

## 7. DESCRIPCIÓN Y CONFIGURACIÓN DEL SISTEMA DE ADQUISICIÓN DE DATOS

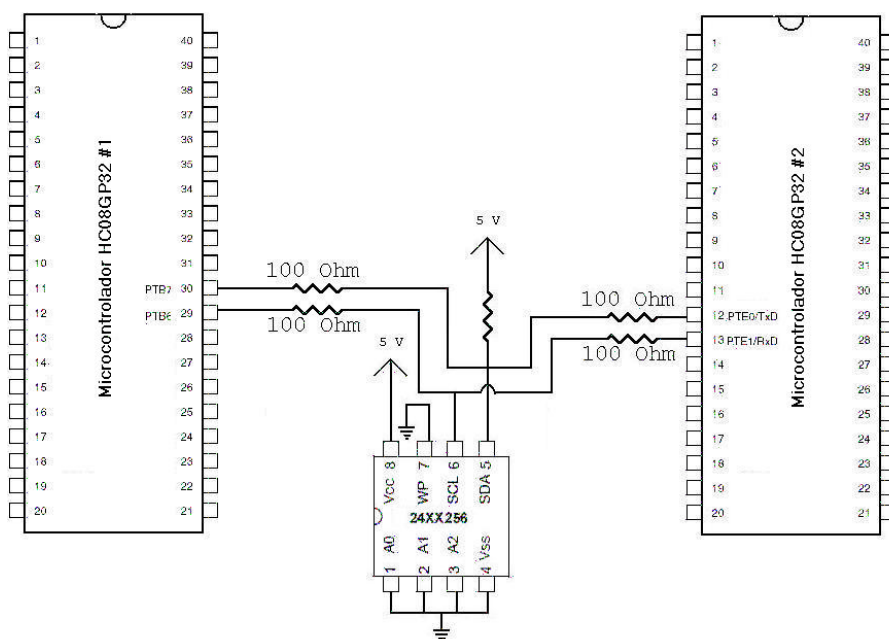
En este capítulo se expone la forma como toda la teoría que se estudió anteriormente se junta de la manera apropiada. Se pretende mostrar la forma como los componentes de hardware y software se enlazan entre sí con el fin de conseguir los objetivos propuestos inicialmente.

### 7.1. HARDWARE

El hardware que compone el sistema es bastante variado. Se presentará cada uno y se explicará su función y configuración.

#### 7.1.1. Microcontroladores MOTOROLA HC08GP32CP:

En el dispositivo creado se usaron 2 microcontroladores MOTOROLA HC08GP32. Uno de ellos se usó para adquirir datos, dibujar en el LCD y escribir en memoria. El otro se dejó exclusivamente para el procesamiento de los datos escritos en la memoria. La figura 53 muestra el modo de conexión entre los microcontroladores y la memoria.



**Figura 53.** Conexión entre los microcontroladores y la memoria EEPROM

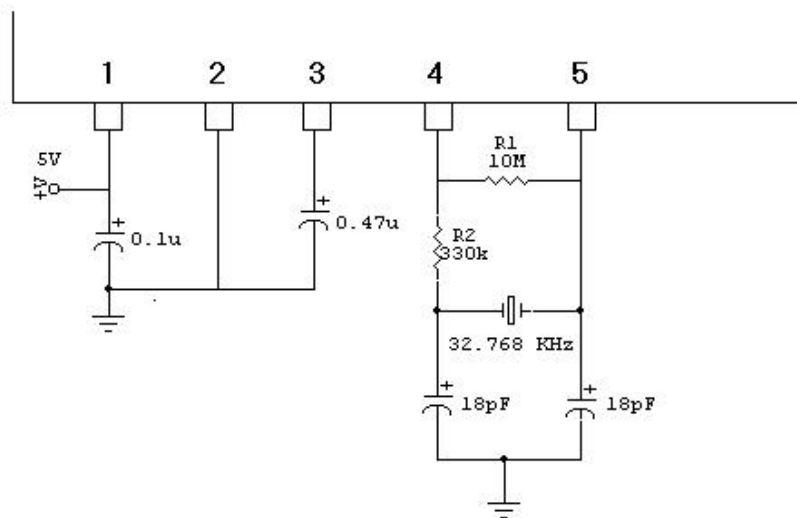
Se observa en la figura unas resistencias de protección entre los puertos del microcontrolador y la memoria. Esto se hace cuando los pines no son de tipo *open drain*, son delicados y hay que tener ciertas consideraciones al respecto.

Otro tipo de consideración es el de mantener un nivel de voltaje alto para cualquier tipo de pin que sea *open drain*. Esto se observa en el pin SDA de la memoria.

El microcontrolador #1 es el maestro y la memoria 24LC256 junto al microcontrolador # 2 son esclavos. Por lo tanto es el microcontrolador #1 quien genera la señal de reloj que entra en el pin SCL de la memoria y al pin PTE1 del microcontrolador #2.

El microcontrolador # 1 se designó como maestro porque es quien administra tareas y es el encargado de la adquisición de datos y de mostrar los resultados en la pantalla. El microcontrolador # 2 es esclavo porque depende del anterior para realizar su tarea de procesamiento de datos. En el apartado 7.2. se explicará la forma en que se programó cada microcontrolador.

Adicionalmente para el microcontrolador #1 se implementó una red de reloj externa para poder usar el PLL. La figura 54 muestra estas conexiones:



**Figura 54.** Circuito externo para usar el PLL del microcontrolador MOTOROLA HC08

#### 7.1.2. Pantalla de Cristal Líquido (LCD)

Este componente permite la visualización de los resultados de adquisición y procesamiento de los datos entregados por el fisiógrafo.

#### 7.1.2.1 Características del LCD:

El LCD usado es marca Hyundai, referencia HG25504. Cuenta con un controlador SED1330F. Este controlador es a quien se le mandan las instrucciones necesarias con el fin que se muestre los datos que se quieren. El LCD tiene una resolución de 128x256 píxeles. Necesita una fuente dual de +5 y -15 voltios. Estos voltajes son para el funcionamiento de la lógica y del contraste, respectivamente. La pantalla es del tipo STN de escala de grises y tiene un ciclo de trabajo de 1/128. Esto se refiere a la frecuencia de refrescamiento de la pantalla. Es un LCD de bajo costo y poderoso por sus comandos de programación.

#### 7.1.2.2. Configuración del LCD.

La apariencia física del LCD usado se muestra en la figura 55:



**Figura 55.** Pantalla de Cristal Líquido Usada en el proyecto

FUENTE: ALL ELECTRONICS. En: Graphic LCD. Estados Unidos, 2004

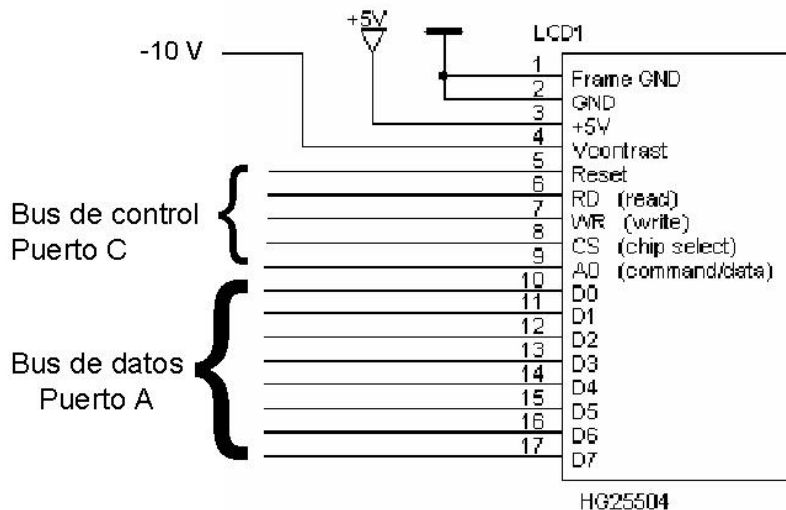
<<http://www.allectronics.com/cgi-bin/category.cgi?category=365&item=LCD-101&type=store>>

[Consulta: 9 de septiembre de 2004]

EL LCD cuenta con 17 pines. A grosso modo sus funciones son:

- Alimentación: pines 1, 2, 3, 4.
- Señales de control: pines 5, 6, 7, 8, 9.
- Bus de datos: pines desde el 10 hasta el 17.

La conexión usada se ilustra en la figura 56:



**Figura 56.** Conexión de la pantalla al microcontrolador

Más detalladamente, las funciones de cada uno de los pines se explican en la tabla 3:

pin	nombre	Función
1	FG	Chasis
2	Vss	Tierra
3	Vdd	Alimentación de 5 voltios para la lógica y el LCD
4	Vo	Contraste de el LCD
5	RES	Reset de LCD
6	RD	Señal de lectura
7	WR	Señal de escritura
8	CS	Señal de chip select
9	A0	Señal de selección de tipo de control
10	Bus de datos 0	Bus de datos
11	Bus de datos 1	Bus de datos
12	Bus de datos 2	Bus de datos
13	Bus de datos 3	Bus de datos
14	Bus de datos 4	Bus de datos
15	Bus de datos 5	Bus de datos
16	Bus de datos 6	Bus de datos
17	Bus de datos 7	Bus de datos

**Tabla 3.** Descripción de los pines del LCD

### 7.1.3. Memoria EEPROM serial:

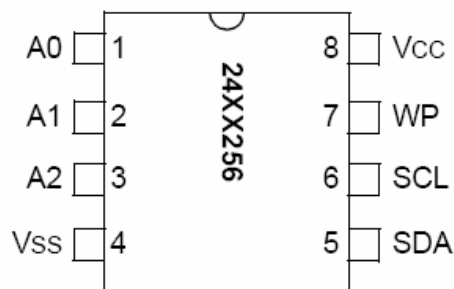
La memoria EEPROM es una parte importante del instrumento realizado. En la memoria se almacenan los datos adquiridos y se realizan las operaciones intermedias del procesamiento en frecuencia.

7.1.3.1. Características: La memoria EEPROM utilizada en el sistema posee las siguientes características:

- Marca Microchip® referencia 24LC256
- Almacena hasta 256 Kbits de información
- Usa protocolo de comunicación serial I2C
- Tecnología CMOS
- Modo de escritura de página de 64 bytes
- 1'000.000 ciclos de lectura/borrado
- protección contra carga electrostática

7.1.3.2. Configuración de la memoria:

La memoria tiene una disposición de pines como muestra la figura 57:



**Figura 57.** Diagrama de los pines de la memoria 24LC256

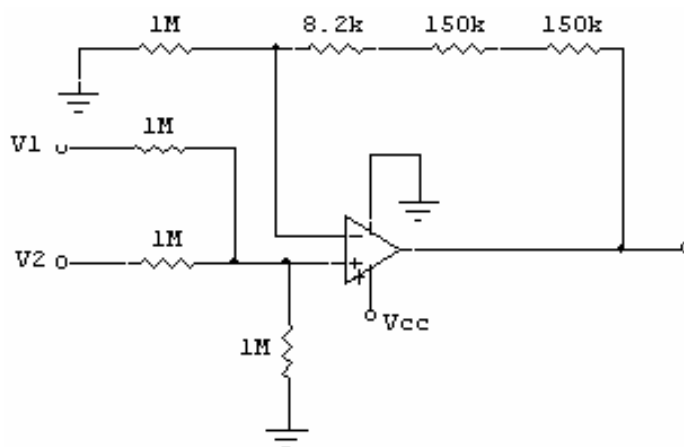
Los pines de este integrado tienen las siguientes funciones:

- A0, A1, A2: Son entradas usadas por la memoria para operaciones con múltiples dispositivos. Los diferentes valores que pueden tener estas entradas son como la identificación que tiene el integrado ante otros. Permite direccionar hasta 8 dispositivos diferentes. En caso de ser otras memorias, se podría aumentar la capacidad de almacenamiento hasta 2Mbit. Como en el proyecto se usa una sola memoria, estas tres entradas se conectan a cero voltios.
- Vss: conexión a tierra



derivaciones. Se requiere de una etapa de acondicionamiento para que sitúe la señal de voltaje entre 0 y 5 voltios.

Se creó un circuito que permite elevar el nivel de DC de la señal original a través de un amplificador operacional en configuración sumador y seguidor de voltaje, tal como se muestra en la figura 59:



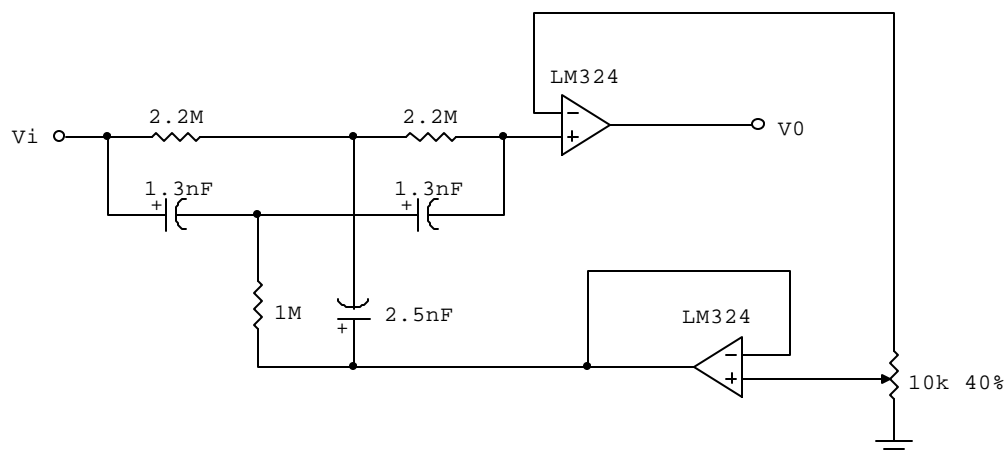
**Figura 59.** Configuración sumadora no inversora usando un amplificador operacional

FUENTE: TAMAYO, Alejandra, ZULUAGA, Luisa. Digitalización del Módulo de Electrocardiografía de un fisiógrafo tipo MK-III-P Narco BioSystems Divisions. Tesis de grado. Medellín, 2004.

[Consulta: Agosto de 2004]

Este circuito se deriva de una investigación previa realizada en el trabajo de grado “DIGITALIZACIÓN DEL MÓDULO DE ELECTROCARDIOGRAFÍA DE UN FISIÓGRAFO TIPO MK-III-P NARCO SCIENTIFIC BIO SYSTEMS DIVISIONS”. El resultado encontrado en ese entonces para este circuito se encontró adecuado y se usará para este trabajo.

En la etapa de acondicionamiento es importante el filtrado de la señal para eliminar componentes de frecuencia indeseadas. Para el caso del fisiógrafo, en las mediciones realizadas se encontró que la señal de salida del monitor de electrocardiografía tenía una componente de frecuencia de 60 Hz. Tomando como referencia el trabajo previamente citado, se usará un filtro elimina banda, el cual se muestra en la figura 60.

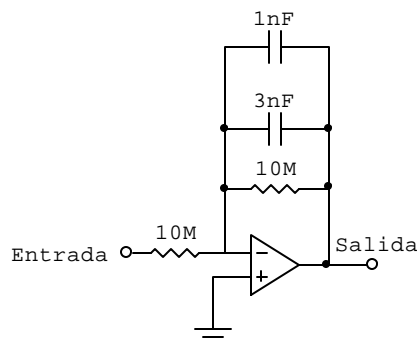


**Figura 60.** Filtro eliminabanda de 60 Hz.

FUENTE: TAMAYO, Alejandra, ZULUAGA, Luisa. Digitalización del Módulo de Electrocardiografía de un fisiógrafo tipo MK-III-P Narco BioSystems Divisions. Tesis de grado. Medellín, 2004.

[Consulta: Agosto de 2004]

Otro filtro implementado fue un filtro pasabajos, el cual me limita el ancho de banda de la señal adquirida. Se implementó un filtro con frecuencia de corte 150 Hz, el cual se ilustra en la figura 61:



**Figura 61.** Filtro de 150 Hz usado para limitar el ancho de banda de la señal adquirida.

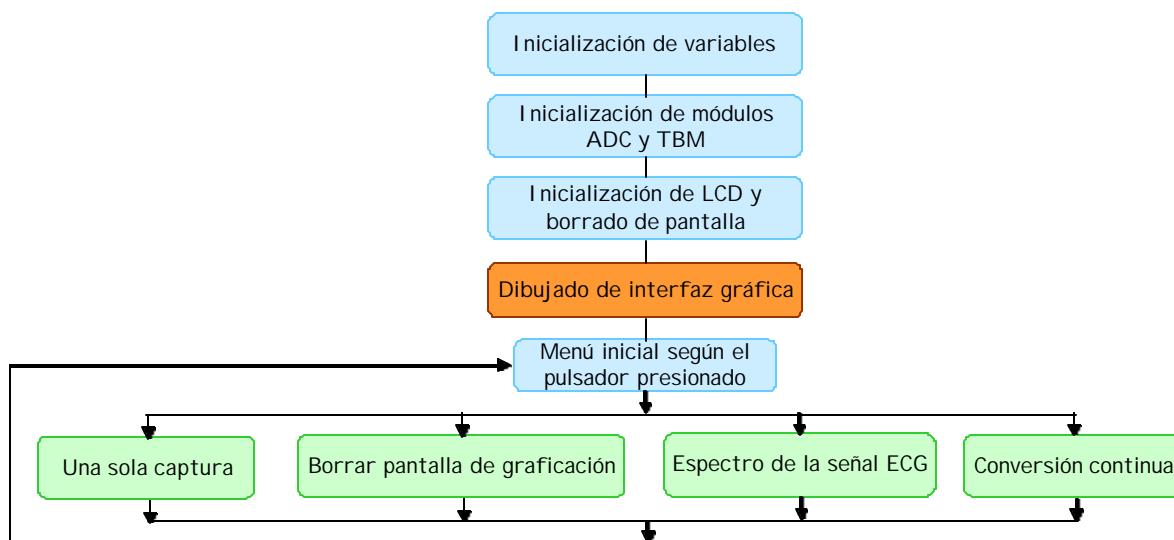
## 7.2. SOFTWARE

### 7.2.1. Microcontroladores MOTOROLA HC08GP32CP:

Desde la perspectiva del software se tiene que cada uno de los dos microcontroladores tienen una función específica, con el objetivo de respetar los tiempos de procesamiento y permitir que las tareas se hagan de una forma más rápida.

7.2.1.1. Microcontrolador #1: Este microcontrolador tiene, de forma general, las siguientes funciones, representadas en la figura 62:

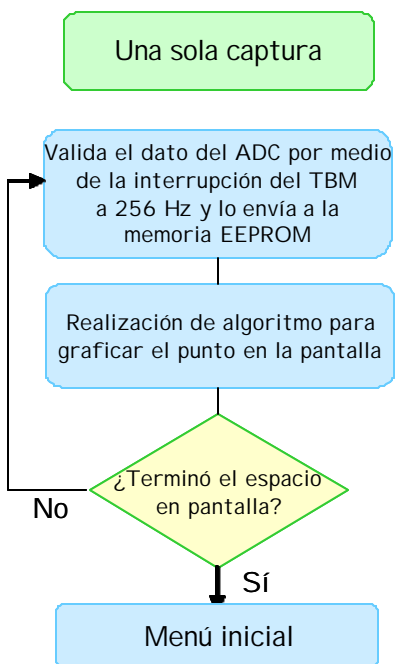




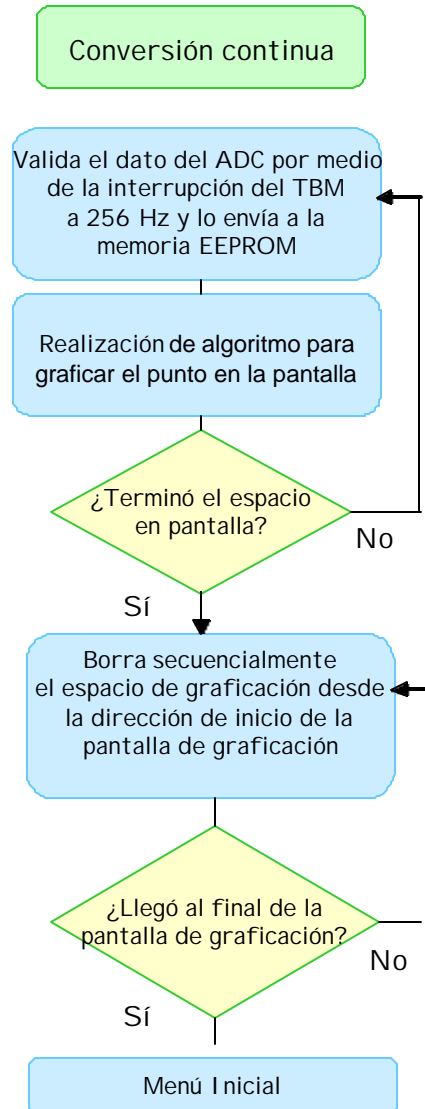
**Figura 62.** Diagrama de flujo general del microcontrolador #1

Más específicamente:

- Realizar una sola captura (figura 63) o una captura en modo continuo (Figura 64). En la primera modalidad, se visualiza sólo un pantallazo de la señal adquirida. En la segunda modalidad, se realizan múltiples capturas de modo indefinido, hasta que el usuario de la orden de finalizar.

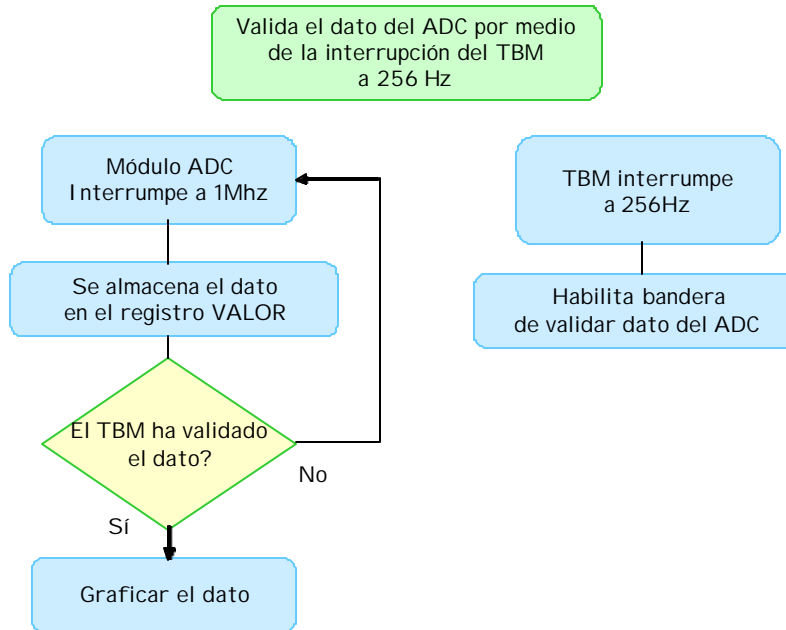


**Figura 63.** Modo de graficación en una sola captura.



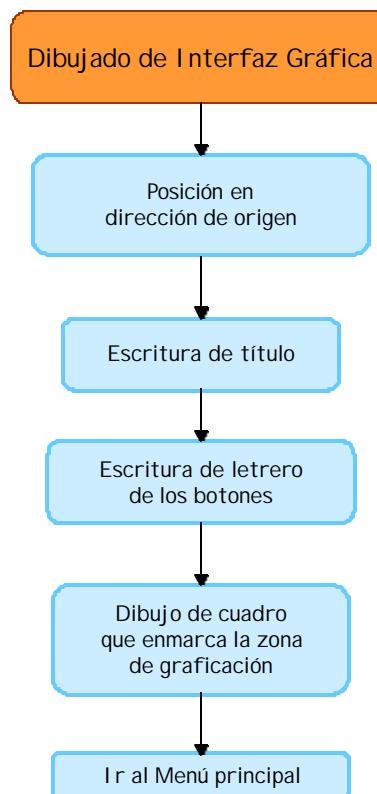
**Figura 64.** Modo de graficación continua.

- Capturar la señal ECG proveniente del Fisiógrafo por medio del conversor análogo a digital, a una tasa de muestreo de 256Hz determinada por el módulo TBM. Esto se muestra en la figura 65:



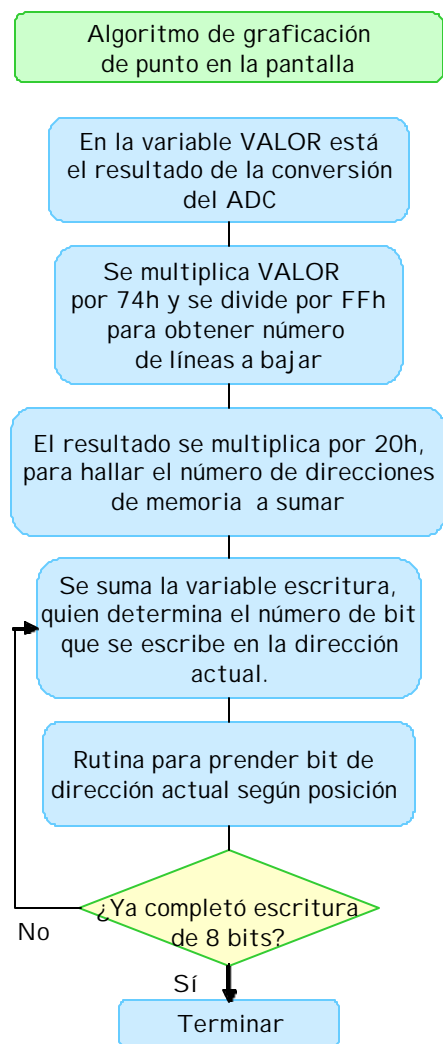
**Figura 65:** Procedimiento de autenticación de dato adquirido por el módulo ADC por parte del TBM

- proveer la interfaz gráfica en el LCD, mostrado en la figura 65:



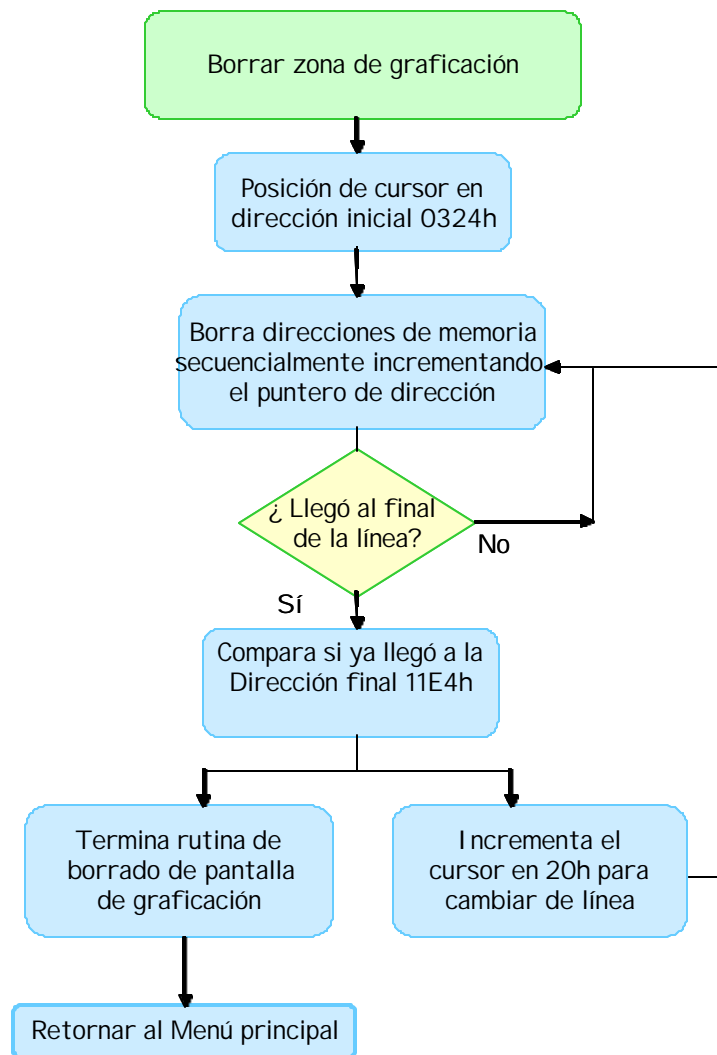
**Figura 66.** Diagrama de flujo para dibujar la interfaz gráfica en el LCD

- Dibujar la señal capturada en el LCD. Este procedimiento se observa en el diagrama de flujo de la figura 67:



**Figura 67.** Procedimiento de graficación de un punto en la pantalla

- Enviar a la memoria los datos capturados para ser procesados por el microcontrolador #2
- Leer los datos ya procesados para ser graficados en el LCD
- Borrar la pantalla de graficación, mostrado en la figura 68:



**Figura 68.** Secuencia de borrado de la zona de graficación

Para la programación de estas funciones fue necesario usar los diferentes módulos que posee el microcontrolador. A continuación, se explicará cuáles se usaron y su configuración.

- *Clock Generator Module*: (CGMC): Este módulo es un multiplicador de frecuencia. Se usó para incrementar la velocidad del bus de datos. A partir de un cristal de 32,768 KHz se incrementó la velocidad del bus de datos hasta 8Mhz. Los registros de este módulo de configuraron de la siguiente manera:

Registro PCTL: 00000010b

Registro PBWC: 10000000b

Registro PMSH: 03h

Registro PMSL: D1h

Registro PMRS: D0h

Registro PMDS: 01h

- *Timebase Module*: (TBM). Este módulo permite programar interrupciones periódicas a unos tiempos seleccionados por el usuario. Para el propósito del dispositivo, el TBM es quien autoriza la autenticación del dato capturado por el ADC. La configuración del módulo se hace a través del único registro:

Registro TBCR: 00110100b

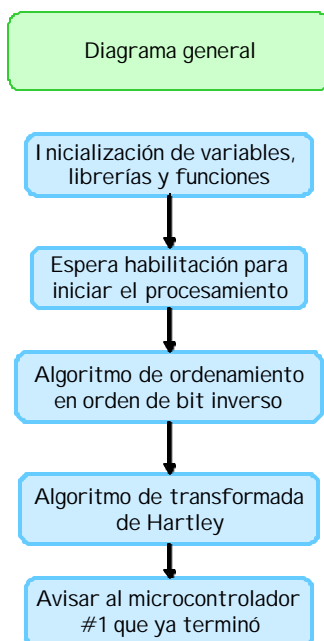
Mediante esta configuración, el TBM interrumpe cada 3.9ms, lo que da una frecuencia de muestreo de 256 Hz.

- *Analog to Digital Converter*: (ADC). El módulo ADC es el encargado de asignar una cuantificación a la señal de entrada proveniente del fisiógrafo. El módulo se configura a través de los siguientes registros:

Registro ADSCR: 60h

Registro ADCLK: 70h: divisor entre 8 para obtener funcionamiento correcto

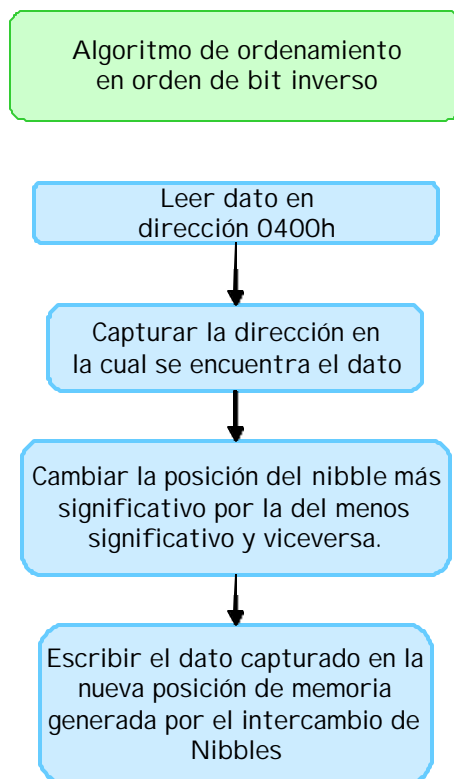
7.2.1.2. Microcontrolador #2: Este microcontrolador tiene las funciones mostradas a través del diagrama de bloques de la figura 69:



**Figura 69:** Funcionamiento global del microcontrolador #2

Más detalladamente:

- Algoritmo de ordenamiento en orden de bit inverso: Este algoritmo pone los datos capturados en el orden necesario para poder realizar las operaciones pertinentes. Desde la dirección 0000h hasta la 03FFh están los datos capturados. A partir de la dirección 0400h se escribirán los datos en orden de bit inverso. La figura 70 muestra esto en detalle:



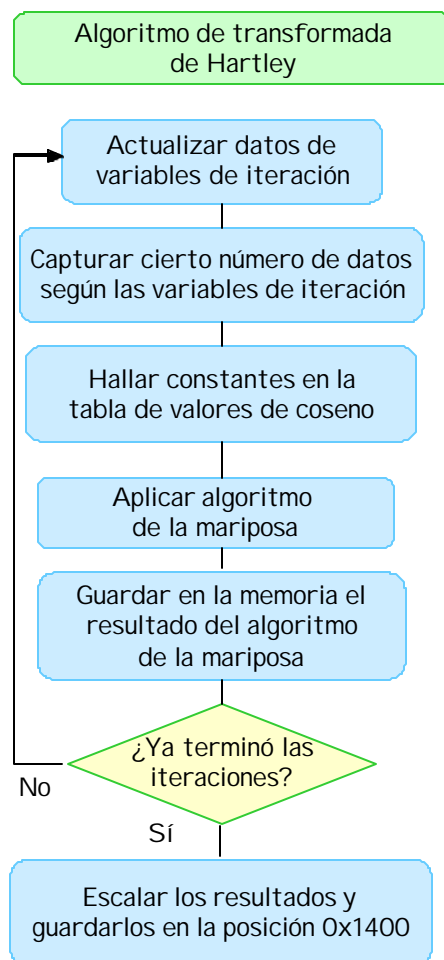
**Figura 70.** Algoritmo de ordenamiento en bit inverso

- Algoritmo de la Transformada de Hartley: Este algoritmo permite hallar las componentes en frecuencia de un conjunto de datos muestreados en el tiempo. La figura 71 ofrece un resumen de este procedimiento. El correcto desempeño de la transformada de Hartley depende de que se tomen la cantidad de datos necesaria para poder efectuarla. Se tiene que:

Frecuencia de muestreo =  $F_s = 256 \text{ Hz}$

Para abarcar una buena cantidad de periodos de señal ECG, se van a adquirir datos por 4 segundos. Por lo tanto:  $256 \text{ Hz} \times 4 \text{ seg.} = 1024 \text{ datos}$

Si en los 4 segundos de recogen 1024 datos, se requiere  $N = 1024$  datos a la entrada del algoritmo de la Transformada de Hartley.



**Figura 71.** Diagrama de flujo de la transformada de Hartley

7.2.2. Pantalla de Cristal Liquido (LCD): El LCD cuenta con ciertos comandos que permiten programar el LCD para que muestre los datos de la manera que se desea. Los comandos de los cuales dispone el LCD son:

- *System set*: Inicializa el LCD y configura los parámetros básicos de operación
- *Disp ON/OFF*: Habilita o deshabilita la visualización, al igual que establece la velocidad de parpadeo del cursor.
- *CSR FORM*: Determina el tipo de cursor que se usa
- *CSR DIR*: configura la dirección en la cual el cursor se autoincrementa
- *MWRITE*: Escribe datos en la dirección del cursor
- *MREAD*: Lee datos de la dirección donde se encuentra el cursor
- *CSRW*: lee el contenido de la dirección donde se encuentra el cursor.
- *CSRR*: lee la dirección donde está el cursor.



De nada sirve tener muchos comandos si no se sabe en dónde se van a ubicar los datos en la pantalla. A continuación se expone el modo como se realizó la asignación de direcciones

#### 7.2.2.1. Asignación de direcciones.

El LCD dispone de tres subpantallas, (*layers* del inglés, niveles), las cuales tienen cada una asociada una dirección de memoria para poder llevar a cabo la labor de graficación. El fabricante especifica que de las tres pantallas, la pantalla en *layer 1* debe ser de modo texto solamente, mientras que la pantalla ubicada en *layer 2* y 3 pueden ser gráficas o de texto. Para los propósitos del dispositivo, *layer 1* se especificó como texto y *layer 2* se configuró como tipo gráfica.

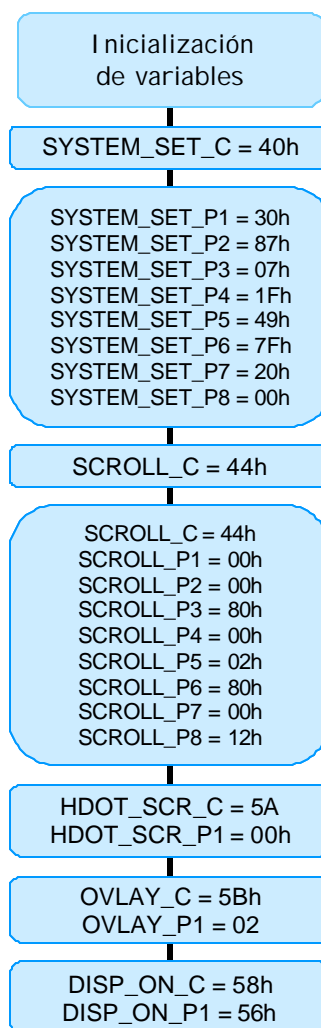
- *Layer 1*: modo texto.
  - Resolución horizontal: 256 píxeles.
  - Resolución vertical: 128 píxeles.
  - Para modo texto, el carácter tiene una resolución de 8 píxeles horizontales por 8 píxeles verticales.
  - Número de caracteres que caben en una línea:
 
$$\frac{\text{resolución horizontal}}{\text{resolución horizontal del caracter}} = \frac{256}{8} = 32 \text{ caracteres}$$
  - Número de líneas que caben horizontalmente:
 
$$\frac{\text{resolución vertical}}{\text{resolución vertical del caracter}} = \frac{128}{8} = 16 \text{ líneas}$$
  - Memoria requerida: 32 caracteres x 16 líneas = 512 bytes
  - Dirección de inicio: 0000h
  - Dirección final: 0000h + 01FFh = 01FFh
- *Layer 2*: modo gráfico.
  - Resolución horizontal: 256 píxeles.
  - Resolución vertical: 128 píxeles.
  - Para modo gráfico, el carácter tiene una resolución de 8 píxeles horizontales por 1 píxel vertical.
  - Número de caracteres que caben en una línea:
 
$$\frac{\text{resolución horizontal}}{\text{resolución horizontal del caracter}} = \frac{256}{8} = 32 \text{ caracteres}$$
  - Número de líneas que caben horizontalmente:

$$\frac{\text{resolución vertical}}{\text{resolución vertical del caracter}} = \frac{128}{1} = 128 \text{ líneas}$$

- Memoria requerida: 32 caracteres x 128 líneas = 4096 bytes
- Dirección de inicio: 0200h
- Dirección final: 0200h + 0FFFh = 11FFh

Ya con las direcciones de inicio y finalización de cada *layer*, se puede proceder a enviar información a esas direcciones para realizar la graficación de datos y escritura de palabras.

7.2.2.2. La inicialización del LCD se realiza según el diagrama de flujo que se ilustra en la figura 72:



**Figura 72.** Diagrama de flujo de la inicialización del LCD

- *System set*: Este comando configura el LCD de la siguiente forma:
  - Habilita generador de caracteres interno
  - Altura de caracter de 8 píxeles
  - Método de partición tipo panel sencillo
  - Sin corrección por *offset*
  - Ancho de caracter de 8 píxeles
  - Rango de dirección horizontal de 32 Bytes
  - Altura de cuadro de 128 líneas
  - Rango de memoria virtual de 0 Bytes
- *Scroll*: determina los intervalos de memoria entre los cuales se encuentran las subpantallas. Hay una subpantalla de texto entre 0000h y 01FFh y la subpantalla gráfica entre 0200h y 11FFh.
- *HDOT SCR*: permite desplazamiento horizontal píxel a píxel. No tiene disponible esta opción.
- *OVLAY*: determina la composición entre las subpantallas y el modo como éstas se superponen. Está en modo OR exclusiva.
- *DISP ON*: prende o apaga la pantalla, al igual que configura la rata de parpadeo del cursor. También permite seleccionar si se desea un cursor en bloque o de línea. El LCD tiene deshabilitado el cursor.

#### 7.2.2.3. Rutinas implementadas:

Para un manejo más modular del LCD, se dotó el programa de rutinas que facilitan el entendimiento del programa y facilidad de operación. Las rutinas implementadas son:

- *Borrar LCD*: sirve para dejar la pantalla limpia para empezar a graficar
- *Borrar pantalla*: Sirve para borrar exclusivamente la sección de la pantalla en donde se dibuja la señal ECG
- *Escribir comando / escribir instrucción*: Es la rutina que tiene por objetivo validar el dato que se tiene en el bus de datos.
- *Dibujar píxel*: es la rutina que activa un píxel en una dirección determinada según el dato adquirido por el microcontrolador a través de conversor análogo digital.

#### 7.2.3. Memoria EEPROM serial:

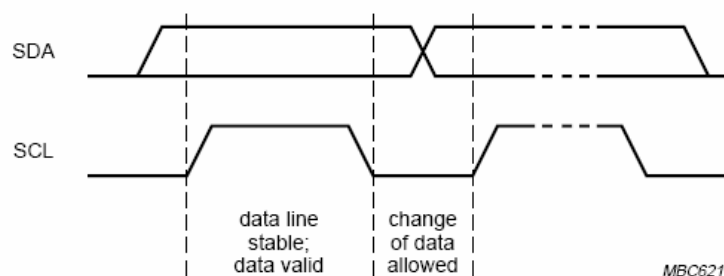
Para poder manejar la memoria fue necesario realizar dos programas. Uno en assembler, el cual se preocupa por recoger datos escritos al LCD y escribirlos en la memoria. Luego de ser procesados los datos, leer de la memoria y llevarlos al microcontrolador para que el se encargue de llevarlos al

LCD. En sí, los programas son muy parecidos porque necesitan implementar por software el protocolo serial I2C. A continuación se expone este protocolo.

#### 7.2.3.1. Protocolo de comunicaciones seriales I2C orientado a la memoria EEPROM serial:

El bus I2C es una interfaz serial en dos alambres implementados en microcontroladores y periféricos diversos. Este protocolo requiere de una línea de reloj (SCL) y una línea de datos (SDA) para efectuar la comunicación. Los dispositivos conectados a este bus pueden ser maestros o esclavos. Un maestro es un dispositivo que inicia la transferencia, genera la señal de reloj y termina la transferencia. Un dispositivo esclavo es el que es direccionado por el maestro. El protocolo I2C permite múltiples maestros en el mismo bus al igual que múltiples esclavos, cada uno con una dirección que lo identifica.

La transferencia de bit se rige según el diagrama de tiempo ilustrado en la figura 73:



**Figura 73.** Transferencia de bit en el bus I2C

FUENTE: PHILIPS. En: The I2C-Bus Specification versión 2.1. [En línea] Estados Unidos, 2000.

<[www.philips.com/i2c](http://www.philips.com/i2c)>

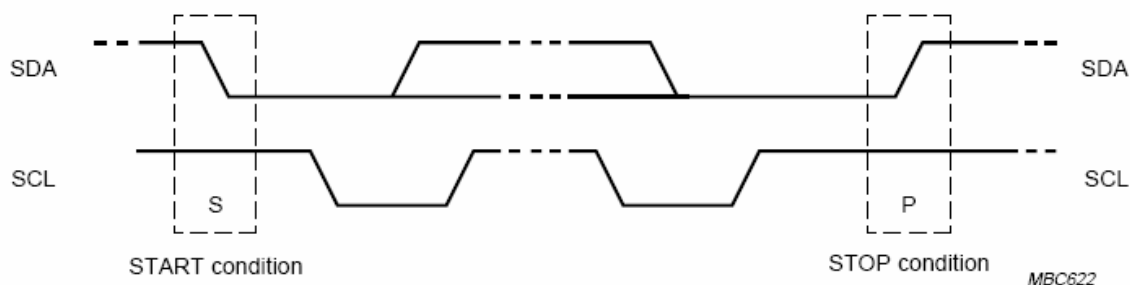
[Consulta: agosto 10 de 2004]

Los datos que se transfieren por la línea de SDA deben ser estables mientras la línea SCL está en alto. Por lo tanto, el nivel lógico en SDA sólo puede cambiar mientras SCL esté en bajo.

Ahora, ¿qué instrucciones son necesarias para poder manejar un bus de datos I2C?

- Condición de START: se usa para iniciar una transferencia de datos. Para realizar un START se debe programar una transición de nivel alto a bajo en la línea de SDA mientras la línea SCL está en nivel alto.
- Condición de STOP: Se usa para interrumpir la transferencia de datos. Para realizar un STOP se debe programar una transición de nivel bajo a alto de voltaje en la línea de SDA mientras la línea SCL está en nivel alto.

La figura 74 muestra gráficamente estas condiciones:



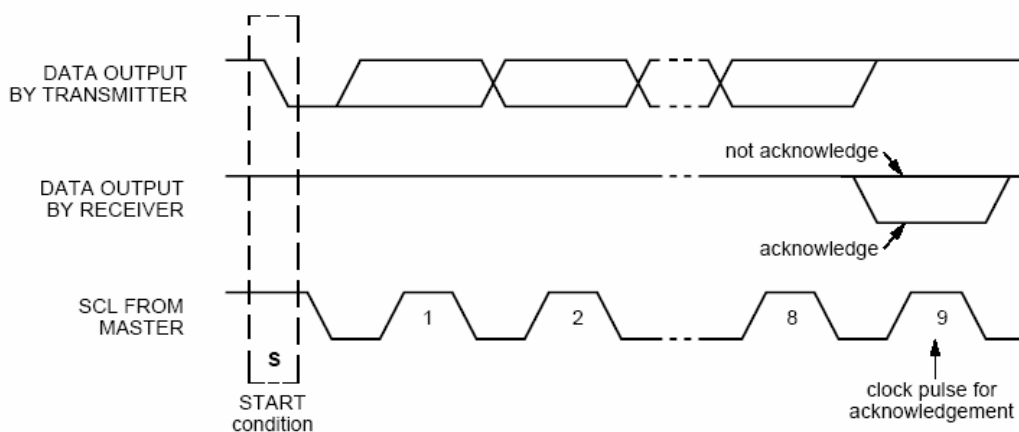
**Figura 74.** Condiciones de START y de STOP

FUENTE: PHILIPS. En: The I2C-Bus Specification versión 2.1. [En línea] Estados Unidos, 2000.

<[www.philips.com/i2c](http://www.philips.com/i2c)>

[Consulta: agosto 10 de 2004]

- Bit de *ACKNOWLEDGE*: Cada byte transmitido debe ir seguido de un bit de acknowledge. Este bit es de carácter obligatorio. Esto consiste en que el receptor (sea maestro o esclavo) debe llevar a nivel bajo de voltaje la línea de SDA mientras la línea de SCL está en alto. La figura 75 muestra el modo como debe llevarse a cabo este procedimiento:



**Figura 75.** Condiciones de START y de STOP

FUENTE: PHILIPS. En: The I2C-Bus Specification versión 2.1. [En línea] Estados Unidos, 2000.

<[www.philips.com/i2c](http://www.philips.com/i2c)>

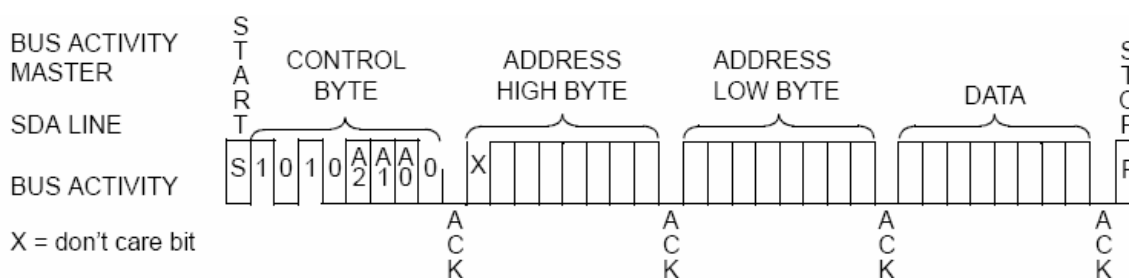
[Consulta: agosto 10 de 2004]

En lo que se refiere específicamente a la memoria 24LC256, se debe seguir cierta secuencia para poder escribir o leer datos.

- Byte de control: Es el primer byte transmitido luego de la condición de START. Los cuatro primeros bits deben ser '1010', sea para operaciones de lectura o escritura. Los siguientes tres bits identifican un dispositivo entre varios esclavos que estén conectados al bus. Para

este caso en el que sólo hay una memoria conectada al bus, sería '000'. El último bit corresponde a la operación que se desea hacer. Si se desea leer de la memoria, se escribe un '1'. Si se desea escribir, se escribe un '0' es este bit.

- Dirección: Luego de especificar el dispositivo con el que se quiere comunicar, y de definir lo que se va a hacer, se debe hacer saber en qué dirección de memoria se va a efectuar la operación. La memoria EEPROM de 250 Kbits tiene un rango de direcciones que va desde 000h hasta 7FFFh. Se necesitan 15 bits para direccionarlo. Por eso se hace necesario enviar 2 bytes para especificar esta posición. En el primer byte, el bit más significativo no interesa su valor. Ya en el segundo byte, todos los bits se tienen en cuenta.
- Escritura: El proceso de escritura puede ser hecho de 2 formas: la escritura de byte individual o la escritura de página.
  - Escritura de byte individual: Luego de enviar el byte de control y la dirección de memoria, se envía un byte de datos. Luego de recibir el acknowledge, se envía una señal de STOP. La figura 76 muestra este procedimiento:



**Figura 76.** Procedimiento para escribir un byte de información

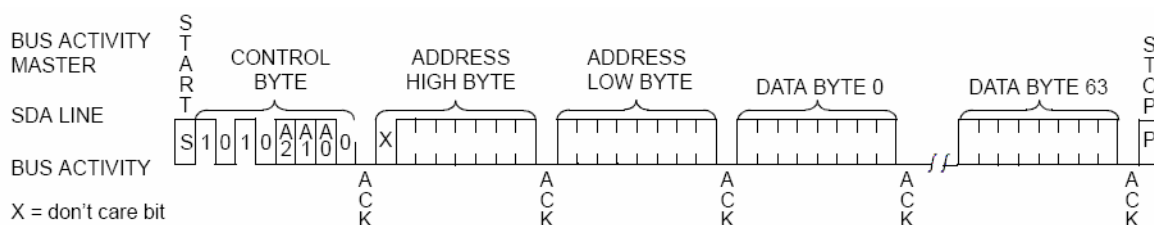
FUENTE: MICROCHIP. En: 24LC256 EEPROM Datasheet. [En Línea] Estados Unidos 2002.

<[www.microchip.com](http://www.microchip.com)>

[Consulta: 10 de agosto de 2004]

Se puede apreciar en la figura que luego de ser transmitido cada byte, se debe esperar a que el esclavo emita la señal de acknowledge.

- Escritura de página: Es igual al de escritura de página individual, solo que luego de enviar el dato y recibir la señal de acknowledge, no se genera condición de STOP, sino que se envía otro dato. No hay que preocuparse por incrementar la dirección donde se está haciendo la escritura en la memoria, la memoria por sí misma incrementa la dirección. Al terminar de transmitir el byte de información número 63, se debe recibir la señal de acknowledge y emitir la señal de condición de STOP. Este proceso se ilustra en la figura 77:



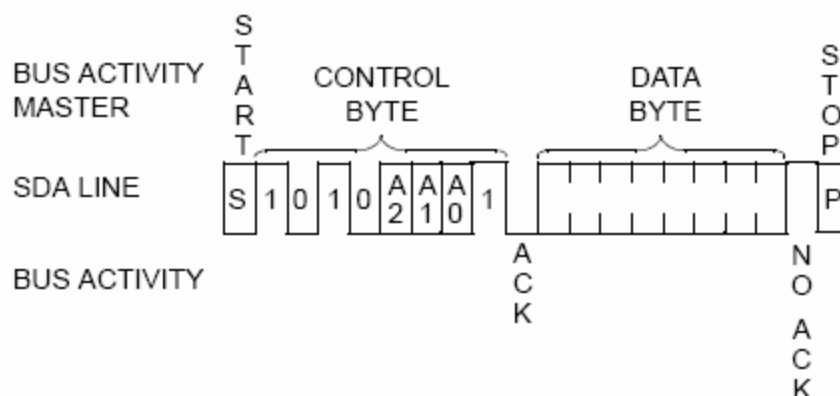
**Figura 77.** Procedimiento para escribir un byte de información

FUENTE: MICROCHIP. En: 24LC256 EEPROM Datasheet. [En Línea] Estados Unidos 2002.

<www.microchip.com>

[Consulta: 10 de agosto de 2004]

- Lectura: La forma como se inician las operaciones de lectura son del mismo modo que se inician las operaciones de escritura, la única diferencia es que el bit menos significativo del byte de control es puesto en '1'. Hay tres modos de lectura, a saber:
  - Lectura de dirección actual: La memoria 24LC256 tiene un contador interno que guarda la dirección previamente accesada incrementada en '1'. Tan pronto como se transmite el byte de control especificando lectura, y luego que el esclavo emite la señal de acknowledge, el esclavo transmite el dato de 8 bits. Al recibir esto, el maestro no emite señal de acknowledge, pero si emite condición de STOP, con el fin de terminar el proceso. La figura 78 explica gráficamente el proceso anterior:



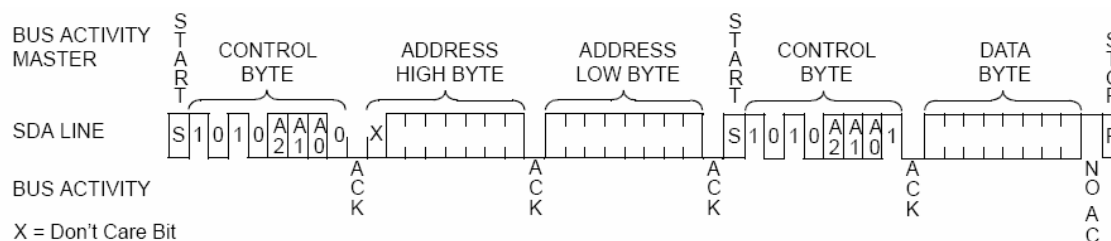
**Figura 78.** Procedimiento para escribir un byte de información

FUENTE: MICROCHIP. En: 24LC256 EEPROM Datasheet. [En Línea] Estados Unidos 2002.

<www.microchip.com>

[Consulta: 10 de agosto de 2004]

- Lectura aleatoria: Para realizar este tipo de lectura, se debe especificar primero la dirección de memoria que se va a leer, usando el bit menos significativo del byte de control en '0'. Luego que la memoria ha puesto su contador en el lugar correcto, se da otra condición de START y el byte de control se configura en modo de lectura. Después de recibido el *acknowledge* el esclavo transmite el byte. El maestro genera condición de STOP para terminar el proceso. La figura 79 muestra más detalladamente este proceso:



**Figura 79.** Procedimiento para escribir un byte de información

FUENTE: MICROCHIP. En: 24LC256 EEPROM Datasheet. [En Línea] Estados Unidos 2002.

<[www.microchip.com](http://www.microchip.com)>

[Consulta: 10 de agosto de 2004]

- Lectura secuencial: Es iniciada de la misma forma que la lectura aleatoria. Cuando el maestro recibe el primer byte de información, en lugar de generar condición de STOP, genera el bit de *acknowledge* y esto permite que el esclavo le trasmita el siguiente byte de información correspondiente a la dirección de memoria siguiente. Si el contador de memoria que tiene la memoria 24LC256 llega al máximo, que es 7FFFh, se desborda e inicia desde la dirección 0000h.

#### 7.2.3.2. Funciones de la memoria EEPROM 24LC256:

Desde el punto de vista del programa, la memoria cumple una función de soporte de información. El microcontrolador no cuenta con la memoria suficiente para almacenar volúmenes grandes de datos. Además, entre las funciones de procesamiento se cuenta con una transformación en frecuencia por medio de la Transformada de Hartley y las operaciones intermedias de esto se guardan en la memoria. Los dos microcontroladores usados en el instrumento creado acceden la memoria de la siguiente manera:



- Microcontrolador 1: Muestra datos de la señal ECG y los escribe en la memoria entre las direcciones 0000h y 03FFh. Luego, después que el microcontrolador 2 termina de procesar los datos, el microcontrolador 1 lee las 1024 componentes en frecuencia, localizadas entre las direcciones 1400h y 17FFh para ser graficadas en el LCD.
- Microcontrolador 2: Lee los datos almacenados por el microcontrolador 1 entre las direcciones 0000h y 03FFh y los ordena en *orden de bit inverso* entre las direcciones 0400h y 07FFh. Los resultados de las operaciones intermedias se almacenan entre las direcciones 0400h – 0BFFh y 0C00h – 03FFh. Luego el vector final lo almacena entre las direcciones 1400h y 17FFh.