

Artículo de

Electrocardiógrafo digital de 12 derivaciones usando un convertidor ADS1298 y filtros digitales

12 lead digital electrocardiograph using an ADS1298 converter and digital filters

1 . Ariel Quezada

RESUMEN

El desarrollo de un electrocardiógrafo de 12 derivaciones usando filtros analógicos implica el uso de una gran cantidad de componentes y presenta problemas relacionados con la variación de sus valores nominales, sus tolerancias y efectos del tiempo y la temperatura. El sistema propuesto utiliza un convertidor ADS1298 de 24 bits de resolución, un microcontrolador y un computador personal que permite capturar la información y guardarla en un formato abierto. Esta información se procesa usando filtros digitales para remover señales de ruido en frecuencias específicas. Al utilizar filtros digitales, se utiliza una cantidad menor de componentes y se evita el efecto de las tolerancias y degradación de los mismos en el comportamiento de los filtros. El sistema se ha desarrollado utilizando solamente herramientas de programación gratuitas y de código abierto.

Palabras clave: Electrocardiógrafo. 12 derivaciones. Procesamiento digital de señal.

ABSTRACT

The development of a 12-lead electrocardiograph using analog filters involves the use of a large number of components and presents problems related to the variation of their nominal values, their tolerances and effects of time and temperature. The proposed system uses an ADS1298 24-bit resolution converter, a microcontroller and a personal computer, used to capture the information and save it in an open format. This information is processed using digital filters to remove noise signals at specific frequencies. When using digital filters, a smaller amount of components is used and the effect of their tolerances and degradation in the behavior of the filters is avoided. The system has been developed using only free and open source programming tools.

Keywords: Electrocardiograph. 12 lead. Digital signal processing.

INTRODUCCIÓN

Tradicionalmente, la adquisición de señales de electrocardiografía se ha realizado usando diseños analógicos en las etapas de amplificación y filtrado. Un problema de los sistemas analógicos es que son voluminosos, puesto que requieren un gran número de componentes. Otro problema es que se hace el diseño tomando en cuenta los valores nominales de los componentes; sin embargo, estos valores varían con la temperatura y se degradan con el tiempo, además de variar entre componentes supuestamente con el mismo valor [1].

Una forma de evitar estos problemas es realizar parte del sistema por software, específicamente la que corresponde al filtrado y usando técnicas de procesamiento digital de señal. Si la señal es muestreada con una resolución muy alta, el contenido de información se mantiene y se puede recuperar usando filtros digitales [2].

Actualmente existen en el mercado componentes específicos para aplicaciones de este tipo, como el ADS1298ECGde Texas Instruments que es un convertidor analógico-digital de 24 bits y 8 canales simultáneos, con amplificadores diferenciales de ganancia programable para cada canal, además de contar con circuitería adicional para generar señales propias del electrocardiógrafo [1].

Asimismo, la disponibilidad de software libre para manejar microcontroladores, programas en PC y procesamiento digital de señales es abundante, con buena documentación en general, y con algunos productos de excelente calidad, en muchos casos mejores que las ofertas de software pagadas. Este es el caso particular de Win-avr (avr-gcc, Programer 's Notepad) y Python (PySerial, Numpy, SciPy, Matplotlib, WxPython).

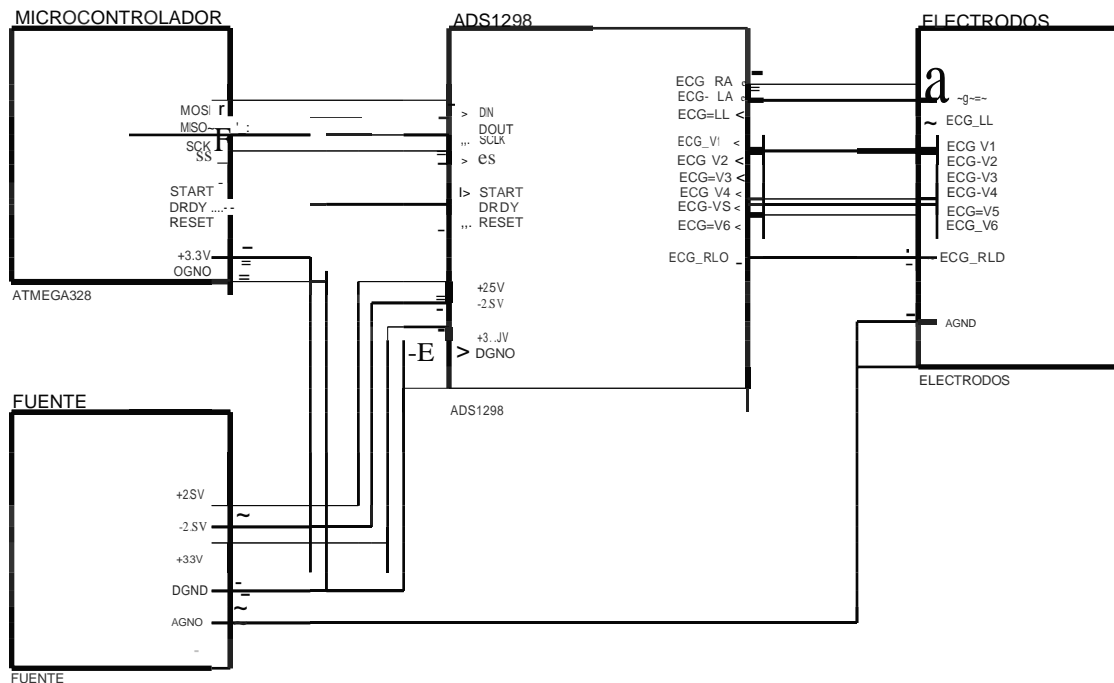
El objetivo de este trabajo es utilizar filtros digitales para reemplazar los filtros analógicos en un electrocardiógrafo de 12 derivaciones y mostrar una aplicación real del procesamiento digital de señales y los beneficios que se pueden lograr al utilizar esta herramienta.

MATERIALES Y MÉTODOS

1.- Hardware

El sistema de adquisición de datos está compuesto por cuatro elementos: fuente de alimentación, microcontrolador, ADS1298y electrodos. Estos elementos están conectados como se muestra en la figura N° 1.

Figura N° 1. Diagrama esquemático.



Fuente: Elaboración propia, enero 2015.

Fuente de alimentación

El sistema requiere de una fuente de alimentación bipolar con voltajes de +2.5V y -2.5V para la parte analógica, y +3.3V para la parte digital.

Se utiliza un transformador que permite alimentar al sistema con 220 VAC y que aísla al paciente de la entrada de voltaje. La salida del transformador entrega 6 VAC, que se pasa por un puente rectificador para volverla positiva.

Este voltaje alimenta los circuitos de los reguladores de voltaje que entregan una salida de +2.5V y +3.3V. Para obtener el voltaje negativo se utilizó un convertidor de voltaje ICL7660, que entrega un voltaje negativo de la misma magnitud que el voltaje que se conecta en su entrada.

Microcontrolador

Se utiliza el microcontrolador ATmega328 funcionando a una velocidad de 7.3728 MHz, alimentado con +3.3V. Trabajar con una frecuencia de 7.3728 MHz permite programar el puerto serial del microcontrolador a velocidades estándar, sin errores en la división de frecuencia. La velocidad estándar máxima que se puede alcanzar con esta frecuencia es de 460.800 baudios.

El microcontrolador se conectó al convertidor ADS1298 mediante el puerto SPI y se encarga de configurar y controlar la adquisición de datos desde el convertidor y enviarlas hacia un computador, mediante su puerto serial.

Convertidor analógico-digital

El convertidor analógico-digital utilizado es el ADS1298, un convertidor sigma-delta de 24 bits y 8 canales simultáneos, que incluye amplificadores de ganancia programables y es capaz de alcanzar una frecuencia de muestreo de 32.000 muestras por segundo. Adicionalmente, incluye varias características que se requieren usualmente para la adquisición de datos de un electrocardiograma (ECG), como ser la elección de cualquiera de las señales para generar la derivación de la pierna derecha (RLD) y el terminal central de Wilson (WCT) (1, 3). Se conecta al microcontrolador a través del bus serial SPI y se alimenta con +2.5V y -2.5V para su parte analógica y con +3.3V para su parte digital.

Los electrodos se conectan a las entradas diferenciales del ADS1298, que previamente pasan por un filtro RC pasa bajos de segundo orden, que es un filtro anti-aliasing, para evitar errores de muestreo en el convertidor.

El ADS1298 se configuró para trabajar con una frecuencia de muestreo de 500 muestras por segundo. Se hizo la lectura simultánea de 8 canales de datos y uno de estado, cada uno con una longitud de 24 bits, es decir, 3 bytes. Por

tanto, la velocidad de envío de datos al computador es de 13.500 bytes por segundo (500 muestras* 9 canales* 3 bytes). Para poder enviar datos a esta velocidad, el puerto serial del microcontrolador se programó para trabajar con una velocidad de 230.400 baudios.

2.- Software

Microcontrolador

El microcontrolador se programó usando la herramienta WinAVR, que utiliza el compilador AVR-GCC, el editor de texto Programmer's Notepad y la herramienta de programación AVRdude. Para grabar el código en el microcontrolador se ha utilizado el programador USBasp.

Protocolo de comunicación serial

Para la comunicación serial entre el microcontrolador y el computador se enviaron los datos en formato binario (que pueden tener cualquier valor entre 0x00 y 0xFF), por lo que es necesario implementar algún sistema de codificación de tramas para poder delimitar el inicio y el final de un paquete de datos que contenga la información de los 8 canales y estado de cada lectura del ADS1298. En el presente caso, se realizó la formación de tramas mediante relleno con octetos (byte stuffing) similar al usado en el protocolo PPP (Point-to-Point Protocol).

El byte usado como delimitador de la trama es 0x7E (0b01111110), que indica que se ha terminado de enviar un paquete de datos. El byte usado como caracter de escape (escape character) es 0x7D (0b01111101).

Cuando dentro de la trama el byte a enviar es igual a un delimitador o un escape, se reemplaza con 0x7D seguido por el resultado de un XOR entre el valor original y 0x20 (0b00100000). Así 0x7D se convierte en el par 0x7D 0x5D y 0x7E se convierte en el par 0x7D 0x5E.

Para recuperar la información original, cuando se encuentra 0x7D en la trama, no se lo toma en cuenta, solamente se toma el byte siguiente y se utiliza el valor resultante de hacer un XOR con 0x20. Así, al encontrar el par 0x7D 0x5D, se convierte en un solo byte 0x7D y el par 0x7D 0x5E se convierte en el byte 0x7E.

Cada vez que se realizaba una lectura del ADS1298, se formaba un paquete modificando todos los valores iguales a 0x7D y 0x7E, agregando al final el delimitador 0x7E y se envió estos datos por el puerto serial.

Programa para computador

Para el programa en el computador se utilizó el lenguaje Python 2.7, con las librerías PySerial, NumPy, SciPy, Matplotlib y WxPython. PySerial se utiliza para la comunicación serial entre el computador y el microcontrolador.

MumPy y SciPyse utilizan para el procesamiento digital de la señal, filtrado y análisis del espectro de frecuencias. Para el filtrado se aplicó la función `scipy.signal.filtfilt`, que es un filtro de adelante-atrás (forward-backward filter). Esta función aplica un filtro lineal dos veces, una hacia adelante y otra hacia atrás, de manera que el efecto combinado de ambos filtrados tiene fase lineal. De esta forma el filtrado no distorsiona la fase de la señal procesada [4].

Con la función `scipy.signal.iirfilter` se define el tipo de filtro que se va a aplicar y en el presente caso se utilizó un filtro rechaza banda de tipo Butterworth.

Matplotlib es una librería que se utiliza para representar gráficamente las señales de ECG. Matplotlib genera gráficas de manera similar a Matlab y se complementa directamente con NumPy y SciPy.

WxPython es la librería que se encarga de la interfaz gráfica de usuario (GUI). Permite definir distintos tipos de objetos gráficos, como ser ventanas, botones, paneles, cuadros de diálogo, pestañas, etc. en un sistema que funciona basado en eventos.

Formato de archivo

Los datos se guardan en un archivo de texto, usando etiquetas para delimitar los diferentes campos (derivaciones, cantidad de datos por canal, muestras por segundo, ganancia).

Si se leen 1024 datos, en el archivo aparecerá el campo: `<nd>1024</nd>`, donde "nd" es el "número de datos". Las muestras por segundo se representan como: `<sr>500</sr>`, donde "sr" es "sample rate" o tasa de muestreo. La ganancia es `<gn>2</gn>`, donde "gn" es "ganancia". Para las derivaciones se usaron las etiquetas:

`-1229,-1294, ... `

`<L2>-2083,-2339, ... </L2>`

`<VI>-739,-2411, ... </VI>`

`<V2>566,-1060, ... </V2>`

`<V3>-227,-1637, ... </V3>`

`<V4>-1815,-3601, ... </V4>`

`<V5>1373,-447, ... </V5>`

`<V6>-1313,-3488, ... </V6>`

Los datos de cada derivación están en modo texto y representan un número de 24 bits y es una cadena de la cantidad de datos que se han leído por cada canal, separados por una coma. El programa guarda los datos separados por comas, entre las etiquetas correspondientes. Para recuperar los datos, se separan los canales usando las etiquetas y se separan los valores usando las comas.

Las derivaciones adicionales se forman mediante operaciones matemáticas, puesto que estas derivaciones se pueden expresar en términos de L1 y L2. Esto es posible porque el convertidor analógico-digital realiza el muestreo simultáneo de todas las entradas. Estos valores no se guardan, sino que se calculan al momento de mostrar las señales en el interfaz gráfico de usuario.

RESULTADOS

Las gráficas de la señal de ECG sin filtrar se muestran en la figura N° 2. En la figura N° 3 se observa la misma señal de ECG, pero luego de aplicar un filtro rechaza banda de 50 Hz.

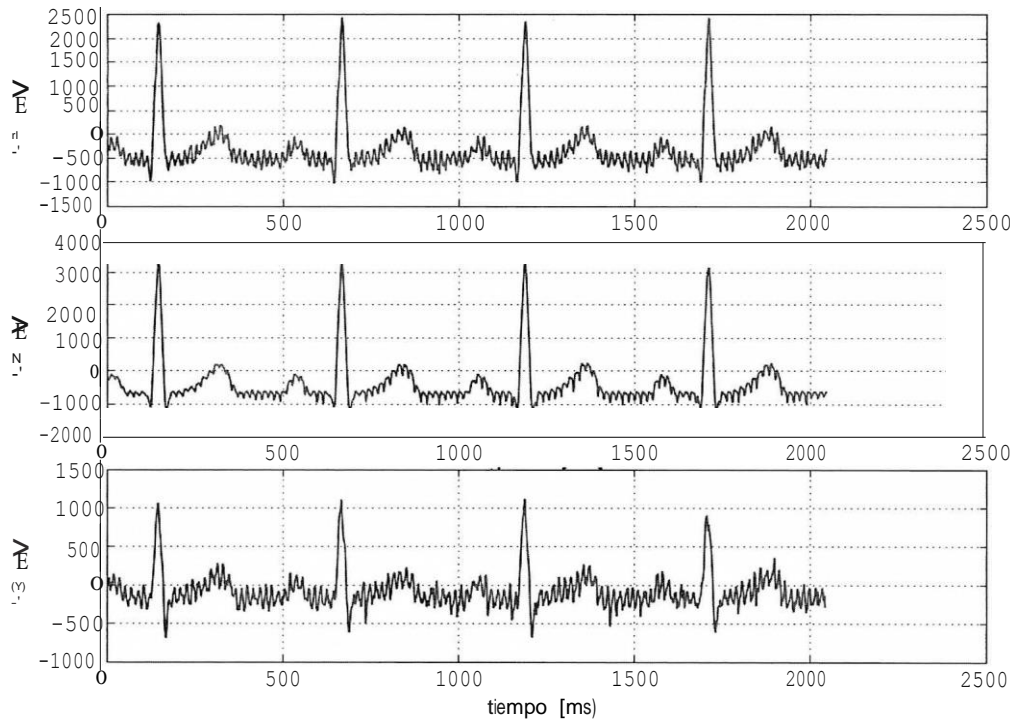
Es posible realizar las lecturas de la señal de ECG usando el sistema propuesto y las señales son claramente reconocibles en las gráficas del computador.

Las señales obtenidas presentan una magnitud importante de ruido en 50 Hz y sus armónicos (100 Hz, 150 Hz). Para determinar el espectro en frecuencia de las señales se aplica una Transformada Rápida de Fourier, que está incluida en el módulo SciPy (fftpack.fft).

El ruido de 50 Hz se logra reducir en gran medida usando el filtro rechaza banda digital en 50 Hz, pero tendría que aplicarse el mismo filtro en las frecuencias de 100 Hz y 150 Hz. Sin embargo, a mayor cantidad de filtros, la señal original se va distorsionando y la velocidad de respuesta del sistema va disminuyendo.

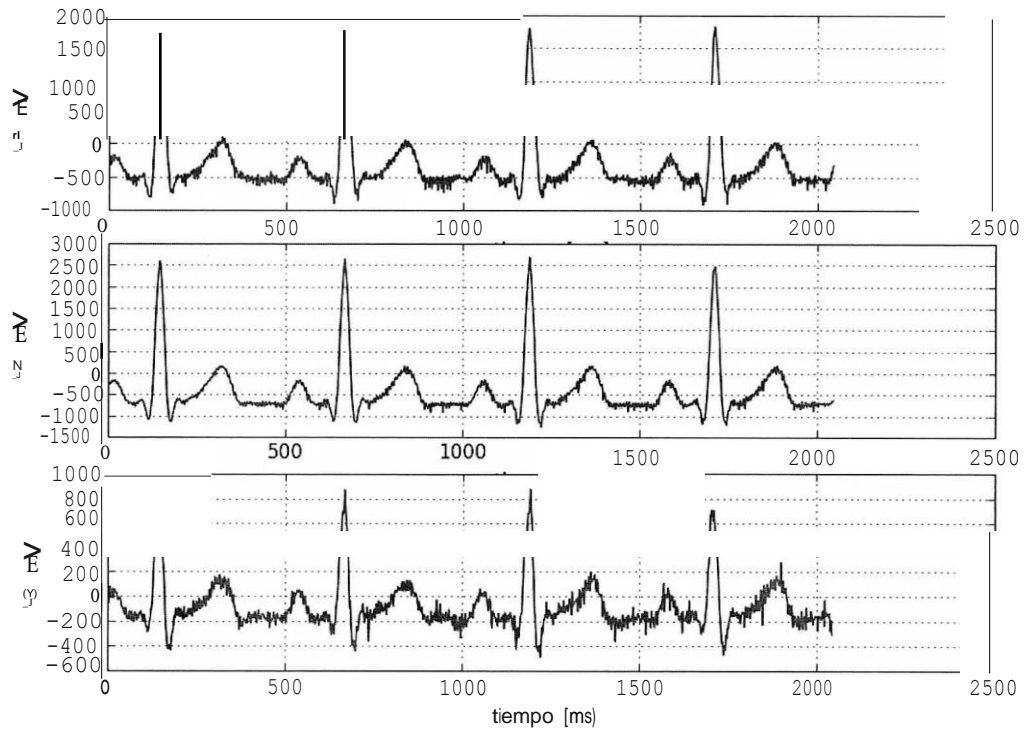
Es importante utilizar un cable blindado y conectar el blindaje correctamente, puesto que, sin el blindaje, el sistema no permite discriminar entre ruido y señal de ECG.

Figura N° 2. Señal de ECG sin filtrar



Fuente: Elaboración propia, enero 2015.

Figura N° 3. Señal de ECG luego de aplicar un filtro rechaza banda en 50 Hz



Fuente: Elaboración propia, enero 2015.

DISCUSIÓN

El prototipo que se ha implementado no está en una placa impresa y la contaminación por ruido es muy significativa, especialmente en la frecuencia de 50 Hz y sus armónicos en 100 Hz y 150 Hz, por lo que se deben utilizar técnicas de reducción de ruido y un sistema de placa impresa de por lo menos cuatro capas, donde se tenga una para alimentación y otra para tierra, para reducir el ruido y mejorar la calidad de la señal.

Este trabajo ha sido la base para el desarrollo de un sistema medidor de vibraciones de máquinas rotativas [5] y un electrocardiograma de 12 derivaciones no simultáneo y con hardware más optimizado [6], que utilizan Python y procesamiento digital de señales.

REFERENCIAS BIBLIOGRÁFICAS.

- [1] Texas Instruments. (2010). Low-Power, 8-Channel, 24-Bit Analog Front-End for Biopotential Measurements (Rev. J). 01/2015. Sitio web: <http://www.ti.com/lit/ds/symlink/ads1298.pdf>
- [2] Texas Instruments. (2010). Analog Front-End Design for ECG Systems Using Delta-Sigma ADCs (Rev. A). 01/2015. Sitio web: <http://www.ti.com/lit/an/sbaa160a/sbaa160a.pdf>.
- [3] Texas Instruments. (2010). ADS1298ECG-FE/ADS1198ECG-FE ECG Front-End Performance Demonstration Kit User's Guide. Rev. 01/2016. Sitio web: <http://www.ti.com/lit/ug/sbaul71c/sbaul71c.pdf>.
- [4] Unpingco, J. (2014). Python for Signal Processing. Suiza: Springer.
<https://doi.org/10.1007/978-3-319-01342-8>
- [5] Torrez, M. y Quezada, A. (2016). Diseño de un Sistema Medidor de Vibraciones para Máquinas Rotativas de 900 rpm hasta 84000 rpm. Journal Boliviano de Ciencias. 12, 23-28.
- [6] Camacho, A., Quezada, A. y Morales, I. (2016). Sistema de Adquisición y Tratamiento Digital de las Doce Derivaciones Cardiacas del Electrocardiograma. Journal Boliviano de Ciencias. 12, 29-33.

Fuentes de financiamiento: Esta investigación fue financiada con fondos de los autores.

Declaración de conflicto de intereses: Los autores declaran que no tiene ningún conflicto de interés.

Copyright (c) 2017 Ariel Quezada



Este texto está protegido por una licencia [Creative Commons 4.0](https://creativecommons.org/licenses/by/4.0/).

Usted es libre para Compartir —copiar y redistribuir el material en cualquier medio o formato— y Adaptar el documento —remezclar, transformar y crear a partir del material— para cualquier propósito, incluso para fines comerciales, siempre que cumpla la condición de:

Atribución: Usted debe dar crédito a la obra original de manera adecuada, proporcionar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que tiene el apoyo del licenciante o lo recibe por el uso que hace de la obra.

[Resumen de licencia](#) - [Texto completo de la licencia](#)