

Monitor de Electrocardiografía a través de una Computadora Personal

M.Sc. Miguel Yapur, myapur@espol.edu.ec

Ing. Jhovany Rodríguez, mjrodrig@espol.edu.ec

Ing. Wladimir Gaibor, wgaibor@espol.edu.ec

Facultad de Ingeniería en Electricidad y Computación (FIEC)

Escuela Superior Politécnica del Litoral (ESPOL)

Campus Gustavo Galindo, vía Perimetral Km 30.5, Guayaquil, Ecuador

Resumen

En el presente proyecto se exponen los principios, el diseño y la construcción de un módulo de adquisición de señales bioeléctricas del corazón (EKG). Se toma una derivación bipolar de la señal eléctrica del corazón (D1, D2 ó D3) y se la transmite hacia una computadora personal (PC) mediante el puerto paralelo. El módulo está compuesto de diferentes bloques: a) bloque de adquisición de la señal cardiaca, b) bloque de filtrado de la señal, c) bloque de amplificación y desplazamiento DC, d) digitalización y transmisión de la señal hacia la PC a través del puerto paralelo. El bloque para la adquisición de la señal se encarga de conseguir una señal con muy poco ruido, luego esta señal es filtrada para eliminar señales de frecuencias no deseadas y se la amplifica para que ingrese al convertidor analógico- digital (ADC) y así obtener la señal digitalizada. A la salida del ADC se colocan buffers para aislar el puerto paralelo del circuito. Mediante la utilización de un software se crea la interfaz de la PC que toma los datos a través del puerto paralelo y los muestra en el monitor. La interfaz se realiza en Visual Basic 6.0, el mismo que mostrará en la pantalla tanto la señal de EKG como la frecuencia cardiaca del paciente.

Palabras Claves: Señal cardiaca, derivación bipolar, puerto paralelo, computador personal.

Abstract

This article presents the design and construction of an EKG (electrocardiographic) acquisition module. A bipolar derivation (D1, D2 or D3) is processed and transmitted to a personal computer (PC) through its parallel port. Hardware is based on the following blocks: an acquisition stage, a filter, an amplifier and an analog to digital converter for the processing of the cardiac signal. The signal/noise ratio is very high, resulting in a low artefactum module. Software is based on Visual Basic 6.0 and allows interfacing the cardiac information and displaying the EKG signal and the cardiac frequency in the monitor of the PC.

1. Introducción

Las enfermedades del corazón son en nuestros días una de las principales causas de muerte. Factores genéticos, hipertensión arterial, diabetes, tabaquismo, tasas altas de colesterol y stress provocan el funcionamiento incorrecto del corazón.

La electrocardiografía es un método clave para el diagnóstico de enfermedades cardiovasculares. En muchos lugares de atención médica, ya cuentan con computadoras para el diagnóstico clínico. La ciencia médica actual cuenta con equipos de diagnóstico indispensables para poder ayudar a la gente que lo requiera, pero la realidad es que estos equipos tales como los electrocardiógrafos, son demasiado costosos para las labores propias del médico clínico.

En este proyecto realizamos un electrocardiógrafo tratando de que la respuesta que brinda sea

exactamente igual a un electrocardiógrafo comercial con materiales de bajo costo y con el beneficio de proporcionar una interfaz gráfica mediante un software, en el cual el galeno podrá visualizar la señal cardiaca en una computadora personal y guardar los datos obtenidos para su correspondiente estudio.

Además de ser un equipo de un costo relativamente bajo, es portátil, necesita sólo dos baterías para funcionar; de esta forma el especialista que no cuente con un equipo profesional tendrá un ahorro considerable de tiempo y podrá de forma expedita controlar a sus pacientes.

Debido al avance de los sistemas computacionales, el electrocardiograma obtenido, podría ser enviado vía e-mail a cualquier otro lugar del país o del mundo; puede también ser guardado en algún medio de almacenamiento con la disponibilidad de llevar la información de un lugar a otro.

2. Diseño analógico

La señal cardiaca es de amplitud muy pequeña (alrededor de 1mV), la cual se encuentra inmersa dentro de otras señales que son de una amplitud mayor como por ejemplo el artefacto producido por la señal de 60Hz. Para poder registrar las señales cardiacas debemos limpiarlas del resto de señales que no interesan, eliminándolas con procedimientos adecuados. Por estas razones la señal que se desea obtener debe recibir un tratamiento especial para que esté libre de ruido.

El proceso de obtener y filtrar la señal cardiaca se realiza en el bloque de “adquisición de señal”, que se detalla a continuación.

2.1. Diagrama de bloques

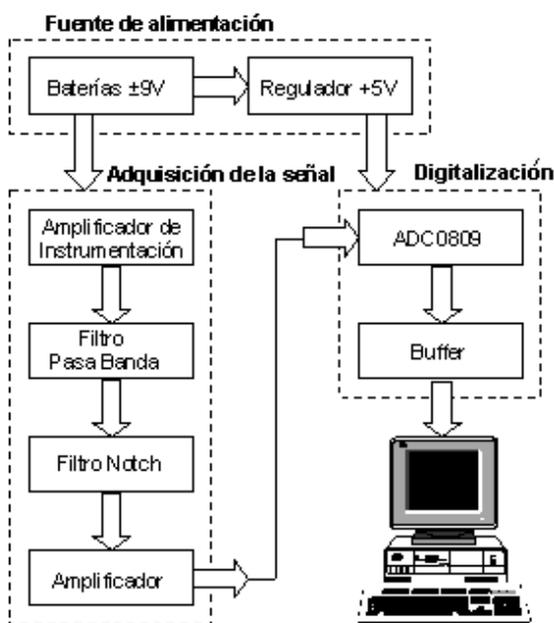


Figura 1. Diagrama de bloques.

2.2. Adquisición de la señal

En la figura 2 podemos ver el circuito que se utilizó para adquirir la señal cardiaca.

Se trata de un amplificador de instrumentación, el cual está construido con dos seguidores de voltaje y un amplificador diferencial. Las pequeñas señales provenientes del corazón pueden ser amplificadas y a la vez se minimizan las señales de ruido, lo cual se logra gracias a la propiedad del amplificador diferencial, de tener muy baja ganancia en modo común. Los opamps utilizados son de tecnología JFET, los cuales presentan una alta impedancia de entrada y una mínima corriente de polarización,

brindando así un margen de seguridad eléctrica para el paciente.

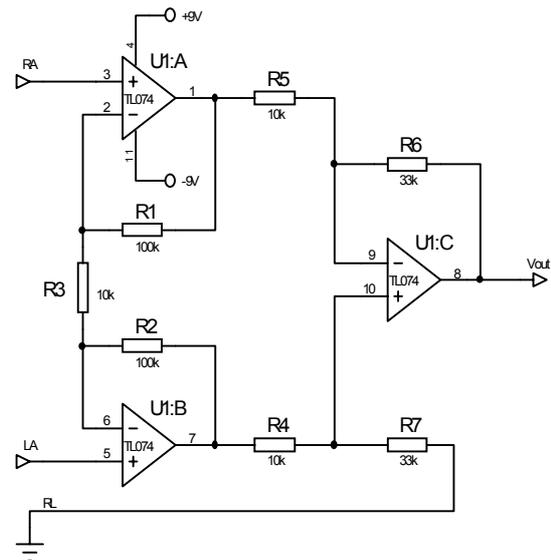


Figura 2. Amplificador de instrumentación.

2.3. Filtro pasa banda

La señal obtenida se ingresa al filtro pasa banda, lo que asegura que esté dentro de la banda especificada por las normas médicas, que está entre los 0.05Hz y 100Hz.

Estudios realizados demuestran que señales arriba de 100Hz no son cardiológicas y además, filtrando frecuencias menores de 0.05Hz eliminamos una diferencia de potencial entre los electrodos y la superficie de la piel que alcanzan niveles de hasta 300mv y que pueden llegar a saturar los circuitos del amplificador. Eliminando estas frecuencias aseguramos una alta ganancia de la señal EKG.

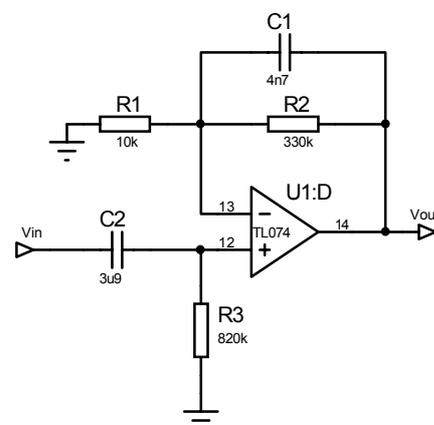


Figura 3. Filtro pasa banda.

2.4. Filtro Notch

Una vez que hemos determinado el rango de frecuencias para la señal que obtenemos, lo que nos queda es ingresar esta señal a un filtro Notch de 60Hz, para eliminar el ruido producido por señales de esta frecuencia que distorsionan la señal de EKG.

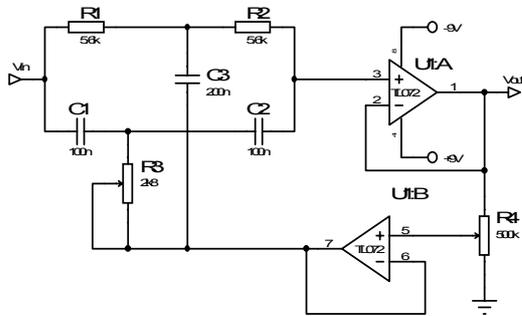


Figura 4. Filtro rechazo de banda (Notch).

2.5. Amplificador

La señal obtenida hasta ahora necesita ser manipulada para que alcance una amplitud comprendida entre 0v y 5v, para poder digitalizarla con el ADC0809, que acepta como entrada sólo señales que estén comprendidas entre esos valores.

Esto se logra con un amplificador sumador inversor que se muestra en la figura 5. Sumamos un voltaje DC de manera que obtenemos una señal positiva y además le damos ganancia para asegurar que esté entre 0v y 5v.

El opamp U1:A es un buffer seguidor de voltaje que sirve para separar esta etapa de la anterior para que la señal no se vea afectada por algún cambio de impedancias.

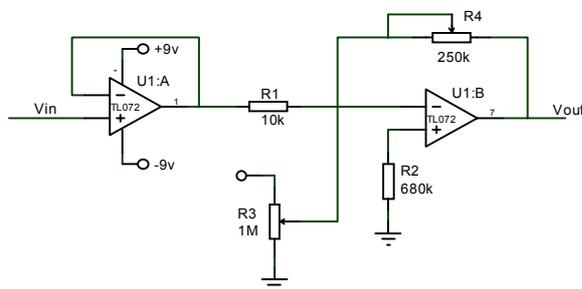


Figura 5. Amplificador.

2.6. Digitalización y buffers

Este bloque está compuesto prácticamente por el circuito integrado ADC0809, que es un convertidor de

señales analógicas comprendidas entre 0v y 5v, a señales digitales de 8 bits.

Con este circuito integrado obtenemos señales digitalizadas necesarias para la comunicación con el computador.

A la salida del ADC0809 se coloca un buffer, que en este caso es el 74LS244, el cual sirve para aislar todo el circuito del computador.

3. Layout

A continuación se muestra el proyecto final en proto board.

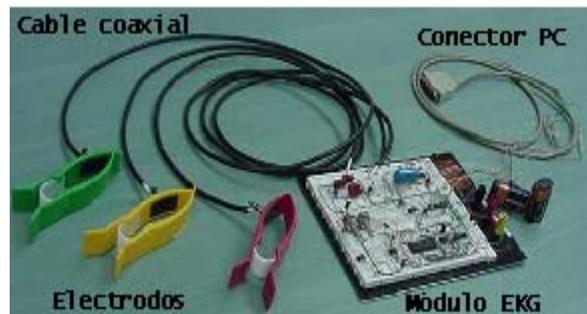


Figura 6. Foto del proyecto.

En la figura 6 podemos observar tres pinzas que contienen los electrodos; éstos se colocan, uno en la mano derecha (RA), otro en la mano izquierda (LA) y el tercer electrodo en el tobillo de la pierna derecha que es la que sirve de referencia (RL).

Los cables que se usan para conectar los electrodos al circuito son cables coaxiales apantallados que ayudan en gran parte a eliminar el ruido.

Para conectar el módulo de EKG al computador personal usamos el cable de la impresora que tiene en un extremo un conector DB25 macho, que se ajusta perfectamente en el puerto paralelo de la PC.

La fuente de alimentación son un par de baterías de 9v, con las que disminuimos enormemente el ruido debido a que el módulo no está conectado a la red eléctrica.

4. Diseño del software

Para diseñar el software debemos tener en cuenta que el puerto de comunicación que utilizaremos es el puerto paralelo de la PC; por lo tanto, aquí hacemos una breve descripción de los conceptos más importantes en lo que se refiere a este puerto de comunicación.

4.1. Registros del puerto paralelo

Se llama paralelo porque tiene un bus de datos de 8 líneas y se pueden escribir en él 8 bits al mismo tiempo. En las computadoras éste se encuentra en la

parte posterior y es un conector DB25 hembra en la mayoría de los casos.

Se puede encontrar una amplia bibliografía que detalla el uso específico de cada terminal con respecto a la impresora; pero lo que nos interesa es conocer en qué terminales podemos escribir datos hacia el hardware y en qué terminales podemos leer datos desde el hardware.

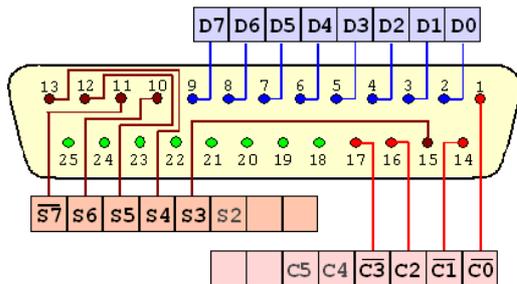


Figura 7. Puerto paralelo.

Como podemos ver en la figura 7, en el puerto paralelo podemos encontrar tres registros: DATOS(D0 - D7), ESTADO(S2 - S7) y CONTROL(C0 - C5).

El puerto DATOS: tiene 8 terminales de salida.

El puerto ESTADO: tiene 5 terminales de entrada.

El puerto CONTROL: tiene 4 terminales de salida

Tierra: Las restantes 8 terminales están aterrizadas.

Entonces, observamos que podemos utilizar el registro de DATOS para escribir hacia el hardware y podemos utilizar el registro de ESTADO para leer datos desde el hardware.

4.2. Lectura y escritura de datos en el puerto paralelo usando Visual Basic 6.0

El programa Visual Basic 6.0 no tiene instrucciones propias para escribir o leer datos del puerto paralelo. Lo que se hace es controlar el puerto por medio de una DLL, que significa librería de enlace dinámico. Las librerías de enlace dinámico son uno de los elementos principales del sistema operativo Windows 9X. En su concepto básico, se tratan de archivos ejecutables independientes que contienen funciones y recursos que pueden ser llamados por los programas y por otras DLL para realizar ciertos trabajos. Una DLL no puede ser ejecutada en forma independiente; entra en acción hasta que un programa u otra DLL llama a una de las funciones de la librería. El término "enlace dinámico" se refiere al hecho de que el código que contiene la DLL se incorpora al programa ejecutable que la llama sólo hasta el momento en que es requerido, en tiempo de ejecución.

Dentro de la DLL tenemos funciones que controlan el puerto paralelo y que pueden ser llamadas desde Visual Basic logrando con esto nuestro objetivo.

En Internet se puede encontrar una gran cantidad de estas librerías; la que nos dio mejor resultado es la INPOUT32.DLL debido a que trabaja bajo cualquier Windows, incluso el XP.

Para utilizar esta librería desde Visual Basic, lo primero que debemos hacer es declararla de la siguiente manera:

```
Public Declare Function Inp Lib "inout32.dll" _
Alias "Inp32" (ByVal PortAddress As Integer) As Integer
```

```
Public Declare Sub Outp Lib "inout32.dll" _
Alias "Out32" (ByVal PortAddress As Integer, ByVal Value As Integer)
```

Para escribir datos en el puerto se hace de la siguiente manera:

Outp DireccionPuerto, dato a escribir

La función Outp está declarada arriba y sirve para escribir en el puerto. La sintaxis de la función es escribir la dirección del registro del puerto al que se va a enviar el dato, seguido de una coma el valor que deseamos cargar en el puerto.

Para leer datos en el puerto se hace de forma similar:

Byte_leído = Inp(DireccionPuerto + 1)

La función Inp también declarada arriba sirve para leer los datos que provienen desde el hardware. La sintaxis de la función es escribir la dirección del registro del puerto en el que deseo leer. Este dato leído se guarda en la variable Byte_leído para uso del programa de la interfaz gráfica.

Para que estas instrucciones declaradas en el módulo principal del programa funcionen correctamente, se tiene que copiar la librería INPOUT32.DLL en el sistema operativo. Para realizar esta acción se ubica primero la carpeta "windows" y luego dentro de ésta se abre la carpeta "system" y aquí se copia la librería.

Cuando se ha cargado la librería en nuestro sistema, el siguiente paso a seguir es conocer la dirección del puerto paralelo y para esto solo basta con revisar las propiedades del puerto LPT1, que en la mayoría de los casos, la dirección "base" es la 378h.

Cuando se habla de la dirección del puerto paralelo se observa el término "dirección base". La razón de esto radica en que el puerto tiene tres registros y la dirección base da información de la dirección del registro de DATOS, la dirección "base+1" se refiere a la dirección del registro de ESTADO y la dirección "base+2" al registro de control. Así dependiendo si se escribe o se lee datos del puerto, tenemos que colocar en el programa la dirección del registro al que se desea acceder.

5. Diagrama de flujo

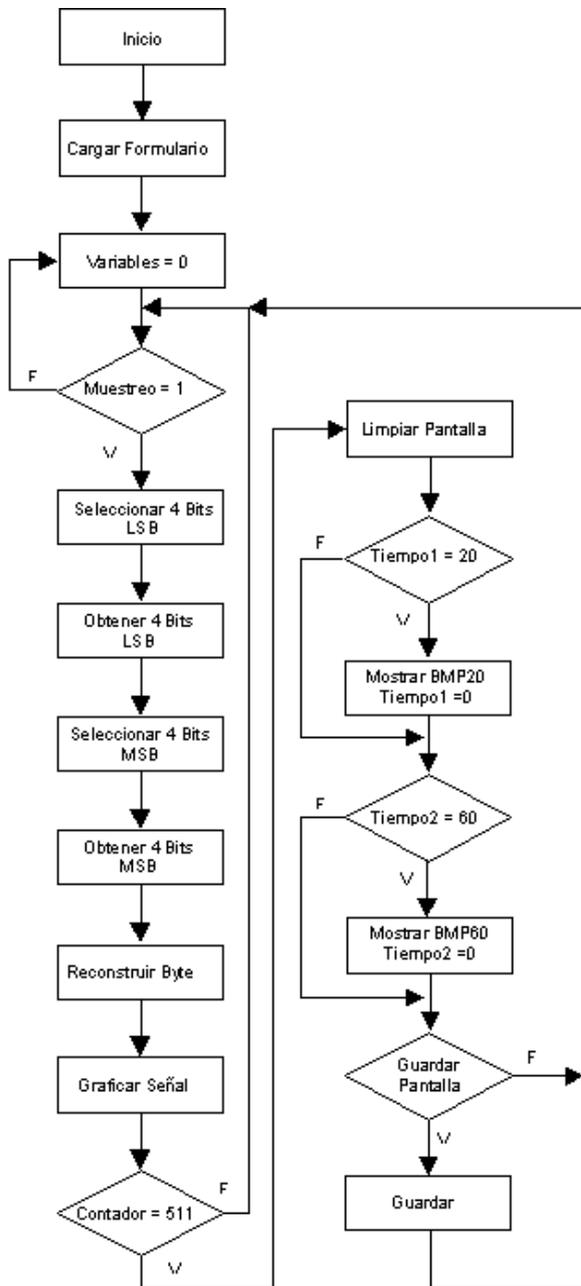


Figura 8. Diagrama de flujo.

6. Resultado final

Como resultado final tenemos un electrocardiograma en nuestra PC de muy bajo costo. La interfaz gráfica permite introducir ciertos datos como el nombre, edad, sexo, peso y la estatura del paciente.

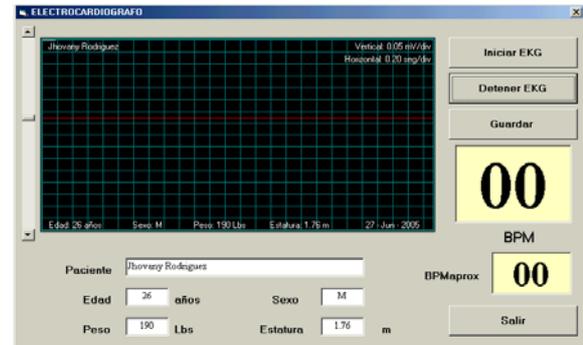


Figura 9. Interfaz gráfica.

En la figura 9 se muestra la interfaz gráfica desarrollada en Visual Basic 6.0. Esta interfaz, a más de mostrarnos la señal electrocardiográfica, nos visualiza también los latidos del corazón por minuto (BPM).

Para comenzar a correr el programa solo debemos hacer un clic en el botón “Activar EKG” y la interfaz comienza a graficar la señal electrocardiográfica como se muestra en la figura 10.

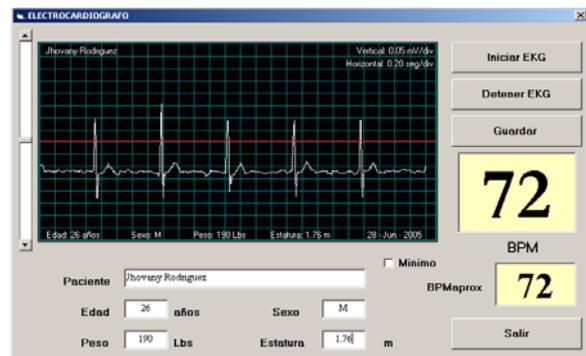


Figura 10. Visualización de la señal cardiaca.

En la figura 10 no se aprecia realmente la nitidez de la señal debido al tamaño del gráfico, pero el resultado es una señal limpia de ruido, en la cual el galeno puede observar con fidelidad el resultado.

Para detener el programa solo basta con hacer un clic en el botón “Detener EKG” y la interfaz congela la imagen obtenida hasta ese momento.

El software además permite guardar el electrocardiograma como un archivo bmp (mapa de bits) en el lugar del disco duro que se desee. La imagen guardada como mapa de bits se presenta en la figura 11.



Figura 11. Electrocardiograma1.

De esta manera tenemos un archivo con extensión bmp en el cual se puede ver claramente el electrocardiograma por el módulo registrado. A continuación se presentan dos archivos bmp que muestran los electrocardiogramas tomados a dos personas en nuestro laboratorio de electrónica médica.



Figura 12. Electrocardiograma2.



Figura 13. Electrocardiograma3.

Como podemos apreciar en las figuras arriba mostradas, los electrocardiogramas obtenidos con nuestro electrocardiógrafo muestran claramente la actividad eléctrica del corazón con lo cual un cardiólogo puede decir el buen o mal funcionamiento de nuestro corazón.

7. Conclusiones

Podemos decir que hemos cumplido con el propósito de construir un electrocardiógrafo de simple diseño, confiable y bajo costo (menor que \$300), con una respuesta de la actividad eléctrica de nuestro corazón tan fiel como la de electrocardiógrafos comerciales.

Una de nuestras metas a futuro es que este dispositivo tenga aceptación entre los médicos de nuestra comunidad a fin de que la comunidad médica pueda hacer uso de esta herramienta que está al alcance de la economía general.

8. Referencias

- [1] Bronzino, J., *The Biomedical Engineering Handbook*, The electrical engineering handbook, U.S.A: CRC Press, Inc., 1995, pp. 181-189.
- [2] Kington, T., and Plonsey R, "*Engineering Contributions to Biophysical Electrocardiography*", New York: IEEE Press, Inc., 1982, pp. 2-4, pp. 70-73, pp. 89-91.
- [3] Maltzahn, W., and Yapur M., "Medical Electronics," Folleto Técnico, Facultad de Ingeniería en Electricidad y Computación, ESPOL, 1987, pp. 1-20.
- [4] Coughlin, R., and Driscoll F, *Amplificadores Operacionales y Circuitos Integrados Lineales*, México: Prentice Hall, 1999, pp. 214-244.
- [5] Jacob, M., *Applications and design with Analog Integrated Circuits*, New Jersey: Prentice Hall, 1982, pp. 359-425.