"03:28:28"," -> "," 0 48 1 20 "  
"03:28:29"," -> "," 0 48 0 21 "  
"03:28:30"," -> "," 0 49 0 23 "  
.....  
"03:29:37"," -> "," 0 51 0 80 "  
"03:29:38"," -> "," 0 51 0 79 "  
"03:29:39"," -> "," 0 52 1 78 "  
"03:29:41"," -> "," 0 52 0 78 "
```

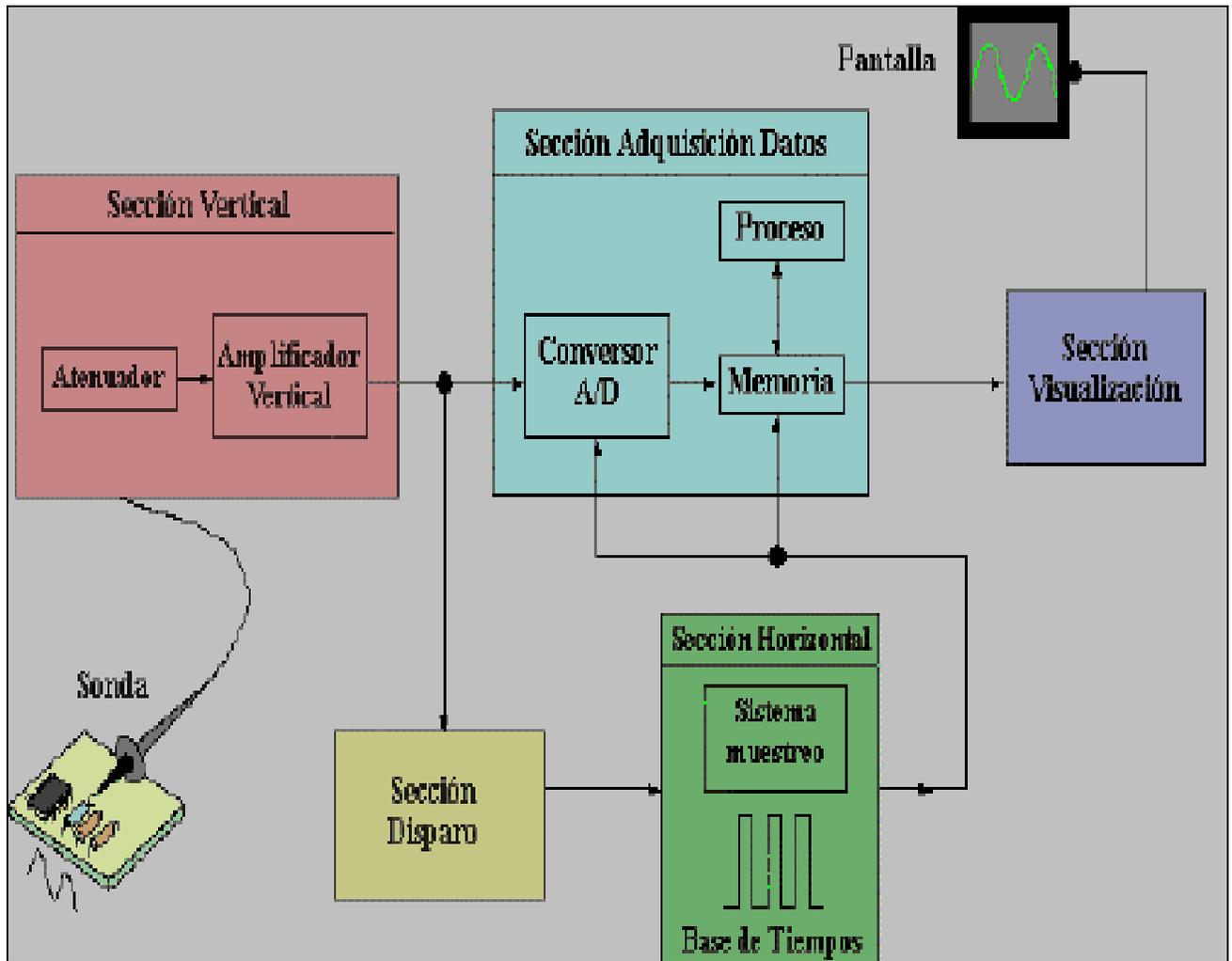


Figura 2.

En cuanto a la comparación entre las aplicaciones realizadas en Matlab y Visual Basic se puede decir que se encontró un error en la adquisición de datos en Matlab que se pudo intuir, se debe a los recursos utilizados por este último, que son mayores a los que utiliza Visual Basic. La frecuencia de adquisición en Visual Basic fue más exacta, se detectó una diferencia de tiempo de milisegundos.

Con las distintas pruebas que se realizaron sobre este prototipo, se comprobó que su comportamiento alcanzó y superó el desempeño esperado.

Con respecto a la adquisición de datos implementada, se puede decir que posee las ventajas de no necesitar un modelo matemático preciso del sistema a controlar, tiene alto rechazo al ruido y permite contemplar situaciones excepcionales del estado del proceso.

Como desventaja se puede decir, que el diseño debe realizarse generalmente con el método de prueba y error.

Algunas de las aplicaciones de los sistemas de adquisición de datos

Area de Energía

En esta área, se han desarrollado proyectos de control de demanda y control de climatización en edificios y centros comerciales (malls). Con aplicaciones que permiten controlar en un computador (control centralizado) por medio de un software HMI, las señales que son tomadas por lo general de un modulo de adquisición de datos o demanda de energía eléctrica según sea el caso.

Algunas de estas aplicaciones fueron hechas en:

- Hotel Intercontinental
- Edificio AETNA
- Edificio Compañía de Seguros de la Construcción
- Hotel Sheraton San Cristóbal
- Mall La Dehesa

Area de Industria Frigorifica

Son múltiples y variados los procesos de Automatización en esta área, algunos de los más frecuentes son controlar y administrar en un computador (control centralizado) por medio de un software HMI, las señales de temperatura entregadas por los sensores ubicados en las cámaras de frío o los procesos de producción controlando también condensadores, compresores y parámetros eléctricos. Estas señales son tomadas por el modulo de adquisición de datos y enviadas por cable o vía radio módem (si es necesario) al HMI, en este software de despliegan las pantallas gráficas con las variables de terreno, así el operador puede tomar decisiones respecto del funcionamiento o temperatura de cada cámara o proceso.

Algunas de estas aplicaciones fueron hechas en:

- Frigorífico Agrisouth - Maule (CDP)
- Frigorífico Santa Adriana - Ovalle (York Refrigeration Chile)
- Frigorífico Chorombo - Melipilla (York Refrigeration Chile)
- Frigorífico SACOR - Punta Arenas (York Refrigeration Chile).
- Frigorífico Pesca Chile - Punta Arenas (Great Ltda)

Area de Industria Pesquera

En los barcos pesqueros de alta mar, es muy necesario conocer las temperaturas de las cámaras de frío y tener controlado minuto a minuto los estados de los distintos componentes mecánicos y eléctricos del barco, es así como a través de un modulo de adquisición de datos se pueden recoger las señales de maquinas, motores, iluminación, y otros que se procesan y se pueden reportar a un software HMI o solo dar alarmas. Algunas de estas aplicaciones fueron hechas en barcos de la flota de la empresa Pesca Chile.

Area Industrial

En el área de procesos industriales, se han desarrollado proyectos de medición y control de caudales de aceites de pescado, medición y control de largos en maquinas de corte de cañerías en plantas de fabricación de cañerías y de medición y control de temperaturas en maquinas para la fabricación de flotadores de mallas de pesca, En estas aplicaciones es común la utilización de modulos de adquisición de datos, y sensores instalados en los procesos. La información es entregada a un software HMI para ser desplegada en la pantalla o puede ser también enviada a un proceso administrativo para su proceso (Ej. facturación). También se han desarrollado controles para maquinas productoras de hielo y para procesos de envasado.

Algunas de estas aplicaciones fueron hechas en:

- Concha y Toro
- Cambiaso Hnos
- Super Pollo
- Química Spes
- Industrias THC
- Industrias Themco
- ICER
- Mainar

Area Hospitalaria

En esta área, se ha desarrollado la aplicación de un control de señales de estado de iluminación y alarmas de un Hospital, las que son tomadas por un modulo de adquisición de datos y monitoreadas por un software HMI que despliega múltiples pantallas gráficas con la información de los distintos sectores del Hospital, de esta manera los operadores tienen un completo panorama del estado de las variables controladas. Esta aplicación se encuentra funcionando en el Hospital Naval en Viña del Mar.

Area Servicios Sanitarios

En esta área en particular, existen aplicaciones en el control de:

- Plantas elevadoras y de tratamiento tanto de agua potable como de aguas servidas.
- Control de niveles de pozos y estanques.

También se han desarrollado aplicaciones con Variadores de Frecuencia para mantener presión y/o nivel constante y la instalación de Partidores Suaves, entre otros.

En la gran mayoría de estos procesos, se han utilizado modulos de adquisición de datos y softwares HMI, además de los sensores e instrumentos necesarios para obtener la información de terreno. La información gráfica puede ser desplegada en un computador o en una pantalla de interface operador.

Algunas de estas aplicaciones fueron hechas en:

Aguas Cordillera

- Planta Elevadora Padre Hurtado
- Planta Elevadora Las Flores
- Planta Elevadora Santa Teresita
- Planta Elevadora Florencio Barrios
- Planta Elevadora La Ermita de San Antonio
- Planta Elevadora El Alba
- Control de Niveles Tranque La Dehesa
- Planta Elevadora Bilbao
- Planta de Tratamiento Padre Hurtado

- Varios controles de pozos y estanques.

BAPA

- Planta de tratamiento de aguas servidas Cantagua
- Planta de Tratamiento Viña Selentia
- Planta de tratamiento Viña Concha y Toro

Aguas Quinta

- Plantas de tratamiento y aguas servidas - Algarrobo.
- Planta Agua Potable San Antonio

ESSBIO

- Seis Plantas Tratamiento Aguas Servidas
- Planta de agua potable - Nacimiento
- Planta de agua potable - Los Alamos

ESSAR

- Planta de agua potable - Pucón
- Sistema de Alarma vía Radio en 17 localidades
- Planta de agua potable - Villarrica
- Planta de agua potable - Lumaco
- Planta de agua potable - Cholchol

SMAPA- Maipú

- Control de Niveles de estanques elevados

Agua Potable Manquehue

- Planta de Bombeo Canal La Dehesa

Biwater

- Telemetría de pozos y estanques en Chicureo

Monitoreo y Análisis de una Central Termoeléctrica

El Objetivo:

Mantener un registro histórico del funcionamiento de una central generadora de energía basada en carbón, para permitir su análisis y la optimización de su proceso, con un bajo costo de desarrollo y mantenimiento.

Introducción

La Central Termoeléctrica de Tasajero (TERMOTASAJERO S.A. E.S.P.) posee una potencia de generación de 150 MW netos. Esta se basa en la combustión de carbón. A máxima potencia, la central consume 53.05 t/h de carbón, suficientes para producir 488 t/h de vapor, que actúan sobre una turbina de 3 etapas. La energía generada por esta central es suficiente para satisfacer la demanda del departamento Norte de Santander en Colombia.

Inicialmente se definieron las funciones para el sistema:

1. Presentar esquemáticos de planta donde se pueda apreciar en tiempo real el comportamiento de las variables de cada proceso presentado.
2. Mantener un registro histórico de las tendencias de las variables del proceso durante un año, permitiendo acceso fácil y rápido a la información almacenada.
3. Llevar el seguimiento de eventos (Secuencia de eventos) para diagnósticos e informes extraordinarios.
4. Informar de estados anormales y de riesgo durante la operación, para permitir una rápida acción correctiva.
5. Trabajar sobre una base de medidores y transmisores ya instalados, tanto de señales análogas como digitales.

El Sistema de Adquisición de Datos

El sistema de adquisición de datos se basa en una red Ethernet, sobre la cual se encuentran seis módulos de comunicaciones FP-1600, cada uno con una carga promedio de 7 módulos de adquisición de datos. A esta misma red se conectan tres estaciones que actúan como clientes, y un servidor que almacena la información del proceso. Este último se conecta a la red empresarial (segmentando la red), para ofrecer la información a través de Internet Explorer 5. El total de señales que se monitorean es de 474: 229 análogas, con rangos de 1 ~ 5, 0 ~ 10 V y 4 ~ 20 mA; y 245 digitales de contactos libres de potencial. Dadas las limitaciones económicas impuestas para el proyecto, y debido a que el sistema de control de esta central es antiguo (basado en relés y tarjetas de baja escala de integración), se optó por adquirir módulos DI-301, que permiten medir señales de contactos libres de

potencial. Esto garantiza que se obtenga la señal de respuesta del propio sistema, en el tiempo exacto que ocurra.

Propiedades Operativas

El software Lookout se emplea como asistente de operación de la central, presentando en 3 estaciones las condiciones de operación en tiempo real. Gracias al poderoso manejo de gráficas de Lookout, se pueden presentar los estados de las variables de proceso en diagramas esquemáticos de los sistemas de planta, navegando a través de estos con un simple click sobre un elemento (ya que se comportan como mapas sensitivos), o por medio de un menú. También se emplean animaciones, así como la activación automática de paneles ante condiciones anormales en el sistema.

El objeto hypertrend del software Lookout permite presentar las tendencias de las señales. Este permite acceso instantáneo tanto en tiempo real como histórico de todos los datos. Gracias a los miembros de datos del mismo, es posible flexibilizar la forma de ver los valores almacenados en la base de datos Citadel. Es posible hacer zoom tanto horizontal como vertical, ocultar y presentar las señales, para así no perder ningún detalle del comportamiento de la central.

Esta flexibilidad es llevada a su máxima expresión, al permitir al usuario definir las variables que desee ver en el hypertrend, definir el valor mínimo y máximo de escala y guardar su propia configuración, para poderla usar posteriormente.

La funcionalidad de los objetos alarm también fue aprovechada para definir 87 mensajes de alarma sonoros, los cuales permiten a los operadores de la central estar conscientes todo el tiempo del funcionamiento de esta.

Adicional a esto, el sistema calcula la eficiencia de la planta en tiempo real, permitiendo al operador mejorar condiciones de combustión, presión, etc.

Resultados Obtenidos

- Optimización del consumo de materia prima.
- Generación de reportes mensuales de eficiencia, con gráficas comparativas de consumo nominal vs. consumo real.
- Generación de reportes diarios de potencia generada, consumo de carbón y agua desmineralizada.
- Disminución drástica del número de disparos de planta, gracias al aviso oportuno de condiciones anormales de operación (cada disparo cuesta como mínimo US\$ 50.000).
- Análisis de fallas ocurridas y de la respuesta de la unidad, gracias a la información almacenada en la base de datos histórica, y la cual anteriormente no estaba a disposición.

- Seguimiento fino de generación, permitiendo mejorar los márgenes de desviación de carga definidos por el Sistema de Interconexión Eléctrica.