

SISTEMAS DE CONVERSIÓN ANALÓGICA DIGITAL

Introducción

La utilización de un ordenador en un sistema de medida pasa necesariamente por la conversión de la señal analógica procedente de los sensores a una señal digital capaz de poder ser entendida por el ordenador. A este proceso se le conoce como Conversión Analógica Digital (ADC).

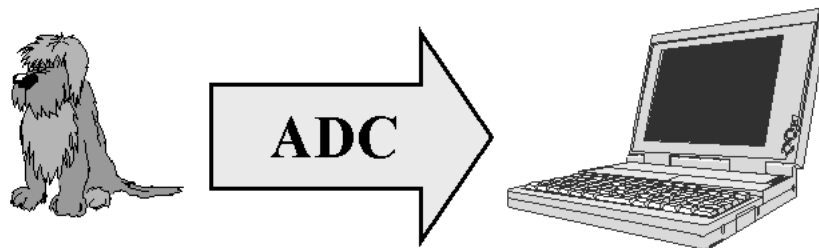


Figura 4-1. Cuando se desea pasar información a un computador se necesita un conversor analógico digital.

Por el contrario, en muchos sistemas de regulación se precisa de una señal analógica de control y para ello se debe realizar el proceso inverso: convertir la señal digital, obtenida con el ordenador a una señal analógica mediante un dispositivo de Conversión Digital Analógica (DAC). Existen muchos diseños para realizar un circuito de conversión analógica digital o viceversa. En general se utilizan circuitos integrados que realizan estas funciones directamente o con un mínimo de circuitería periférica. Sin embargo, no todos son aptos para una aplicación concreta y por ello deben estudiarse cuidadosamente sus características y elegir el que mejor se adapte a nuestras necesidades. La precisión y la velocidad de muestreo son los más evidentes, pero la simplicidad del diseño, la estabilidad frente a condiciones adversas y el consumo suelen ser muchas veces las cualidades que condicionan todo el diseño. El coste es otro factor a tener en cuenta, en general es posible disponer de circuitos ampliamente utilizados en la industria a un precio muy razonable.

Conversión digital analógico

Un convertidor digital analógico es un dispositivo que genera una señal (en corriente o en tensión) proporcional a la palabra digital presente en sus entradas. El convertidor digital analógico más sencillo que se puede concebir consta simplemente de una tensión de referencia y de un conjunto de resistencias que entran en circuito en función de que su correspondiente interruptor este conectado o no. La tensión de salida del amplificador operacional viene dada por:

$$V_0 = V_{REF} \left(\frac{B_0}{8R} + \frac{B_1}{4R} + \frac{B_2}{2R} + \frac{B_4}{R} \right) Rr$$

donde V_o es la tensión de salida de operacional, V_{Ref} es la tensión de referencia y R_r la resistencia de realimentación del amplificador operacional. S_0, S_1, S_2, S_3 son los valores lógicos (0 o 1) de los correspondientes bits.

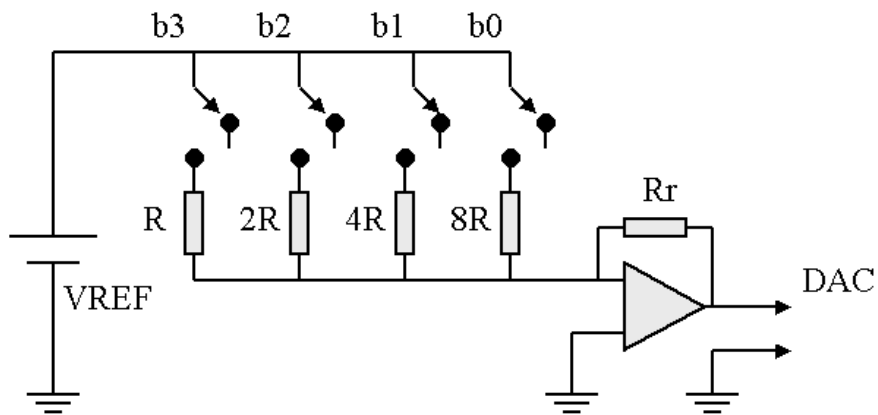


Figura 4-1. La figura presenta un convertor digital analógico de 4 bits realizado con un amplificador operacional y cuatro resistencias.

La realización de un convertor con esta tecnología requiere sólo disponer de un conjunto de resistencias de precisión de valores $R, 2R, 4R, \dots 2^n R$. Este requerimiento es muy difícil de conseguir cuando el número de bits aumenta. En realizaciones pequeñas se suele acudir a acoplar en paralelo (o serie) grupos de resistencias iguales. Una astucia permite realizar convertidores digital analógicos de alta resolución con sólo dos conjuntos de resistencias de precisión de valores R y $2R$, como todas las resistencias son iguales, el conjunto es mucho más fácil de producir industrialmente. Para ello se utiliza un circuito derivado del anterior con una red de resistencias $R-2R$ a la entrada del amplificador operacional.

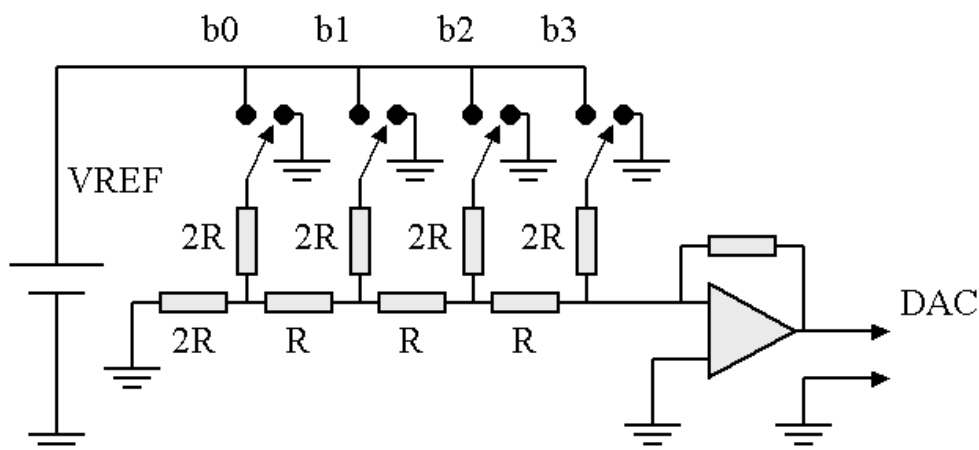


Figura 4-3. La figura presenta un convertor digital analógico de 4 bits realizado con un amplificador operacional y una red de resistencias $R-2R$.

La mayoría de los convertidores digital analógico (y analógico digital) trabajan con este dispositivo. La precisión de las distintas resistencias y su variación con la temperatura limitan este tipo de convertidores a resoluciones por debajo de 16 bits.

En general, los conversores digital analógico suelen ser por diseño unipolares. Para su utilización como dispositivos bipolares se desplaza la tensión de salida mediante un segundo amplificador operacional y una corriente auxiliar derivada de la propia tensión de referencia.

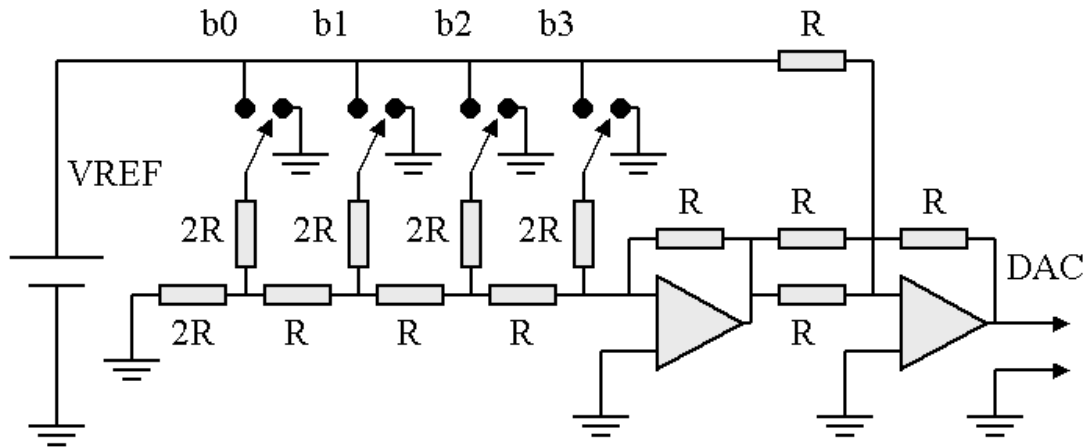


Figura 4-4. Conversor digital analógico de 4 bits bipolar con red de resistencias R-2R.

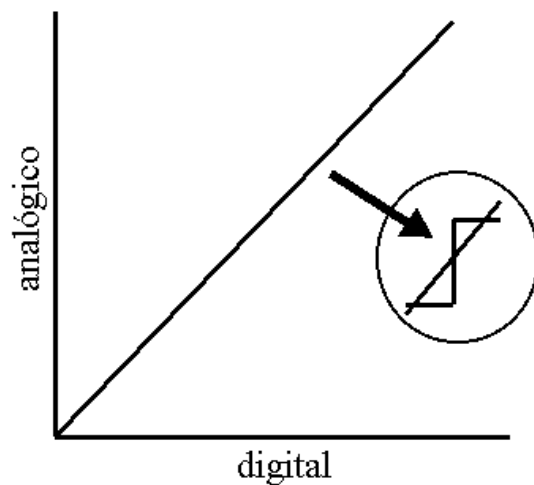
La utilización practica de cada uno de ellos depende del tipo seleccionado, debiéndose respetar escrupulosamente las condiciones de diseño. Una precaución a tener en cuenta es la estabilidad de la tensión de referencia, especialmente su variación con la temperatura o el envejecimiento. Hay que tener presente que para un simple conversor de 12 bits la estabilidad de la referencia debe ser mejor de 2^{-12} .

Errores de digitalización

Las cualidades de un conversor digital analógico se expresan en función de sus discrepancias con el comportamiento que debería presentar un conversor digital analógico ideal:

Figura 4-5. Error de cuantificación

En un conversor digital analógico ideal la relación entre la entrada digital (números) y la salida analógica (voltios) es lineal. La resolución es igual al incremento más pequeño que puede experimentar la señal digital que es igual al cambio del bit menos significativo. El error es siempre menor a 1 bit menos significativo.



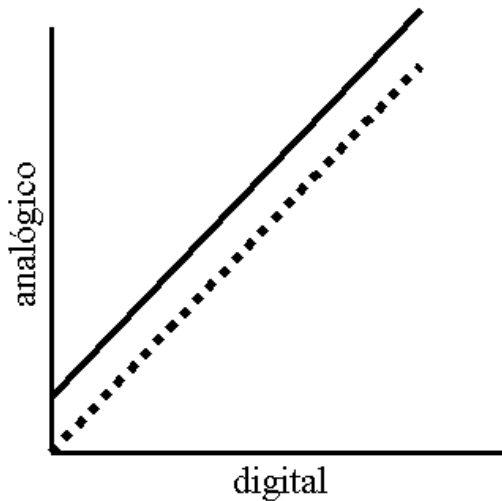


Figura 4-6. Error de cero (offset)

En un conversor digital analógico real la señal analógica está desplazada con respecto al valor que le correspondería en el conversor ideal. Este desplazamiento equivale a que para una entrada digital igual a cero se tiene un valor de la tensión de salida del conversor (error de cero). En general, puede compensarse mediante las técnicas habituales aplicadas a los amplificadores operacionales. Es importante destacar que el offset varía notablemente con la temperatura.

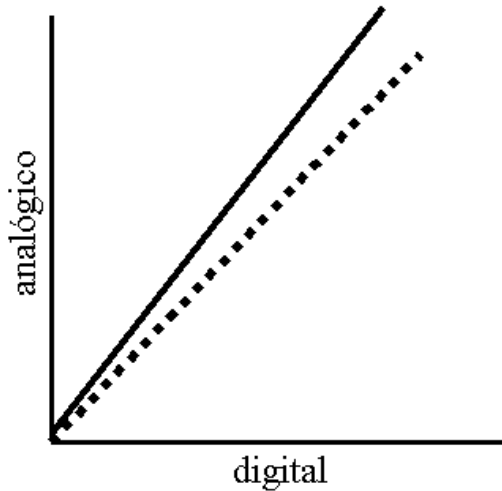


Figura 4-7. Error de amplificación

La discrepancia entre la salida real y la salida teórica aumenta con el valor de la entrada. Ello se debe a que la red de resistencias de los amplificadores operacionales de salida no está bien ajustada. Este error es fácilmente corregible y todos los circuitos de conversión digital analógica presentan estas resistencias accesibles desde el exterior. Otra causa de este error es la fuente de la tensión de referencia, la cual debe ajustarse con sumo cuidado.

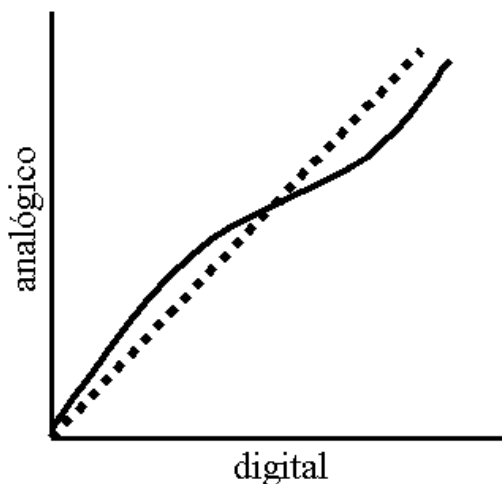
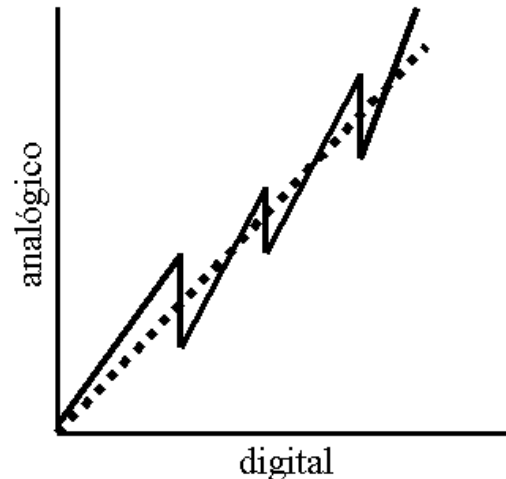


Figura 4-8. Error de linealidad

La relación entre la señal digital y la señal analógica discrepa ligeramente de la dependencia lineal que deberían presentar. Se debe a pequeñas discrepancias entre los valores de las distintas resistencias que componen la red R-2R. En general un mismo fabricante suministra un mismo tipo de conversor con distintas precisiones en linealidad. Sólo se puede compensar por software.

Figura 4-9. Error de monotonicidad

A una señal creciente en la entrada digital no siempre le corresponde una señal de salida creciente. Este error se debe a desajustes en la red R-2R y no se puede corregir. Es especialmente acusado en las series económicas de los convertidores de alta resolución. En general son dispositivos de muchos bits pero en los que sólo los más significativos están correctamente ajustados. A esto debemos añadir la dependencia de la temperatura de estas características. Otros factores importantes al seleccionar un determinado tipo de convertidor es la velocidad máxima de operación y el requerimiento energético de todo el sistema.



Al seleccionar un adecuado convertidor digital analógico se deben tener presente los márgenes que presentan estos errores, teniendo en cuenta que la información dada por el fabricante corresponde a datos de laboratorio y en la práctica se presentan valores ligeramente superiores. Hay que tener presente que la precisión de un convertidor analógico digital ideal es de +/-1 bit menos significativo lo que exige precisiones superiores al 0.05% para un convertidor de 12 bits en todos los componentes.

Conversión analógico digital (ADC)

El convertidor analógico digital es el componente básico para que un ordenador pueda realizar la medida de la señal eléctrica analógica suministrada por el elemento sensor. Estos elementos pasan de una señal que varía continuamente a una señal que lo hace a saltos (resolución) y sólo cada cierto tiempo (muestreo). Las distintas limitaciones que presenta un determinado tipo de convertidor analógico digital son las mismas ya definidas al tratar los convertidores digital analógico. Sin embargo la falta de monotonicidad se suele denominar en este caso como pérdida de código (*missing code*).

El convertidor analógico digital más sencillo consiste en una serie de circuitos comparadores ajustados cada uno de ellos a niveles de tensión de referencia sucesivamente cada vez más elevado, obtenidos mediante un divisor resistivo de una misma fuente de alta estabilidad. Al aplicar una tensión a la entrada del convertidor, conmutan todos aquellos convertidores cuya tensión de referencia sea menor que la entrada. Un circuito lógico codifica la salida de los comparadores en un número en el formato binario deseado. La gran velocidad de muestreo y la cualidad de que la conversión se realiza con un sólo pulso de reloj, hace que a este tipo de convertidores se les conozca como *convertidores flash*.

El elevado número de comparadores que se precisan para realizar un convertidor de este tipo hace que sólo existan en catálogo unos pocos circuitos de este tipo, todos ellos de poca resolución (8 bits). La elevada velocidad de conversión (>10MHz) que puede alcanzarse con ellos les hace aptos para el procesado digital de señales de alta

frecuencia, como el radar o el vídeo digital. En geofísica, este tipo de conversores sólo se utiliza en las técnicas de georadar. Los indicadores de nivel en escala logarítmica que equipan muchos instrumentos son conversores de este tipo con 16 comparadores distribuidos en una escala logarítmica de niveles sucesivos.

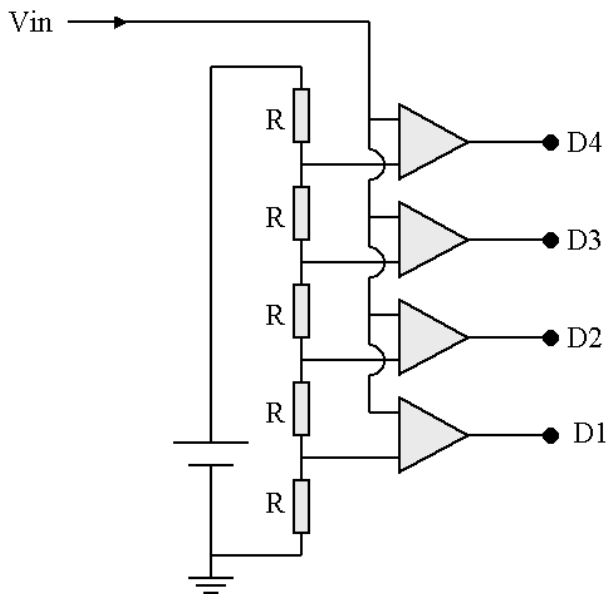


Figura 4-10. Conversor analógico digital de 4 niveles por comparación directa de la señal de entrada con cuatro tensiones de referencia obtenidas mediante un divisor resistivo.

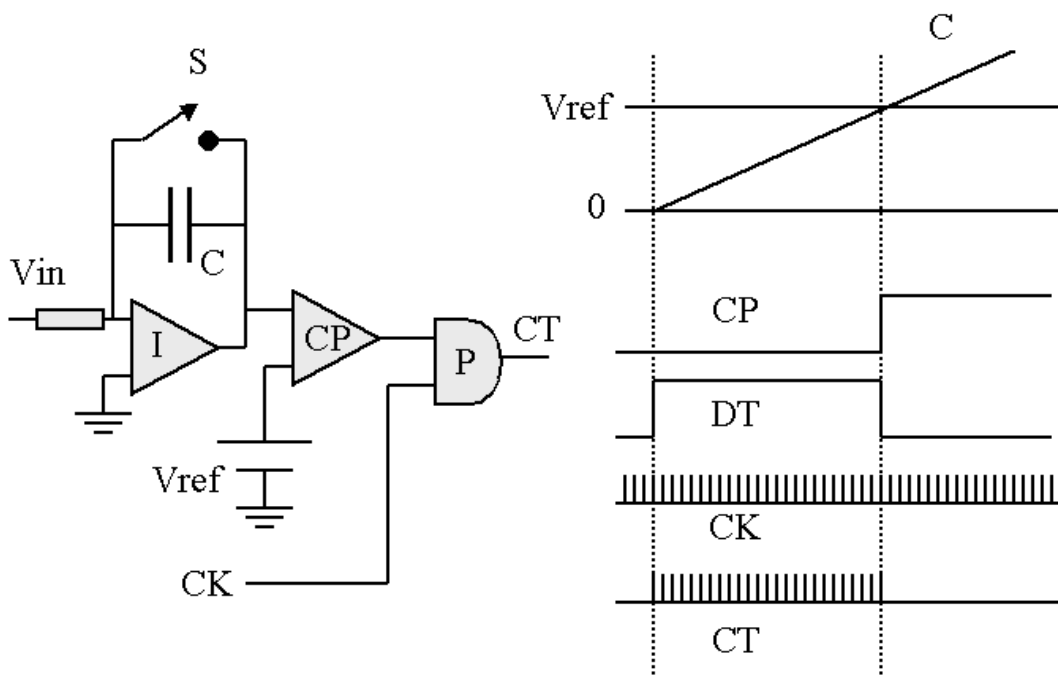


Figura 4-11. Conversor analógico digital de integración: consiste en medir el tiempo que tarda en cargarse un condensador C a corriente constante. I integrador, CP comparador, P puerta para el conteo de pulsos, S interruptor de inicio de conversión.

El problema de precisar muchos elementos de precisión para realizar un conversor analógico digital ha llevado a desarrollar otra familia de conversores que sólo requieren una referencia de tensión estable: son los conversores de *rampa* o de *integración* basados en cargar un condensador mediante una corriente proporcional a la tensión de entrada. Estos conversores, por las condiciones de diseño, presentan una excelente linealidad y elevada precisión, pudiéndose alcanzar fácilmente 16 bits a bajo costo. Sin

embargo, difícilmente pueden hacer más de unas pocas muestras por segundo (modelos de 12 bits y superiores). Una cualidad intrínseca del diseño de estos convertidores es su gran capacidad de rechazo del ruido, especialmente el inducido por la red de suministro eléctrico (50Hz o 60Hz). Para ello hay que diseñar el sistema para que el tiempo de integración sea múltiplo del periodo de la red.

La conversión se realiza midiendo el tiempo transcurrido (T) entre el momento en el que se abre el interruptor S y el momento en el que la carga del condensador de integración C hace conmutar al comparador COMP. La medida del tiempo se realiza contando pulsos de reloj (CK). La dificultad de este tipo de convertidores radica en su calibración y en el error introducido por las derivas que pueda presentar el sistema de amplificador operacional (integrador) y el comparador. Una solución es utilizar un convertidor de doble rampa.

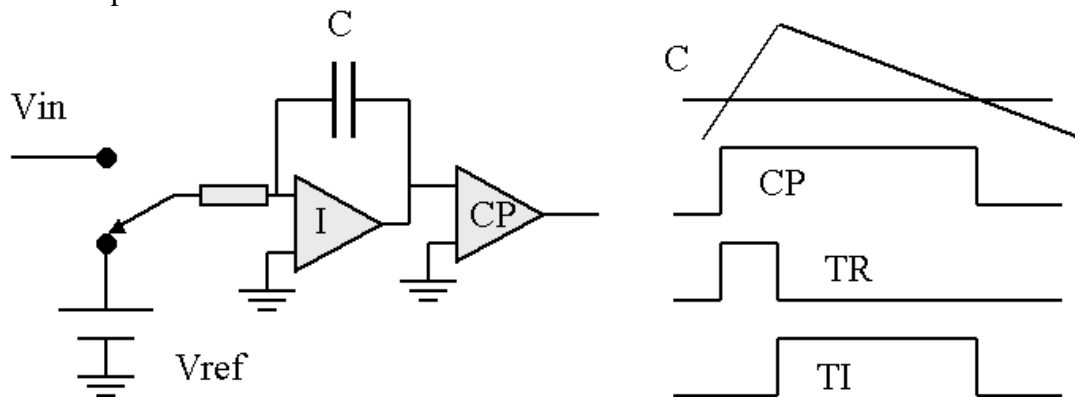


Figura 4-12. Convertidor analógico digital de doble rampa. Se carga el condensador C con una corriente de referencia durante un tiempo fijo TR y se mide el tiempo necesario para descargarlo con la corriente obtenida con la señal de entrada Vin.

Este tipo de convertidores comparan el tiempo necesario para cargar el condensador C mediante la corriente suministrada por la tensión a medir (Vin.) con el tiempo necesario para descargarlo hasta el nivel inicial (Ref.) mediante una corriente conocida generada por la fuente de referencia. La lógica de control conmuta la entrada del integrador entre la tensión a medir y la fuente de referencia. Inicialmente la entrada está conectada a potencial cero (masa).

En general todos los voltímetros digitales que se encuentran en el comercio operan bajo este principio, pues son dispositivos de una elevada precisión y estabilidad intrínseca y de muy bajo costo, pues no requieren ajustes críticos. Un convertidor analógico digital muy utilizado de doble rampa es ICL7109. Este circuito CMOS posee varias entradas de control que simplifican extraordinariamente su operación en múltiples configuraciones, tanto conectado directamente al bus del sistema de datos como su conexión directa a un dispositivo serie (UART) lo que le permite funcionar en modo remoto. De este convertidor existen versiones preparadas para activar directamente un visualizador de cristal líquido (LCD) como el 7106 o de diodos LED.

Otra familia de convertidores analógico digital se diseñan a partir de un convertidor digital analógico del tipo R-2R, cuya salida se compara con la tensión a medir mediante un circuito comparador. Una pequeña lógica de control varía la señal digital presente en la entrada hasta que la salida del convertidor iguala a la tensión a medir. La realización más

sencilla es utilizar un contador bidireccional conectado a la entrada del conversor digital analógico.

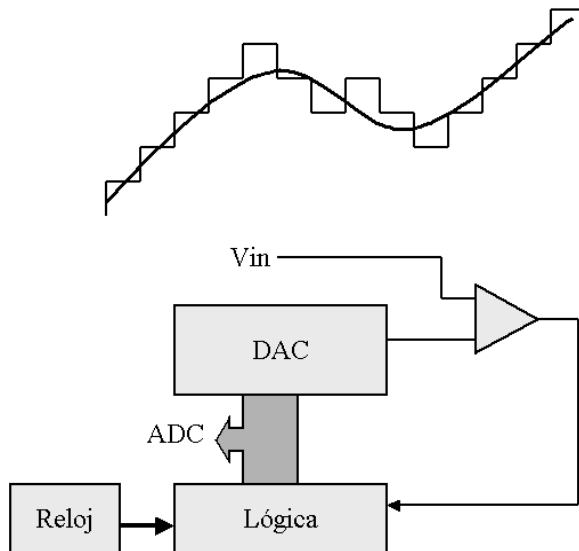


Figura 4-13. Convertor analógico digital de seguimiento. La salida de un conversor digital analógico se compara con la señal de entrada, según sea mayor o menor se incrementa o decrementa un contador asociado al conversor digital analógico.

Según el comparador indique un valor superior o inferior, se incrementa o decrementa el valor del contador (*convertor de seguimiento*). Cuando la señal varía más rápidamente que la velocidad del conversor, los datos leídos quedan sistemáticamente por debajo o por arriba del valor correcto. Además, este tipo de conversor tiene que invertir un tiempo considerable hasta que el contador alcanza el valor correcto.

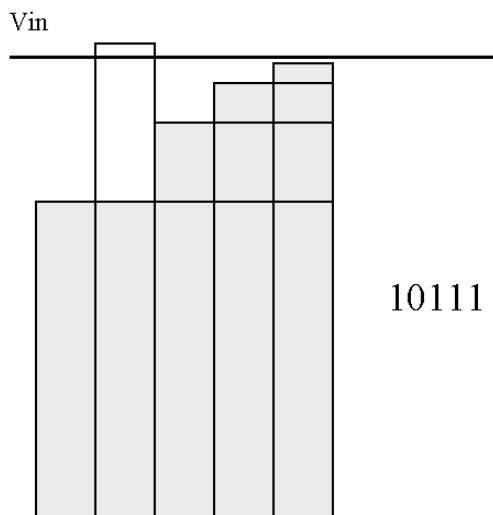


Figura 4-14. Conversión analógico digital por aproximación sucesiva. Un circuito similar al utilizado en los convertidores de seguimiento, pero ahora, cada vez que se realiza una conversión se ensayan los sucesivos bits empezando por el más significativo. Según se obtenga un valor menor o mayor se añade o no el bit al control del conversor digital analógico. El tiempo necesario (número de pulsos de reloj) es siempre igual al número de bits más uno.

Una solución a este problema es utilizar un algoritmo de aproximación sucesiva: se ensaya inicialmente con el bit más significativo, si es mayor este bit permanece a 0 durante todo el proceso y si es menor a 1. Seguidamente se ensaya con el segundo bit más significativo y se va repitiendo el proceso. El número de ensayos a realizar es igual al número de bits a convertir más uno, mientras que en un convertidor de contador puede llegar a ser de 2^n ensayos. Es muy importante que la señal de entrada no varíe durante todo el tiempo que dura la conversión, ya que un error en una sola de las comparaciones provoca que todo el código de salida sea erróneo. Para evitar que la señal de entrada varíe durante la conversión (unos pocos microsegundos) se utilizan circuitos de

muestreo y retención o cuando el espectro de la señal de entrada lo permite un simple filtro pasa bajos que limite la velocidad de variación de la señal por debajo de 1 bit menos significativo en el tiempo que dura una conversión. Los conversores analógico digitales disponibles en forma de circuitos integrados, como ADC0801 requieren muy pocos componentes externos: como son la referencia de tensión, los condensadores de desacoplo y el sistema de reloj (un cristal de cuarzo, una red RC o en generador externo). Las señales de control permiten seleccionar el circuito (CS), iniciar una conversión (WR), o leer los datos (RD). El estado del conversor (en reposo o realizando un conversión) se conoce mediante una señal fin de conversión (EOC o INTR). Muchos conversores R-2R son unipolares, para su utilización en modo bipolar se desplaza el cero a la mitad de la tensión de referencia, para ello basta con disponer de un divisor resistivo conectado a la entrada de conversor, entre la tensión de referencia y la entrada de señal (salida del amplificador de entrada). La rapidez, precisión y estabilidad de los conversores de aproximación sucesiva hace que estos sean los más utilizados en la gama de 8 a 16 bits y velocidades entre 10Hz y 1MHz. A velocidades menores se utilizan conversores de *doble rampa* y a mayores, conversores de tipo *flash*.

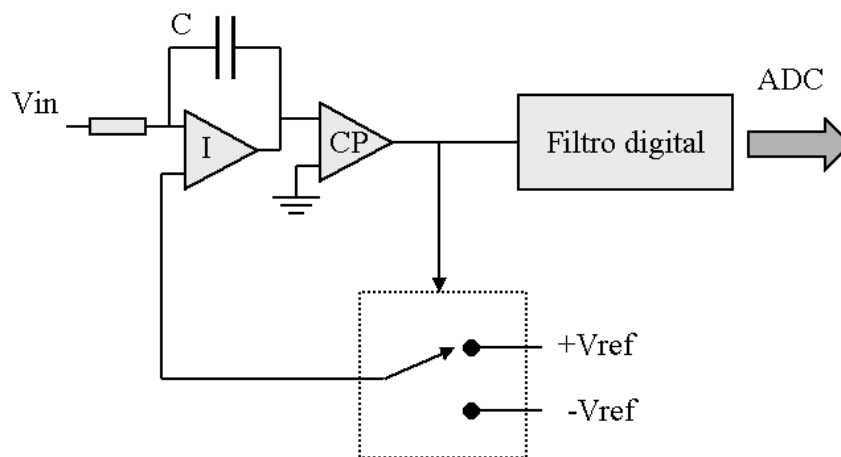


Figura 4-15. Conversor analógico digital *delta-sigma* se trata de un conversor con un integrador en el que a la señal de entrada se suma o resta una tensión de referencia. Este proceso se hace muy rápidamente, de forma que se obtienen una secuencia de 1 y 0 según el número de transiciones hayan sido necesarias para igualar a la tensión de entrada. Un filtro digital reduce la velocidad de muestreo al valor deseado, aumentando paralelamente la resolución.

Es importante proteger mediante diodos la entrada del conversor frente a tensiones que superen la alimentación o de polaridades inversas. Un amplificador operacional dispuesto a la entrada del conversor permite trabajar con una elevada impedancia de entrada, amplificar la señal de entrada o disponer de un filtro pasa bajos o pasa altos.

Conversores de alta resolución

La resolución de los conversores analógico digitales está limitada a 16 bits, ya que la tecnología actual no permite ajustar y mantener los distintos componentes del conversor con precisiones superiores a 2^{-16} . Además, el propio ruido térmico originado en los componentes, contactos y conductores es también mayor a 2^{-16} . Estas resoluciones resultan insuficientes en algunos casos, como es la sismología, en la que el margen de variación de la señal puede superar ampliamente el 2^{16} , sin embargo la señal a procesar presenta un espectro limitado y bastante bien definido que permite obtener una

resolución mayor utilizando un método de sobre-muestreo y un procesado digital posterior de los datos. Este tipo de conversores, que en banda limitada pueden alcanzar 24 bits de margen dinámico, se basan todos ellos en un conversor de un sólo bit operando a alta velocidad en seguimiento continuo de la señal.

El circuito básico consta de un amplificador diferencial, seguido de un integrador y un comparador, Según la salida del comparador indique mayor o menor se inyecta una carga dQ o $-dQ$ obtenida a partir de la misma referencia de tensión. De este modo a la salida del comparador se obtiene una secuencia de 1 y 0 que corresponde a las variaciones de la señal de entrada. Esta secuencia digital se procesa en tiempo real mediante un filtro digital. Este proceso de señales permite reducir el ruido del sistema y obtener información válida correspondiente a variaciones de señal inferiores al salto de 1 bit en el conversor. La eficacia de este tipo de conversores radica en el elevado número de conversiones que realizan por segundo (200KHz) que el proceso digital reduce a menos de 200Hz. El proceso de filtrado se realiza en dos pasos: en el primero se promedia la señal y se reduce en un factor 100 o superior (diezmado). Este filtro digital presenta una respuesta del tipo $\text{sen}(x)/x$ (figura). El segundo filtro es un filtro pasa bajos de corte muy abrupto. Este tipo de conversores siempre proporcionan el código de salida en serie, con el reloj ligado a la frecuencia aparente de muestreo, aunque es posible con ciertas limitaciones operar con reloj externo. En caso de que se desee un código paralelo basta con disponer de dos/tres registros de desplazamiento de 8 bits.

Por su propia estructura estos conversores no permiten multiplexar la señal de entrada, funcionando siempre como conversores de seguimiento continuo de la señal de entrada. Un cambio brusco en la entrada requiere varios ciclos de medida. Además, cada vez que se aplica la tensión de alimentación entran en un ciclo de auto-calibrado, que también es accesible externamente. Un ciclo de lectura especial permite conocer los coeficientes del filtro y las constantes de calibración. Todos estos conversores poseen diversas ganancias, aunque no son compatibles todas ellas con los modos de operación en alta resolución. El costo de este tipo de conversores es relativamente bajo, ya que se utilizan masivamente en los equipos de sonido digital. Las características de estos conversores dependen fuertemente de la velocidad de muestreo elegida y de la amplificación elegida.

Conversores analógico digital de aplicación general

Actualmente hay multitud de conversores analógico digitales, resultando en ocasiones difícil decidirse por un modelo concreto. Los criterios de selección pueden ser:

- Facilidad y seguridad de suministro
- Condiciones de operación
- Velocidad de muestreo
- Resolución
- Salida de datos serie o paralelo
- Señales necesarias para su operación
- Consumo
- Precio

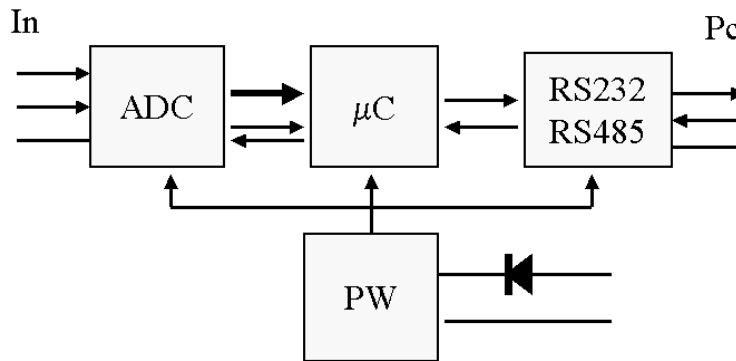


Figura 4-16. La utilización de un convertor analógico digital requiere un mínimo de componentes auxiliares para que pueda funcionar de forma sencilla. Se parte de un convertor analógico digital (ADC), operado por un microcontrolador (μC) que proporciona los datos del convertor en un formato estándar aceptado por los computadores. En muchos casos se requiere un adaptador de nivel (RS232/RS485) para acoplar el dispositivo a la línea de transmisión y a la entrada del computador. Una fuente de alimentación protegida contra cortocircuitos e inversiones de polaridad completará el circuito.

Es muy importante analizar la facilidad y seguridad de suministro, pues la implementación de un convertor analógico digital supone un considerable esfuerzo en realización de placas de circuito impreso y en desarrollo del software para su operación y no podemos estar sujetos a que en pocos meses sea descatalogado. Se debe tener presente que en el entorno de los volcanes activos las condiciones de operación son muy extremas, por consiguiente deberemos elegir circuitos que presenten buena estabilidad térmica y soporten temperaturas extremas. La velocidad de muestreo y la resolución son dos parámetros fáciles de elegir, pues los fabricantes ofrecen sus productos bajo estos dos parámetros. La salida serie o paralelo depende del tipo de interfase que se disponga, en general vamos a trabajar con interfases serie que permiten una fácil conexión a un microcontrolador mínimo. El número de señales necesarias para su operación condiciona el número de líneas de entrada salida que el microcontrolador deberá disponer para manejar el convertor. El consumo es importante en equipos que deban operar aislados, aunque la mayor contribución en ese campo es el sistema de telemetría o de almacenamiento de datos. En general no hay grandes diferencia de precio entre los convertidores de características similares.

En general precisaremos convertidores lentos y de alta resolución como son los convertidores de tecnología *delta-sigma*. Solo en aplicaciones muy especiales utilizaremos convertidores de menor resolución (12 bits) como puede ser en el caso de que el consumo sea una condición primordial o que las características de las señales no justifiquen el empleo de un convertor de alta resolución. Aunque, dada la poca diferencia de precio, es preferible decidirse por un convertor de alta resolución y utilizarlo en todas las aplicaciones, con ello se simplifica el mantenimiento y el tiempo invertido en desarrollos.

En muchos casos es posible utilizar módulos comerciales para realizar la adquisición de datos, pero en general no han sido concebidos para trabajar en el campo, requiriendo una potencia excesiva y no siempre adaptados a condiciones extremas. En general, estos módulos están diseñados para operar con computadores de sobremesa o portátiles de elevadas prestaciones por lo que su operatividad sobre el volcán es reducida.

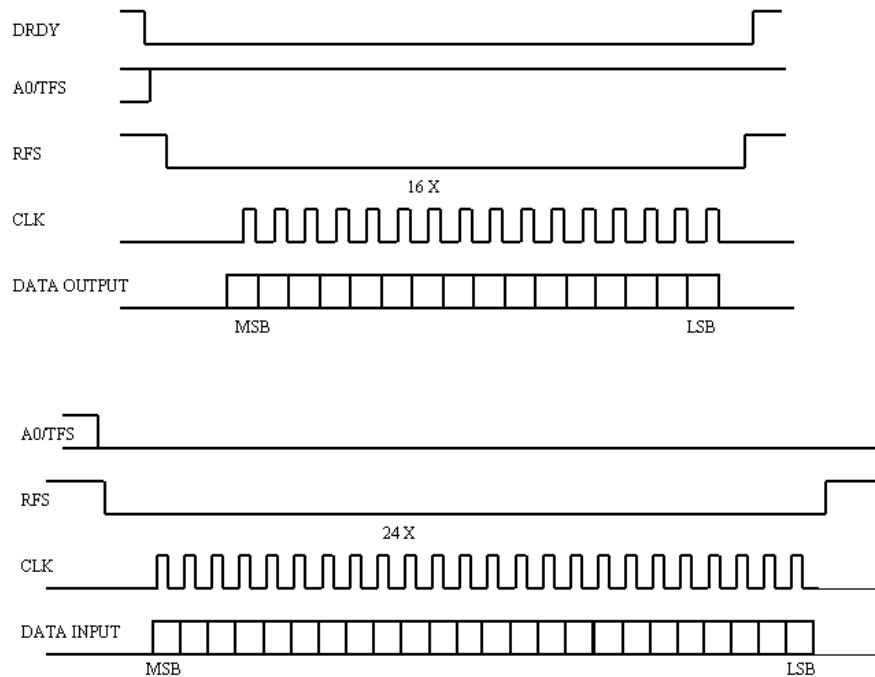


Figura 4-18. Señales necesarias para controlar el AD7710. Posee una interfase serie debiéndose primero enviar un mensaje de configuración y después leer los datos al finalizar cada medida.

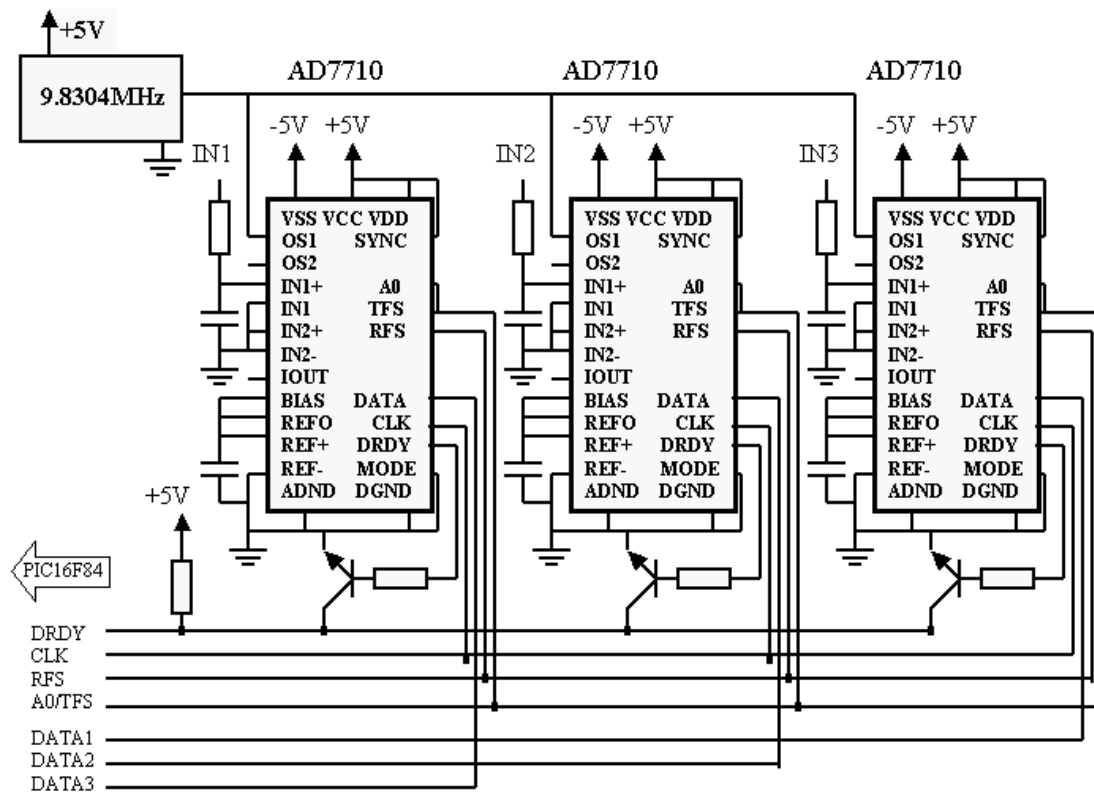


Figura 4-19. También es sencillo construir un módulo para tres canales. Como los convertidores delta-sigma no permiten multiplexar rápidamente las señales de entradas se necesita un convertidor para cada canal que compartan las líneas de control. Se utiliza un transistor (BC107) para mezclar las tres señales de fin de muestra.

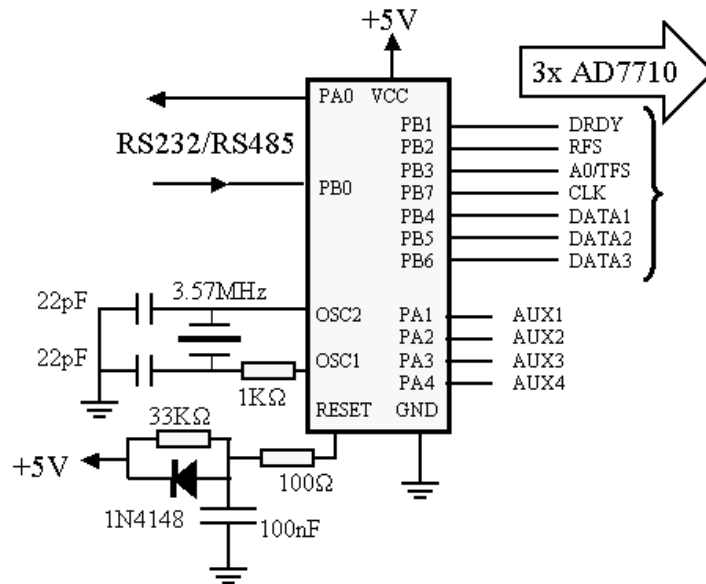


Figura 4-5. Se utiliza un sólo microcontrolador PIC16F84 para manejar los tres convertores AD7710.

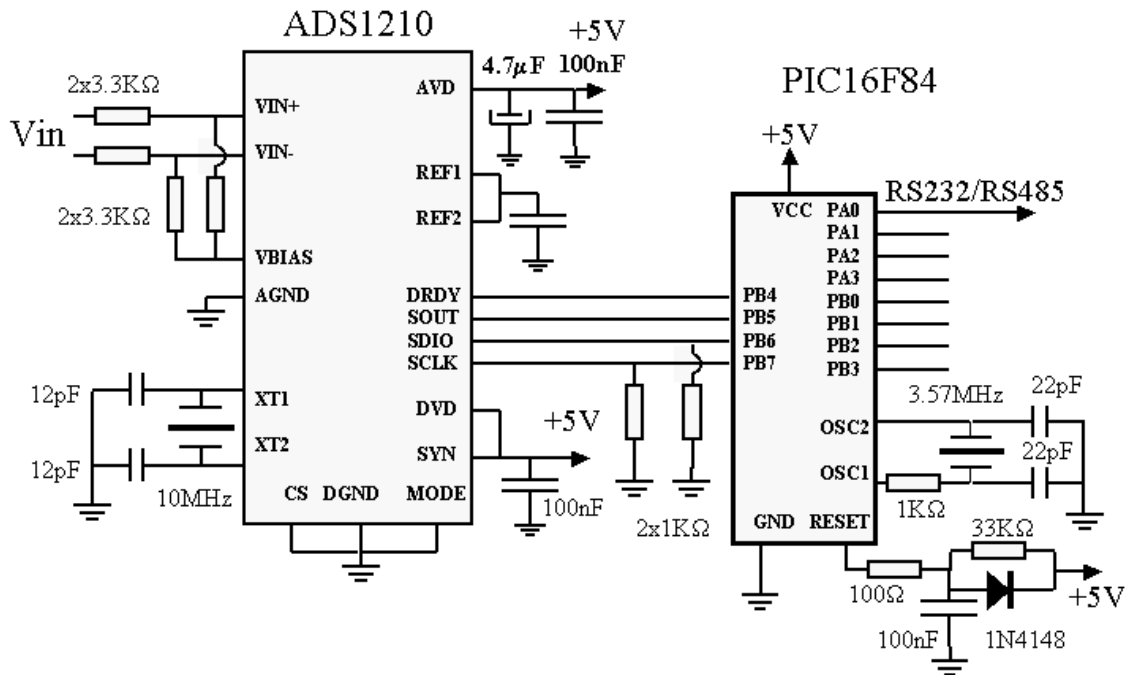


Figura 4-20. Módulo de adquisición de datos realizado con un convertor analógico digital ADS1210 y un microcontrolador PIC16F84. Obsérvense las dos resistencias de 10 KΩ a masa en las líneas de reloj SCLK y datos SDIO. Estas resistencias evitan que durante el arranque el convertor interprete la inicialización de las entradas y salidas del microcontrolador como comandos.

ADS1210

Se trata de otro convertor de tecnología delta-sigma de 24 bits de resolución producido por Burr-Brown. Precisa menos líneas de control, de hecho puede operar con sólo dos líneas, pero el software es más sencillo con cuatro líneas. Este convertor sólo acepta señales positivas, lo que obliga a disponer de un circuito de entrada para cambiar el nivel. En muchos casos puede ser suficiente emplear un doble divisor resistivo como se

muestra en el esquema. En otros deberemos incluir un amplificador operacional, que habrá que seleccionar con mucho cuidado para mantener la tasa de ruido dentro de límites aceptables. También dispone de un preamplificador de ganancia variable (1 a 16) y diversos modos de operación que permiten buscar un compromiso entre consumo, precisión y frecuencia de muestreo. Sin embargo no es posible cubrir todo el rango de 24 bits en todas las escalas ni modos de operación. Por consiguiente debe estudiarse detalladamente cual es el punto de operación deseado, en este sentido es preferible utilizar el AD7710.

La operación es similar al AD7710, se debe enviar primero un comando, especificando la frecuencia de muestreo y la ganancia, y después leer los tres Bytes de datos. Es importante prestar atención al hecho que los comandos deben enviarse sólo cuando lo autoriza el convertor (DRDY bajo), en caso contrario puede producirse un comportamiento errático del convertor. La necesidad de utilizar un puente resistivo a la entrada para poder manejar señales negativas obliga a operar siempre con elevadas ganancias en el convertor, pues en caso contrario es imposible aprovechar todo el margen dinámico. También debe cuidarse la secuencia de arranque, polarizando a masa las líneas de entrada al convertor para evitar que se confunda la programación de estas líneas, por parte del microcontrolador, como comandos

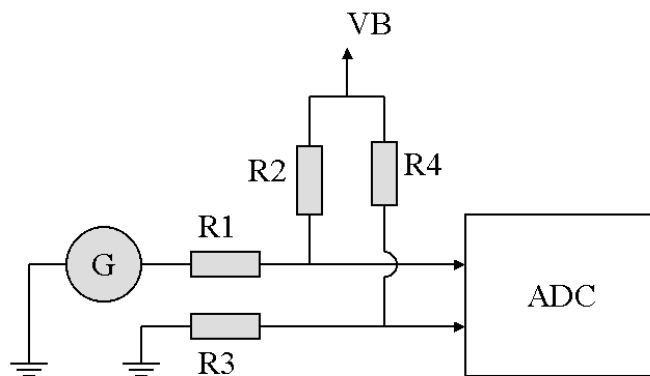


Figura 4-21. La utilización de un puente resistivo para polarizar adecuadamente la entrada diferencial de forma que pueda aceptar señales negativas debe hacerse teniendo en cuenta la resistencia interna del generador de señal G. La resistencia R3 debe ser igual a $R1 + G$ y $R2 = R4$.

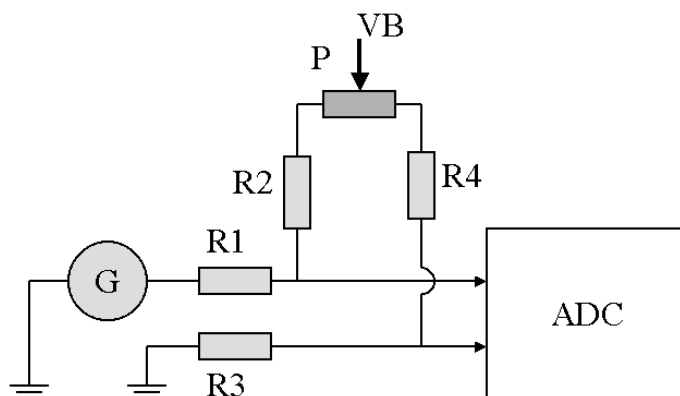


Figura 4-22. La impedancia de salida de la funete se puede compensar fácilmente con un potenciómetro P.

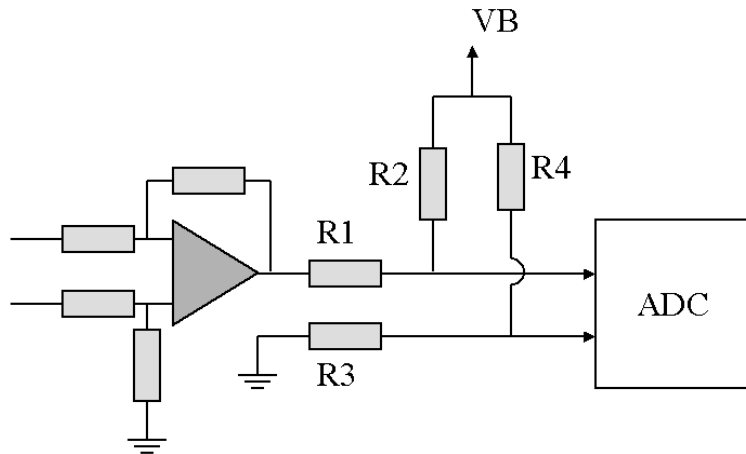


Figura 4-23. En muchos casos es necesario utilizar un amplificador diferencial alimentado simetricamente (+5V, -rV) que proporciona una muy baja impedancia de salida y de esta forma el sistema es independiente de la impedancia de la fuente.

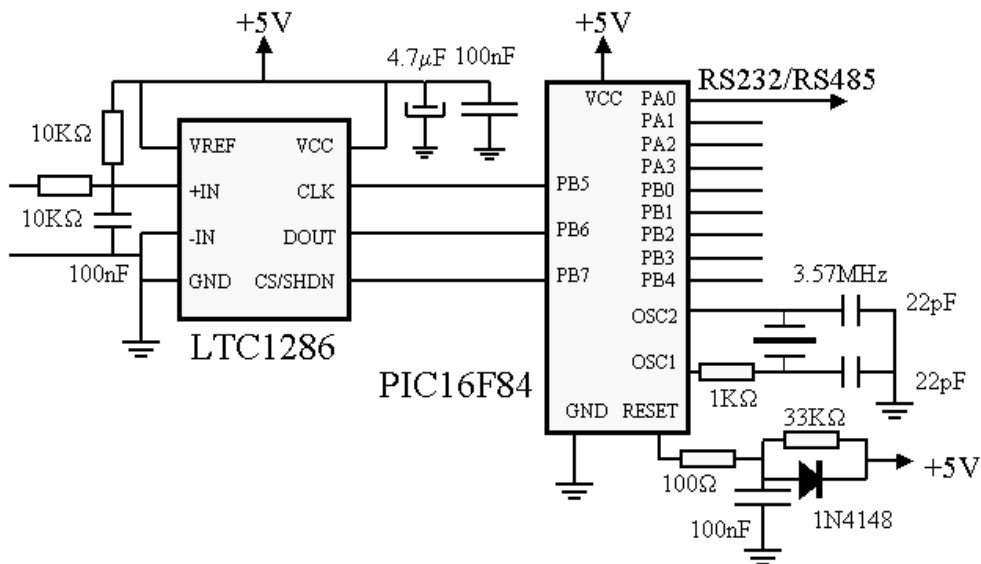


Figura 4-24. Conversor de 12 bits de aproximación sucesiva, interfase serie y bajo consumo. Se utiliza la misma fuente de alimentación como tensión de referencia. El microcontrolador debe ser también de bajo consumo (PIC16FL84).

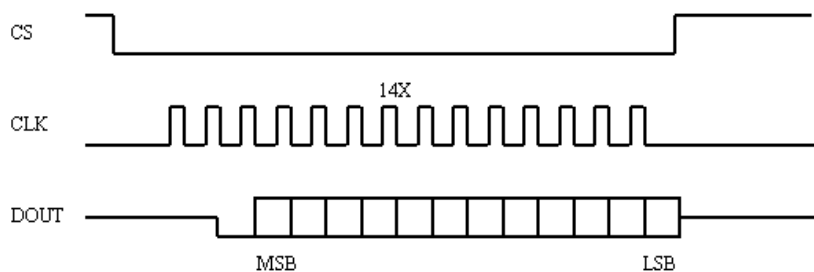


Figura 4-25. La interfase serie es muy sencilla, basta con seleccionar el conversor (CS) y enviar doce pulsos de reloj.

Conversores de 12 bits.

La diferencia de precio ya no justifica utilizar en aplicaciones de baja frecuencia conversores de baja resolución, salvo en casos en que el consumo sea primordial. En general 12 bits es una resolución insuficiente en sismología, incluso operando con sensores de baja calidad. Igualmente en los inclinómetros es poca sensibilidad salvo que se disponga de un complejo mecanismo de ajuste de cero. Los conversores de aproximación sucesiva de 12 bits son componentes muy sencillos de operar, con una interfase serie de dos o tres hilos, por la que envían ráfagas con los doce bits en serie. El consumo es muy reducido, pero cuando se requiere una referencia de tensión de precisión o un preamplificador esta ventaja puede quedar muy reducida. Estos conversores se utilizan mucho en equipos autónomos de adquisición de datos a muy baja velocidad, por ejemplo un dato a la hora y en los que no se requiere mucha precisión. Este tipo de instrumento puede permanecer mucho tiempo en situación de espera con consumo muy reducido, se conecta toma el dato y vuelve a reposo. En estos casos este tipo de conversor es el ideal pues sólo consume en el momento en que está realizando la medida, volviendo automáticamente al estado de bajo consumo.

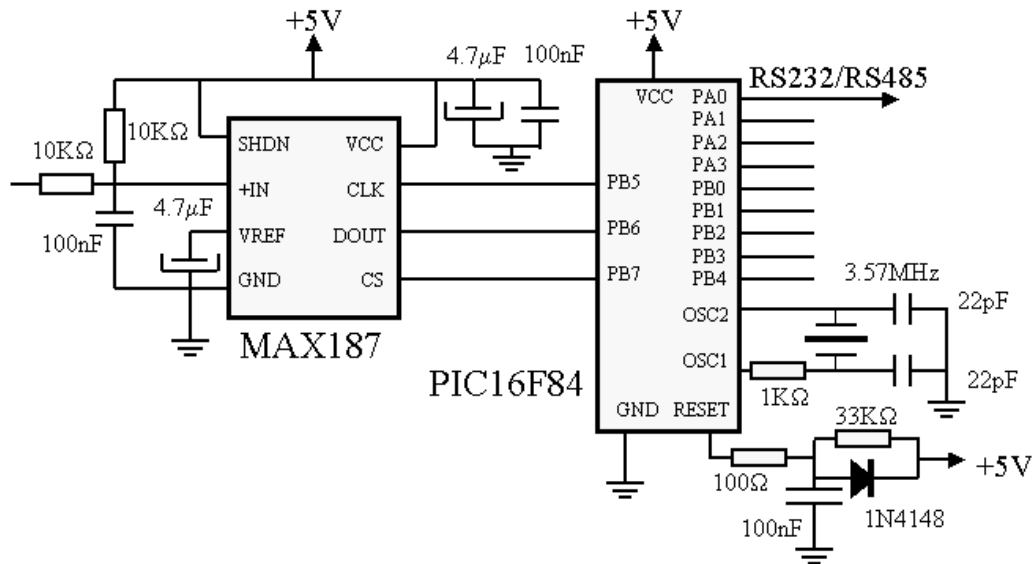


Figura 4-26. Otro conversor de 12 bits de aproximación sucesiva, con interfase serie y bajo consumo. También utiliza la misma fuente de alimentación como tensión de referencia. El microcontrolador debe ser también de bajo consumo (PIC16FL84).

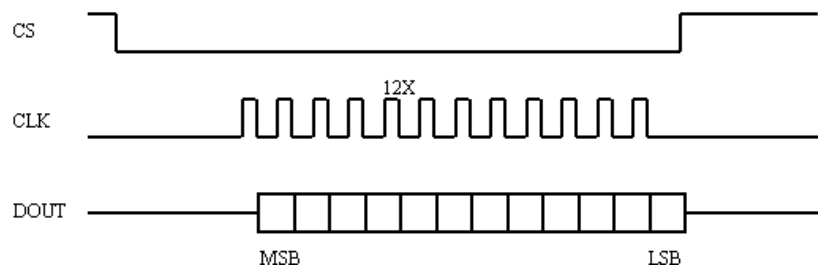


Figura 4-27. La interfase serie es semejante a la utilizada por el MAX187, basta con seleccionar el conversor (CS) y enviar catorce pulsos de reloj.

Convertor de 8 canales y 12 bits.

En muchos casos se precisa medir varias señales sin requerir una precisión excesiva, bien porque se trate de señales con mala relación señal ruido o por la propia naturaleza de las señales como es el caso de una estación meteorológica. En estas circunstancias resulta conveniente utilizar un convertor analógico digital con varias entradas. Comercialmente se encuentran muchos convertores de este tipo, siendo generalmente de 8 canales de entrada como es el MAX186 que tiene una resolución de 12 bits. Este convertor es de muy sencilla operación y permite solucionar fácilmente las situaciones donde se requiere medir muchas señales.

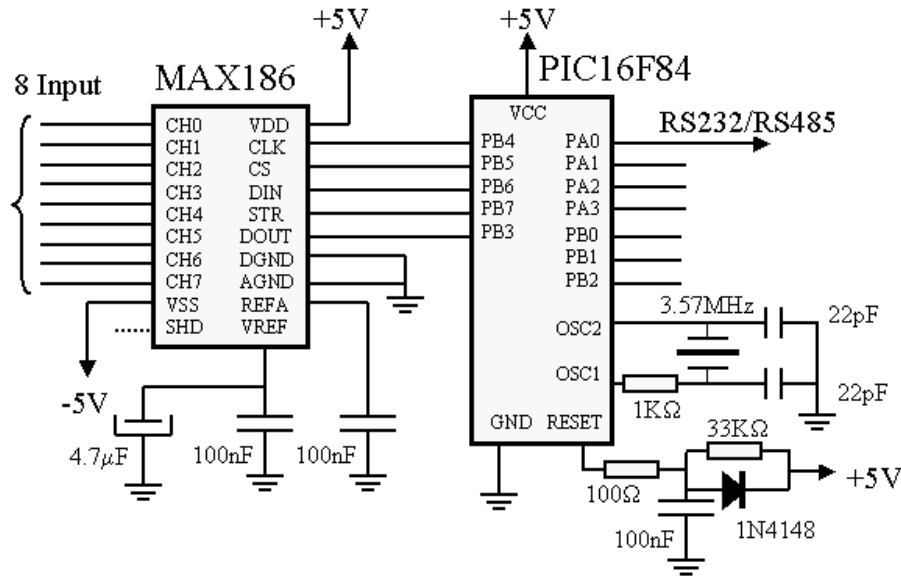


Figura 4-28. Se utiliza un convertor de aproximación sucesiva, de 12 bits y 8 canales de entrada (MAX186) controlado por un microcontrolador PIC16F84. De esta forma se obtiene una salida serie estándar que podemos conectar directamente al puerto RS232 del computador.

La configuración propuesta se basa en la utilización de un convertor analógico digital de 12 bits y 8 canales de entrada (MAX186), capaz de operar a 100KHz de frecuencia de muestreo y referencia de tensión incorporada internamente. Es necesario disponer un sistema de filtros *antia-aliasing* en cada una de las entradas para limitar la señal por debajo de la mitad de la frecuencia de muestreo. Se utiliza un microcontrolador PIC16F84 para proporcionar todas las señales de control y obtener una salida directamente compatible con el computador.

Como en todos los convertores de alta resolución es necesario que la masa analógica y la masa digital tengan sólo un punto en común y que este sea precisamente el pin correspondiente a la masa analógica del convertor. Igualmente es importante incluir condensadores de desacoplo en las alimentaciones de todos los chips. Esto es especialmente importante en las alimentaciones del convertor. Estos condensadores deben disponerse lo más próximos posible al correspondiente pin. También debe desacoplarse la tensión de referencia. Finalmente, los conductores correspondientes a la señal de entrada no deben circular cerca de señales digitales. Una fuente de ruido importante es la generada en los propios circuitos integrados del filtro antialiasing debido a que el ruido aumenta considerablemente con la frecuencia y el convertor utiliza un tiempo muy corto para adquirir el dato, por lo que no tiene tiempo de cancelar el ruido al promediar la señal. Para corregir este efecto debe utilizarse un pequeño filtro

RC dispuesto lo más cerca posible del correspondiente pin de entrada. Los valores recomendados para operar con frecuencias de reloj del orden de 1MHz son 220Ω y 10nF. El filtro antialiasing deberá diseñarse para una frecuencia de corte del orden de un tercio o un cuarto de la frecuencia de muestreo. En general, cuando se trata de adaptar una red analógica ya existente al registro digital, suelen servir el mismo sistema de filtros, debiendo sólo incluir un pequeño atenuador o amplificador para adaptar el nivel de señal al requerido por el conversor, tratando de optimizar la relación señal ruido. Es importante cuidar en sistemas FM que la subportadora de audio quede suficientemente atenuada por el filtro, ya que al tener mucha mayor sensibilidad el sistema digital, estas señales parásitas pueden llevar a inhibir la correcta operación del algoritmo de detección.

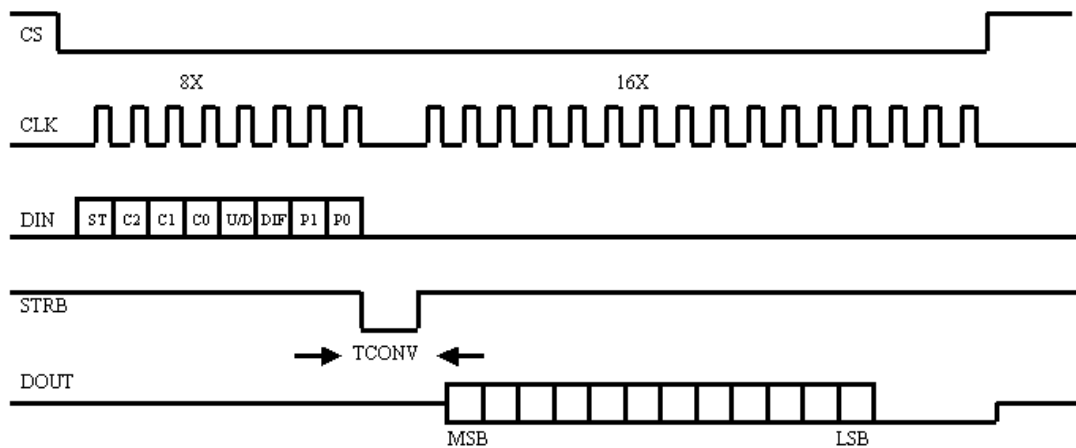


Figura 4-29. El MAX186 posee una línea serie de entrada de comandos y otra de salida de datos, funcionando ambas bajo el mismo reloj. Primero el microcontrolador envía una ráfaga de 8 bits con un bit de sincronismo, el número del canal que se desea medir, la polaridad de la señal (unipolar o bipolar) y si se desea una configuración de entradas diferencia o referida a masa. Seguidamente el conversor realiza la medida (STRB a nivel bajo) y al finalizar (STRB sube a nivel alto) se pueden leer el dato en una ráfaga de 16 bits. Al finalizar la medida de todos los canales, el microcontrolador ordena los datos de acuerdo al formato deseado (binario, ASCII) incorporándole la cabecera necesaria para su identificación y los transmite al computador en serie a 9600 baudios.

Fuentes de alimentación

Los conversores analógico digitales y sus elementos de control requieren 5V (3 V en las realizaciones de muy bajo consumo) para la alimentación digital y en muchos casos una tensión negativa -5V. Dado que los límites de la señal de entrada al conversor son +5V, es recomendable alimentar toda la sección analógica con +8V y -8V para evitar que los operacionales corten las excursiones de la señal en las proximidades de 5V. Unos simples diodos protegen las entradas del conversor en caso de excursiones excesivas. El bajo consumo permite utilizar una pequeña fuente conmutada doble para obtener todas las tensiones necesarias a partir de la propia alimentación del sistema de Pc. También pueden utilizarse fuentes conmutadas integradas del tipo MAX638 para la rama positiva y un ICL7660 (o un MAX660) para la rama negativa del conversor. Además deberemos disponer de las tensiones de alimentación para el sistema de filtros, que pueden obtenerse también mediante fuentes conmutadas o reguladores serie convencionales, en función de lo crítico que sea disminuir el requerimiento energético. En aquellos casos donde ya existe una alimentación simétrica a 12V o 15V puede utilizarse directamente

con sólo incluir un reguladores serie positivo (LM7805) y negativo (LM7905) para el conversor, y adoptar las precauciones elementales de desacoplar cuidadosamente mediante condensadores las líneas de alimentación en su entrada a la unidad de adquisición.

La fuentes de alimentación se deben diseñar de forma que todo el equipo funcione con una alimentación única de 12V. Es importante que acepte un amplio margen de funcionamiento, pues cuando se utilizan baterías con generadores eólicos o paneles solares las tensiones pueden variar entre 9V y 17 V. El consumo de todo el sistema suele ser inferior a 20mA, mucho menor que el requerido por el Pc. Hay que cuidar especialmente las conexiones al sistema central de alimentación, ya que no es raro que las diferentes fuentes conmutadas generen un ruido excesivo para la resolución del conversor: la solución radical es utilizar fuentes independientes, si ello no es posible deben utilizarse cables gruesos e independientes, especialmente para la masa, para las distintas secciones (telemetría, registradores gráficos, filtros, conversor y ordenador. Cuidar también las conexiones a la batería y de esta al sistema de carga. En bastante frecuente que un equipo opere satisfactoriamente con baterías y plantee problemas al conectar el cargador. Controlar que el cargador sin batería no suministre una tensión superior a la máxima que puedan soportar los distintos circuitos. Una desconexión errónea de la batería, estando conectado el cargador, puede destruir todo el equipo. Es conveniente disponer a la entrada de un diodo serie que permita el paso de varios amperios y que sirva de protección ante la conexión invertida de la batería. Esta es la causa principal de la destrucción de los instrumentos utilizados en geofísica. En casos de que se desee trabajar con niveles muy bajos de tensión de la batería (menos de 10V) se puede disponer de una protección alternativa en base a un fusible y un diodo en paralelo. Este diodo debe disponerse en la rama positiva, ya que en caso contrario tendríamos una caída de potencial del orden de 0.7V con respecto a masa.

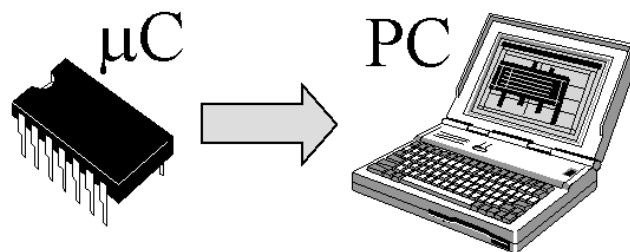


Figura 4-30. En un sistema de adquisición de datos, el conversor analógico digital estará controlado por un microcontrolador (μC) que transmitirá la información a un computador (PC). Habrá que desarrollar el software para ambos sistemas. Los programas para el sistema de microcontroladores se construyen en ensamblador utilizando las herramientas propias de cada dispositivo. Por el contrario, los programas para los computadores se escriben utilizando lenguajes de alto nivel (C, BASIC, JAVA, etc.) compilándolos para el correspondiente sistema operativo.

Software para el control de conversores analógico digital

La realización de un conjunto instrumental para el seguimiento de la actividad volcánica implica conjugar múltiples elementos, tales como sensores, condicionadores de señal, conversores analógico digitales, enlaces telemétricos, computadores para adquisición de datos y para su análisis. Este sistema estará soportado por un conjunto de elementos físicos (hardware) y una serie de aplicaciones informáticas (software) que deberán

operar conjuntamente. En general, estas aplicaciones deberemos desarrollarlas, pues difícilmente se van a poder encontrar paquetes comerciales para el análisis de las señales propias de la vigilancia de volcanes. Otro aspecto importante es que habitualmente el sistema estará constituido por varias plataformas, desde módulos electrónicos con microcontroladores a redes formadas por computadores trabajando bajo distintos sistemas operativos. Esto implica que el software correspondiente a cada uno de estos elementos se deberá desarrollar utilizando distintos lenguajes y herramientas. Hay que tener presente al diseñar el software que en cualquier momento puede ser necesario modificarlo, integrando nuevos componentes o eliminando ciertas funciones. También deberá tenerse presente que el sistema pueda ser operado con personal sin especial formación, por ejemplo durante una crisis, por lo que las interfases de usuario deben ser fácilmente asimilables por el operador.

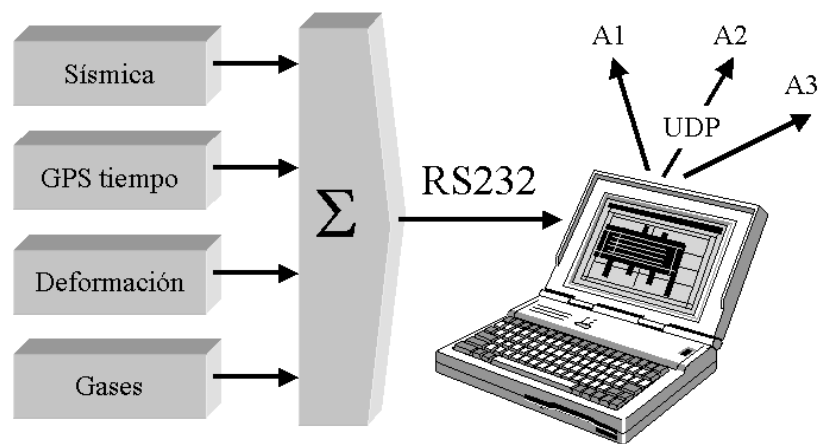


Figura 4-31. En un sistema de adquisición de datos llegan datos procedentes de varios sensores, con secuencias de medida distintas. La figura representa un sistema típico en seguimiento de la actividad volcánica, donde se registra la actividad sísmica, la deformación y la emisión de gases. Los datos llegan al computador por una única línea de telemetría, el procesador identifica cada ráfaga y la envía a la correspondiente aplicación (A1,A2,A3) mediante mensajes bajo protocolos UDP o TCP/IP.

Sistemas multiparamétricos. Formato de los datos

En un sistema de seguimiento de la actividad volcánica se integran datos de muy distinta índole, como es sísmica, deformación, gases, temperaturas, campo magnético, etc. Por ello al sistema de adquisición le llegan datos de varios tipos, con secuencias distintas, utilizando puertos distintos o compartiendo el mismo puerto y enlace de telemetría. En estos casos es conveniente que cada dato sea independiente, pues de esta forma, el fallo de un sistema no supone ningún problema para el resto. Una solución es dotar a cada dato de una cabecera que permita su fácil identificación. Para facilitar la decodificación y evitar errores, se ha optado por una cabecera de dos bytes, el primero permite identificar el inicio de la secuencia, el segundo el tipo de dato. Todos los mensajes terminan con un byte de fin de secuencia. De esta forma se reducen las posibilidades de error sin sobrecargar el sistema y sin necesidad de utilizar enlaces bidireccionales. Para evitar las pérdidas de datos por interferencias, especialmente frecuentes en telemetría radio, se transmite cada dato independientemente, de esta forma en caso de interferencias sólo perdemos un dato y no un paquete. Si se utilizan enlaces

bidireccionales de alta velocidad puede ser conveniente cambiar este modelo y pasar a utilizar paquetes de datos con estructuras más complejas. Sin embargo, los enlaces situados sobre el volcán son en general muy sencillos y de carácter unidireccional. El formato para la transmisión de los datos se ha optimizado para que sea fácil su implementación en microcontroladores con muy poca memoria y que operan a frecuencias lo más bajas posibles para disminuir el consumo energético. Para la cabecera (hexadecimal AA) y el fin de mensaje (hexadecimal 3C) se han elegido caracteres cuya representación en binario sea fácilmente identificable por el microcontrolador mediante un análisis bit a bit de las secuencias de entrada. El carácter para la identificación del canal puede ser cualquiera, disponiéndose así de 256 posibilidades. En algunos casos es necesario mejorar la inmunidad frente a interferencias que pueden provocar el fallo de algún bit, por ejemplo cuando se utiliza una telemetría radio, para evitarlo es suficiente incluir uno o dos bytes para el control de la paridad, esta información se inserta antes del carácter de fin de secuencia.

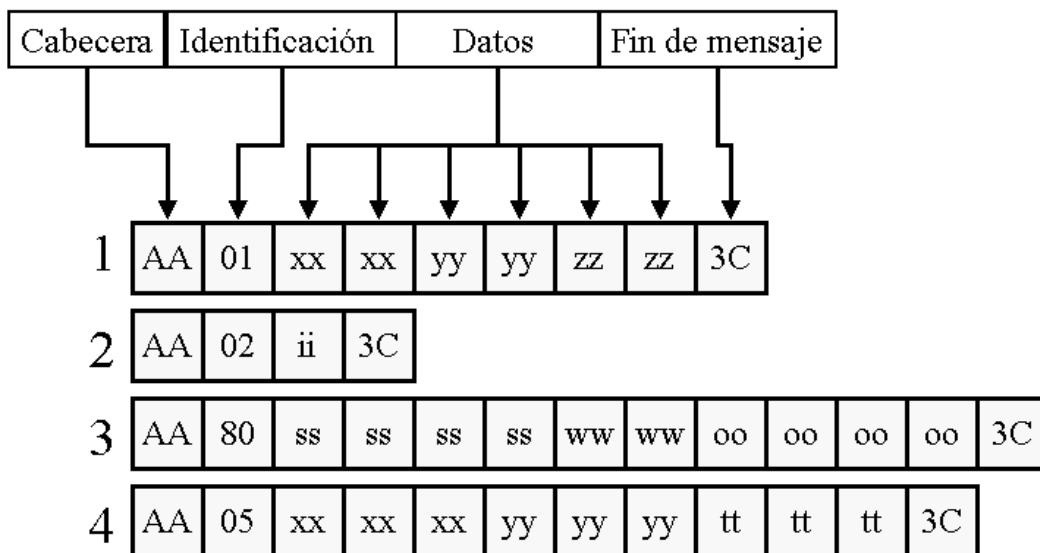


Figura 4-32. Ejemplo de formato de datos en la que cada secuencia binaria posee la misma estructura: cabecera, identificación, datos y fin de mensaje. La figura muestra los formatos para sísmica de tres canales y 16 bits (1), para información sobre el instrumentos (2), para mensajes de tiempo GPS (3) y para deformación con dos componentes de inclinación y la temperatura en 24 bits (4). El código hexadecimal AA (10101010) corresponde a la cabecera, 3C (00111100) a fin de mensaje, 01, 02, 80, 05 son identificativos de dato.

Procesado de los datos

Una vez los datos llegan al computador se pueden estructurar en paquetes, que se envían a las distintas aplicaciones para su procesamiento y análisis. Estas aplicaciones pueden estar soportadas en el mismo computador, en computadores distintos enlazados por una red local o en sistemas remotos a los que se accede mediante Internet. En los dos primeros casos podemos utilizar protocolos sencillos como son los mensajes UDP, en caso de acceso a través de Internet se requiere un protocolo TCP/IP o derivados o utilizar mensajería tipo correo electrónico.

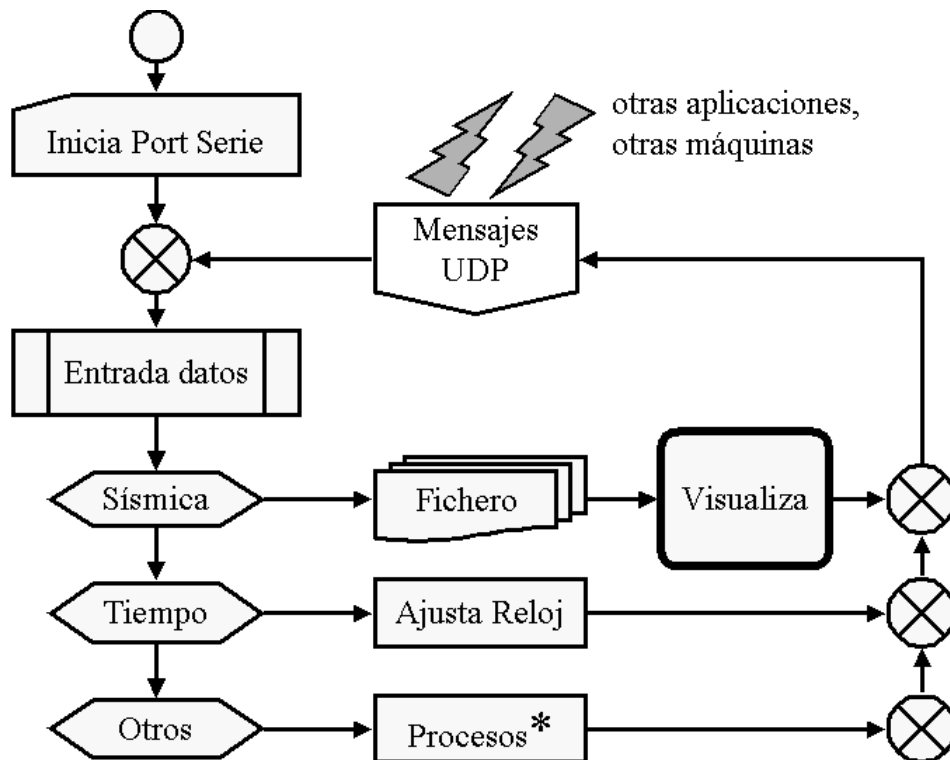


Figura 4-33. Diagrama bloque de un sistema de adquisición de datos multiparamétrico. La figura corresponde a un sistema de registro sísmico que comparte el enlace telemétrico con deformación, gases y un receptor GPS para sincronismo horario.

Es importante cuidar la respuesta del sistema frente a un error en el tráfico de mensajes. Debe evitarse que el fallo de un mensaje, por ejemplo la interrupción en la llegada de algún tipo de datos, provoque el colapso del sistema. Este tipo de errores es muy frecuente cuando se utilizan protocolos TCP/IP cuando una de las máquinas no está disponible. Además, deberemos contar con sistemas alternativos, por ejemplo los distintos computadores deben poseer capacidad de almacenaje suficiente para varios días, de forma que ante cualquier rotura del servicio de mensajería, los datos no se pierdan y puedan recuperarse por otros métodos. Es importante que la arquitectura del sistema se pueda modificar fácilmente, desde la estructura más sencilla y a la vez más cuidada, utilizada para la vigilancia en periodos de actividad estacionaria del volcán, a la necesaria para el seguimiento de una crisis volcánica donde la instrumentación varía en función de la actividad del volcán y de los medios disponibles en cada momento. Por ejemplo, en la atención de una crisis es frecuente que se vayan incorporando al sistema de vigilancia equipos procedentes de otros organismos e instituciones, nacionales o extranjeras, y cada uno de ellos con sus propias peculiaridades. Si el sistema se diseña teniendo esto presente, después es muy fácil adaptarlo a las distintas circunstancias. De otro modo es imposible hacerlo durante el desarrollo de la crisis y la integración de los distintos datos deberá hacerse manualmente en el mejor de los casos. En general, la sísmica será la aplicación más crítica, pues es la que maneja mayor volumen de datos y velocidades de adquisición más elevadas. Por ello suele implementarse en los computadores más próximos al sistema de entrada. Las otras aplicaciones suelen instalarse en otras máquinas con acceso más lento. También es frecuente que el análisis de los datos de sísmica requieran utilizar otra máquina para no sobrecargar o perturbar al sistema de adquisición.

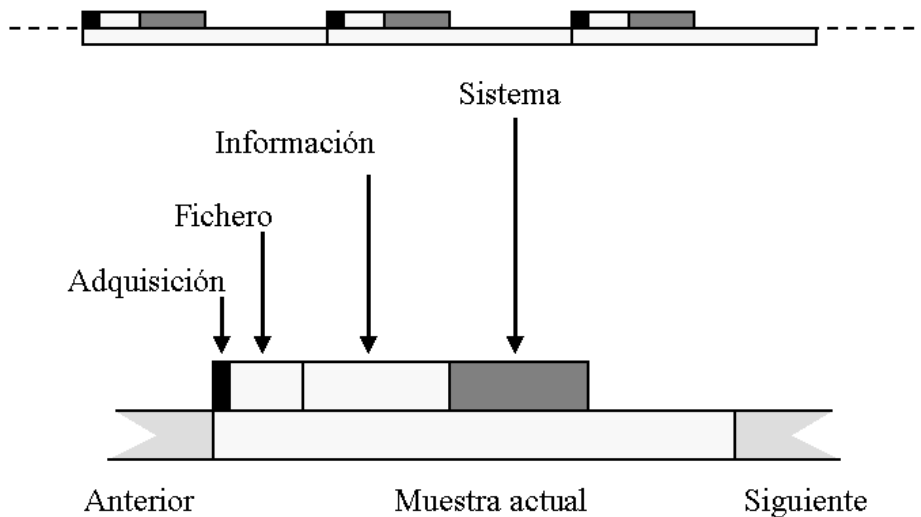


Figura 4-34. En sistemas de adquisición de datos hay que tener siempre presente que los datos llegan de forma continua y no es posible interrumpir la secuencia. El tiempo necesario para su procesado debe ser mucho menor que el transcurrido entre dos muestras sucesivas. El tiempo de procesado incluye la adquisición, la transferencia a ficheros, la información al usuario y las necesidades propias del sistema. Este último es muy importante y difícilmente controlable en sistemas operativos como el WINDOWS.

Sistemas operativos

Se hace necesario conjugar dos realidades, por una parte necesitamos un sistema estable, rápido y que opere en tiempo real, en el que sea fácil desarrollar aplicaciones o modificar las existentes. Pero también que este sistema sea fácilmente manejable por los usuarios, que en caso de crisis puede ser que no todos ellos tengan la formación adecuada para manejar computadores con sistemas específicos. También tenemos que considerar la necesidad de que los sistemas aislados sean de muy bajo consumo y que puedan operarse al aire libre. Desde un punto de vista exclusivamente técnico elegiríamos una máquina UNIX (o uno de sus derivados como LINUX) pero que la mayoría del personal con el que podemos contar como operadores desconocen, en este sentido es preferible utilizar un sistema como el WINDOWS y derivados. Para los sistemas de bajo consumo deberemos utilizar máquinas con sistemas operativos propietarios como el EPOC32. La mejor solución es utilizar sistemas mixtos, donde las partes críticas operen bajo UNIX y reservando el sistema WINDOWS para los computadores de interfase con el usuario no especializado. Ténganse presente que el sistema WINDOWS está especialmente concebido para aplicaciones de ofimática y ocio, siendo difícil su utilización en sistemas de tiempo real, al no permitir la interacción directa con el hardware, al menos de forma sencilla. Sin embargo, no se puede ignorar que todos los usuarios de computadores están familiarizados con su uso.

Sistemas de microcontrolador

Se presentan, a modo de ejemplo, las funciones básicas para el control de convertidores analógico digitales de diversas tecnologías. En todos los casos se utiliza un microcontrolador PIC16F84 que genera todas las señales necesarias para la correcta operación del convertidor y transfiere los datos en serie de acuerdo con un formato RS232

estándar. Los programas que se presentan en este capítulo están escritos utilizando el nemotécnico propio del lenguaje ensamblador del PIC16F84. Es importante utilizar en todos los programas de adquisición un sistema de reinicio automático (*watchdog*) para evitar que el sistema se quede colgado. Este tipo de dispositivos utilizan un temporizador independiente que debe ser puesto a cero periódicamente, en cuanto se produce el desbordamiento del contador reinicia el sistema. El microcontrolador PIC16F84 lleva incorporada esta función y es necesario sólo incluir la instrucción *clrwdt* después de cada ciclo de medida.

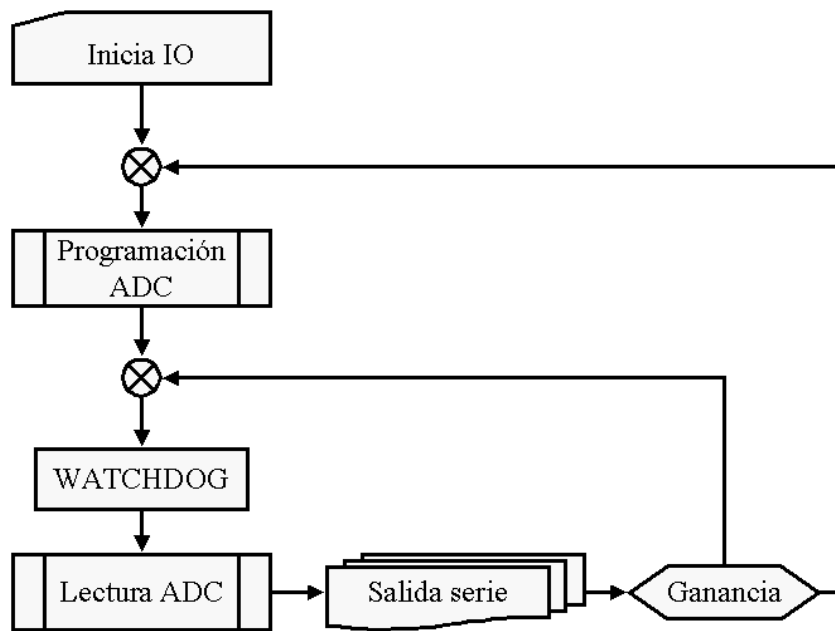


Figura 4-35. Diagrama bloque del programa de control de un convertidor analógico digital de tecnología delta sigma. En primer lugar hay que iniciar las líneas de entrada salida (Inicia IO), seguidamente programar el registro de control del convertidor (programación ADC). El ciclo de medida reinicia el WATCHDOG cada vez que va a tomar un dato, lee el registro de datos del convertidor (lectura ADC), transmite los datos (salida serie) y comprueba si hay algún cambio en los parámetros (ganancia).

Transmisión en serie

En todos los casos se emplea la misma función para la transmisión serie de los datos. El microcontrolador utiliza un cristal de cuarzo de 3.579 MHz que permite obtener con buena precisión los tiempos requeridos para la interfase serie con velocidades de transmisión estándar. Para generar las distintas velocidades (4800, 9600, 19200 y 38400 baudios) se ha escrito un función que produce el retardo necesario mediante un simple contador. En algunas aplicaciones se debe utilizar otra línea para la salida de los datos, pues por necesidades de la realización del circuito impreso, la línea 1 del port B se utiliza para otra función. Igualmente si se utiliza una interfase RS232 como el MAX232 se deberá invertir los niveles de salida (permutar líneas de código `bcf Port_B, DX` y `bsf Port_B, DX`).

Asignación de líneas

```

; PORT SERIE E/S y velocidad
;

```

```

;CLK 3.579
;4800=.58 9600=.26 19200=.11 38400=.3+nop
;
DX equ 1 ;Port_B
BAUD equ .26

```

Salida serie

```

Xmtr bsf Port_B,DX
movwf XmtReg
movlw 08H
movwf Count1
nop
nop
nop
nop
nop
X_next call Delay
rrf XmtReg,Same
btfss STATUS,CARRY
goto X_1
bcf Port_B,DX
goto X_2
X_1 bsf Port_B,DX
X_2 decfsz Count1,Same
goto X_next
call Delay
bcf Port_B,DX
call Delay
call Delay
retlw 0

```

Retardo

```

Delay movlw BAUD
movwf DlyCnt
red1 decfsz DlyCnt,Same
goto red1
retlw 0

```

Programa de control del convertor AD7710 mediante un microcontrolador PIC16F84

Se presenta las funciones para el control de un convertor AD7710 mediante un microcontrolador PIC16F. Se utiliza un cristal de cuarzo de 9.8304 para poder obtener valores enteros para los divisores correspondientes a las frecuencias de muestreo de 200 Hz (0060H), 100 Hz (00C0H), 50 Hz (0180H) y 25 Hz (0300H), pudiéndose seleccionar entre dos de ellas (bit 3 de GAIN). El convertor funciona en modo continuo, leyéndose el registro de salida cada vez que finaliza una conversión. Los datos se encuentran en las memorias ADC1, ADC2. La amplificación interna del convertor (bits 0, 1, 2) está en la memoria GAIN.

Asignación de líneas

```

DAT equ 0 ;Port_B
DRDY equ 1 ;Port_B
RFS equ 2 ;Port_B
TFS equ 3 ;Port_B
A0 equ 4 ;Port_B

```

```
CLK equ 5 ;Port_B
```

Adquisición

```
adc bcf Port_B,CLK
movlw 10H
movwf Count1
adc0 btfsc Port_B,DRDY
goto adc0
bsf Port_B,A0
bcf Port_B,RFS
nop
adc1 nop
bcf STATUS,CARRY
rlf ADC1,Same
rlf ADC2,Same
btfsc Port_B,DAT
bsf ADC1,0
bsf Port_B,CLK
nop
bcf Port_B,CLK
nop
decfsz Count1,Same
goto adc1
nop
bsf Port_B,RFS
movlw 001H
addwf ADC1,Same
btfsc STATUS,CARRY
incf ADC2,Same
movlw 080H
addwf ADC2,Same
adc2 btfss Port_B,DRDY
goto adc2
retlw 0
```

Programación

```
prog bsf GAIN,3
clrf ADP1
clrf ADP2
clrf ADP3
movf GAIN,W
andlw 00FH
movwf ADP1
bcf STATUS,CARRY
rlf ADP1,Same
rlf ADP1,Same
movlw 001H ; código muestreo
movwf ADP2
movlw 086H ;código muestreo
movwf ADP3
btfss GAIN,4
goto prog2
bcf STATUS,CARRY
movlw 003H
movwf ADP2
movlw 00DH
movwf ADP3
prog2 movlw 18H
movwf Count1
bsf STATUS,RP0
```

```

movlw 0C2H
movwf TRISB
bcf STATUS,RP0
call Reta
bcf Port_B,CLK
bcf Port_B,TFS
bsf Port_B,RFS
bcf Port_B,A0
call Reta
prog1 rlf ADP3,Same
rlf ADP2,Same
rlf ADP1,Same
btfss STATUS,CARRY
bcf Port_B,DAT
btfsc STATUS,CARRY
bsf Port_B,DAT
call Reta
bsf Port_B,CLK
call Reta
bcf Port_B,CLK
call Reta
decfsz Count1,Same
goto prog1
bsf Port_B,DAT
call Reta
bsf STATUS,RP0
movlw 0C3H
movwf TRISB
bcf STATUS,RP0
call Reta
bsf Port_B,TFS
bsf Port_B,RFS
bsf Port_B,A0
bcf GAIN,3
retlw 0

```

Retardo

```

Reta movlw RETADC
movwf DlyCnt
red1 decfsz DlyCnt,Same
goto red1
retlw 0

```

Programa de control del conversor ADS1210 mediante un microcontrolador PIC16F84

Se presenta las funciones para el control de un conversor ADS1210 mediante un microcontrolador PIC16F. Se utiliza un cristal de cuarzo de 9.8304 para poder obtener valores enteros para los divisores correspondientes a las frecuencias de muestreo de 100 Hz y 50 Hz (0180H). El conversor funciona en modo continuo, leyéndose cada vez que finaliza una conversión. Los datos se encuentran en las memorias ADC1, ADC2. La amplificación interna del conversor (bits 0, 1, 2) está en la memoria GAIN. El divisor para 50 muestras por segundo con un cuarzo de 9.8304 MHz es 0186H. El conversor opera con 24 bits de resolución, dejando al programa de explotación la reducción al número de bits deseado. En la función de lectura del conversor (*adc*) se envía primero un comando de lectura y tras una espera equivalente a la transmisión de un byte se leen los tres bytes de datos. Para la programación se envía un comando de programación y tras la espera se envían los cuatro bytes con los parámetros de operación.

Asignación de líneas

```
SDOUT equ 0 ;Port_A I
DRDY equ 1 ;Port_A I
SDIO equ 2 ;Port_A O/I
SCLK equ 3 ;Port_A O
```

Adquisición

```
Adc btfsc Port_A,DRDY
    goto adc
    movlw 0C0H
movwf ADP1
movlw 008H
movwf Count1
padc1 rlf ADP1,Same
btfss STATUS,CARRY
bcf Port_A,SDIO
btfsc STATUS,CARRY
bsf Port_A,SDIO
bsf Port_A,SCLK
call Reta
bcf Port_A,SCLK
call Reta
decfsz Count1,Same
goto padc1
movlw 008H
movwf Count1
padc2 call Reta
call Reta
decfsz Count1,Same
goto padc2
movlw 018H
movwf Count1
adc1 bcf STATUS,CARRY
rlf ADC1,Same
rlf ADC2,Same
rlf ADC3,Same
bsf Port_A,SCLK
call Reta
btfsc Port_A,SDOUT
bsf ADC1,0
bcf Port_A,SCLK
call Reta
decfsz Count1,Same
goto adc1
retlw 0
```

Programación

```
prog btfss Port_A,DRDY
goto prog
prog1 btfsc Port_A,DRDY
goto prog1
movlw 064H
movwf ADP1
movlw 0C2H
movwf ADP2
movf GAIN,W
andlw 007H
movwf ADP3
bcf STATUS,CARRY
```

```

rif    ADP3,Same
rif    ADP3,Same
movlw  001H    ; constante
movwf  ADP4
movlw  07FH    ;constante
movwf  ADP5
movlw  008H
movwf  Count1
movlw  002H
movwf  Count2
prog2  rif     ADP5,Same
rif    ADP4,Same
rif    ADP3,Same
rif    ADP2,Same
rif    ADP1,Same
btfss  STATUS,CARRY
bcf    Port_A,SDIO
btfsc  STATUS,CARRY
bsf    Port_A,SDIO
bsf    Port_A,SCLK
call   Reta
bcf    Port_A,SCLK
call   Reta
decfsz Count1,Same
goto   prog2
movlw  008H
movwf  Count1
prog3  call   Reta
call   Reta
decfsz Count1,Same
goto   prog3
movlw  020H
movwf  Count1
decfsz Count2,Same
goto   prog2
retlw  0

```

Retardo

```

Reta  movlw 008H
movwf Count3
Reta1 nop
decfsz Count3,Same
goto  Reta1
retlw 0

```

Programa de control del conversor MAX186 mediante un microcontrolador PIC16F84

Se presenta las funciones para el control de un conversor MAX186 mediante un microcontrolador PIC16F. Se trata de un conversor de 12 bits de resolución, 8 canales de entrada y salida serie. Este conversor es muy útil para equipar pequeños sistemas multiparamétricos, como puede ser una estación meteorológica, donde no se requiere una elevada resolución. Como este tipo de conversores son muy rápidos, los datos no se almacenan en el microcontrolador, sino que se transmiten inmediatamente al sistema de adquisición de datos.

Asignación de líneas

```

DOUT equ 3 ;Port_B

```

```

SCLK equ 4 ;Port_B
CS equ 5 ;Port_B
DIN equ 6 ;Port_B
SSTRB equ 7 ;Port_B

```

Adquisición

```

adc bcf Port_B,CS
    bsf Port_B,DIN
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    btfsc Canal,0
    bsf Port_B,DIN
    btfss Canal,0
    bcf Port_B,DIN
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    btfsc Canal,2
    bsf Port_B,DIN
    btfss Canal,2
    bcf Port_B,DIN
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    btfsc Canal,1
    bsf Port_B,DIN
    btfss Canal,1
    bcf Port_B,DIN
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    bcf Port_B,DIN ;bipolar
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    bsf Port_B,DIN ;entrada single
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    bsf Port_B,DIN ; internal clock
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    bcf Port_B,DIN ;internal clock
    bsf Port_B,SCLK
    bcf Port_B,SCLK
    bcf Port_B,DIN
movlw 010H
movwf Contar
    clrf ADCL
    clrf ADCH
adc2 btfss Port_B,SSTRB
    goto adc2
adc3 bsf Port_B,SCLK
    bcf Port_B,SCLK
    bcf STATUS,CARRY
    rlf ADCL,Same
    rlf ADCH,Same
    btfsc Port_B,DOOUT
    bsf ADCL,LSB
    decfsz Contar,Same
    goto adc3
    bsf Port_B,CS
    retlw 0

```

Software para computadores PC

El software para el sistema de computadores depende lógicamente de cada uno de los tipos de datos que lleguen al computador y lo que se quiera hacer con ellos. Sin embargo, hay una parte común que es la entrada de los datos en el sistema. En general, esa función se realiza a través de una puerta serie, aunque ya en sistemas más grandes pueda hacerse mediante un puerto USB, utilizando una interfase serie RS232 a USB. También pueden utilizarse otro tipo de interfases, sin embargo la interfase serie es la más sencilla de implementar en el hardware externo, especialmente en los sistemas de telemetría radio o cable. Una vez los datos han entrado en el computador allí son procesados por la misma aplicación que gestiona la entrada o remitidos como mensajes UDP o TCP/IP a otras aplicaciones. El problema más común es como gestionar adecuadamente el puerto serie, pues este tipo de función no está suficientemente documentada en la mayor parte de libros de programación, ya que la tendencia actual del desarrollo computacional es la ofimática y el ocio, donde no se plantea el problema de la adquisición de datos.

Gestión del puerto serie en DOS

En este sistema se dispone del control absoluto del puerto serie. El procedimiento consiste en activar la interrupción del puerto serie con los parámetros de comunicación deseados (identificación del puerto, velocidad, número de bits, bits de parada y paridad) y cada vez que llega un byte se produce una interrupción que transfiere el control a una función para su procesamiento.

```
void interrupt com_int(...)
{
    disable();
    if((inportb(portbase_IIR) & RX_MASK)==RX_ID)
    {
        if(((endbuf + 1) & SBUFSIZ - 1)==startbuf)
            SError = BUFOVFL;
        ccbuf[endbuf++] = inportb(portbase_RXR);
        endbuf &= SBUFSIZ - 1;
    }
    outportb(ICR, EOI);
    enable();
}

int serial_out(char x)
{
    while((inportb(portbase_LSR)&XMTRDY)==0)
        disable();
    outportb(portbase_TXR,x);
    enable();
    return(0);
}

int chkccb(void)
{
    return(abs(endbuf-startbuf));
}

char read_byte(void)
```

```

{
char res;
  disable();
  res = ccbuf[startbuf++];
  startbuf %= SBUFSIZ;
  enable();
  return(res);
}

int initserial(void)
{
char c;
int divisor,setting,pnum;
switch(port)
{
  case COM1:
    portbase = COM1BASE;
    break;
  case COM2:
    portbase = COM2BASE;
    break;
  default :
    return(-1);
}
portbase_LCR = portbase + LCR;
portbase_IIR = portbase + IIR;
portbase_RXR = portbase + RXR;
portbase_MSR = portbase + MSR;
portbase_MCR = portbase + MCR;
portbase_LSR = portbase + LSR;
portbase_TXR = portbase + TXR;
portbase_IER = portbase + IER;
if(speed==0)return(-1);
else divisor = (int)(115200L/speed);
if(portbase==0) return(-1);
disable();
c = inportb(portbase_LCR);
outportb(portbase_LCR, (c | 0x80));
outportb(portbase + DLL,(divisor & 0x00FF));
outportb(portbase + DLH,((divisor >> 8) & 0x00FF));
outportb(portbase_LCR, c);
enable();
if(portbase==0) return(-1);
if(bits < 5 || bits > 8) return(-1);
if(stopbits != 1 && stopbits != 2) return(-1);
if(parity !=NO_PARITY&&parity !=ODD_PARITY&&parity !=EVEN_PARITY)return(-1);
setting = bits-5;
setting |= ((stopbits==1)? 0x00 : 0x04);
setting |= parity;
disable();
outportb(portbase_LCR, setting);
enable();
endbuf = 0;
startbuf = 0;
oldvects[0] = getvect(0x0B);
oldvects[1] = getvect(0x0C);
setvect(0x0B,com_int);
setvect(0x0C,com_int);
pnum = (portbase==COM1BASE ? COM1 : COM2);
disable();

```

```

c = inportb(portbase_MCR) | MC_INT;
outportb(portbase_MCR, c);
outportb(portbase_IER, RX_INT);
c = inportb(IMR) & (pnum==COM1 ? IRQ4 : IRQ3);
outportb(IMR, c);
enable();
c = inportb(portbase_MCR) | DTR | RTS;
outportb(portbase_MCR, c);
return 0;
}

void closeserial(void)
{
int c;
disable();
c = inportb(IMR) | ~IRQ3 | ~IRQ4;
outportb(IMR, c);
outportb(portbase_IER, 0);
c = inportb(portbase_MCR) & ~MC_INT;
outportb(portbase_MCR,c);
enable();
outportb(portbase_MCR,0);
setvect(0x0B,oldvects[0]);
setvect(0x0C,oldvects[1]);
}

```

Gestión del puerto serie en WINDOWS 9x

En este sistema operativo, no se tiene acceso directo al puerto, sino que se abre un fichero con las características del puerto y en ese fichero se van acumulando los datos que llegan al puerto. Hay que ir leyendo los datos de este fichero para evitar que se produzca un desbordamiento. Las funciones siguientes proporcionan un buen control sobre las comunicaciones serie en Windows 9x, deben hacerse cambios para que trabajen en Windows 2000. También pueden utilizarse controladores adaptados a los distintos lenguajes y sistemas de desarrollo de software. Además, los buffer deben ser muy grandes, pues estos sistemas operativos pueden estar varios segundos sin atender una determinada aplicación.

```

HANDLE hComm = NULL;
TRead *ReadThread;
COMMTIMEOUTS ctmoNew = {0}, ctmoOld;

void __fastcall TRead::Execute()
{
    DWORD dwError,dwEvent;
    COMSTAT cs;
    FreeOnTerminate = true;
    SetCommMask(hComm, EV_RXCHAR);
    while(1)
    {
        while(1)
        {
            if(Terminated)return;
            ClearCommError(hComm,&dwError,&cs);
            if(cs.cbInQue)ReadPort();
            else break;
        }
        if(WaitCommEvent(hComm,&dwEvent,NULL))
    }
}

```

```

        {
            if(dwEvent&EV_RXCHAR)ReadPort();
        }
    }
}

__fastcall TRead::TRead(bool CreateSuspended)
    : TThread(CreateSuspended)
{
}

void __fastcall inicia_serie(void)
{
    char inicia_port[32];
    DCB dcbCommPort;
    sprintf(inicia_port,"COM%i",port)

    hComm=CreateFile(inicia_port,GENERIC_READ|GENERIC_WRITE,0,0,OPEN_EXISTING,0,0)
;
    if(hComm == INVALID_HANDLE_VALUE)
    {
        MessageDlg("Problems in Serial PORT, Programm
Abort",mtError,TMsgDlgButtons()<<mbOK,0);
    }
    else
    {
        GetCommTimeouts(hComm,&ctmoOld);
        ctmoNew.ReadIntervalTimeout = MAXDWORD;
        SetCommTimeouts(hComm,&ctmoNew);
        dcbCommPort.DCBlength = sizeof(DCB);
        GetCommState(hComm,&dcbCommPort);
        sprintf(inicia_port,"%i,N,8,1",baudios);
        BuildCommDCB(inicia_port,&dcbCommPort);
        SetCommState(hComm,&dcbCommPort);
        ReadThread = new TRead(false);
    }
}

void __fastcall ReadPort(void)
{
    DWORD dwBytesRead;
    char InBuff[2000];
    ReadFile(hComm,InBuff,999,&dwBytesRead,NULL);
    .....
}

```

Gestión del puerto serie en EPOC32

En este sistema operativo, no se tiene acceso directo al puerto, sino que se abre un fichero con las características del puerto y en ese fichero se van acumulando los datos que llegan al puerto. La lectura se realiza en bloques de la longitud deseada, de esta forma es muy fácil deshacer los paquetes de datos, leyendo primero el byte correspondiente a la cabecera y después a la identificación. Seguidamente se leen de dos en dos los bytes correspondientes a los datos y finalmente el byte de fin de ráfaga.

```

PROC rsset:(hComm%,baud%,parity%,data%,stop%,hand%,term&)
    LOCAL frame%,srchar%(6),dummy%,err%

```

```

frame% = data% -5
IF stop%=2 :frame%=frame% OR 16 :ENDIF
IF parity% :frame%=frame% OR 32 :ENDIF
srchar%(1)=baud% OR (baud%*256)
srchar%(2)=frame% OR (parity%*256)
srchar%(3)=(hand% AND 255) OR $1100
srchar%(4)=$13
REM iniciación de la puerta serie
POKEL ADDR(srchar%(5)),term&
Err%IOW(hComm%,7,srchar%(1),dummy%)
IF err% :RAISE err% :ENDIF
ENDP

REM leer un buffer en puerta serie
bufferlen% = 1
address& = ADDR(buffer$)
ret% = IOREAD(hComm%,UADD(address&,1),bufferlen%)

REM escribir un buffer en puerta serie
IOWRITE(hComm%, UADD(address&,1),bufferlen%)

```

Recomendaciones

Estas funciones están estrechamente ligadas a un desarrollo concreto. Cualquier cambio, por pequeño que sea, implica modificaciones importantes en el software de control, por ello deben considerarse como simples ejemplos. Para evitar tener que estar escribiendo siempre las mismas funciones es importante realizar los diseños procurando respetar al máximo las asignaciones de entrada y salida. Es necesario consultar la información sobre los distintos lenguajes de programación y realizar muchos ensayos hasta poder construir las funciones que realmente se ajustan a nuestras necesidades. En general se encuentran bastante bien documentados los sistemas de programación para microcontroladores, sin embargo es difícil encontrar buenos ejemplos para las aplicaciones de adquisición de datos en computadores operando con WINDOWS.

La transferencia de datos al disco es función del tipo de computador, del sistema operativo y del lenguaje utilizado. Las opciones más frecuentes son escribir los datos uno a uno o en bloques, que por facilidad suelen hacerse con los datos de un segundo o de un minuto, de esta forma es fácil mantener el sincronismo temporal, pues cada actualización del fichero se hace con los datos correspondientes a un segundo exacto. Un método para hacer esto es ir almacenando los datos en una matriz, al momento de transferir los datos al disco se ajusta el número de datos, suprimiendo o interpolando datos. En general, se pierden (o sobran) unos pocos datos por minuto debido a interferencias en telemetría radio o por desajustes en las velocidades de muestreo de los conversores analógico digitales. De esta forma, cada fichero contiene el mismo número de datos, y el interpolar o suprimir un dato no supone ningún problema para los análisis posteriores.