

DISEÑO Y CONSTRUCCIÓN DE UN DATA LOGGER UNIVERSAL CON COMUNICACIÓN RS-232 Y USB HACIA UNA PC

Quito Avila Mónica, Ing.
Corrales Luis, PhD
Escuela Politécnica Nacional

1. RESUMEN

El objetivo de este trabajo fue diseñar y construir un Data Logger que posibilite la adquisición de datos desde transmisores comerciales que entregan señales de voltaje de 0 a 10 V, 0 a 5 V o señales de corriente de 4 a 20 mA. El equipo permite el ingreso del tiempo de muestreo, así como también la transferencia de los datos almacenados a una PC sea vía el puerto RS 232 o vía el Puerto USB.

Para la descarga de los datos desde el Data Logger se desarrolló un programa en la PC con Microsoft Visual Basic 6.0. Las pruebas realizadas al equipo demostraron que éste es capaz de trabajar con diferentes tipos de transmisores, la selección del tiempo de muestreo va desde 0.1 s hasta 24 horas, y puede comunicarse con una PC vía RS 232 o USB simultáneamente.

Los datos son guardados en la PC en un formato que permite leerlos desde otros programas como el EXCEL. Desde la PC se podría guardar los datos en una memoria FLASH USB. Al ser una solución propia, se podrá modificar tanto el hardware como el software para ajustarse a cualquier otro tipo de demanda.¹

2. ABSTRACT

This paper aim was to design and build a Data Logger that makes possible the acquisition of data from commercial transmitters that work with voltages ranging from 0 to 10 V, 0 to 5 V or currents ranging from 4 to 20 mA. The equipment allows the selection of the sampling time and the transferring of the data stored to a PC via the RS-232 or USB ports.

The design was based on a PIC 16C774 which sports a 12 bits A/D converter, and features a serial communication port.

For the USB communication, the integrated module USBMOD2 was used. The data are stored in a 64 Kbytes EEPROM memory.

mony_quito@yahoo.es

To upload the data from to Data Logger to the PC, a program in Microsoft Visual BASIC 6.0 was developed. Tests carried out proved the system was able to work with different types of transmitters, the sampling time can be selected from 0,1 s up to 24 hours, and it can communicate with a PC via RS 232 or USB, simultaneously.

The data is stored in the PC in a format that it makes possible to be read from other programs like EXCEL. The data can also be downloaded into a USB FLASH memory. Being this a local solution, this can be modified at will, to suit any other type of demand.

3. INTRODUCCIÓN

La mayoría de empresas o industrias necesitan en sus sistemas diferentes tipos de almacenamiento de datos. La posibilidad de tener datos históricos, hacer proyecciones y predicciones en base a los datos guardados es lo que impulsa su utilización.

Si bien un monitoreo diario, semanal o mensual, de una o de diferentes variables podría hacerse en forma manual, una mejor alternativa es contar con un sistema que colecte los datos de manera automática y controlada.

Las dificultades que se presentan para la captura de datos ha motivado la búsqueda constante de sistemas que faciliten la recolección. Dentro de este esfuerzo se enmarca este proyecto: diseñar una alternativa para muestrear en forma automática señales analógicas, emitidas por diferentes transmisores comerciales, garantizando así la diversidad del dispositivo. Adicionalmente, un programa desarrollado para una PC ofrecerá la posibilidad de transmitir los datos desde el dispositivo hacia la PC, sea por el puerto serial o por el nuevo puerto de comunicación USB.

4. METODOLOGÍA DE DISEÑO

Un Data Logger es un instrumento de registro electrónico que almacena las variables de un proceso medidas en el tiempo. Un Data Logger

se caracteriza por una serie de parámetros que permiten definir su utilización. Los parámetros se fijan a partir de un conjunto de funciones que debe cumplir el mismo, de entre las que destaca el nº de canales de entrada, tanto analógicos y/o digitales, y los márgenes dinámicos de entrada y salida que deben tener dichos canales.

Existen versiones comerciales que capturan datos sea de temperatura, humedad relativa, intensidad de la luz, voltaje, presión, flujo, entre otros. Partiendo de esta realidad, se pensó más bien en un Data Logger que sea de propósito general y, por lo mismo, se pensó en uno que pueda aceptar las señal provista por transmisores comerciales, los cuales es conocido entregan señales normalizadas.

Se pensó también en un instrumento, de naturaleza “portátil”. Esto quiere decir que debía ser diseñado para que consuman la menor potencia posible y pueda trabajar desde baterías de forma continua, el mayor tiempo posible.

SEÑAL DE ENTRADA

Puesto que uno de los objetivos de este trabajo es hacer que el Data Logger trabaje con transmisores industriales comerciales, esto quiere decir que debía poder trabajar con señales analógicas normalizadas, siendo 4 – 20 mA, 0 – 10 V_{DC} y 0 – 5 V_{DC} los valores más empleados.

CAPACIDAD DE LA MEMORIA

Los Data Logger que se comercian localmente, por lo general no permiten modificar el periodo de muestreo; adicionalmente, están limitados por la cantidad de memoria interior que poseen. Por consiguiente, antes de especificar la cantidad de memoria requerida, fue necesario calcularla sobre la base de que este sistema debía posibilitar la selección del tiempo de muestreo. Por ejemplo: para una aplicación que requiere periodos de muestra de una por segundo y la prueba dura una hora, el Data Logger debe poder guardar 3600 muestras (1 muestra/segundo x 1 hora x 3600 segundos/1hora x numero de bytes). Se investigó entonces el mercado para seleccionar una memoria que sea de gran capacidad y, al mismo tiempo, que no consuma mucha energía.

SOFTWARE DE ADQUISICIÓN DE DATOS

Si bien el Data Logger puede almacenar datos, eventualmente estos deben ser transferidos a

una PC para su procesamiento. Con este propósito se decidió desarrollar una aplicación que permita bajar y guardar los datos medidos en un formato que pueda ser leído por aplicaciones como el Excel, por dar un ejemplo.

Resumiendo las características señalads, se partió de un esquema como el que se muestra en la Figura 1.

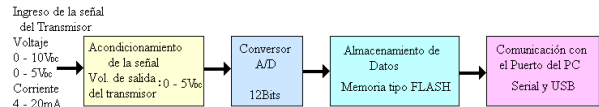


Figura 1. Diagrama de bloques funcional del Data Logger

5. DISEÑO Y CONSTRUCCIÓN DEL HARDWARE DEL DATA LOGGER

El hardware de Data Logger incluye: la fuente de alimentación, el acondicionamiento de las entradas analógicas, el sistema de almacenamiento de datos y el circuito para la transmisión de datos.

La fuente fue diseñada considerando sobre todo el concepto de portabilidad y que garantice su funcionamiento, en cualquier circunstancia. Al final se construyó una fuente de 5V_{DC} para el funcionamiento del PIC y de 24V_{DC} para los transmisores de corriente.

Adicionalmente, se decidió dotarle de dos alternativas de alimentación: con AC y mediante una batería interna para que cumpla con el objetivo de ser portátil.

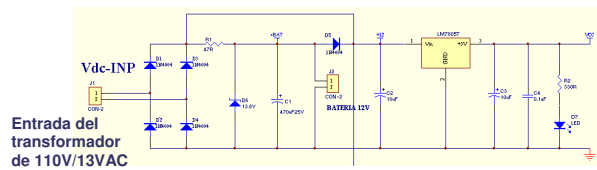


Figura 2. Esquema del circuito de alimentación del sistema Data Logger

ACONDICIONAMIENTO DE LAS ENTRADAS ANALÓGICAS

En el Data Logger se diseñaron circuitos que aceptan valores normalizados de voltaje (0 – 10V_{DC}) y corriente (4 – 20mA), y se decidió que sean digitalizadas con una resolución de doce bits (Convertor A/D 12Bits), para ampliar su espectro de aplicaciones a aquellas que requieren de gran precisión.

ACONDICIONAMIENTO DE SEÑAL DE VOLTAJE

Para trabajar con señales de voltaje de 0 - 5V_{DC} y 0 - 10V_{DC} se hizo necesario diseñar un circuito de acondicionamiento que permita obtener una señal de voltaje adecuada para ser digitalizada. Para esto se seleccionó un amplificador operacional de ganancia elevada que posee una impedancia de entrada alta y una baja impedancia de salida. Buscando permitir que se conecte con una gran cantidad de circuitos para proveerlo de versatilidad, el amplificador esta conectado como circuito seguidor unitario.

ACONDICIONAMIENTO DE SEÑAL DE CORRIENTE

Igualmente, se diseñó un circuito para que maneje la señal de corriente, normalmente de 0 - 20mA o de 4 - 20mA. El acondicionador de señal convierten las señales de corriente a señales de voltaje por medio de una resistencia de precisión (Vea Figura 3). El voltaje que resulta puede de esta forma ser digitalizado.

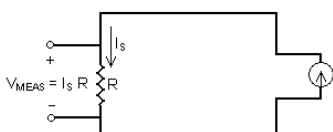


Figura 3. Las señales de corriente de 0 a 20 mA o 4 a 20 mA se convierten en Señales de voltaje pasando por un resistor de alta precisión.

CIRCUITO DE CONTROL

Para el control del sistema se decidió trabajar con un PIC seleccionado en concordancia con las necesidades del mismo.

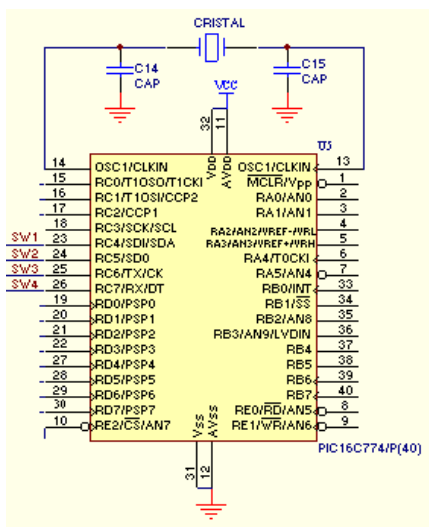


Figura 4. Esquema del PIC16C774

El PIC16C774 EPROM es un microcontrolador con 10 canales análogo/digital de 12 bits de resolución dando una solución completa a señales mixtas. Además posee 4096x14 palabras de memoria de programa y 256 bytes usados para memoria RAM con 5 MIPS de rendimiento @ 20MHz. Destacan dos módulos de PWM y dos puertos seriales. La sincronización del puerto serial puede ser configurado como: SPI interfaz periférico serial, bus I2C Inter-Integrated Circuit y como USART Universal Synchronous Asynchronous Receiver Transmitter, también conocido como interfaz de comunicación serial SCI.

ALMACENAMIENTO DE LOS DATOS

Para almacenar los datos dentro del Data Logger se seleccionó la memoria 24LC64

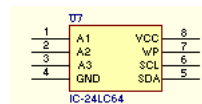


Figura 5. Memoria 24LC64

El circuito integrado EEPROM 24LC64 es una memoria reprogramable y borrable que contiene 8192 localidades de memoria con 8 bits cada una, (8K X 8) (64K bit) y opera con una fuente de alimentación desde +1.8V a +5.5V. La capacidad de escritura de una página es de 32bytes por dato, aproximadamente se puede almacenar 4081bytes.

El tamaño del bus de datos es de ocho líneas equivalente a 512Kbits de direcciones, de ahí que el tamaño estándar es de 8 pines.

INTERFAZ DE COMUNICACIÓN

Para posibilitar la comunicación entre el Data Logger y la PC se recurrió al USBMOD2

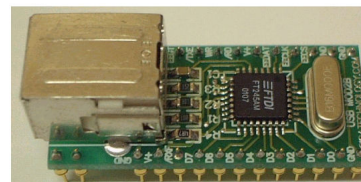


Figura 6. El USBMOD2

El USBMOD2 es un modulo integrado de bajo costo para la transferencia de datos desde un periférico y una PC. Su velocidad es de 8 millones de Bits (1 Megabyte) por segundo. Esta basado en el interfase FTDI FT8U245 USB FIFO que facilita la comunicación.

La velocidad de transferencia paralela de este IC facilita las interfaces con CPU y otros

dispositivos que manejen mapas de memorias con su CPU, I/O, DMA y control de p rticos I/O.

El USBMOD2 es ideal para desarrollo r pido de prototipos Plug and Play.

Sus caracter sticas m s relevantes son:

Emplea un simple modulo de alta velocidad para aplicaciones USB UART; esta basado en FTDI FT8U245 USB FIFO – IC de Transferencia r pida de datos paralelos; empaquetamiento Standard 32 pines tipo DIP; conector integrado USB tipo B; cristal y todos los componentes pasivos integrados en la misma tarjeta; se puede conectar una memoria tipo EEPROM externa y provee fuente desde el USB con una corriente m xima de 50mA.

Otro CI importante en la comunicaci n es el IC FT8U245

Ofrece una soluci n simple para transferencia de datos v a USB en un solo chip; env a y recibe datos hasta una velocidad de 1 MByte/s; posee un buffer de 384 bytes para recepci n y 128 bytes para transmisi n; tiene una interfaz simple para CPU o MCU; no se necesita conocimientos especiales sobre el USB, ni conocer el manejo de protocolos USB; esto lo hace autom ticamente el IC. El p rtico virtual COM - FTDI y los drivers de estos eliminan la necesidad de drivers espec ficos para el USB. Tiene un multiplicador de reloj integrado de 6Mhz – 48Mhz que ayuda en aplicaciones de FCC y CE; no requiere de regulador externo, tiene integrado un regular interno de 3.3V; trabaja con especificaciones USB 1.1.; permite la conexi n de EEPROM externa para almacenamiento de datos.

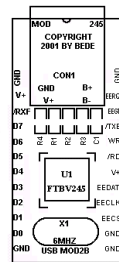


Figura 7. Diagrama del m dulo USBMOD2.

CIRCUITO DEL PIC

Operaciones de Entrada (Pulsadores)

Para entrada de datos y seteo de par metros se utiliza 4 pulsadores descritos como UP, DOWN, SI Y NO. Para que el PIC lea correctamente estos, su p rtico de I/O correspondiente se configur  colocando un 1L

en el bit correspondiente del registro TRIS del p rtico para que act e como entrada. Un 0L hace que act e como salida. Al momento de iniciarse el PIC todos sus puertos quedan como entradas y sus conversores A/D en modo an logo, por lo que es preciso definir tambi n el modo de trabajo.

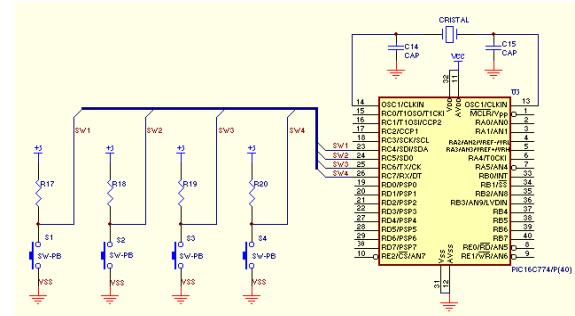


Figura 8. Hardware para los pulsadores

En el hardware de la Figura 8 se observa que se han colocado 4 pulsadores al puerto C a trav s de resistencias pull up, lo que garantiza que cuando no se encuentra pulsado el pin no se halle flotando, y as  d  un nivel l gico estable.

Convertor A/D

Este elemento fija muchas de las caracter sticas del Data Logger. Cuanto mayor sea el proceso de conversi n, mayores ser n las posibles frecuencias de muestreo [1]. Las se ales de entrada han de ser muestreadas seg n el criterio de Nyquist, por lo que es importante que el convertidor anal gico-digital pueda convertir la se al en palabras digitales en el menor tiempo posible; un proceso r pido adquiere m s valores en un tiempo dado que uno lento y esto permite el poder representar mejor las se ales originales.

Otro par metro muy importante en el convertidor anal gico-digital es la resoluci n, que se puede definir como el n mero de bits que utiliza el convertidor para representar la se al anal gica.

La resoluci n del convertidor debe ser suficientemente alta para detectar el m nimo cambio de tensi n exigido.

En la actualidad existen diferentes tipos de convertidores anal gico-digitales. El m s popular es el de aproximaciones sucesivas, ya que ofrece la m xima velocidad y resoluci n.[2] Este tipo de convertidor es caracter stico del PIC16C774.

Por lo tanto, considerando el teorema de Nyquist y el m todo de comparaci n, se

configuró el pórtico correspondiente A/D como análogo, en modo de 12 bits, tiempo de muestreo 50 µs.

El circuito final que funcionó como lo indicado se muestra en la Figura 9.

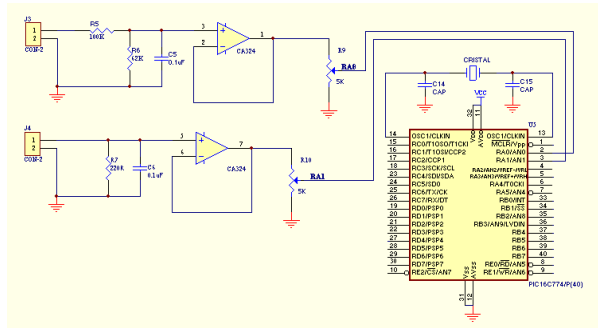


Figura 9. Hardware utilizado para el acondicionamiento de las señales de entrada.

Almacenamiento en la EEPROM

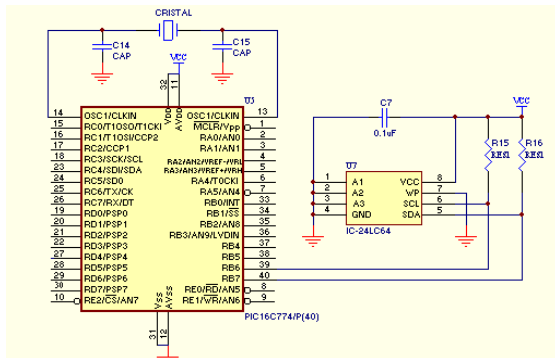


Figura 10. Almacenamiento en la memoria EEPROM.

La Figura 10 muestra las conexiones que hacen posible el almacenamiento de datos en la memoria. Dentro de los varios tipos de comunicación serial síncrona se seleccionó el protocolo I²C [4].

En el diseño del Data Logger se considera la comunicación síncrona porque la transmisión de datos (SDA) se realiza sincronizada con una señal de reloj (SCL), que permitirá el almacenamiento de datos en forma ordenada.

Módulo LCD

El LCD presenta preguntas necesarias en la ejecución del programa del PIC, por ejemplo, el número total de muestras.

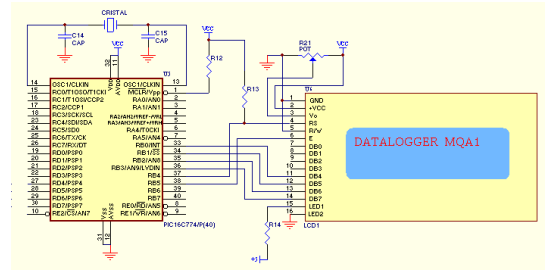


Figura 11. Hardware para el LCD

Para dar soporte al módulo LCD se seleccionó un circuito microcontrolador especializado en el manejo de datos y señales de control necesarias para desplegar el conjunto de caracteres ASCII. La lógica de control se diseño tal que mantenga la información en la pantalla hasta que ella sea sobrescrita o borrada de la memoria RAM de datos [5].

Comunicación Serial

En la actualidad los microcontroladores muy a menudo se comunican con PCs, generalmente para aprovechar el poder de procesamiento y gráficos de la PC. Si bien lo actual es una conexión vía USB, no muchos dispositivos industriales soportan este tipo de comunicación. Por lo mismo, para que este equipo tenga una mayor versatilidad en cuanto a comunicación, se decidió proveerlo tanto de comunicación RS 232 como USB.

Puerto RS 232

El diseño del circuito para la transmisión de datos por medio del pórtico serial (RS – 232) requirió diseñar un circuito que pueda manejar las señales eléctricas de los puertos serie RS232. [7]

Un 0 lógico esta entre +3 y +25 V.

Un 1 lógico esta entre -3 y -25 V.

La región entre -3 y +3 V no esta definida.

El voltaje en circuito abierto no debe exceder los 25 V, en referencia a tierra.

La corriente en cortocircuito no debe exceder los 500mA.

No se descuidó en el diseño que la comunicación RS232 es asincrónica. Esto significa que no hay señal de reloj asociada a la señal de datos sino que cada palabra es sincronizada empleando un bit de inicio (start bit) y un bit de parada (stop bit).

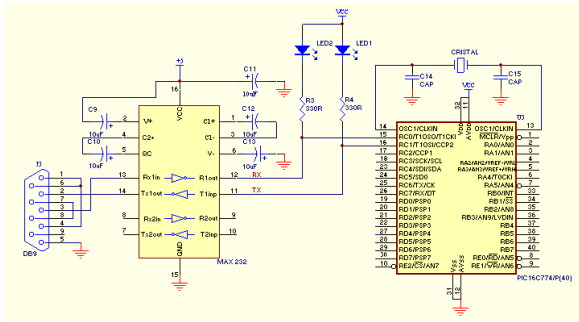


Figura 12. Hardware utilizado para la comunicación serial.

Puertos USB [8].

El PC adolece de ciertas falencias como la escasez de líneas de interrupción IRQs y canales de acceso directo a memoria DMA.

Esta carencia se resolvió dotando al PC del bus de alta velocidad USB (Universal Serial Bus) que es un bus serie, bidireccional y de bajo costo que ofrece las características ideales de Plug and Play (PnP) y facilidad de conexión y desconexión.

El bus USB soporta intercambio de datos entre la PC y un amplio conjunto de periféricos. Todos los periféricos conectados comparten el ancho de banda del bus por medio de un protocolo de arbitraje basado en testigos ("Tokens"). El bus permite conexión y desconexión dinámica, es decir, que los periféricos se conectan, configuran, manipulan y desconectan mientras el sistema principal y otros periféricos permanecen en funcionamiento.

La topología del bus USB es tipo estrella y se organiza por niveles. En un bus USB existen dos tipos de elementos: Principal ("host") y dispositivos; a su vez, los dispositivos pueden ser de dos tipos: concentradores y funciones. Algunos dispositivos pueden ser de los dos tipos al mismo tiempo. Por ejemplo, una pantalla USB (función) puede ser a su vez un concentrador con dos o más conexiones auxiliares para conectar otros dispositivos.

Los concentradores ("Hubs") son el centro de una estrella, y sirven para conectar al sistema principal, con otro hub o con una función. Cada hub puede proporcionar 500 mA con una alimentación de hasta 2.5 W de cada uno de los dispositivos a él conectados.

Una función es un dispositivo capaz de transmitir o recibir datos o información de control en un bus USB, suele conectarse como un dispositivo independiente enlazado por un

cable de menos de 5 metros, a un puerto del hub o directamente al sistema principal.

Que un hub pueda estar conectado a otro hub, significa que pueden conectarse dispositivos en cascada; el sistema soporta un total de 127 dispositivos. Una característica importante es que el PC principal o el concentrador proporcionan la energía necesaria a la función por el cable de conexión, lo que evita la necesidad de fuentes de alimentación independientes.

El protocolo de comunicación utilizado es de testigo, que guarda cierta similitud con el sistema Token-Ring de IBM. Puesto que todos los periféricos comparten el bus y pueden funcionar de forma simultánea. La información es enviada en paquetes; cada paquete contiene una cabecera que indica el periférico a que va dirigido. Existen cuatro tipos de paquetes distintos: Token, Datos, Handshake, y Especial, el máximo de datos por paquete es de 8; 16; 32 y 64 Bytes respectivamente.

El funcionamiento está centrado en el host, todas las transacciones se originan en él; es el controlador host el que decide todas las acciones, incluyendo el número asignado a cada dispositivo (esta asignación es realizada automáticamente por el controlador "host" cada vez que se inicia el sistema o se añade, o elimina, un nuevo dispositivo en el bus), su ancho de banda, etc. Cuando se detecta un nuevo dispositivo es el host el encargado de cargar los drivers oportunos sin necesidad de intervención por el usuario.

El sistema utiliza cuatro tipos de transacciones que resuelven todas las posibles situaciones de comunicación. Cada transacción utiliza un mínimo de tres paquetes, el primero es siempre un Token que avisa al dispositivo que puede iniciar la transmisión.

Transferencia de control ("Control transfer"). Ocurre cuando un dispositivo se conecta por primera vez. En este momento el controlador de host envía un paquete "Token" al periférico notificándole el número que le ha asignado.

Transferencia de pila de datos ("Bulk data transfer"). Este proceso se utiliza para enviar gran cantidad de datos de una sola vez. Es útil para dispositivos que tienen que enviar gran cantidad de datos cada vez, como escáneres o máquinas de fotografía digital.

Transferencia por interrupción ("Interrupt data transfer"). Este proceso se utiliza cuando se

solicita enviar información por el bus en una sola dirección (de la función al host).

Transferencia de datos isócrona ("Isochronous data transfer"). Este proceso se utiliza cuando es necesario enviar datos en tiempo real. Los datos son enviados con una cadencia precisa ajustada a un reloj, de modo que la transmisión es a velocidad constante.

Cables y conectores

El cable de bus USB es de 4 hiloscuya función se indica a continuación:

Tabla 1. Distribución de pines

Pin	Nombre	Descripción	Color
1	VBUS	+ 5 VCD	rojo
2	D-	Data -	azul
3	D+	Data +	amarillo
4	GND	Tierra	verde

Existen dos tipos de cable: apantallado y sin apantallar. En el primer caso el par de hilos de señal es trenzado; los de tierra y alimentación son rectos, y la cubierta de protección (pantalla) solo puede conectarse a tierra en el principal. En el cable sin apantallar todos los hilos son rectos. Las conexiones a 15 Mbps y superiores exigen cable apantallado.

Tabla 2. Conexiones del USB

AWG	mm Ø	long. máx.
28	0.321	0.81 m
26	0.405	1.31 m
24	0.511	2.08 m
22	0.644	3.33 m
20	0.812	5.00 m

Se usan dos tipos de conectores, A y B. Ambos son polarizados (solo pueden insertarse en una posición) y utilizan sistemas de presión para sujetarse. Los de tipo A utilizan la hembra en el sistema principal, y suelen usarse en dispositivos en los que la conexión es permanente (por ejemplo, ratones y teclados). Los de tipo B utilizan la hembra en el dispositivo USB (función), y se utilizan en sistemas móviles (por ejemplo, cámaras fotográficas o altavoces). En general la hembra de los conectores A están en el lado del host (PC) o de los concentradores (hubs), mientras las de tipo B están del lado de los periféricos.

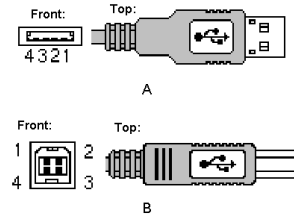


Figura 13. Conectores USB

USBMOD2

El IC USBMOD2 permite obtener los datos de manera sencilla sin necesidad de conocimientos especiales sobre el USB, ni conocer el manejo de protocolos USB, esto lo hace automáticamente el IC.

Para la obtención de datos a través del USBMOD2 se debe considerar los siguientes pines:

/RD (pin de entrada): Si esta en bajo habilita los bytes de datos en D0...D7 que corresponde al bus de datos bidireccional Bit #0 al Bit #7 respectivamente, a la transición alta saca el siguiente byte de datos si existe.

WR (pin de entrada). En la transición de alto a bajo escribe el byte de datos en D0...D7

/TXE (pin de salida). Cuando esta en bajo puede escribirse datos caso contrario no.

/RXF (pin de salida). Cuando esta en alto, no lee datos y cuando esta en bajo pueden leerse los datos disponibles (Ver Figura 14).

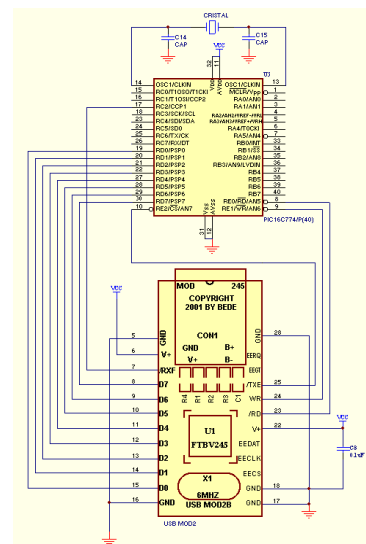


Figura 14. Hardware utilizado para la comunicación USB.

6. DESARROLLO DEL SOFTWARE DE COMUNICACIÓN DE DATOS.

DISEÑO DEL SOFTWARE DEL PIC

Este programa se diseño para realizar las funciones siguientes: ingresar la fecha y hora en que se inicia el muestreo, ingresar el tiempo de muestreo, seleccionar el tipo de señal a muestrear, voltaje o corriente, guarda los datos en la memoria, y tener acceso a ellos por medio de la comunicación serial y USB.

Para lograr estas funciones se desarrollo el programa principal cuyo diagrama de flujo se muestra en la Figura 19 siguiente.



Figura 15. Programa Principal

Para la programación del PIC se realizó un análisis de las características de cada uno de los elementos que intervienen en el sistema Data Logger y se utilizó para el desarrollo el compilador PICBASICPRO. Varias subrutinas fueron desarrolladas para lograr producir un software que produzca una interfaz lo más amigable de usar.

En la Figura 16 se da el algoritmo base para la transmisión de datos.



Figura 16. Algoritmo de la Comunicación Serial RS-232

Es importante saber la función que realiza el USBMOD2, ya que al ser un módulo integrado tiene ciertos parámetros que deben ser configurados, como el pin RXF que indica si el buffer esta con datos o no. Si este buffer esta con datos se debe dar la instrucción para el inicio de comunicación.



Figura 17. Algoritmo de la Comunicación Serial USB

SUBROUTINA TRANSFERENCIA DE DATOS

Esta subrutina permite transferir los datos hacia la PC.

Para el inicio de transferencia de datos se envía una marca o un código que permita dar el inicio de comunicación puede ser este un "1" o "*" desde la PC; dentro del software del PIC se establecerá ese código.

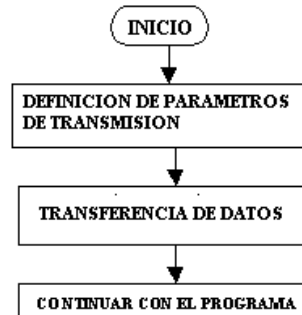


Figura 18. Algoritmo para la transferencia de datos

DESARROLLO DEL SOFTWARE DE COMUNICACIÓN DE DATOS

Para el desarrollo del Software del proyecto Data Logger, se empleó el programa Microsoft Visual Basic 6.0, por ser el más práctico para este tipo de aplicaciones.

Al inicio del programa muestra una pantalla en la cual se puede seleccionar dos funciones a realizar, continuar y salir.



Figura 19. Pantalla principal

Al seleccionar “Continuar” se despliega otra pantalla que permite realizar otras funciones.

Si se elige “Salir” abandona el programa.



Figura 20. Pantalla de funciones

Al seleccionar Adquirir se da el inicio a la transferencia de datos desde el Data Logger a la PC. Este software presenta dos opciones adicionales:

La primera es para abrir con Excel y la otra es para Grabar a Flash.

El archivo es tipo texto y se graba con el nombre de datos.xls, el nombre está dado por el programa.

Al seleccionar “Regresar” se despliega la pantalla principal.

Si se elige “Salir” abandona el programa.

PROGRAMA EN VISUAL STUDIO BASIC 6.0

Este programa permite obtener los datos de una manera sencilla. En base a comandos se puede pasar de una pantalla a otra; después de adquirir los datos se envía a una hoja de cálculo de EXCEL, y/o a grabar en la Memoria

Flash USB. La Figura 21 indica el algoritmo del programa desarrollado en Visual Studio Basic.

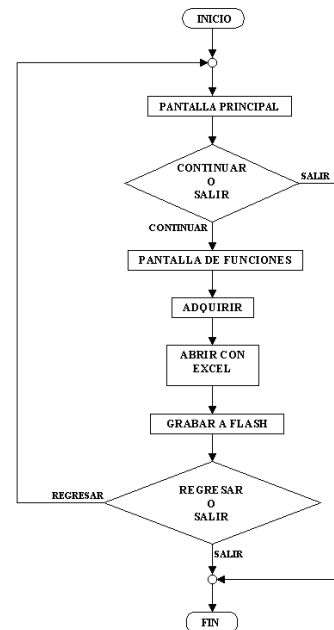


Figura 21. Algoritmo del Programa en Visual Studio Basic

Visual Studio ofrece herramientas que facilitan el ensamblaje del programa obteniendo así un programa ejecutable que permitirá ser instalado en cualquier computador sin que sea necesario instalar previamente Visual Studio.

7. CONCLUSIONES

En la industria es muy difundido el uso de transmisores que midan las variables de un proceso. Al entregar valores de corriente o voltaje normalizados, favorecen y facilitan la creación de dispositivos de propósito general, uno de los objetivos del trabajo presente. Pero si bien el DATA LOGGER que aquí se ha construido, está listo para trabajar con cualquier transmisor, muy bien pueda ser empleado para que se acople a sensores o equipos de medición de diseño propio, con la única condición que generen corriente o voltajes estándar.

Para el funcionamiento en otros rangos de corriente o voltaje o tiempos de muestreo diferentes a los implementados, el haber producido una solución local, posibilita alterar tanto el software como el hardware para ajustar este pequeño sistema a otras aplicaciones. Una conclusión valiosa que se puede extraer de lo hasta aquí dicho es la conveniencia de que la Universidad promueva y apoye el desarrollo de tecnología propia.

8. BIBLIOGRAFÍA

- [1] Ver Pág. 188, Curso de Electrónica V; Técnica de Medición y Regulación
- [2] Ver Pág. 193, Curso de Electrónica V; Técnica de Medición y Regulación
- [3] Ver Pág. 304. Curso de Electrónica III; Módulos de la μ electrónica
- [4] Ver Pág. 52, Manual Microprocesadores PIC
- [5] Ver Pág. 67, Manual Microprocesadores PIC
- [6] Para mayor información ver:
<http://www.lvr.com>
<http://www.monografias.com/trabajos5/transdat/transdat.shtml#muestre>
- [7] Para mayor información ver:
<http://www.redcientifica.com/doc/doc200207030001.html>
<http://www.euskalnet.net/shizuka/rs232.htm1>
- [8] Para mayor información ver
http://www.zator.com/Hardware/H2_5_3.htm# o
<http://www.usb.org/>

9. BIOGRAFÍA



Mónica del Rocío Quito Avila, Nació en Quito, Ecuador en el año de 1975. Obtuvo el título de bachiller en Humanidades Modernas especialización Físico-Matemático en el Colegio Experimental “24 de Mayo” en 1992. Sus estudios superiores lo realizó en la Escuela Politécnica Nacional en Quito obteniendo su título en Ingeniería Electrónica y Control en 2005. Actualmente se desempeña como profesora en la Escuela Politécnica Javeriana del Ecuador.