

# Implementación de monitores ECG, mediante $\mu$ Converters

Por Enrique Company Bosch y Eckart Hartmann

De Analog Devices.  
Artículo cedido por  
Arrow Iberia Electrónica

*Este artículo sugiere algunas ideas para una implementación de bajo coste de un monitor ECG. Se propone un circuito que ejecuta conversión A/D, filtrado digital, y amplificación digital usando un Microconverter. Un Chip que integra un ADC, un microcontrolador y memoria Flash.*

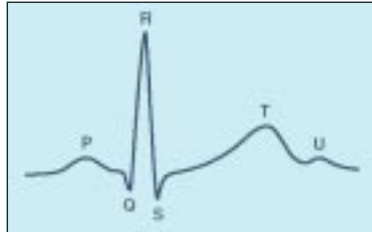


Figura 1. Señal ECG

Un electrocardiograma (ECG) es un registrador de la actividad eléctrica en la superficie corporal generada por el corazón. La información se recoge mediante electrodos situados en localizaciones específicas del cuerpo. La señal ECG se caracteriza por seis picos y valles nombrados con letras sucesivas del alfabeto P, Q, R, S, T y U (ver fig. 1).

## Requisitos de un Electrocardiógrafo

El front end de un ECG debe ser capaz de gestionar señales extremadamente débiles desde 0.5 mV a 5.0 mV, combinadas con un componente DC de hasta  $\pm 300$ mV que resulta del contacto electrodo-piel, además de una componente en modo común de hasta 1.5V, que resulta de la diferencia de potencial entre los electrodos y masa. El ancho de banda útil de una señal ECG,

dependiendo de la aplicación puede oscilar desde 0.5 Hz a 50 Hz, para monitorización en unidades de cuidados intensivos y hasta 1 kHz para detección de marcapasos. Una aplicación ECG estándar tiene un ancho de banda de 0.05 Hz a 100 Hz.

Las señales ECG se pueden degradar por varias clases de ruido:

Interferencias procedentes de la línea de alimentación; Ruido producido en los electrodos de contacto con la piel; Señales EMG producidas por contracciones musculares; Derivas de la línea de referencia (baseline) por cambios en la impedancia piel-electrodo o debidas a la respiración; interferencias electromagnéticas y ruido de alta frecuencia inducido por otros equipos.

Por todo ello es necesario intercalar un filtro apropiado si queremos obtener una señal, lo más fidedigna posible.

## Mejorando la arquitectura básica de un ECG típico

La figura 2 muestra un diagrama de bloques de un electrocardiógrafo típico de un solo canal. Se puede ver que todo el filtrado se hace en el dominio analógico, mientras que el microprocesador, microcontrolador o DSP se usa principalmente para las comunicaciones. De este modo, las propiedades de cómputo de un núcleo digital no están fácilmente disponibles para tratar con la señal en su estado original. Los sofisticados filtros analógicos pueden generar un gran coste al diseño global debido a su poca flexibilidad y el espacio, coste y consumo que requieren.

El *signal chain* se puede simplificar usando un Microconverter ADuC842, el cual permite combinar ADCs, filtros y microprocesador en un único integrado. Entre las ventajas adicionales están la flexibilidad de implementación de filtros y aislamiento en el dominio digital.

El diseño propuesto se muestra en la figura 3.

## Procesado analógico de la señal de entrada

El front end analógico usa la aproximación típica con un amplificador de instrumentación (IA) y un amplificador operacional para el circuito del pie derecho. El IA es el AD620, un amplificador de instrumentación de bajo coste y alta eficacia, con excelentes prestaciones en continua: CMR  $\gg 100$  dB hasta 1KHz, máxima tensión de offset de 50mV, baja corriente de polarización (1 nA máx), y bajo ruido en tensión (0.28 mV desde 0.1 Hz a 10 Hz).

El AD620 requiere una única resistencia externa  $R_g$  para establecer la ganancia, la ecuación es:

$$[Gain = 1 + 49.4 \text{ k ohms}/R_g]$$

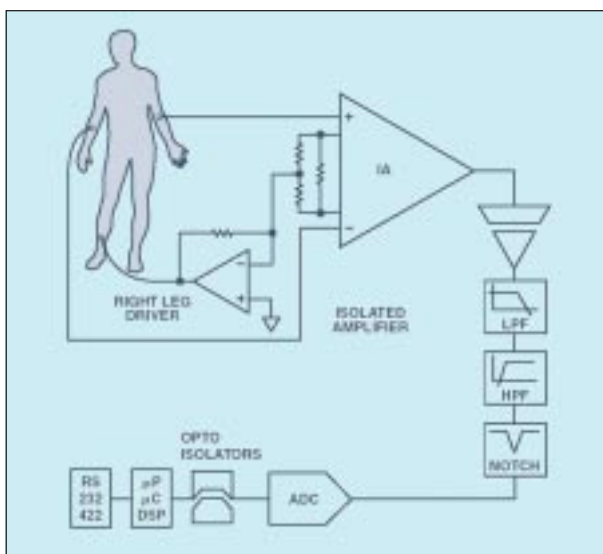
Para evitar la saturación de salida, la ganancia utilizada viene limitada por el *swing* de salida y la máxima tensión de entrada en el IA.

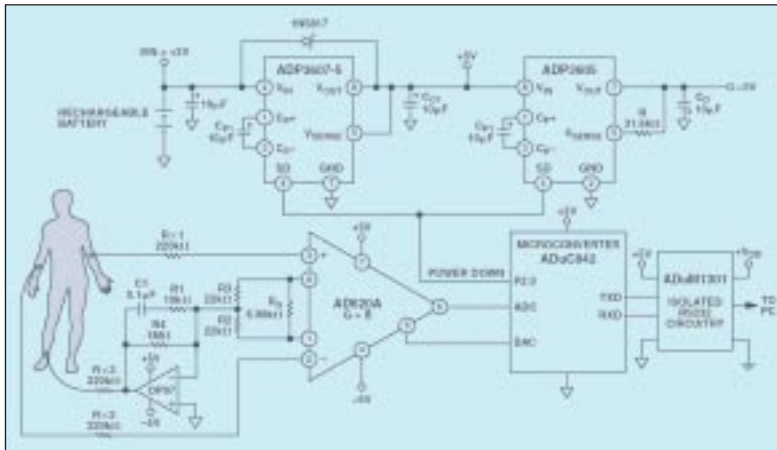
Si alimentamos a  $\pm 5$  V, el *swing* de salida del AD620 es de  $\pm 3.8$ V; la entrada máxima es  $\pm 5$ mV además de un offset DC variable de hasta  $\pm 300$ mV, permitiendo una ganancia máxima de 12.45. Escogemos una ganancia menor de 8 ( $\pm 1\%$ ), con una  $R_g = 6,980 \text{ k}\Omega$ .

El operacional usado en el circuito del pie derecho es el OP97, un amplificador operacional de alta precisión y bajo consumo con un extremadamente alto rechazo al modo común (114 dB mínimo). Este circuito invierte, en la pierna derecha, las interferencias en modo común con el objeto de cancelarlas.

El operacional tiene una ganancia en tensión de 90,9, es decir:

Figura 2. Electrocardiógrafo típico de un solo canal





$R4/R2 \parallel R3 = 1 \text{ M}\Omega/11 \text{ k}\Omega$ ,  
 con filtrado paso bajo con frecuencia de corte de 150 Hz para lograr estabilidad:

$$f_{-3 \text{ dB}} = 1/(2\pi \times 10 \text{ k}\Omega \times 0.1 \mu\text{F})$$

### Aislamiento digital

Es necesario un aislamiento digital entre el interface RS232 y el PC. El circuito de aislamiento es el ADuM1301, un circuito de aislamiento bi-direccional basado en la tecnología iCoupler de Analog Devices, una tecnología que acaba con las dificultades encontradas normalmente en los diseños con optoacopladores (funciones de transferencia no lineales, etc.) También logra mayor velocidad de transmisión con menor consumo que con optoacopladores.

El ADuM1301 tiene tres canales de aislamiento independientes, usándose en este caso dos de ellos, uno para transmitir y otro para recibir datos (una capacidad adicional del ADuM1301, no necesitada aquí, es la capacidad para habilitar/deshabilitar la entrada/salida de datos). La fuente de alimentación para el lado de medición del ADuM1301 se obtiene del regulador/elevador ADP3607-5, que proporciona una salida fija de 5 V positivos.

La alimentación del lado del PC está totalmente aislada del circuito. Se obtiene del PC (como en este caso) o de otra fuente diferente.

### Circuito de alimentación

La alimentación aislada se suministra mediante una batería recargable. Para soportar una señal de entrada bipolar, necesitamos una fuente dual para el AD620 y el OP97. El regulador/elevador ADP3607-5 y el inversor ADP3605 proporcionan una alimentación dual regulada que suministra tensiones reguladas positivas y negativas desde una única batería de 3 V.

El ADP3607 es capaz de proporcionar hasta 50 mA. Trabajando con una tensión de entrada tan baja como 3 V, tiene una salida fija a 5 V. Está también disponible en un formato ajustable con rango de 3 V a 9 V a través de una resistencia externa. Se puede obtener una tensión incluso mayor con una etapa externa mediante componentes pasivos.

El inversor de tensión ADP3605 con tensión de salida regulada es capaz de entregar hasta 120 mA. Hay versiones con salida fija a -3 V (ADP3605-3) o ajustable mediante resistencias externas con rango desde -3 V a -6 V. (Se puede conseguir una tensión negativa mayor mediante una etapa externa, como con el ADP3607). Se necesita una alimentación de -5 V a partir de una tensión de entrada de +5 V, así que establecemos una  $R = 31,6 \text{ kohms}$  ( $\pm 1\%$ ), usando la ecuación siguiente:

$$V_{out} = -1,5R/9,5 \text{ k}\Omega$$

Ambas alimentaciones ( $\pm 5 \text{ V}$ ) se generan mediante bombas de carga capacitivas, las cuáles no pueden generar tensiones peligrosas, incluso bajo condiciones de fallo, ya que no requieren de inductancias. Estos dispositivos se caracterizan por un modo shutdown, el cual permite al Microconverter dejarlos en modo de bajo consumo cuando el sistema no está en uso.

Además del aislamiento digital, las resistencias serie, Rx1, Rx2, y Rx3, proporcionan protección al paciente, cumpliendo con la AAMI (Association for the Advancement of Medical Instrumentation) para niveles seguros de corriente. Estos estándares requieren que las corrientes RMS de masa o de defecto deben ser menores de  $50 \mu\text{A}$ .

### Procesado de la señal

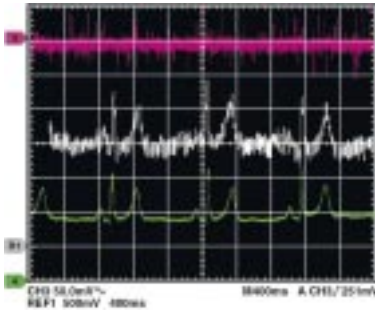
El Microconverter ADuC842 está perfectamente indicado para las principales tareas de procesamiento de la señal. Se caracteriza por un rápido ADC de 12-bit y otros periféricos analógicos de altas prestaciones, un núcleo 8052 de 1 ciclo de reloj por ciclo máquina, memoria flash de código de 62 kB incorporada, y otros útiles periféricos (ver figura 4). Los componentes clave del Microconverter para este diseño son el ADC y el núcleo 8052. El ADC convierte la salida analógica del amplificador de instrumentación a una señal digital. El software escrito para el núcleo 8052 procesa la señal digitalizada para obtener los datos de la señal ECG final.



Figura 3. Configuración ECG propuesta

Figura 4. Diagrama de bloques simplificado del ADuC842

Figura 5. Trazas osciloscópicas (ver texto)



Cómo en muchos diseños de Microconverters, el software incluye complejo código de alto nivel en C y rutinas sensibles al tiempo escritas en ensamblador. En este caso, la implementación de filtros pasa banda y filtros notch es en C, mientras que el ADC se controla mediante ensamblador. El código en ensamblador, combinado con la velocidad del convertidor, permite la acumulación de múltiples muestras, mejorando la resolución efectiva del ADC más allá de los 12 bit usuales.

La figura 5 da una buena indicación de la efectividad del Microconvertidor. La señal de arriba es la señal aplicada desde el amplificador de instrumentación al ADC. La señal de en medio muestra los resultados iniciales conseguidos mediante el filtrado con código C, mientras que la señal inferior muestra el resultado final después de múltiples conversiones, usando código en ensamblador.

### Filtros realizados en código C

La señal adquirida se filtra digitalmente en el Microconvertidor. Para este propósito, se diseñaron dos filtros IIR (Infinite Impulse Response) de segundo orden, basados en una frecuencia de muestreo de 500Hz. Un filtro notch se diseñó para suprimir la interferencia a 50Hz. El procedimiento escogido fue el método de polo cero, con una frecuencia del notch de 50Hz y un ancho del notch de 10Hz. Para lograr esto se requirió la siguiente función de transferencia:

$$H(z) = \frac{1 - 1.618z^{-1} + z^{-2}}{1 - 1.5164z^{-1} + 0.8783z^{-2}}$$

Esta función de transferencia se puede convertir dentro de un algoritmo programable recursivo:

$$NOut_k = NIn_k - 1.618NIn_{k-1} + NIn_{k-2} + 1.5164NOut_{k-1} - 0.8783NOut_{k-2}$$

En esta ecuación el subíndice, k, significa el valor actual, k-1 significa el valor en el instante previo, y así sucesivamente.

Necesitamos ahora transformar esta ecuación a código. La codificación en C fue la elección automática para este procesamiento de aritmética intensiva, ya que la programación en ensamblador nos habría llevado mucho tiempo. La implementación directa de las ecuaciones de filtrado no sería adecuada con el ADuC842, ya que no está pensado para operaciones en coma flotante. Afortunadamente podemos escalar los coeficiente (p.e. 4096) e implementar el código notch como:

```
iNOut = (4096L*iNIn-6627L*iNIn1
+4096L*iNIn2+6211L*iNOut1
-3598L*iNOut2)/4096;
```

Esto implementa un filtro de segundo orden. Aunque se pueden calcular filtros de mayor orden, en la práctica parece más sencillo poner en cascada filtros de segundo orden.

El segundo filtro es un filtro Butterworth pasabanda con una frecuencia de corte baja de 0.05Hz y una frecuencia de corte alta de 100Hz. La función de transferencia y el algoritmo recursivo son:

$$H(z) = \frac{0.4206 - 0.4206z^{-2}}{1 - 1.1582z^{-1} + 0.1582z^{-2}}$$

$$BOut_k = 0.4206BIn_k - 0.4206BIn_{k-2} + 1.1582BOut_{k-1} - 0.1582BOut_{k-2}$$

La implementación en código C sería:

```
iBOut = (1723L*iBIn-1723L*iBIn2
+4745L*iBOut1
-650L*iBOut2) /4096;
```

Las salidas se pueden escalar cambiando los coeficientes de las entradas. Así mismo, si las señales son todas positivas, la división por 4096 se realiza sencillamente mediante 12 desplazamientos a la derecha.

La implementación mostrada en el listado 1, está desarrollada para 5 filtros pasa banda y 2 filtros notch.

### Procesado en ensamblador

Las funciones principales realizadas en ensamblador son para medir la señal de entrada a intervalos regulares y para asegurar que los cálculos en C se repitan a la velocidad requerida de 500 veces por segundo. El Timer0 se programó para correr de manera continua y generar interrupciones en intervalos de 1ms. Cada interrupción restaura el Timer0, realiza una conversión ADC, e incrementa una variable, c2ms, usada para sincronizar el código C. En esta etapa de desarrollo del código, las primeras líneas de código C fueron:

```
while(c2ms<2); //Usado en 1ª fase
c2ms = 0;
iAdc00 = iAdc0;
```

Inicialmente, c2ms es 0, y el código C esperará en la línea mientras (c2ms < 2); Después de 1ms, ocurre la interrupción del Timer0, y c2ms se incrementa a 1. Después de otro milisegundo, c2ms se incrementa a 2. Salimos del while y el código continúa reseteando el contador c2ms a 0 y haciendo los cálculos de filtrado. Después, el código C desplaza los resultados a través de la cadena de variables para la siguiente iteración del bucle. La parte final del bucle es el printf(...), que envía los resultados al PC para mostrar la gráfica. El procesamiento de los datos en el PC está más allá del ámbito de este artículo siendo tan sofisticado como el diseñador desee. Esta solución produjo la gráfica blanca de la figura 5.

```

while(1)
{
while(c25ms<64); //Wait for 64 measurements to be done.
iIn1 = iAdc0>>3; //save accumulated measurement.
iAdc0 = 0; //Zero for new measurement accumulation.
c25ms = 0; //Reset synchronization timer.
// 3 Band pass 0.05 - 100Hz fs=500 first 2 with gain of 4 each.
iB01 = (6591L*iIn1-6591L*iIn0+4745L*iB01L-
6591L*iB02L)>>12L;
iB02 = (6591L*iB01-6591L*iB02+4745L*iB01L-
6591L*iB02L)>>12L;
iB03 = (1723L*iB02-1723L*iB03+4745L*iB01L-
6591L*iB02L)>>12L;
iB04 = (1723L*iB03-1723L*iB04+4745L*iB01L-
6591L*iB02L)>>12L;
iIn1 = (1723L*iB03-1723L*iB04+4745L*iIn1-6591L*iIn2)>>12L;
// 2 notch filters.
iB01 = (4096L*iIn1-4096L*iIn0+4096L*iIn2+6311L*iB01L-
3598L*iB02L)>>12L;
iB02 = (4096L*iB01-4096L*iB02+4096L*iB02L+6311L*iB01L-
3598L*iB02L)>>12L;

iIn2 = iIn1; //save delayed values for filters.
iIn1 = iIn0;
iB02L = iB01L;
iB01L = iB02L;

iIn2 = iIn1L;
iIn1 = iIn0L;
iB02L = iB01L;
iB01L = iB02L;

iB02L = iB01L;
iB01L = iB02L;
iB02L = iB01L;
iB01L = iB02L;
}
//Other delayed values not shown.

if(iIn>24000) iDec += 1; //Control A/D20 output level.
if(iIn<8000) iDec -= 1;
iOut1 = (iDec)<<8;
DAC1H = (iOut1)>>8;
DAC1L = (iOut1)<<8;

if((iB30+iOfs)>3000) iOfs += 1; //Control output level.
if((iB30+iOfs)<1000) iOfs -= 1;
iOut = ((iB30+iOfs)<<8);
DAC1H = (iOut)>>8;
DAC1L = (iOut)<<8;
if((c2+43)) printf("%4d\n",iOut); //Output to PC.
}
    
```

```

IntT0:   push    ACC
        push    PSW
        clr     T0          ;Stop T0.
        mov    TH0,#0f0h   ;Reload 1/32ms.
        mov    TL0,#0f0h
        setb   T0          ;Restart T0.
        mov    a,ADCDAT0L
        rrl   a,iAdc0+3
        mov    a,iAdc0+1,a
        mov    a,ADCDAT0H
        andl  a,#0fh
        addc  a,iAdc0+2
        mov    iAdc0+1,a
        clr   a
        addc  a,iAdc0+1
        mov    iAdc0+1,a
        mov    AOCCON1,#0
        mov    AOCCON2,#10h ;Start a conversion.
        inc   c25ms        ;Increment ms counter.
IntT0H: pop    PSW
        pop    ACC
        ret
    
```

Para mejorar el resultado, la velocidad de generación de interrupciones del Timer0 se aceleró a 1/32 ms, y el dato se acumuló en iAdc0, para utilizar múltiples medidas en vez de una única. Al mismo tiempo, el while se cambió a while(c2ms < 64), de manera que el código C esperaría la

acumulación de 64 medidas antes de cada bucle de filtrado. El valor en iAdc0 se salva en iAdc00 para un nuevo procesamiento, y el iAdc0 es puesto a cero-listo para acumular las siguientes 64 medidas. Esta solución mejorada produce la gráfica verde de la figura 5.

## Ganancia

La ganancia de la señal es un factor muy importante en un diseño ECG. En el diseño descrito, depende de varios factores. La ganancia analógica se fija en 8X, tal como se discutió anteriormente.

Lo siguiente fue una ganancia de 64X resultante de acumular 64 medidas de esta señal. Hay una pérdida de señal de 8X a través del código iBln = iAdc0 >> 3; y finalmente, una doble ganancia de 4X debido al escalado de las dos primeras ecuaciones para el filtrado pasabanda.

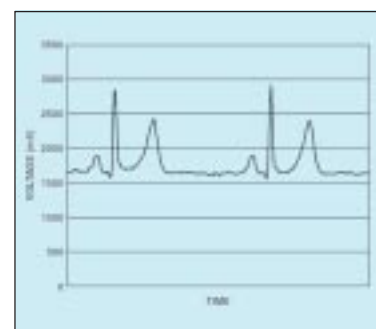
Esto genera una ganancia total de:

$$G = (8 \times 64 / 8) \times 4 \times 4 = 1024$$

ganancia típica en circuitos ECG.

## Conclusión

La figura 6 muestra los resultados con una configuración Einthoven lead I. Se observan los buenos resultados obtenidos a pesar de la simplicidad de la electrónica utilizada, de todas maneras un diseñador dedicado podría mejorar los resultados de manera significativa.



Listado 1 Rutina (parcial) en lenguaje C de la aplicación descrita

Figura 6. Gráfica de una medida real

Listado 2. Código ensamblador

Otras mejoras adicionales se podrían lograr mediante la implementación en código de distintas frecuencias de filtrado y otras características especiales.

La memoria de código del ADuC842 es Flash, permitiendo actualizaciones después de la fabricación del producto, según las necesidades de los pacientes. □