



UNIVERSIDAD NACIONAL DE
MAR DEL PLATA



FACULTAD DE INGENIERÍA

Sistema Autónomo de Adquisición de Datos Portátil

Data Logger

ESTEBAN J. SARAVIA

DIRECTOR: RAÚL RIVERA
WALTER GEMIN

FECHA: NOVIEMBRE 2004

LUGAR: LABORATORIO DE PROCESOS Y MEDICIÓN DE SEÑALES



RINFI es desarrollado por la Biblioteca de la Facultad de Ingeniería de la Universidad Nacional de Mar del Plata.

Tiene como objetivo recopilar, organizar, gestionar, difundir y preservar documentos digitales en Ingeniería, Ciencia y Tecnología de Materiales y Ciencias Afines.

A través del Acceso Abierto, se pretende aumentar la visibilidad y el impacto de los resultados de la investigación, asumiendo las políticas y cumpliendo con los protocolos y estándares internacionales para la interoperabilidad entre repositorios



Esta obra está bajo una [Licencia Creative Commons Atribución- NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).



UNIVERSIDAD NACIONAL DE
MAR DEL PLATA



FACULTAD DE INGENIERÍA

Sistema Autónomo de Adquisición de Datos Portátil

Data Logger

ESTEBAN J. SARAVIA

DIRECTOR: RAÚL RIVERA
WALTER GEMIN

FECHA: NOVIEMBRE 2004

LUGAR: LABORATORIO DE PROCESOS Y MEDICIÓN DE SEÑALES

INDICE

Resumen.....	1
Introducción.....	2
Anteproyecto.....	4
Proyecto.....	6
1 Fuente de alimentación de bajo consumo.....	6
1.1 Diseño.....	6
1.2 Funcionamiento en 5 volts.....	7
1.3 Funcionamiento en modo sleep.....	7
2 Microcontrolador PIC16F873.....	8
2.1 Programación del microcontrolador.....	10
2.2 Oscilador / Modo sleep.....	10
2.3 Interrupciones.....	11
2.4 Timer1 y Timer0.....	11
2.5 Conversor A/D.....	12
2.6 Puertos de Entrada/Salida.....	13
2.7 Puerto Serie Sincrónico.....	14
3 Reloj de Tiempo Real.....	15
3.1 Oscilador / Divisor CD4060.....	15
3.2 Uso del Timer1.....	15
4 Adquisición de Datos.....	17
4.1 Tiempo en espera.....	17
4.2 Intervalo entre muestras.....	17
4.3 Toma de un dato.....	18
4.4 Modo en Pausa.....	18
5 Memoria de Datos.....	20
6 Tipos de Sensores.....	22
6.1 Sensor de Temperatura LM335.....	22
6.2 Sensor de Humedad HIH3610.....	23
6.3 Sensores Alineales.....	24

7	Interfase de comunicación con la PC.....	26
7.1	I ² C (Inter-Integrated Circuit).....	27
7.2	I ² C Mediante RS-232.....	28
8	Software de Aplicación.....	30
8.1	Programación del Data Logger.....	31
8.2	Lectura de Datos.....	34
8.3	Calibración y Configuración de Sensores.....	36
9	Operación y Mantenimiento.....	38
9.1	Aspecto del Data Logger.....	38
9.2	Conexión de los Sensores.....	39
9.3	Conexión a la PC.....	39
9.4	Recomendaciones para su Correcto Uso.....	40
9.5	Especificaciones Técnicas.....	41
	Resultados y Conclusiones.....	42
	Bibliografía.....	43
	Anexos	
A	Circuitos Esquemáticos.....	44
A.1	Fuente de Alimentación.....	44
A.2	Circuito Principal.....	45
A.3	Cable de Conexión PC.....	46
B	Programa del Microcontrolador (diagrama de flujo).....	47

RESUMEN

Se desarrolló un sistema de adquisición y registro automático de datos de cuatro canales destinado al monitoreo y reporte de cambios en las condiciones ambientales. El mismo fue solicitado la Facultad de Ciencias Agrarias.

Dicho sistema consta principalmente de un microcontrolador de la línea PIC de Microchip, una memoria EEPROM serie para el almacenamiento de los datos y la posibilidad de conectar hasta cuatro sensores. Este dispositivo es alimentado por una batería de 9 volts, lo que junto con su tamaño reducido le permite ser fácilmente transportado.

La lectura de los datos se efectúa mediante una PC, a la que se accede mediante su puerto serie. El software desarrollado permite la lectura de los datos recopilados en el dispositivo y su presentación en pantalla. Estos datos a su vez pueden ser convertidos en un formato de planilla de cálculos Excel. El usuario puede ingresar y configurar el tipo de sensor que desee, mientras se ajuste a las especificaciones de entrada de cada canal. También puede definir los intervalos de tiempo entre muestras para cada uno de los sensores y el tipo de sensor conectado.

La capacidad de almacenamiento es de 16300 muestras, y puede programarse su lectura en intervalos desde 1 segundo hasta 6 hs.

El consumo es muy reducido. Fijándose un intervalo entre muestras de 1 minuto y utilizando una batería alcalina de 9V y 500mAh, la autonomía es de aproximadamente un año.

El sistema permite puede registrar en forma independiente variables tales como temperatura, humedad relativa, intensidad de la luz, voltaje, presión, etc.

INTRODUCCIÓN

¿QUE ES UN DATA LOGGER?

Un Data Logger es un dispositivo electrónico que registra mediciones en el tiempo, provenientes de diferentes sensores. Luego cada medición es almacenada en una memoria, junto con su respectiva fecha y hora.

En general los Data Loggers son pequeños y alimentados por baterías, y están conformados por un microprocesador, una memoria para el almacenamiento de los datos y diferentes sensores. La mayoría utilizan a la PC como interfase para programar al dispositivo y leer la información recolectada.

PRINCIPALES CARACTERÍSTICAS

Como punto de partida se buscó desarrollar un sistema que fuera competitivo frente a productos similares existentes en mercado, con respecto al costo y a las características más relevantes.

Para ello se propuso investigar dichos productos en cuanto a sus especificaciones, compararlas y marcar las más importantes. Luego de esto se obtuvo la conclusión de que el sistema debería cumplir con las siguientes características:

- Ser portátil, es decir, funcionar con baterías y tener un peso y tamaño que le permita ser transportado con facilidad.
- Tener un bajo consumo: hay que tener en cuenta que este tipo de sistemas pueden trabajar durante semanas, meses y hasta años, por lo tanto es muy importante el consumo y la capacidad de las baterías.
- Intervalos de muestreo programables y con la mayor flexibilidad posible, pudiendo ser desde cantidades en segundos hasta en horas. Esto permite registrar variables con diferentes velocidades de variación.
- Tener una buena capacidad de almacenamiento de datos. En este punto entran en juego las características anteriores, por lo tanto hay que determinar una cierta cantidad de memoria, teniendo en cuenta que la duración de las baterías depende del consumo en el tiempo y el tiempo de trabajo dependerá del tiempo entre muestras y la capacidad de memoria.

- Bajo costo de los componentes y disponibilidad. Es muy importante ya que para ciertas aplicaciones se podrían necesitar varios equipos, o bien, puedan ser utilizados en lugares donde corran el riesgo de ser destruidos.
- Una interfase con el usuario a través de una PC, donde el usuario pueda programar o leer el dispositivo de una manera sencilla y rápida, utilizando aplicaciones que le sean familiares para analizar la información.

APLICACIONES

Los Data Loggers pueden ser construidos para controlar todo tipo de datos ambientales. La temperatura y la humedad son las más comunes. Según se requiera pueden servir para otras aplicaciones. Por ejemplo han sido utilizados por muchos años en los servicios meteorológicos para medir la humedad, presión atmosférica, los niveles de precipitación, etc. Del mismo modo, los museos los usan para medir y controlar las condiciones de conservación de obras de arte.

Pero no solo son dispositivos estáticos, también pueden ser transportados, por ejemplo, para registrar la cadena de frío de una carga desde una planta hasta el centro de distribución.

ANTEPROYECTO

TECNOLOGÍA UTILIZADA

Ya se han establecido las partes principales que conforman un Data Logger (microcontrolador, memoria y sensores), ahora se analizarán más en detalle cada una de estas partes, buscando las mejores alternativas para su desarrollo.

El microcontrolador es la parte fundamental del sistema, pero también la más costosa. Para este desarrollo se utilizó el microcontrolador PIC16F873 de la firma Microchip por tener una buena relación entre costo y prestaciones, el cual posee, entre otras características:

- un conversor A/D de 10 bits, con 5 canales de entrada,
- puertos de comunicación serie sincrónicos (I²C) y asincrónicos (RS232),
- modo sleep de muy bajo consumo,
- timers o contadores que pueden ser incrementados por un oscilador externo,
- buena capacidad de memoria de programa,
- bajo costo,
- disponibilidad en el mercado local, esto es muy importante si se busca reproducir el sistema en cantidades.

La memoria de datos debe ser del tipo no volátil o EEPROM, ya que esta tecnología permite retener la información aún sin alimentación. El microcontrolador posee una memoria de este tipo, pero es de tamaño reducido, por lo tanto es necesario una memoria externa de mayor capacidad, lo que implica una interfase de comunicación con ella.

Esta puede ser serie o paralelo. Una interfase paralelo es rápida y fácil de implementar, pero necesita varias líneas de transmisión. Una serie requiere muy pocas líneas y es la ideal para esta aplicación, ya que la velocidad no es un factor importante.

La memoria EEPROM serie utilizada es la 24LC256 (de 32Kbytes) de la marca Microchip, la cual posee una interfase de comunicación sincrónica denominada Inter-Integrated Circuit (I²C), siendo compatible con el microcontrolador seleccionado.

Los sensores son la otra pieza fundamental, ellos se encargan de traducir una medida en un valor de tensión que ingresará al conversor A/D, de aquí la importancia de la resolución del conversor (10 bits en este caso).

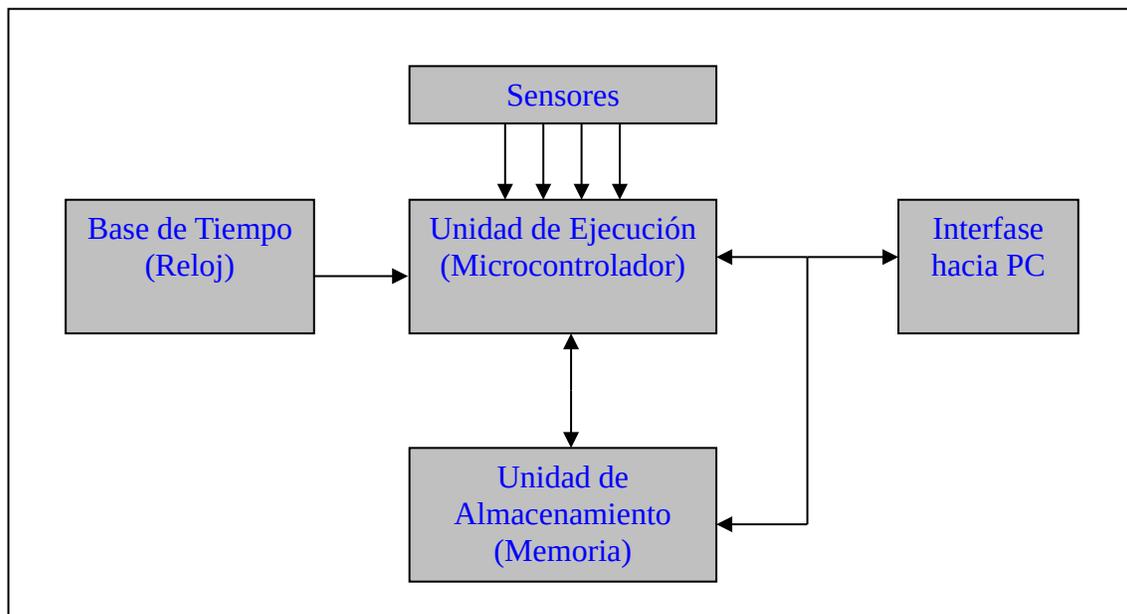
Pero esta conversión se debe hacer en un instante de tiempo programado, por lo tanto es necesario un circuito que genere una señal de reloj que ingresará al microcontrolador, para mantener un sincronismo con la fecha y hora actual.

Cuando la información ha sido recolectada en la memoria EEPROM, deberá ser transportada hacia la PC para su presentación, para esto es necesario una interfase de comunicación con la misma. Una comunicación serie RS232 sería una buena opción, ya que es compatible con el microcontrolador, pero implica que los datos deben ser transportados de la memoria al microcontrolador (interfase I²C) y luego del microcontrolador hacia la PC (interfase RS232). Esta operación se puede tornar lenta si tenemos en cuenta que la cantidad de datos a transferir es importante.

Para tener una mayor velocidad en la transferencia de datos la mejor opción es implementar una interfase de comunicación desde la PC a la memoria directamente. Pero la PC no posee una interfase I²C, por lo tanto se decidió emularla por medio de la interfase serie RS232, existiendo la posibilidad de hacer lo mismo por el puerto paralelo de la PC.

DIAGRAMA EN BLOQUES

El siguiente diagrama en bloques representa la forma en como se conectarán las partes.



PROYECTO

1.FUENTE DE ALIMENTACIÓN DE BAJO CONSUMO

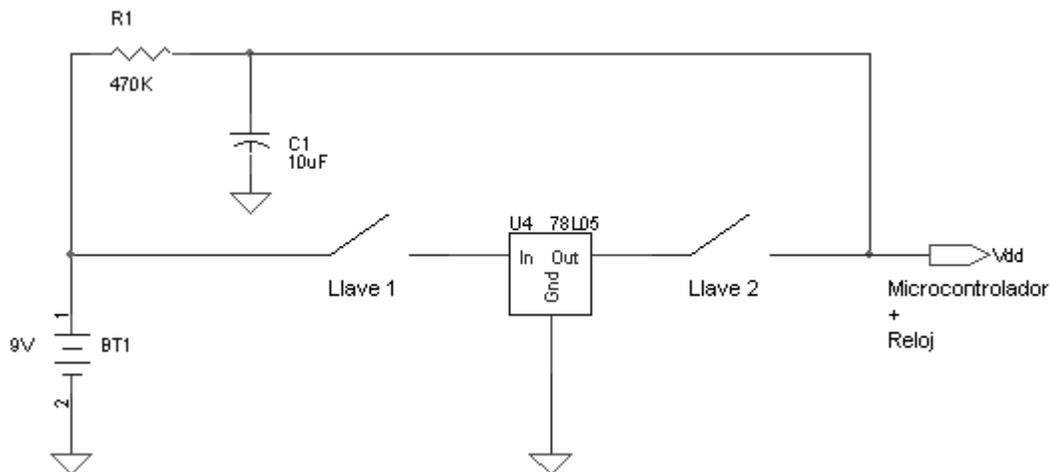
El ahorro de energía es la característica que diferencia un Data Logger de otros sistemas de adquisición de datos. El método de operación fundamental de un Data Logger radica en el hecho de que la mayoría del tiempo se encuentra apagado. Solamente se enciende cuando debe tomar y almacenar un dato, y luego se vuelve a apagar. En otras palabras, el dispositivo por sí solo debe administrar el consumo de corriente.

Para asegurar una mayor duración de la batería, este sistema permite que las diferentes partes que componen el dispositivo alimenten por separado, por medio de llaves comandadas por el microcontrolador, permitiendo regular el consumo en las diferentes etapas de operación.

Solamente existe un pequeño consumo en forma constante, perteneciente a la etapa de reloj, que se encarga de mantener el tiempo actualizado, e indicar al microcontrolador que ha pasado el intervalo de tiempo programado y que debe tomar las muestras.

1.1.DISEÑO

El siguiente esquema representa el diseño de la fuente de alimentación:



Se alimenta por una batería de 9 volts alcalina (IEC-6LR61), que según el manual del fabricante ofrece una capacidad de 500mAh (miliamperes-hora). **La batería debe ser reemplazada antes de que su tensión caiga por debajo de los 7 volts.**

El sistema opera con una tensión de 5 volts, por ello se utiliza el regulador de tensión 78L05 (U4), el cual entrega una corriente máxima de 100mA y requiere de una corriente de polarización de aproximadamente 3,5mA.

Las dos llaves de la figura conectan o desconectan por completo al regulador de tensión del resto del circuito. Estas llaves se abren y cierran en forma simultánea y son comandadas por el microcontrolador.

1.2.FUNCIONAMIENTO EN 5 VOLTS

Cuando el microcontrolador cierra las llaves se generan 5 volts en el punto Vdd, además esto produce la carga del capacitor C1 a 5 volts. En esta instancia el consumo se debe principalmente a la polarización del regulador y a las etapas que se conecten a la salida del mismo (memoria, sensores, etc.).

Hay que notar que del mismo punto Vdd toman alimentación la etapa de reloj y el microcontrolador, por lo tanto se utiliza esta tensión regulada como tensión de referencia para el conversor A/D.

1.3.FUNCIONAMIENTO EN MODO SLEEP

Una vez que se han tomado las muestras, el microcontrolador abre las llaves y pasa a modo sleep (ver capítulo 2.2), siendo el modo en el cual se encuentra la mayoría del tiempo.

De esta forma el capacitor funcionará ahora como regulador de tensión, alimentando a la etapa de reloj desde la batería a través de la resistencia R1. En este momento el consumo se debe prácticamente esta etapa, que es de aprox. 11 μ A, despreciando el consumo del microcontrolador en modo sleep.

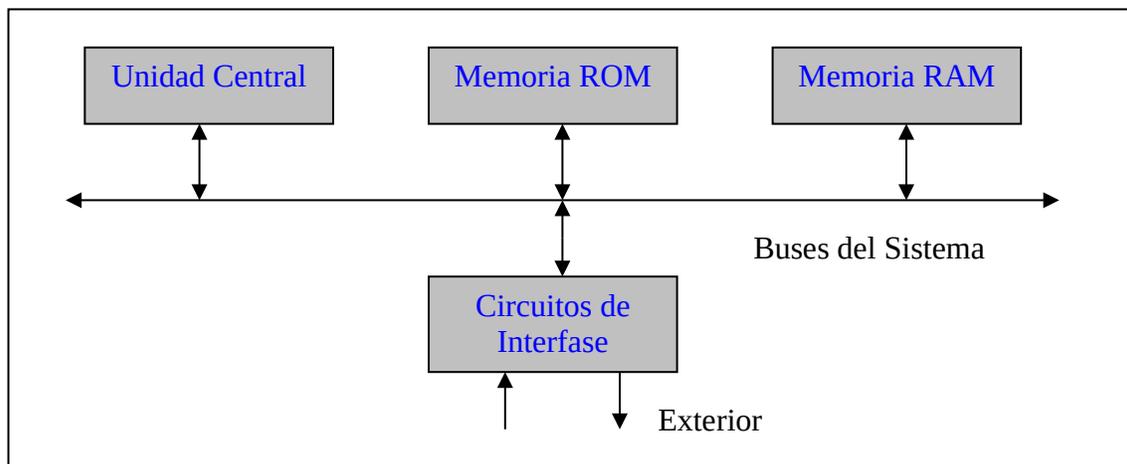
Esta corriente de 11 μ A produce una caída de tensión en R1 de 5.2volt, por lo tanto la tensión en el punto Vdd (en modo sleep) será de $9 - 5.2 = 3.8$ volts aproximadamente.

El valor de capacidad C1 ha sido calculado para poder suministrar la energía necesaria para el encendido del microcontrolador, que luego cerrará nuevamente las llaves.

2.MICROCONTROLADOR PIC16F873

Un microcontrolador se utiliza cuando se tiene la necesidad de sistemas “inteligentes” o bien, programables. Un ejemplo muy simple es el programador de una lavadora, el cual debe controlar una cierta cantidad de elementos con ciclos perfectamente definidos, pero variables en función del programa seleccionado.

La figura presenta el esquema típico de cualquier microcontrolador.



La presencia de estos elementos básicos es indispensable y aunque no se presenten tan claramente como en este esquema, siempre existen en todo sistema programable.

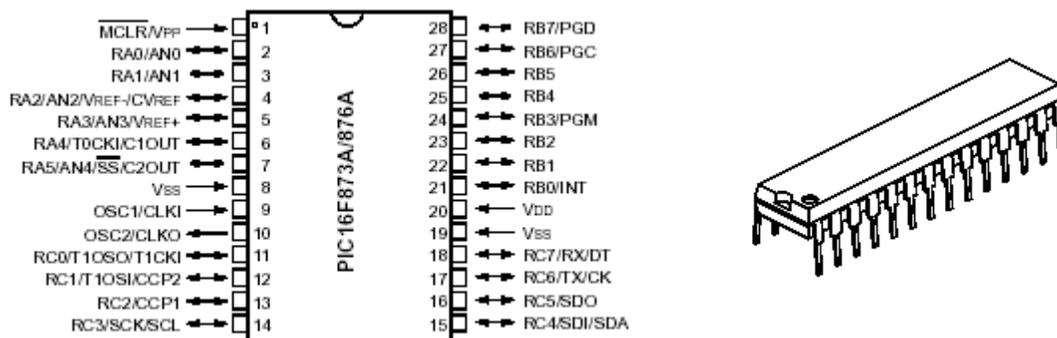
La unidad central, generalmente constituida por un microprocesador más o menos evolucionado, ejecuta el programa que da vida a la aplicación. El programa se almacena en un segundo elemento, que es la memoria ROM. Esta memoria puede constituirse de diferentes formas: UVPROM, EEPROM u OTPROM, cualquiera que sea la que se utilice es una memoria no volátil desde donde se ejecutará el programa una vez alimentado el sistema.

Para poder trabajar correctamente, un microprocesador necesita almacenar datos temporales en alguna parte, y para ello existe la memoria RAM, que no necesita ser de grandes dimensiones.

El último elemento y que, generalmente, es el más importante en una aplicación es todo lo concerniente a los circuitos de interfase con el mundo exterior.

Un microcontrolador integra en un único encapsulado lo que antes necesitaba una o varias decenas de elementos discretos. Como resultado de su aplicación, se obtiene una evidente disminución en el volumen del hardware y del circuito impreso. Esta integración también ha tenido como consecuencia inmediata la simplificación del diseño del circuito impreso, ya que no es necesario llevar los buses de direcciones y de datos de un componente a otro. Además, la cantidad de conexiones entre componentes/zócalos o componentes/circuito impreso también disminuye, lo que aumenta la fiabilidad del sistema. Así mismo, la disminución de componentes reduce el consumo global de todo el sistema.

La siguiente figura muestra la disposición de pines del microcontrolador PIC16F873 fabricado por la empresa Microchip, utilizado en este proyecto.



A continuación se enumeran sus características más significativas:

- Memoria de programa: 4096 palabras de 14bits
- Juego de Instrucciones: 35 (de 14 bits)
- Memoria de Datos RAM: 192 bytes
- Memoria de Datos EEPROM: 128 bytes
- Líneas digitales de entrada/salida: 22
- Conversor A/D de 10 bits con 5 canales de entrada
- Timers: 2 de 8 bits y 1 de 16 bits
- Interfase Serie Sincrónica compatible I²C y SPI

En los incisos siguientes se explicará como se aprovecharon estas características en el proyecto.

2.1.PROGRAMACIÓN DEL MICROCONTROLADOR

En los microcontroladores, como cualquier sistema programado, hay que disponer de una herramienta o medio de desarrollo para escribir los programas y probarlos antes de instalarlos y hacer funcionar el sistema.

En este caso se utilizó el entorno de desarrollo Mplab, el cual es distribuido por la empresa Microchip. Mediante esta aplicación se ingresan todas las instrucciones que conforman el programa, utilizando el juego de instrucciones propio del microcontrolador.

Una vez comprobado el programa, se compila, generando un archivo que contiene el código que se grabará físicamente en el dispositivo. Para ello se utilizó el programador de dispositivos PonyProg, siendo necesaria la construcción del circuito que se conecta al puerto serie de la PC y que permite transferir el programa desde la PC al microcontrolador.

2.2.OSCILADOR/MODO SLEEP

La frecuencia de trabajo del microcontrolador es un parámetro fundamental a la hora de establecer la velocidad de ejecución de instrucciones y el consumo de energía. Cuanto menor sea la frecuencia de trabajo, menor será el consumo. Pero también hay que tener en cuenta que a este ritmo se ejecutarán las instrucciones, trabajará el conversor A/D, se transferirán los datos a la memoria, etc.

Por estas razones, se eligió una frecuencia de trabajo de 4 MHz, correspondiéndole un ciclo de instrucción de 1 μ seg, debido a que cada instrucción tarda en ejecutarse 4 períodos de reloj. Todas las instrucciones del PIC se realizan en un ciclo de instrucción, menos las de salto que tardan el doble. El consumo promedio del microcontrolador trabajando a esta frecuencia es de 1,5mA a 3mA.

El PIC además permite un modo denominado Sleep, en el cual desconecta el oscilador, por lo tanto detiene la ejecución de instrucciones y queda apuntado a la próxima instrucción a ejecutar. El resto del dispositivo sigue activo, solamente detiene la ejecución. De esta forma el consumo disminuye notablemente, siendo éste menor a 1 μ A.

Esta característica es muy útil para el proyecto, ya que ofrece un muy bajo consumo mientras se espera durante los intervalos de muestreo.

2.3. INTERRUPTIONES

Las interrupciones son desviaciones del flujo de control del programa originadas asincrónicamente por diversos eventos que no se hallan bajo la supervisión de las instrucciones. Dichos eventos pueden ser externos al sistema, como la generación de un flanco o nivel activo en una entrada digital del microcontrolador, o bien internos, como el desbordamiento de un contador.

Si el microcontrolador se encuentra en modo sleep y se produce una interrupción se enciende el oscilador y se comienza a ejecutar la rutina de atención a la interrupción.

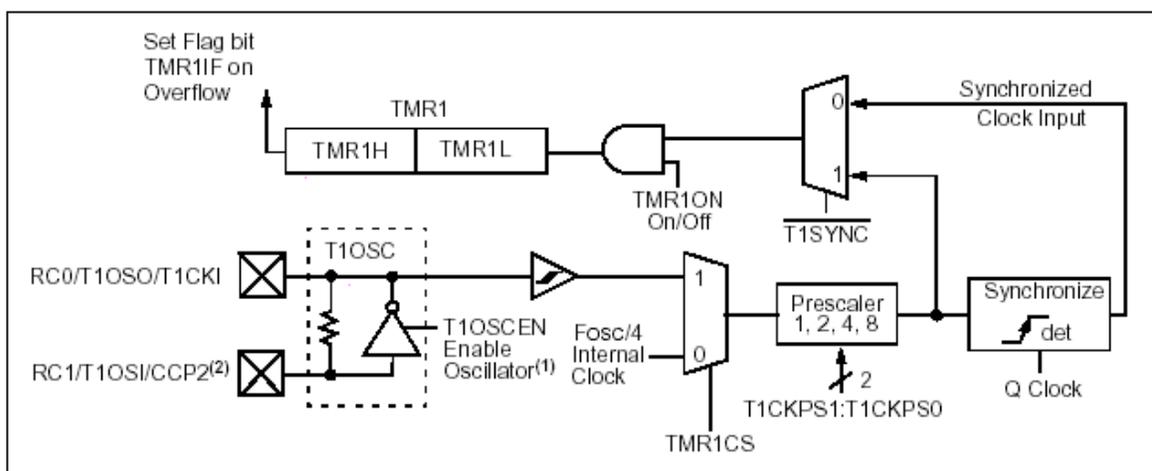
En este sistema se diferencian dos tipos de interrupciones, una debida al desbordamiento del Timer1, que indica la toma de muestras.

La otra fuente de interrupción es generada por la PC para enviar una orden al microcontrolador, para su programación o en la lectura de los datos. Se efectúa mediante un flanco positivo en el pin 3 del conector DB9 del Data Logger.

2.4. TIMER1 Y TIMER0

Estas partes del microcontrolador pueden cumplir dos funciones, primero como temporizadores (timers) para determinar intervalos concretos de tiempo, o bien como contadores de impulsos que se producen fuera del microcontrolador. Si estas tareas se asignaran al programa principal, quitarían mucho tiempo al procesador, que puede ocuparlo en otras actividades.

El PIC16F873 posee dos timers, el timer1 de 16 bits y el timer0 de 8 bits. En la figura se muestra la estructura interna del timer1, similar a la del timer0.



En su función como contador los pulsos ingresan por la pata T1CKI e incrementan el registro TMR1 de 16 bits, realizando con anterioridad la configuración correcta. Este registro se incrementa hasta el valor 65535 (FFFFh), y en el pulso siguiente pasa al valor cero, provocando además un alto el bit TMR1IF lo que produce una interrupción.

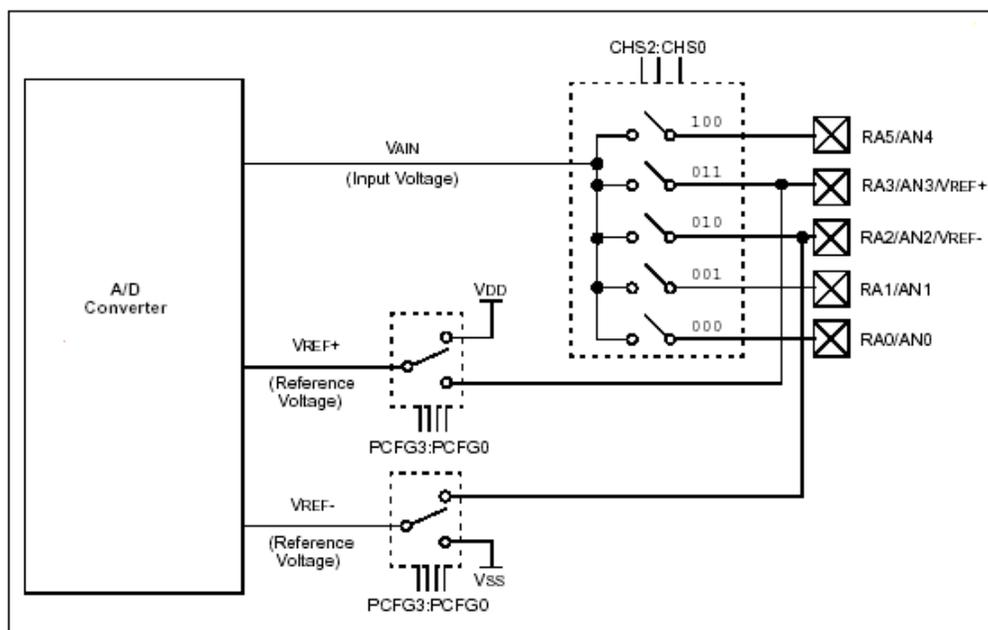
Como temporizador, se carga primero el registro TMR1 con un valor inicial, y si se conoce la frecuencia de los pulsos se puede determinar con exactitud el tiempo que llevará hasta que se produzca la interrupción. En este caso es útil también la opción que permite incrementar el registro TMR1 mediante el oscilador interno ($F_{osc}/4$).

En este proyecto se utiliza el timer0 con el oscilador interno para generar retardos de tiempo conocidos, como ser: tiempo de establecimiento de la fuente de alimentación, generación de sonido por el parlante, tiempo de escritura de memoria eeprom, etc.

El timer1 se utiliza solamente con un oscilador externo para determinar los intervalos de tiempo entre muestras.

2.5.CONVERSOR A/D

Esta unidad se encarga de transformar un valor de tensión analógica en un valor equivalente digital de 10 bits. Como se ve en la siguiente figura, posee 5 canales de entrada y la posibilidad de ingresar tensiones de referencia en forma externa.



En este proyecto las tensiones de referencia son: $V_{ref+}=V_{dd}=5\text{volts}$ y $V_{ref-}=V_{ss}=\text{Masa}$. Por lo tanto se obtiene una resolución de 5 milivolts en todos los canales, teniendo en cuenta los 10 bits del conversor.

El tiempo de conversión por cada bit (T_{AD}), se establece en base al período del oscilador principal (T_{OSC}). En este caso $T_{AD}=8 \cdot T_{OSC}=8 \cdot (1/4 \text{ Mhz})=2 \mu\text{seg}$.

Por cada conversión de 10 bits se necesitan al menos 12 T_{AD} , por lo tanto, el tiempo de conversión por cada dato es de 24 μseg .

Otros datos importantes ofrecidos por el fabricante son: El tiempo de adquisición, que corresponde a la carga del capacitor de sample and hold, debe ser mayor a 20 μseg y la máxima impedancia de la fuente a medir es 2,5K Ω . Si se cumplen estas especificaciones, se consigue la exactitud del conversor.

2.6.PUERTOS DE ENTRADA/SALIDA

Este microcontrolador posee un total de 22 líneas digitales de entrada/salida, pero estas líneas se encuentran agrupadas en puertos. El puerto A, posee 6 líneas y el puerto B y C poseen 8 líneas cada uno.

Algunas de estas líneas poseen otras funciones además de la nombrada, por ejemplo las líneas del puerto A también se utilizan como entradas al conversor A/D. La funcionalidad de cada una de las líneas se establece en el programa del microcontrolador.

A continuación se listan todas las líneas de entrada/salida y su uso en el proyecto:

Puerto A:

Línea	Pin	Descripción
RA0	2	Entrada analógica del canal 1
RA1	3	Entrada analógica del canal 2
RA2	4	Entrada analógica del canal 3
RA3	5	Entrada analógica del canal 4
RA4	6	Detección de estado "En Pausa"
RA5	7	Entrada analógica de tensión de batería / 2

Puerto B:

Línea	Pin	Descripción
RB0	21	Entrada de interrupción externa (PC)
RB1	22	Sin uso
RB2	23	Conecta alimentación al canal 1
RB3	24	Conecta alimentación al canal 2
RB4	25	Conecta alimentación al canal 3
RB5	26	Conecta alimentación al canal 4
RB6	27	Conecta divisor de tensión de batería, para su medición
RB7	28	Conecta regulador de tensión (5 volts)

Puerto C:

Línea	Pin	Descripción
RC0	11	Entrada el Timer1
RC1	12	Sin uso
RC2	13	Sin uso
RC3	14	Serial clock (SCL), para interfase I ² C
RC4	15	Serial data (SDA), para interfase I ² C
RC5	16	Sin uso

RC6	17	Sin uso
RC7	18	Señal de sonido hacia parlante, de frecuencia 2Khz.

2.7.PUERTO SERIE SINCRÓNICO

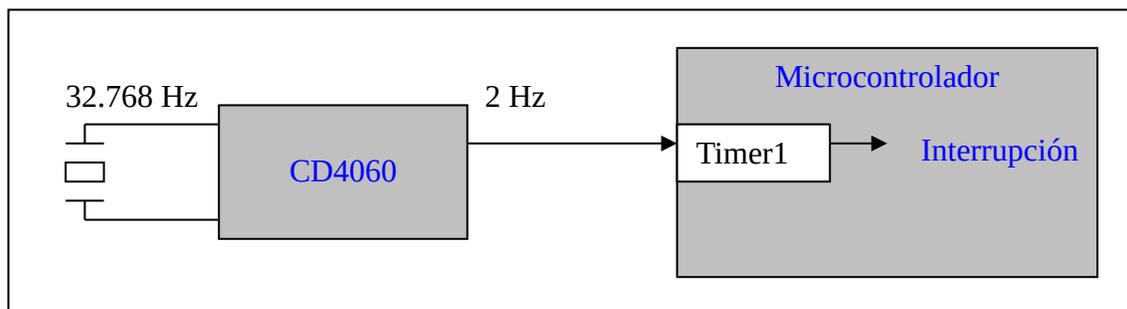
Este módulo es muy útil para realizar una comunicación entre el microcontrolador y otros periféricos, estos pueden ser: memorias serie, registros serie, displays, conversores A/D, etc. El módulo soporta dos protocolos, el denominado Serial Peripheral Interface (SPI) y Inter-Integrated Circuit (I²C).

Esta vía de comunicación es utilizada para transferir información desde y hacia la memoria de datos externa, y el protocolo utilizado es el I²C, ya que esta memoria EEPROM soporta dicho protocolo.

3.RELOJ DE TIEMPO REAL

Esta parte del sistema es la encargada de mantener el sincronismo con la fecha y hora real, para tomar las muestras en el instante programado por el usuario. Esto implica que esta etapa debe funcionar constantemente y poseer el menor consumo posible.

Dicha etapa está implementada por un circuito oscilador/divisor y por el Timer1 disponible en el microcontrolador. La siguiente figura indica su conexionado:



3.1.OSCILADOR/DIVISOR CD4060

El circuito integrado CD4060 es un contador binario de 14 etapas que además posee un oscilador interno, con la posibilidad de funcionamiento a cristal externo.

El cristal utilizado es de 32.768Hz, de bajo costo y muy utilizado por los relojes de pulsera, que provee buena estabilidad en frecuencia. El hecho de utilizar este valor de frecuencia se debe a que el CD4060 puede dividir la frecuencia de entrada en cantidades

fijas, desde 2^4 hasta 2^{14} veces, por lo tanto si dividimos 32768Hz por el valor máximo $2^{14}=16.384$ se obtiene una frecuencia de salida de 2Hz, o bien dos ciclos por segundo.

El CD4060 permite una tensión de alimentación de 3 a 15 volts. El consumo medido del circuito funcionando a esta frecuencia y alimentado con 4volts es de $11\mu\text{A}$.

3.2.USO DEL TIMER1

El contador utilizado propio del microcontrolador es el denominado Timer1 de 16 bits, el cual se incrementará por medio de la señal proveniente del CD4060, es decir, una vez cada medio segundo. Para que esto suceda, el Timer1 debe estar configurado en el modo asincrónico.

Este contador permite que se le ingrese un valor de cuenta inicial (desde 0 hasta 65535). Una vez activo se incrementa en uno al ritmo mencionado y al sobrepasar el valor 65535, vuelve a cero, generando además un pulso de interrupción que enciende el microcontrolador. De esta manera se pueden generar intervalos de tiempo conocidos con solo programar el Timer1 con un valor inicial correcto.

Por ejemplo, si se carga al Timer1 con el valor 0, se producirá una interrupción luego de 65536 cuentas, y como cada cuenta se produce cada medio segundo, habrán transcurrido 32768 segundos desde que se activó el timer, hasta la interrupción. Este intervalo es el máximo posible utilizando esta configuración, y equivale a 9 horas, 6 minutos y 8 segundos.

Para conseguir un intervalo de tiempo definido por el usuario, el sistema se basa en la siguiente ecuación: $N = 65536 - (2 \cdot T)$. Donde T es el intervalo de tiempo en segundos y N es el número que se cargará en el registro del Timer1.

4.ADQUISICIÓN DE DATOS

La toma de muestras se debe realizar en un instante de tiempo definido por el usuario. El equipo se sincroniza con la fecha y hora de la PC en el momento de la programación, y luego haciendo uso del timer1 visto en el capítulo anterior, determina el intervalo de tiempo hasta la primer toma de muestras.

Esto implica que el sistema no necesita llevar un registro actualizado del calendario, sino que solo registra la fecha inicial y genera intervalos de tiempo bien conocidos.

4.1.TIEMPO EN ESPERA

Como su nombre lo indica, en este modo el Data Logger espera por la primer toma de muestras.

Para entender mejor este punto se plantea el siguiente ejemplo: El usuario programa el equipo para comenzar con la toma de muestras a la hora 12:00:00 y en ese instante en su PC es la hora 11:43:20, luego la diferencia de tiempo son 1000 segundos. Inmediatamente se programa al timer1 para que cuente esa cantidad de tiempo, es decir, la interrupción sucederá a la hora 12:00:00.

4.2.INTERVALO ENTRE MUESTRAS

El usuario puede programar diferentes intervalos de muestreo para cada canal, para ello define un intervalo mínimo que se utiliza como base de tiempo para los diferentes canales, es decir, multiplicando esta fracción mínima una cantidad entera de veces se consiguen los intervalos buscados.

Por ejemplo si se busca un intervalo de muestreo de 1 minuto para el canal 1 y de 5 minutos para el canal 2, el intervalo mínimo será de 1 minuto (60 segundos) puesto que es múltiplo para ambos canales.

El intervalo mínimo define entonces cada cuanto tiempo el microcontrolador se enciende (sale de modo sleep) y procede con la toma de muestras. Siguiendo el ejemplo para el canal 1 se tomará un dato cada vez que se encienda y para el canal 2 cada 5 veces que esto suceda.

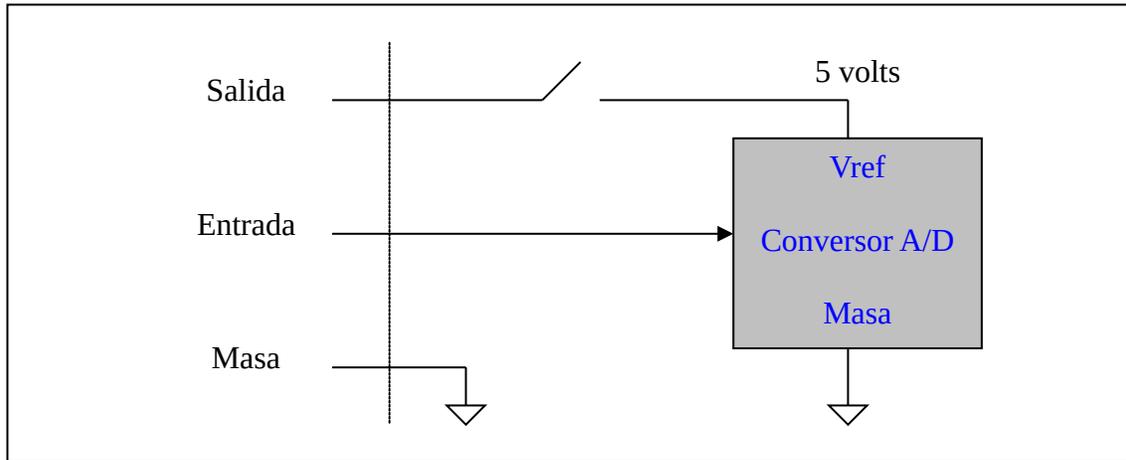
Para mantener el sincronismo inmediatamente después de la interrupción proveniente del tiempo en espera, se carga al timer1 con el valor correspondiente al intervalo mínimo. En la siguiente interrupción se vuelve a cargar al timer1 con el intervalo mínimo, hasta que se complete el tiempo de trabajo programado.

4.3.TOMA DE UN DATO

Cuando corresponda tomar un dato en cierto canal, el sistema procede de la siguiente manera:

- 1) Se cierra la llave de la figura, conectando 5 volts al pin “Salida”, lo que alimenta al sensor que se encuentre instalado.
- 2) Se espera durante 5 milisegundos, dando tiempo a que se establezcan las corrientes.
- 3) Se convierte el valor de tensión analógica en el pin “Entrada” en su correspondiente digital de 10 bits, que luego se almacena en memoria.

- 4) Vuelve a abrir la llave, desconectando los 5 volts hasta la próxima toma de muestras.

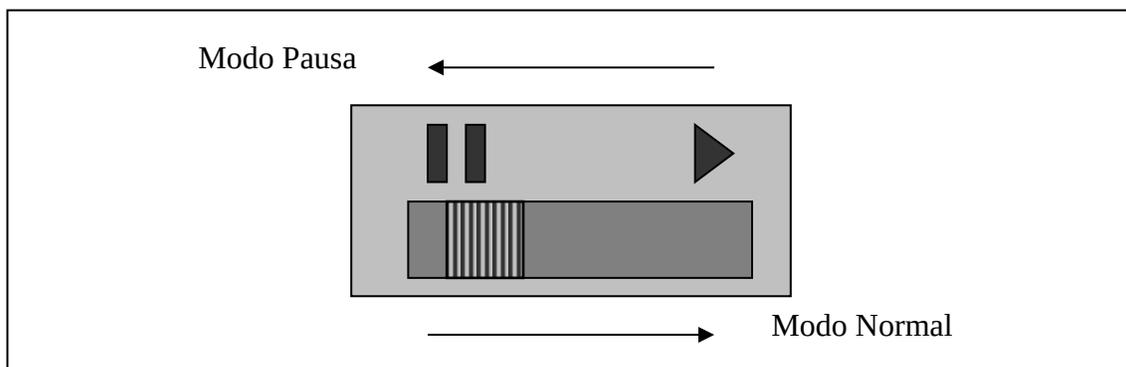


4.4 MODO EN PAUSA

Si el equipo se encuentra funcionando en su lugar de trabajo, el usuario puede verificar su estado, para esto desconecta los sensores y transporta el Data Logger hacia la PC para su lectura de datos.

Este modo existe para poder diferenciar esta instancia de su funcionamiento normal, puesto que si tomara una muestra sin sensor conectado, el dato adquirido sería erróneo.

Para poder seleccionar este modo el usuario debe modificar la posición de la tecla que se encuentra en el frente del equipo, como indica la siguiente figura.

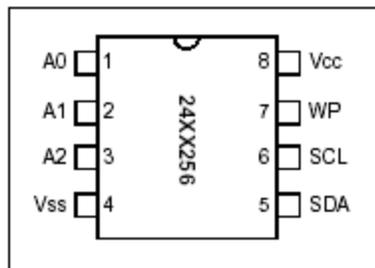


Además este modo se selecciona automáticamente en el momento de conectar el cable del Data Logger a la PC.

5. Memoria de Datos.

Siguiendo con la descripción del diagrama de bloques básico, a continuación se explica el funcionamiento de este dispositivo y su aplicación.

El circuito integrado se denomina 24LC256 de la firma Microchip, es una memoria del tipo EEPROM, con comunicación serie I²C y tecnología CMOS. La siguiente figura ilustra su disposición física.



El rango de tensión de alimentación va desde 1,8 a 5,5 volts, teniendo un bajo consumo. Las líneas A0, A1 y A2 configuran la dirección del dispositivo, lo que permite

conectar hasta 8 memorias en el mismo bus serie. La línea WP permite protección contra escritura.

Esta memoria tiene una capacidad de 32.768 bytes, que son utilizados para almacenar los datos digitales provenientes de la toma de muestras. Cada uno de estos datos es de 10 bits y se agrupa en 2 bytes de la siguiente manera.

Byte parte alta								Byte parte baja							
BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT	BIT
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
Pausa	Fin de Datos	Canal 4	Canal 3	Canal 2	Canal 1	10 bits del conversor A/D Valor digital de la Muestra									

Los 6 bits mas altos se utilizan para identificar el canal al que corresponde el dato, si el equipo está en modo pausa o bien para marcar el último dato adquirido.

La primer zona de memoria EEPROM se destina al almacenamiento de los parámetros de la programación del Data Logger.

En conclusión se pueden guardar 16.300 datos provenientes de los sensores, siendo este dato de utilidad para poder calcular la autonomía del equipo.

Por ejemplo si se programa para que los cuatro canales tomen muestras cada 1 minuto, la capacidad de memoria se repartirá en cuatro partes iguales de 4.075 datos, lo que implica una autonomía de 4.075 minutos (2 días 19 hs. 55 min.) hasta completar la memoria.

Otro ejemplo: canal 1 toma muestras cada 1 minuto y canal 2 cada 3 minutos, por lo tanto un 75% de la memoria (12.225 datos) utilizará el canal 1 y un 25% (4.075 datos) el canal 2; y la memoria estará completa al cabo de 12.225 minutos (8 días 11 hs. 45 min.).

6.TIPOS DE SENSORES

Los sensores son la parte crítica del sistema, ya que su exactitud y precisión determinarán la utilidad de los resultados obtenidos. Este sistema se planteó desde un principio para poder soportar diferentes tipos de sensores, siempre que se adapten a las especificaciones de los canales de entrada y a la forma en como se toman los datos (ver capítulo 4.3). A continuación se detallan dichas especificaciones y se enumeran ejemplos sobre como implementar ciertos tipos de sensores de temperatura y humedad.

Como se explicó anteriormente el equipo trabaja con una tensión de 5 volts, la cual está disponible para el sensor en el momento de la toma de muestras y esa misma tensión se toma como referencia para el conversor A/D. Un dato importante del conversor es su impedancia de entrada, la cual es de $100K\Omega$ en este caso.

Por lo tanto para poder adaptar un sensor al Data Logger se debe tener en cuenta lo siguiente:

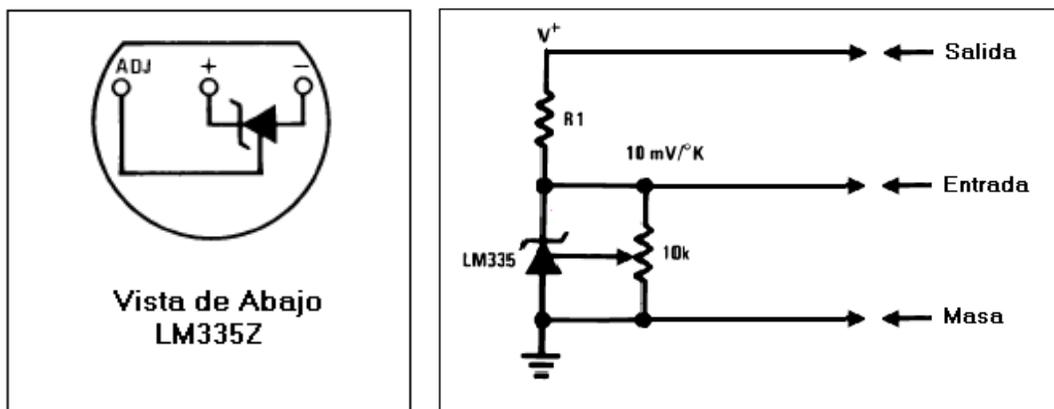
- El sensor debe poder alimentarse con 5 volts y tener un consumo menor a 10mA.
- La variable medida debe traducirse en un valor de tensión analógico entre 0 y 5 volts.
- La impedancia de entrada del conversor no debe afectar en gran medida la exactitud del sensor.
- La resolución del conversor A/D (de 10 bits) es de $4,89\text{mV}=5\text{V} / 1023$.

6.1. SENSOR DE TEMPERATURA LM335

El siguiente ejemplo muestra como se implementó el sensor de temperatura LM335 de una manera sencilla, utilizando los datos suministrados por el fabricante National Semiconductor.

Este circuito integrado funciona como un diodo zener, cuya tensión inversa es directamente proporcional a su temperatura absoluta, con una relación de $10\text{mV}/^\circ\text{K}$. Posee una impedancia dinámica menor a 1Ω y opera con una corriente de polarización de $400\mu\text{A}$ hasta 5mA . Permite medir en un rango de -40°C hasta 100°C , con un error de aproximadamente 1°C .

La figura muestra la disposición de pines y la forma de conectarlo al Data Logger.



De esta manera se genera en la pata “Entrada” del Data Logger una tensión que representa la temperatura en el sensor. Por ejemplo una tensión de 2,9815volts equivale a una temperatura de $298,15^\circ\text{K} = 25^\circ\text{C}$ (recordar que $T_{\text{K}}=273.15 + T_{\text{C}}$).

La resistencia R1 es de $2,2\text{K}\Omega$, necesaria para polarizar al sensor con 1mA aproximadamente y el preset de $10\text{K}\Omega$ permite su calibración. Esta configuración permite

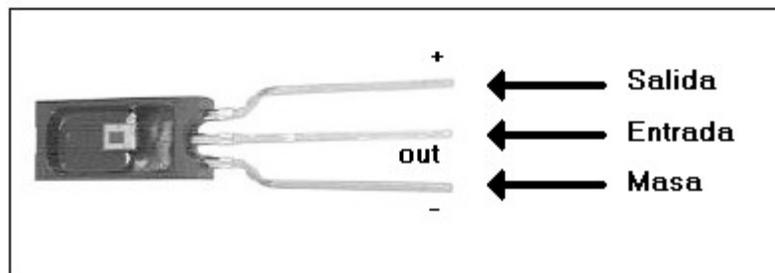
una resolución de 0.5°C gracias a las características del conversor A/D (ver capítulo 2.5).

Por último la ecuación que representa la temperatura T (en °C) en función de la tensión de Entrada (V_e) es la siguiente:

$$(1) \quad T = -273,15 + 100 \cdot V_e$$

6.2. SENSOR DE HUMEDAD HIH3610

Este sensor es fabricado por la firma Honeywell, pensado para este tipo de aplicaciones. En la figura se observa la forma directa de su conexión al Data Logger.



De forma similar al LM335, el HIH3610 produce una tensión proporcional con la humedad relativa de forma lineal, con una exactitud de un 2%. Alimentado con 5 volts posee un consumo de 200µA. Su tensión de salida responde a la siguiente ecuación:

$$V_e = 0,031.RH(\%) + 0.8$$

O bien:

$$(2) \quad RH(\%) = 322,58 \cdot V_e - 258,06$$

Donde V_e es la tensión en la pata “Entrada” y RH es la humedad relativa. El fabricante provee también una ecuación para la compensación por temperatura:

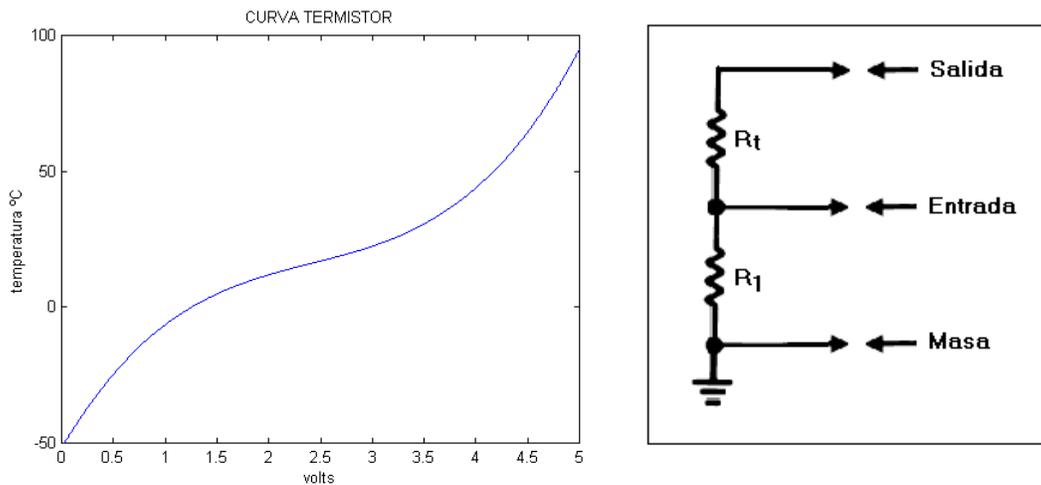
$$RH_{REAL} = RH_{SENSOR} / (1,0546 + 0,00216 \cdot T(^{\circ}C))$$

6.3. SENSORES ALINEALES

En los puntos anteriores se han enumerado dos aplicaciones de sensores lineales, pero no siempre esto es así, pues existen numerosos sensores con características alineales (pe termistores).

Para poder adaptar estos sensores al sistema se deberá encontrar la ecuación que represente a la variable en función de la tensión medida (0 a 5 volts). Esta ecuación es a veces ofrecida por el fabricante directamente o en forma de curvas, de otra forma puede obtenerse a partir de ensayos o bien a través de simulaciones en programas de cálculos matemáticos (pe Matlab).

La figura muestra la curva para un termistor (R_t) y el circuito asociado.



Una transferencia del tipo alineal puede aproximarse por la siguiente ecuación en series de potencias:

$$X=C_0+C_1.V+C_2.V^2+C_3.V^3$$

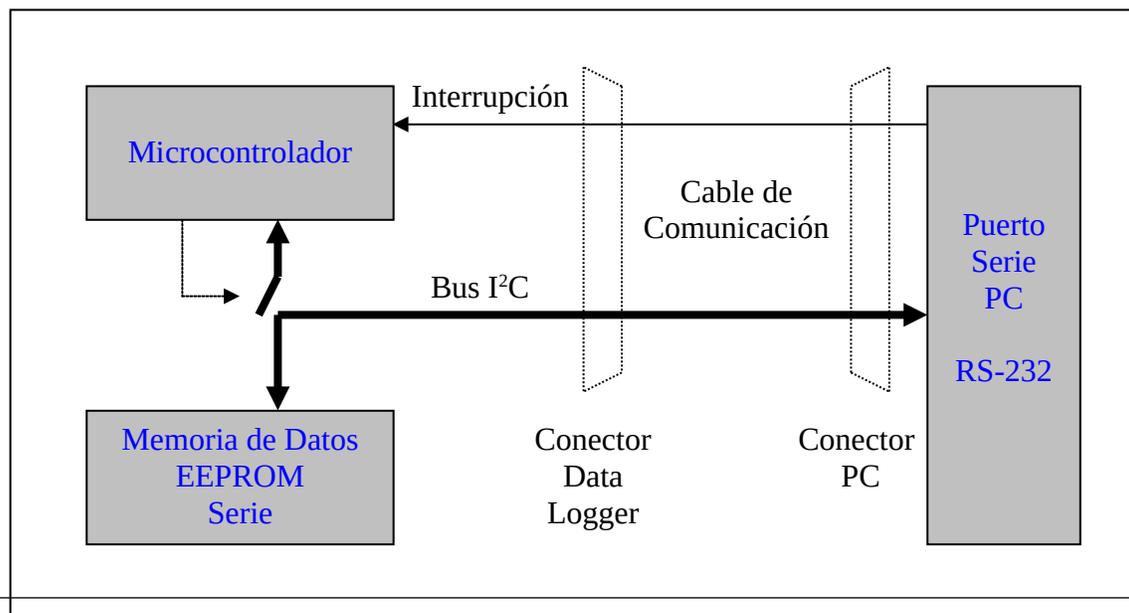
Donde V es la tensión entre 0 y 5 volts y X es la variable a medir. Si aplicamos esta fórmula en la ecuación (1) se ve que $C_0 = -273,15$, $C_1 = 100$, $C_2 = 0$ y $C_3 = 0$. De igual forma en la ecuación (2) $C_0 = 322,58$, $C_1 = -258,06$, $C_2 = 0$ y $C_3 = 0$.

La curva de la figura anterior responde a los valores de $C_0 = -51,7126$, $C_1 = 64,8414$, $C_2 = -22,8073$ y $C_3 = 3,1421$, calculados utilizando Matlab.

7. INTERFASE DE COMUNICACIÓN CON LA PC

Esta interfase es el medio por el cual el Data Logger se comunica con la PC y finalmente con el usuario, por lo tanto debe ser confiable y segura.

En este caso la PC posee una comunicación directa con la memoria de datos serie del equipo, como indica la figura.



Este modo de funcionamiento permite una rápida lectura de los datos almacenados, sin intervención del microcontrolador. Para lograr esto, él puede desconectarse del bus con el fin de no producir colisiones. Siempre que el sistema se encuentre en modo Pausa se libera el bus I²C, es por ello que se selecciona este modo automáticamente al conectar el cable de comunicación.

Al momento de enviar una orden al Data Logger, se escribe un comando en la memoria y luego se envía un pulso por la línea de interrupción que le indica al microcontrolador que debe leer y ejecutar la instrucción.

Mediante este comando el sistema puede realizar lo siguiente:

- Comenzar con la ejecución de las tareas programadas por el usuario, sincronizándose con el reloj de la PC.
- Tomar una muestra de los sensores en forma instantánea y de la tensión de batería.

A continuación se describe brevemente el funcionamiento de la interfase I²C, y su implementación a partir del protocolo RS232.

7.1.I²C (INTER-INTEGRATED CIRCUIT)

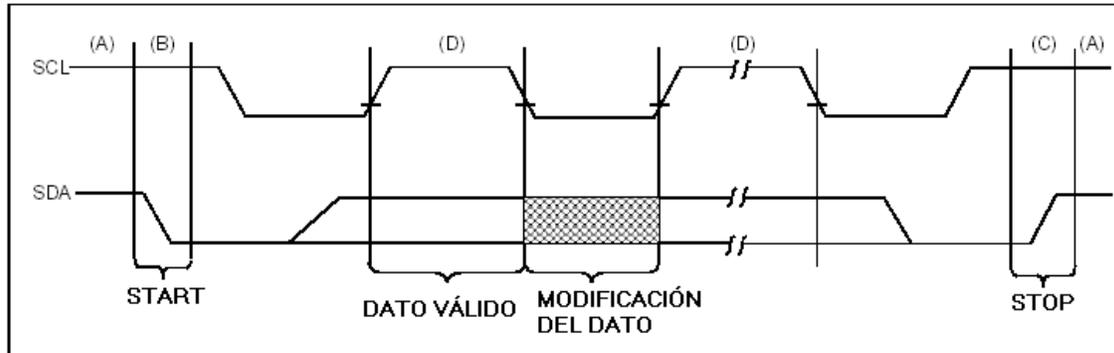
Esta interfase se logra con solo dos líneas de conexión, una para los datos (SDA) y otra para la sincronización (SCL), de aquí su denominación de interfase serie sincrónica. Siempre conecta un equipo “maestro” con uno o varios “esclavos”, donde cada uno posee una dirección de identificación.

Ambas líneas de conexión son del tipo open-collector, por lo tanto requieren resistencias de pull-up hacia Vcc (10KΩ típica).

La línea SDA (Serial Data) es bidireccional usada para la transferencia de direcciones y datos. Durante una transmisión de datos normal, SDA puede cambiar su valor solo cuando SCL está en nivel bajo. Los cambios con SCL en alto están reservados para las condiciones de START y STOP.

La línea SCL (Serial Clock) es controlada por el “maestro” y es utilizada para la sincronización de la transmisión desde y hacia un dispositivo “esclavo”.

La figura describe una secuencia de transferencia de datos en el bus serie.



La zona (A) representa el estado normal del bus, es decir, sin actividad.

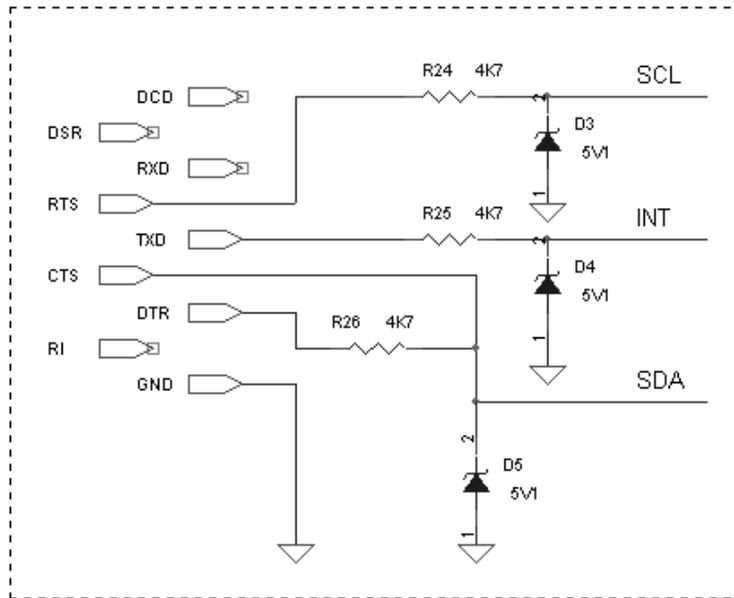
La zona (B) indica una condición de START, que indica el comienzo de una transmisión. Finalmente debe indicarse el fin de la secuencia con una condición de STOP (C).

El estado de la línea SDA representa un dato válido cuando, luego de un START, SDA se mantiene estable durante el período en nivel alto de la señal SCL. Es por ello que SDA se debe cambiar durante el nivel bajo de SCL. Se transmite un bit por cada ciclo de clock.

La cantidad de bytes transmitidos es controlada por el dispositivo “maestro”.

7.2.I²C MEDIANTE RS-232

Como se sabe la PC no posee una interfase serie con protocolo I²C. La figura muestra como implementar esta interfase utilizando el protocolo RS232.



Solo tres de las nueve líneas del conector DB9 de la PC son salidas, ellas son: RTS, TXD y DTR. Las otras cinco son entradas y el pin 5 corresponde a masa. La norma especifica para las salidas en nivel alto una tensión de 5 a 15 volts y en nivel bajo de -15 a -5 volts. Para las entradas se reconoce un nivel alto entre 3 y 25 volts, un nivel bajo entre -25 y -3 volts. Es por esto que agregan diodos zener para ajustar estos niveles de salida en valores de 0 y 5 volts, manejando así las mismas tensiones que la memoria de datos.

En este caso la PC funciona como dispositivo maestro, por lo tanto genera la señal de clock en SCL, la cual se efectúa alternando el estado de esta línea.

La salida DTR se emplea para enviar datos hacia la memoria (escritura), y por la línea CTS de entrada se leen los datos de la memoria.

Hay que notar que no se utilizan las líneas RXD y TXD para la comunicación como lo haría RS232, es decir, el protocolo se emula utilizando las líneas de control, las cuales son manipuladas por el software del Data Logger. Solamente se envía un pulso por la línea TXD, que produce una interrupción en el microcontrolador.

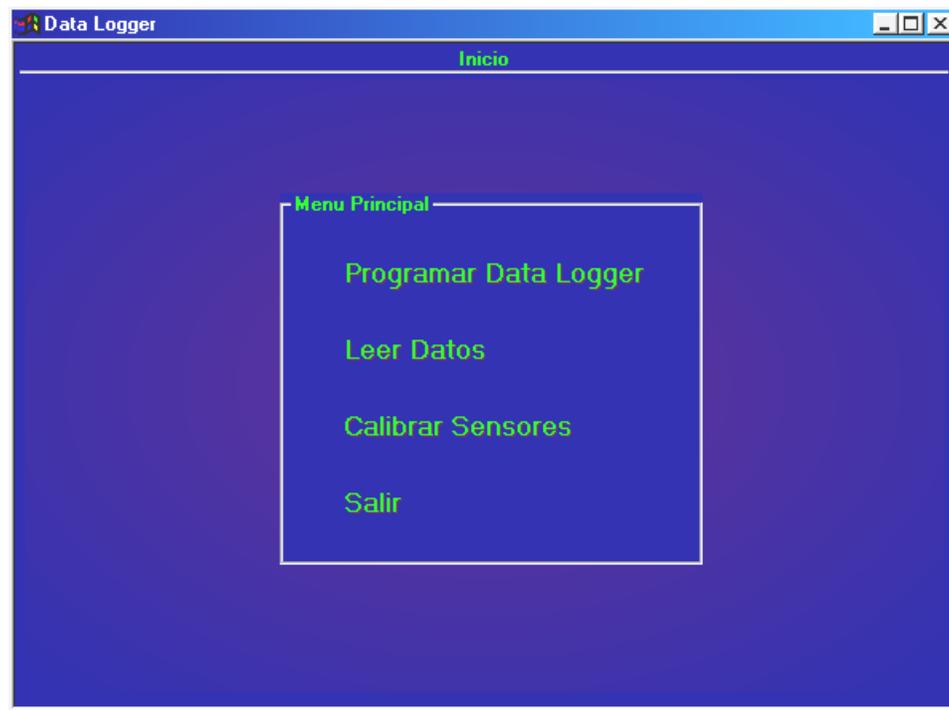
Como alternativa podría efectuarse la comunicación utilizando el puerto paralelo de la PC, obteniendo un resultado similar con solo modificar el software.

8.SOFTWARE DE APLICACIÓN

Este software permite al usuario controlar todas las posibilidades que ofrece el sistema. Fue desarrollado bajo *Visual Basic 6.0*, este entorno es una excelente herramienta de programación que permite crear aplicaciones propias (programas) para ser ejecutadas

bajo *Windows*. Visual Basic permite crear ventanas, botones, menús y cualquier otro elemento de *Windows* de una forma fácil e intuitiva.

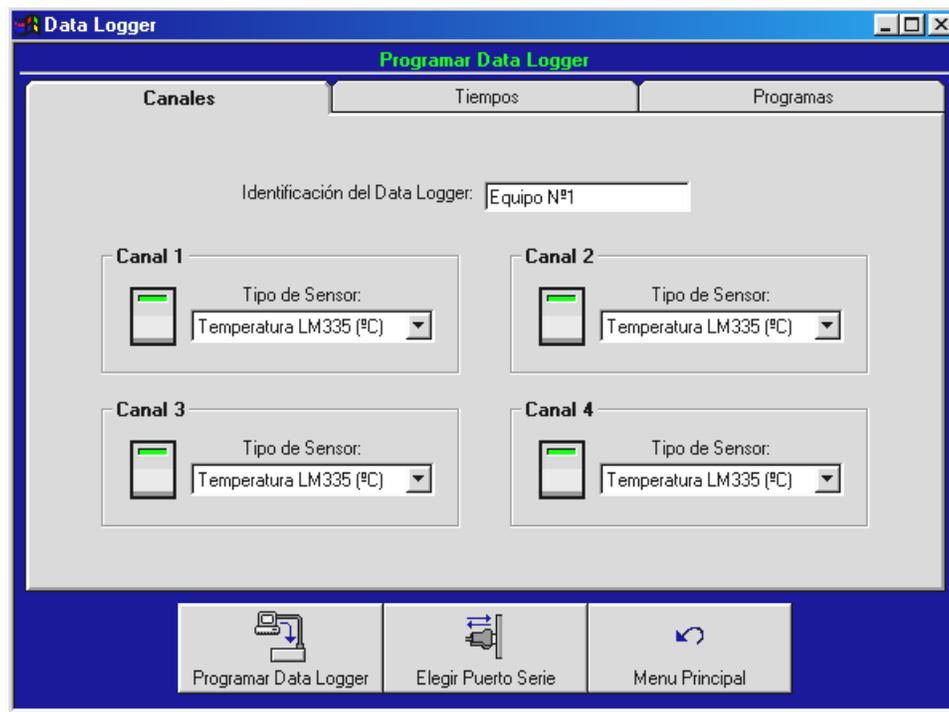
La siguiente ventana se presenta al ejecutar el programa. Para acceder a los diferentes menús debe colocar el mouse sobre el texto, dentro de la casilla “Menú Principal”.



Todos los menús poseen la misma estructura visual, en la parte superior de la ventana se indica el nombre del menú. En la parte central un formato tipo fichero permite seleccionar las diferentes opciones y en la parte inferior se ubican botones de comando que permiten realizar eventos.

8.1.PROGRAMACIÓN DEL DATA LOGGER

Luego de presionar el mouse sobre “Programar Data Logger” en el menú principal aparece la siguiente ventana.

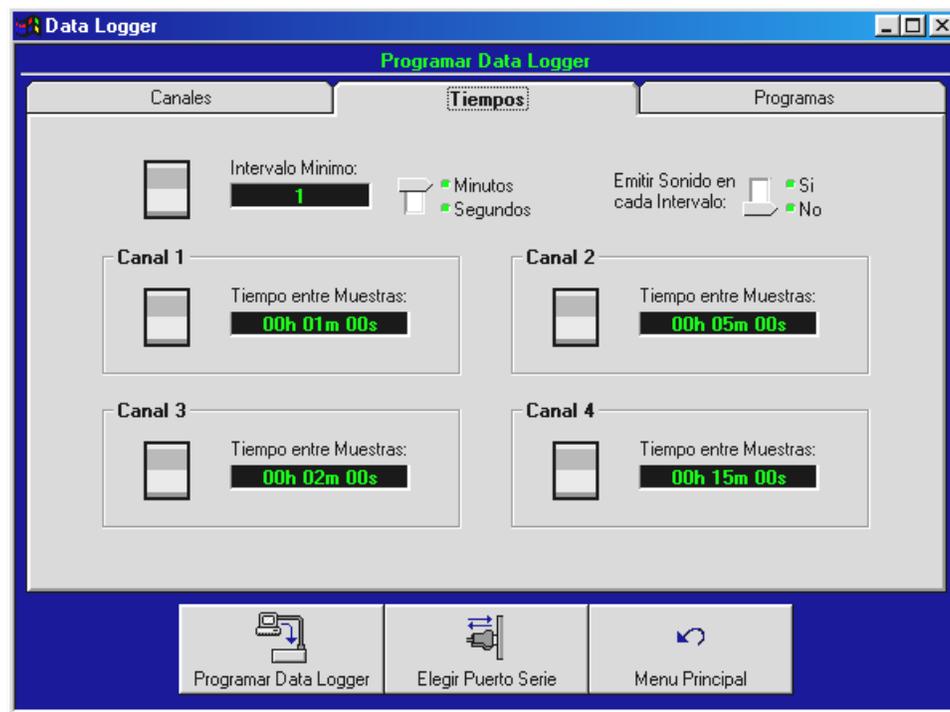


Para realizar una programación se debe pasar por las tres fichas, la primer ficha “Canales” permite:

- ingresar un nombre de identificación del equipo,
- indicar que canal será utilizado, activando o desactivando cada uno de los switches correspondientes dentro de cada recuadro,
- seleccionar el tipo de sensor para el canal activo, desde una lista.

En la figura se ha identificado al Data Logger con el nombre “Equipo N°1” y se utilizarán cuatro sensores iguales de temperatura.

En la ficha “tiempos” se indican los intervalos entre muestras para cada uno de los canales activos, utilizando los switches para modificar dichos valores.



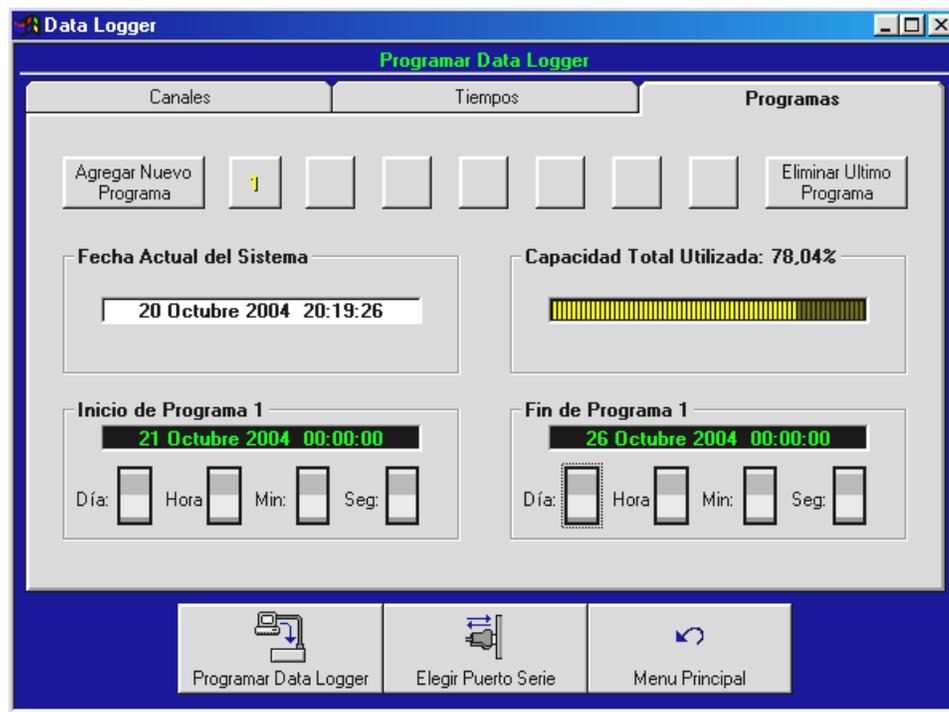
Primero se debe indicar el *intervalo mínimo*, el cual es un factor común a los cuatro canales. Con el selector en *Segundos* se puede definir un tiempo desde 1 a 120 segundos, y con el selector en *Minutos* un tiempo de 1 a 60 minutos.

Presionando los switches de cada recuadro se modifica el tiempo entre muestras a partir del intervalo mínimo, permitiendo un valor máximo de 6 horas.

El segundo selector permite la emisión de un sonido “beep” por cada intervalo mínimo, esto es útil en ciertos casos para verificar el sincronismo con la hora de la PC. Pero hay que tener en cuenta el incremento en el consumo que esto provoca.

En la figura anterior se ha seleccionado un intervalo mínimo de 1 minuto, para el canal 1 un minuto de intervalo entre muestras, para el canal 2, 5 minutos (5 veces el intervalo mínimo), el canal 3 con 2 minutos y el canal 4 con 15 minutos.

La tercer ficha “Programas” muestra lo siguiente.



En esta instancia se determina el tiempo de operación del Data Logger. Esto se efectúa indicando una fecha de inicio y una de culminación de tareas. Además permite hasta siete programas de funcionamiento.

Inicialmente se muestra el primer programa, indicando la fecha del sistema. A partir de esta se puede determinar el inicio del primer programa. Al modificar la fecha de fin de programa se indicará la capacidad de memoria utilizada, según la programación de tiempos realizada con anterioridad.

Presionando el botón “Agregar Nuevo Programa” se indica el nuevo número del programa dentro de las siete casillas superiores, permitiendo al usuario volver a determinar una fecha de inicio y de fin, a partir de la fecha de finalización del programa anterior.

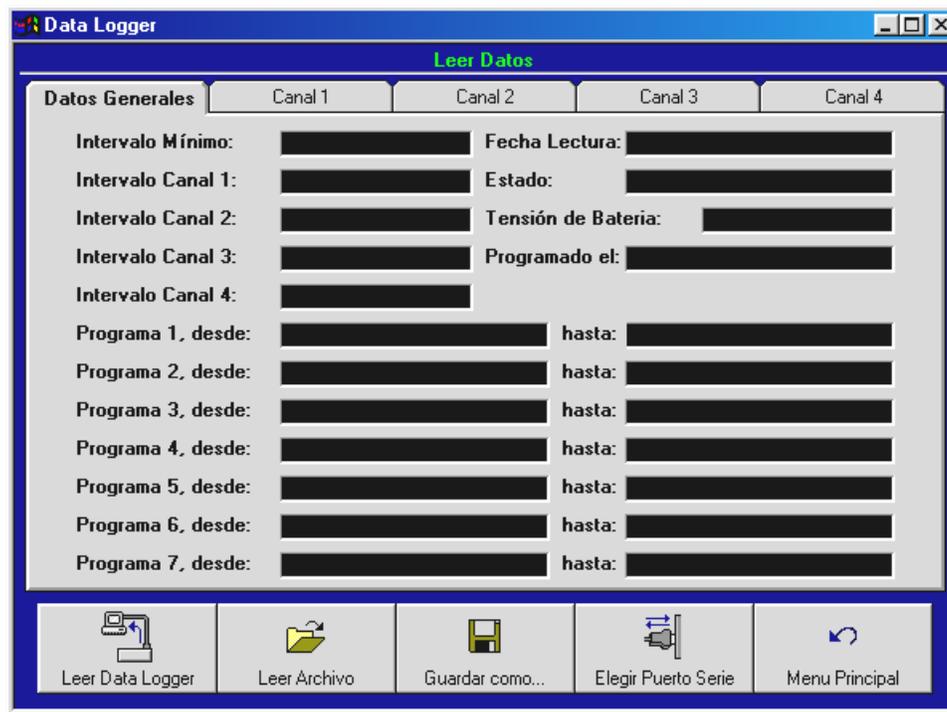
Realizada la configuración, siguiendo los pasos anteriores, se debe presionar el botón “Programar Data Logger” que finalmente dará la orden al equipo que esté conectado a la PC. Esta aplicación dará aviso por cualquier fallo en la comunicación, permitiendo al usuario indicar el número del puerto serie donde ha conectado al Data Logger (COM1 o COM2).

Si la programación fue exitosa el Data Logger emitirá un sonido de confirmación, luego el software preguntará al usuario si desea guardar los datos de la programación realizada en un archivo de texto.

8.2. LECTURA DE DATOS

Esta parte del programa está destinada a la lectura y presentación de los datos al usuario. Los datos pueden ser obtenidos desde el Data Logger o bien desde un archivo.

Inicialmente se presenta la siguiente ventana.



Presionando el botón “Leer Data Logger”, se verifica la conexión al equipo y luego se leen todos los datos almacenados en el mismo, así como los datos de su programación. Una barra en la parte inferior indicará el progreso en la lectura.

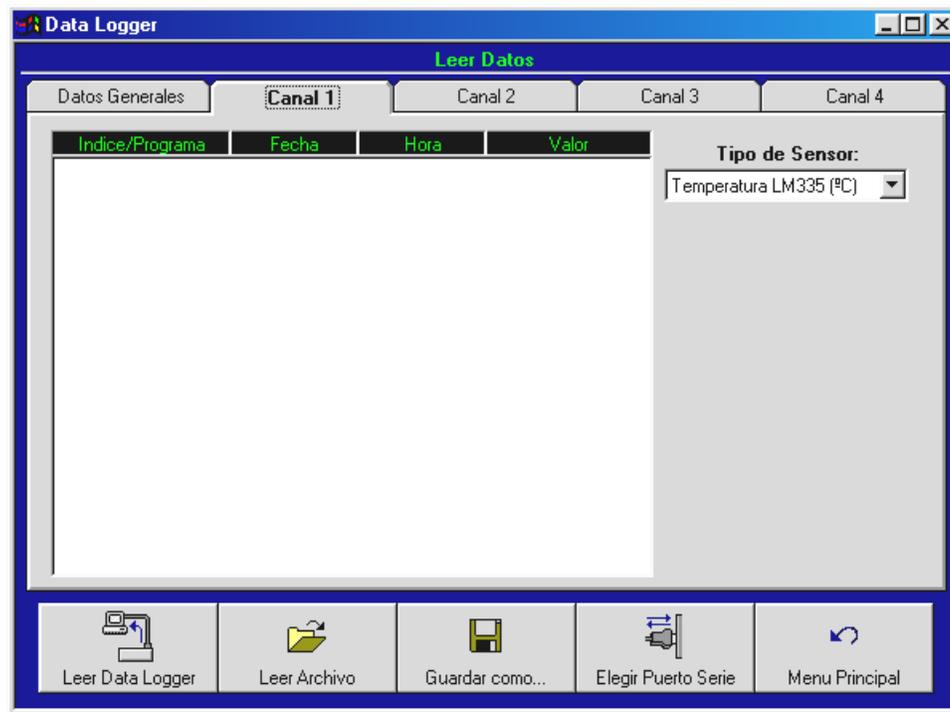
El botón “Leer Archivo” permite cargar datos que fueron guardados con anterioridad, en un formato de archivo propio de este sistema (DLG).

Finalizada la lectura de datos se completarán todos los cuadros de texto que se ven en la figura anterior, con los datos de programación.

Además si el Data Logger se encuentra ejecutando un programa se indicará en el ítem *Estado*, caso contrario se mostrará el texto “sin programa”. La indicación del nivel de

batería corresponde al instante de su lectura.

Seleccionando las fichas restantes, por ejemplo “Canal 1”, aparece una lista con toda la información recopilada correspondiente al canal.

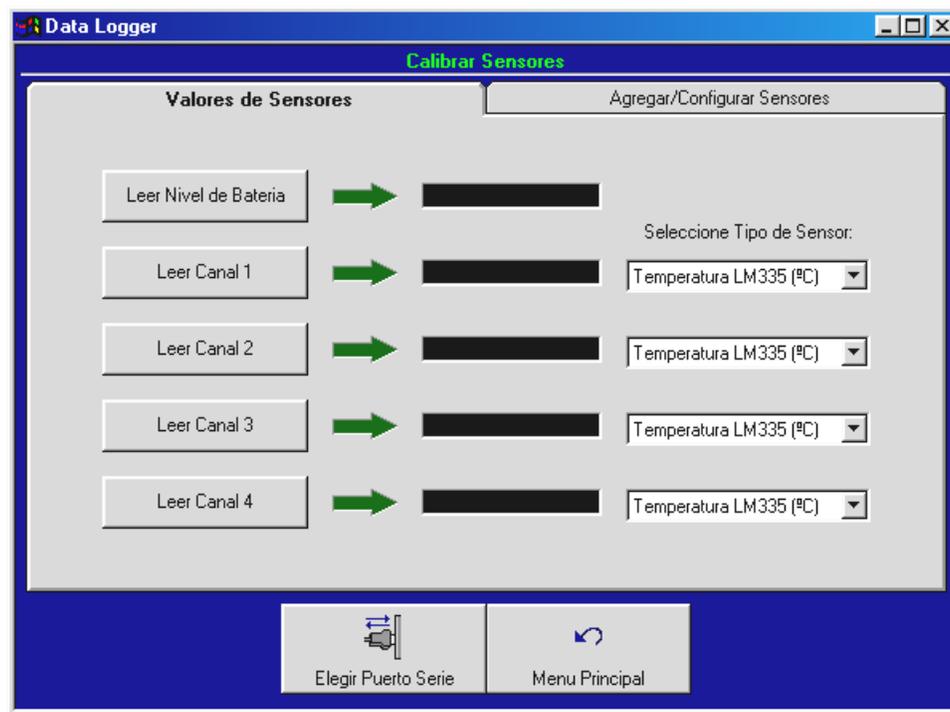


Al costado de la lista se indica el tipo de sensor, con la posibilidad de modificarlo. Si se modifica se volverá a calcular la lista de datos.

Finalmente toda esta información podrá ser guardada en un archivo DLG o en un formato CSV compatible con la planilla de cálculos Excel, mediante el botón “Guardar como...”.

8.3.CALIBRACIÓN Y CONFIGURACIÓN DE SENSORES

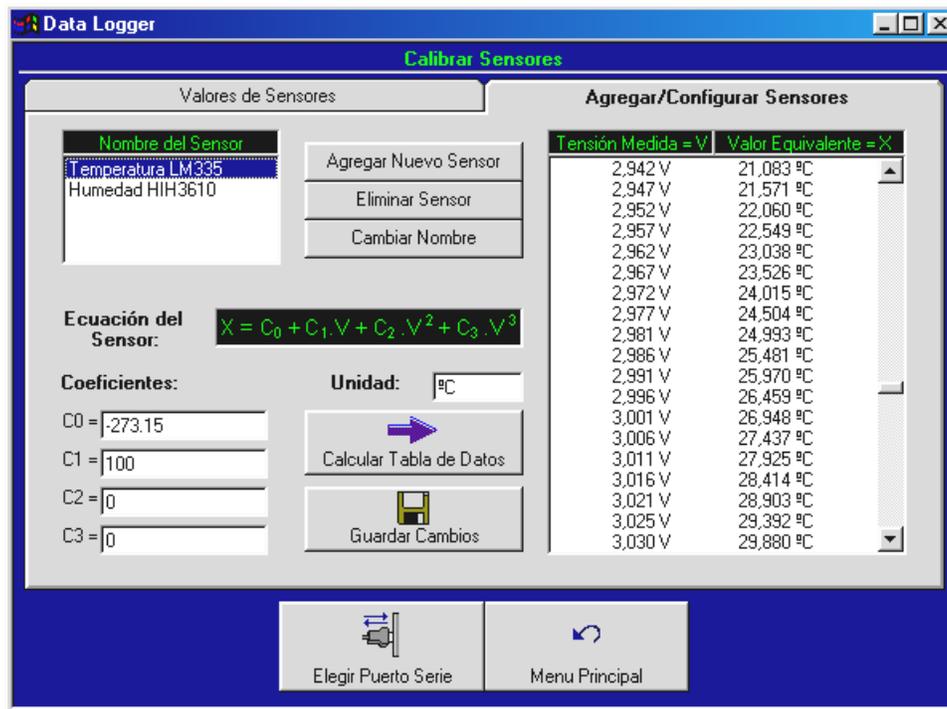
La pantalla que se presenta es la siguiente.



Esta es una forma inmediata de conocer el estado de un sensor que esté conectado al equipo. Solo es necesario presionar el botón correspondiente, a la izquierda de la pantalla, lo que genera una comunicación con el equipo y presentación del dato obtenido.

Cada flecha se encenderá mientras se realiza la comunicación. A la derecha se selecciona el tipo de sensor conectado y según éste se calcula el dato.

Si se desea agregar un nuevo tipo o modificar las características de algún sensor de la lista se debe pasar a la siguiente ficha.



Como se analizó en el capítulo 6, el Data Logger lee un valor de tensión entre 0 y 5 volts y lo convierte a un valor digital de 10 bits. Luego con este valor de tensión el software deberá calcular el dato correspondiente.

Para esto se le permite al usuario crear una tabla de correspondencias, de acuerdo a la *ecuación del sensor*. Es decir puede ingresar los coeficientes de la ecuación presentada y un texto que indicará la unidad de la variable medida.

Luego de presionar el botón “Calcular Tabla de Datos”, se harán efectivos los cambios mostrando la tabla resultante.

Mediante el botón “Guardar Cambios” se almacenan las modificaciones en un archivo de nombre Sensores.dat, con el fin de mantener la configuración cuando se vuelva a cargar el programa.

Si se desea actualizar el programa con una lista de sensores modificada en otra PC, se deberá copiar este archivo en el mismo directorio donde se encuentre la aplicación DataLogger.exe.

9. OPERACIÓN Y MANTENIMIENTO

En el capítulo anterior se ha visto todo lo referente al software de aplicación, a continuación se describen aspectos sobre el uso y mantenimiento del equipo, así como sus especificaciones técnicas.

9.1. ASPECTO DEL DATA LOGGER

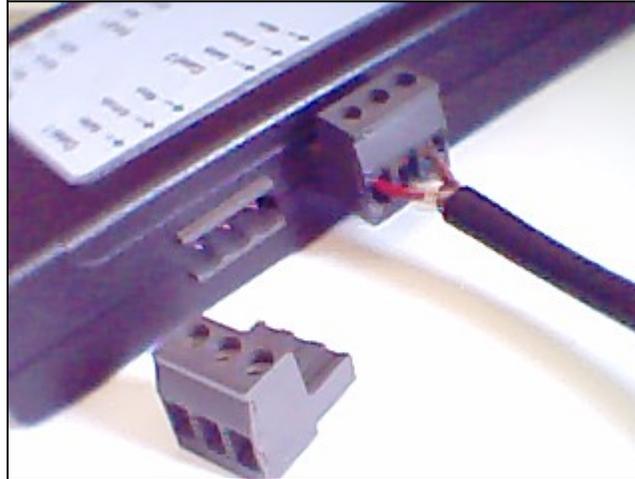
En la figura se muestran fotos del equipo, de su frente y su parte trasera. En la parte superior se encuentran la ficha de conexión hacia la PC y el selector de modo. En ambos costados se distinguen las conexiones para los sensores, teniendo en el frente una indicación referente a cada línea.

En la parte inferior, se encuentra el alojamiento para la batería, siendo accesible por el usuario.



9.2. CONEXIÓN DE LOS SENSORES

La figura muestra los bornes que permiten el conexionado de los sensores. Esta pieza se conecta en una sola posición en cualquier canal del equipo, luego los cables provenientes del sensor se atornillan a la misma.



Se emplea esta técnica de conexionado, pensando en que el equipo pueda funcionar dentro de un *gabinete estanco*, es decir, que proteja al Data Logger de factores climáticos si es que debe trabajar en intemperie. De esta manera, solo el cable debe atravesar el gabinete a través de algún tipo de prensa-cable, para mantener la estanqueidad.

Dentro del gabinete, el cable se atornilla a los bornes y luego estos bornes se conectan al equipo. Si el gabinete se fija en una posición, junto con los sensores, esto permite conectar o desconectar con facilidad el Data Logger.

9.3. CONEXIÓN A LA PC

Como primer paso al comenzar a utilizar el sistema, el usuario deberá conectar el equipo a la PC, utilizando el cable provisto a tal efecto.

Importante: El cable de conexión *NO* es del tipo pin a pin, es decir, fue diseñado únicamente para esta aplicación.

El extremo del cable correspondiente a la ficha macho DB9, se conecta al Data Logger como indica la figura.



El otro extremo (ficha DB9 hembra) se conecta al puerto serie de la PC (COM1 o COM2).

El usuario puede verificar la conexión desde la aplicación, presionando el botón “Elegir Puerto Serie” y luego el botón “Probar Conexión”, donde además permite cambiar el número de puerto de ser necesario.

9.4.RECOMENDACIONES PARA SU CORRECTO USO

- Antes de conectar el equipo a la PC, seleccionar el modo Pausa desde la tecla situada en la parte superior del Data Logger.
- Si el equipo se encuentra funcionando, antes de desconectar los sensores seleccionar el modo Pausa.
- Verificar el nivel de tensión de batería antes de poner en funcionamiento el equipo. Para esto, seleccionar la opción Calibrar Sensores desde el menú principal y luego presionar el botón “Leer Nivel de Batería”.

Si el nivel indicado es menor a 7,5 volts, debe reemplazar la batería.

9.5.ESPECIFICACIONES TÉCNICAS.Alimentación

Batería Utilizada: 9volts alcalina (IEC-6LR61).

Consumo en modo Sleep: 10 μ A.

Autonomía: 1 año, a 1 minuto de intervalo entre muestras.

Canales de Entrada

Cantidad: 4 (cuatro).

Rango de tensión de entrada: 0 a 5volts.

Impedancia de Entrada: 100K Ω .

Resolución del conversor A/D: 10 bits.

Tensión de Salida: 5 volts, durante 5 mseg. en cada muestreo.

Corriente Máxima de Salida: 10 mA.

Adquisición

Tiempo entre Muestras: Programable en cada Canal, desde 1 segundo hasta 6 horas.

Memoria de Datos: EEPROM serie.

Capacidad de Memoria: 16300 muestras.

Conexión a PC

Interfase con PC: Puerto Serie, RS-232.

Protocolo de Comunicación: Inter-Integrated Circuit (I²C)

Aspecto

Dimensiones (con bornes): 25mm (alto) x 100mm (ancho) x 145mm (largo).

Peso (con batería): 160gr.

RESULTADOS Y CONCLUSIONES.

Como resultado de este proyecto se obtuvo un sistema completo para la adquisición de datos conocido como Data Logger, el cual está listo para ser implementado en una aplicación que determine el usuario. Este sistema tiene características que le permiten ser competitivo en el mercado, entre ellas se destacan su bajo costo, tamaño y peso reducido, capacidad de ser portátil, cuatro canales de entrada, fácil programación y uso, etc. Teniendo como principal ventaja la característica de ser un sistema en donde su aplicación finalmente la define el usuario y no la impone el sistema.

Todos los componentes del equipo fueron seleccionados en base a su disponibilidad en el mercado local, lo que permite una fácil reproducción. Como componente central se utilizó el PIC16F873 de Microchip que gracias a su capacidad de ser un dispositivo programable, le permitió al sistema ser autónomo.

Como conclusión se consiguió un sistema versátil y eficiente con posibilidades de ampliar su capacidad y funcionalidad en diseños futuros. Entre estas se podrían nombrar el agregado de canales digitales de entrada/salida, toma de decisiones en base a los datos analógicos recopilados, control desde un sitio remoto, etc. Pero siempre teniendo como base el sistema mostrado en este informe.

BIBLIOGRAFÍA.

- Apuntes de la Cátedra “Adquisición Digital de Señales”, especialmente lo relacionado con Data Loggers e Interfase RS232. Autor: Raúl R. Rivera.
- PICmicro™ Mid-Range MCU Family Reference Manual, obtenido vía Internet.

<http://www.microchip.com/stellent/idcplg?>

http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1335&dDocName=en010236

- Hojas de Datos del microcontrolador PIC16F873, obtenido vía Internet.
- Hojas de Datos de la memoria EEPROM serie, obtenido vía Internet.
- Hojas de Datos del oscilador/divisor CD4060, obtenido vía Internet.
- Información sobre características de Data Loggers existentes, obtenida vía Internet en las siguientes páginas:

<http://www.veriteq.com/temperature/sp-1000-2x.htm>

<http://www.evidencia.biz/espanol/what-is/index.htm>

- Notas de Aplicación (Application Notes) enumeradas a continuación, incluidas en el CDROM “Microchip Technical Library, first edition 2000”, cedido por el Laboratorio de Procesos y Mediciones de Señales.

Código: AN567, Título “Interfacing 24LCXXB Serial EEPROMs to the PIC16C54”

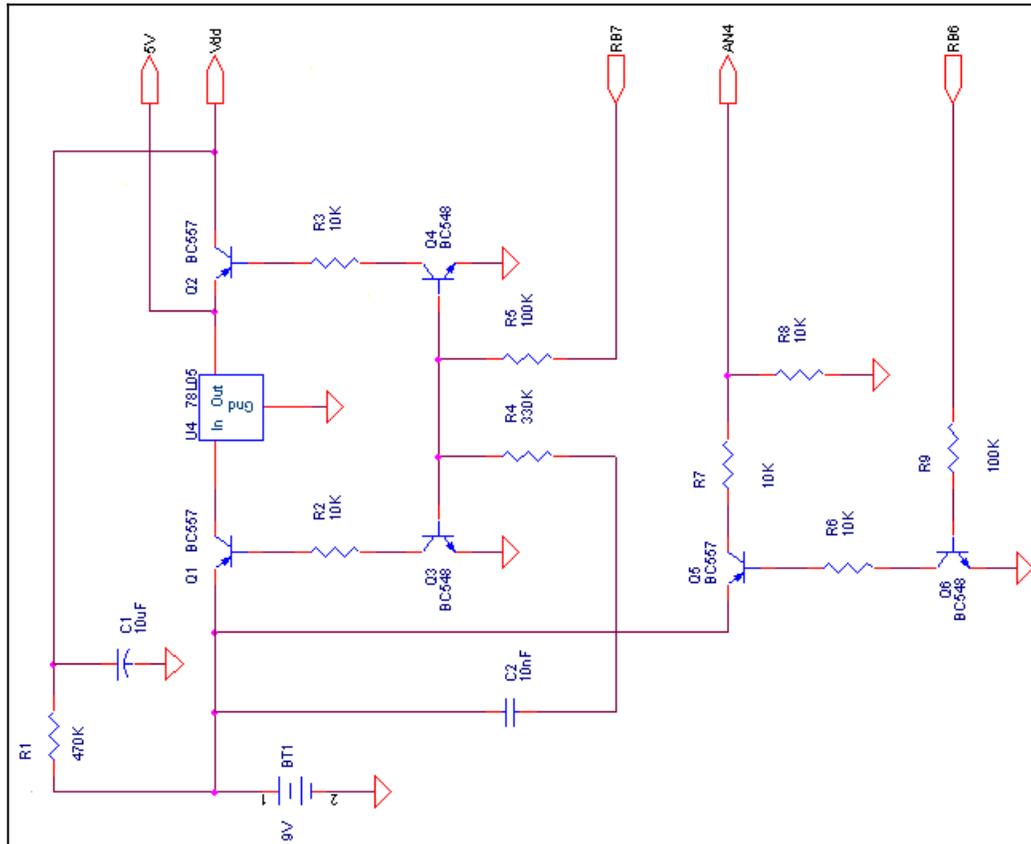
Código: AN580, Título “Using Timer 1 in Asynchronous Clock Mode”

Código: AN582, Título “Low-Power Real Time Clock”

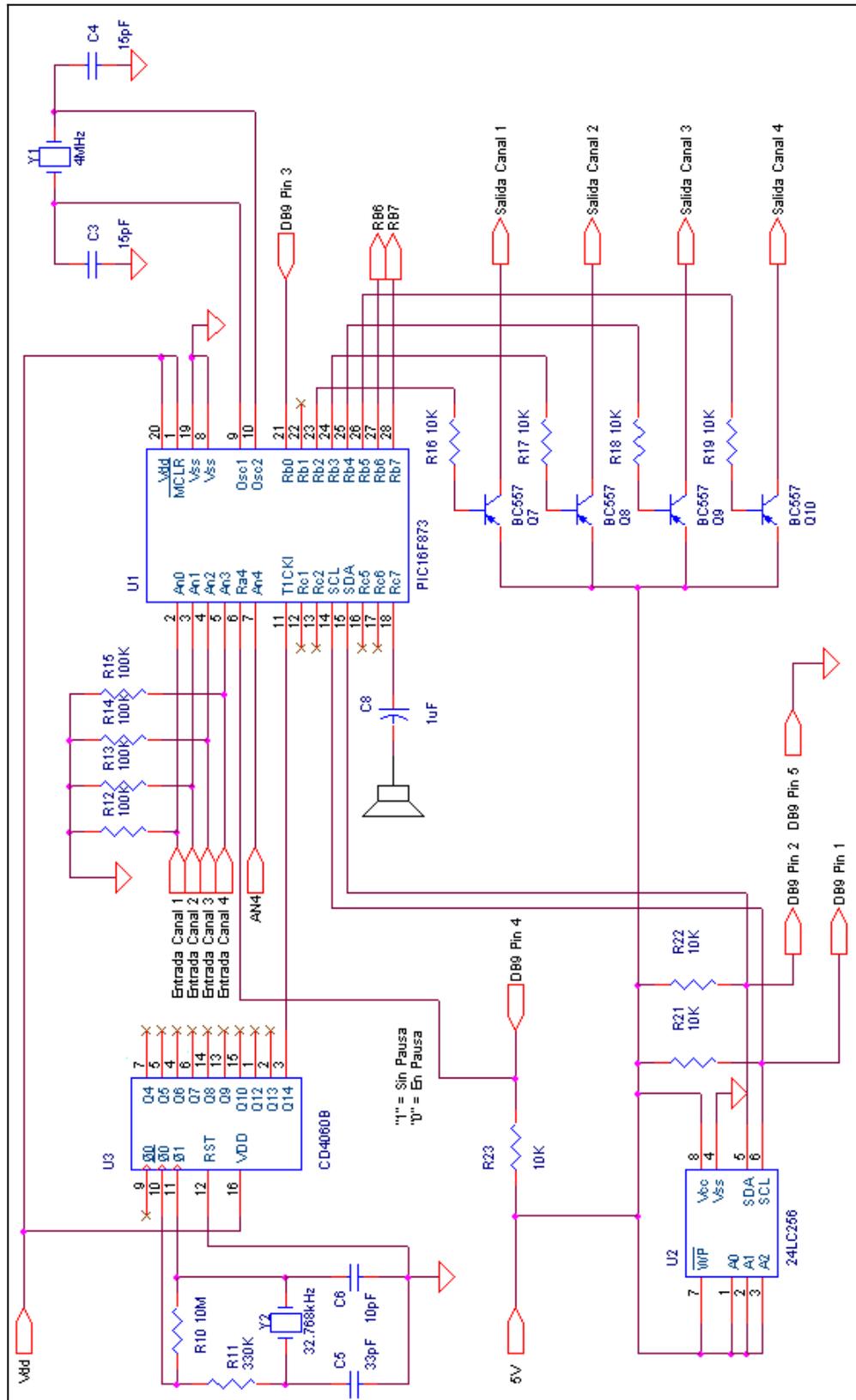
ANEXO A: CIRCUITOS ESQUEMÁTICOS

Los siguientes circuitos conforman todo el sistema electrónico, cuyos esquemas fueron diseñados utilizando el software OrCAD 9.1.

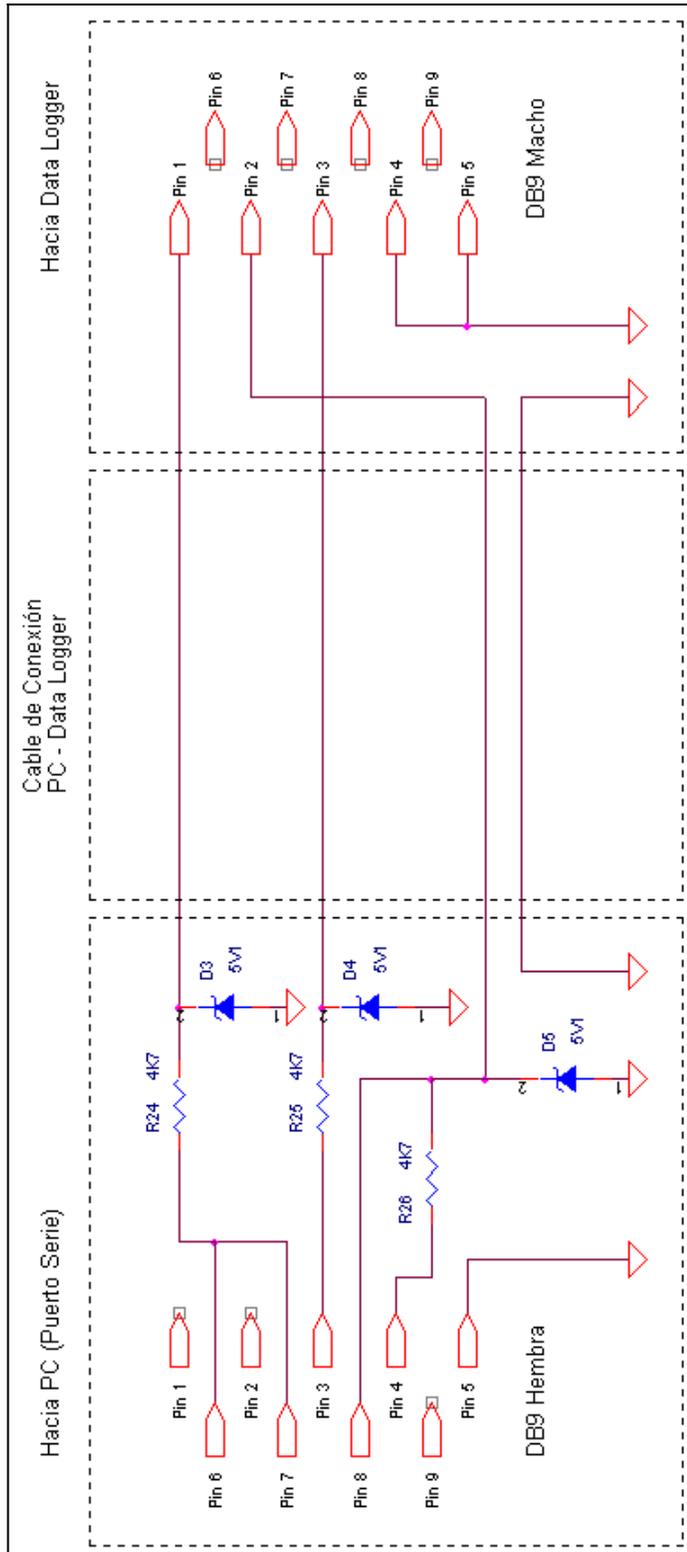
A.1.FUENTE DE ALIMENTACIÓN



A.2.CIRCUITO PRINCIPAL

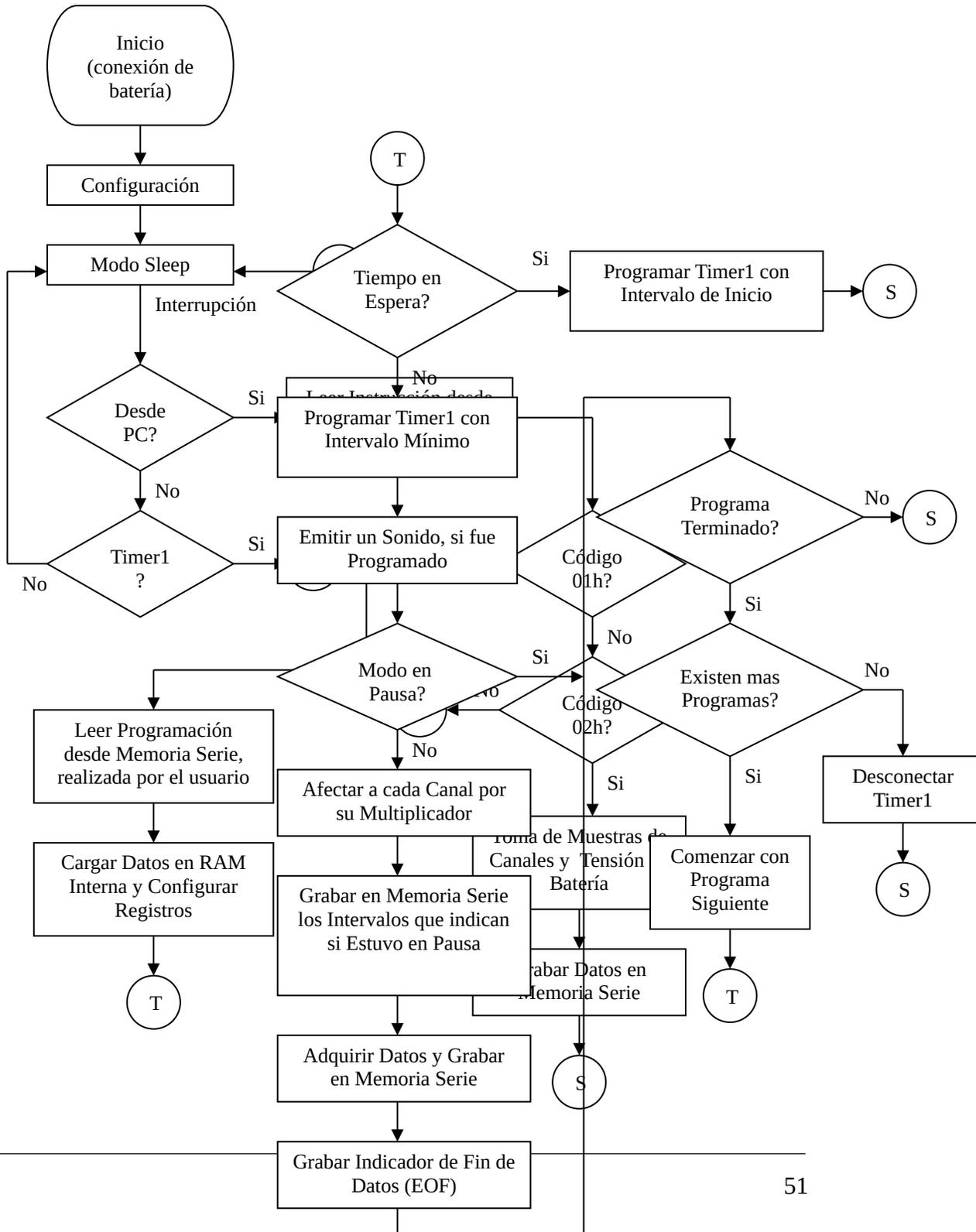


A.3.CABLE DE CONEXIÓN PC



ANEXO B: DIAGRAMA DE FLUJO

A continuación se muestra un diagrama de flujo simplificado del programa que ejecuta el microcontrolador. El código completo se encuentra en el archivo datalogger.asm, escrito mediante el software MPLab de Microchip.



AGRADECIMIENTOS

A los Ingenieros Raúl Rivera y Walter Gemin por prestarme parte de su tiempo, por su ayuda y su insistencia en la mejora del proyecto (especialmente Walter).

Al Ingeniero José Capurro por proponer la idea al departamento.

Al Sr. Martín Ongarini por su guía y ayuda en la utilización del microcontrolador, y principalmente por disponer de la memoria serie.

A mis viejos por bancarme tanto tiempo y a mi novia por ayudarme con la redacción.