



Biblioteca Universitaria

La consulta de este documento, que se lleva a cabo mediante claves de identificación y responsabilidad personal, es posible exclusivamente para fines de estudio personal o investigación. No se autoriza a reproducir su texto más que en forma de breves citas entrecomilladas, indicando el nombre del autor y la fuente. Por tanto, no se permite descargar, copiar, transformar ni grabar su contenido.

Dispositivo acelerómetro microcontrolado

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Carrera

DISPOSITIVO ACELERÓMETRO
MICROCONTROLADO

(Microcontrolled accelerometer device)

Para acceder al Título de

INGENIERO TÉCNICO DE TELECOMUNICACIÓN

Autor: Javier del Hierro González

06 - 2009

Dispositivo acelerómetro microcontrolado

TÍTULO	Dispositivo Acelerómetro Microcontrolado			
AUTOR	Javier del Hierro González			
DIRECTOR	Mario Mañana Canteli			
TITULACIÓN	INGENIERO TÉCNICO DE TELECOMUNICACIÓN	26-06-2009	06-2009	TOMO I DE I

Dispositivo acelerómetro microcontrolado

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL PROYECTO FIN DE CARRERA

Realizado por: Javier del Hierro González

Director del PFC: Mario Mañana Canteli

Título: “Dispositivo acelerómetro microcontrolado”

Title: “Microcontrolled accelerometer device”

Presentado a examen el día:

para acceder al Título de

**INGENIERO TÉCNICO DE TELECOMUNICACIÓN,
ESPECIALIDAD EN SISTEMAS ELECTRÓNICOS**

Composición del Tribunal:

Presidente (Apellidos, Nombre):

Secretario (Apellidos, Nombre):

Vocal (Apellidos, Nombre):

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del PFC
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Proyecto Fin de Carrera N°
(a asignar por Secretaría)

Resumen

En este proyecto se describe un sistema que medirá la aceleración, en tres ejes, de cuerpos en movimiento. El objetivo fundamental de la medición es poder obtener gráficas que permitan estudiar el comportamiento de deportistas, enfermos de Parkinson... El sistema de medición se realizará de manera tal que no interfiera en la actividad normal del individuo en cuestión. Así como también con la suficiente autonomía y sencillez de manejo, que procure un cómodo y fácil manejo.

Agradecimientos

Quiero agradecer su ayuda a todas aquellas personas que más o menos me han facilitado el camino para que este proyecto llegara a buen fin.

A mi director de proyecto, que siempre ha confiado en mí, y eso me ha servido para tener más seguridad en mi trabajo, y no sólo con este proyecto. Gracias.

A profesores de la Universidad de Cantabria, que me han enseñado una gran cantidad de conocimientos y valores para aplicar tanto en mi futuro profesional como en el día a día como persona.

A mis compañeros, por haberme acompañado en el camino y por compartir buenos y malos momentos.

A toda la gente de foros e internet que siempre me han prestado su ayuda de forma desinteresada, y que no sólo resultaron de inspiración y guía, sino también de compañía en muchos momentos difíciles en que no se lograba ver la luz.

A todos vosotros, y a todos los que me haya podido dejar por descuido en el tintero, muchísimas gracias, este proyecto es para vosotros.

**Cuando me equivoco me ayudáis, cuando dudo me aconsejáis
y siempre que os llamo estáis a mi lado.**

**Este proyecto está dedicado a mi padre, a mi madre y a mi
hermana. Gracias, os quiero.**

Lista de contenidos

1	Introducción.....	11
1.1	Organización de la memoria.....	12
2	Descripción del objetivo del proyecto.....	13
2.1	Descripción.....	13
2.2	Especificaciones.....	13
2.2.1	Requisitos de usuarios.....	13
2.2.2	Requisitos de sistema.....	14
2.3	Ámbito.....	16
3	Solución propuesta.....	17
3.1	Descripción.....	17
3.2	Hardware.....	17
3.2.1	Medida de las fuerzas sobre tres ejes.....	17
3.2.1.1	Acelerómetro triaxial MMA7260QT.....	20
3.2.2	Microcontroladores.....	22
3.2.2.1	Microcontrolador PIC 16F873.....	25
3.2.3	Memorias.....	30
3.2.3.1	Memoria EEPROM 24LC256.....	31
3.3	Software del PIC.....	31
3.4	Software externo.....	32
4	Simulación del prototipo.....	36
5	Diseño y construcción del circuito.....	37
6	Posibles mejoras.....	41
7	Documentación anexa.....	42
7.1	Anexo1. Código programa PIC.....	42
7.2	Anexo2. Código programa VB.....	43
7.3	Anexo3. Capas del circuito.....	46

Lista de gráficos

Figura1. Simplificación del modelo de transductor.....	21
Figura2. Diagrama de bloques Von Newman.....	23
Figura3. Diagrama de loques Harvard.....	24
Figura4. Esquemático del patillaje PIC16F876/873.....	25
Figura5. Esquema de simulación en Proteus.....	36
Figura6. Ventana de diálogo Orcad.....	37
Figura7. Esquema del circuito.....	38
Figura8. Ventana de diálogo Orcad	38
Figura9. Ventana de diálogo Orcad	38
Figura10. Esquema del circuito.....	39
Figura11. Esquema del circuito final.....	40

Lista de tablas

Tabla1. Descripción de los pines g-Select.....	21
Tabla2. Resumen de instrucciones.....	26
Tabla3. Resumen de instrucciones.....	26
Tabla4. Resumen de instrucciones.....	27

1 Introducción

La idea de este sistema nació a partir de la necesidad de estudiar a qué fuerzas está sometido un deportista durante el desarrollo de su actividad física, de manera desatendida y prolongada. Este concepto planteaba varias cuestiones.

El sistema debería tener una velocidad de lectura y procesamiento de datos adecuados, acordes a un número de muestras por segundo, que permitan construir gráficas reales. Este hecho a su vez genera otras necesidades. Se precisa un dispositivo de memoria para el almacenamiento de datos muy elevado.

Además, la adquisición de datos por parte del sistema requiere de una (al menos) interfaz de usuario para poder visualizar, manipular e interpretar la información. Luego debería existir algún hardware de comunicación con un medio externo al sistema, como lo puede ser un ordenador.

Cada nueva necesidad genera nuevas opciones, toma de decisiones, valoraciones de integración de nuevos componentes en el sistema, pruebas de estos y efectos colaterales sobre los elementos ya existentes.

Esta memoria describirá todo el proceso de desarrollo del sistema. Desde su nacimiento como idea, las simulaciones como prototipo, distintas posibilidades a las decisiones tomadas, hasta la valoración de los resultados obtenidos mostrando posibles evolutivos al sistema desarrollado.

1.1 Organización de la memoria

En primer lugar, se describe en profundidad el objetivo del proyecto mostrando las necesidades claras que tiene el sistema. Con ellas se definen los requisitos de usuario, a partir de los cuales, teniendo en cuenta las restricciones que marcan los objetivos principales del sistema, se definen los requisitos del sistema. Una vez definidos los requisitos se realiza una especificación completa del sistema, expresando claramente sus funcionalidades y su alcance; qué hace y qué no hace.

Tras la especificación del producto, se muestra un breve estado del arte en la actualidad. No sólo del producto definido en comparación con otros que tengan similar definición, sino también de las tecnologías que se pueden usar en el sistema para cumplir las especificaciones definidas.

Una vez vistas las distintas opciones y las posibles soluciones existentes en la actualidad para desarrollar el proyecto, se detalla la solución propuesta. Cómo se llega a ella, razones de las decisiones tomadas y casos de uso del sistema. Después y a partir del análisis realizado se diseña un prototipo virtual. Con este diseño se procede a la realización física del prototipo para su entrada en pruebas.

En la siguiente parte se detalla el montaje del prototipo, con su manual de usuario, así como las pruebas realizadas con él comprobando que se cumplen los requisitos del sistema definido.

Estas pruebas producen unos resultados que se evalúan comprobando que el sistema cumple los requisitos definidos llegando el objetivo marcado. Los resultados se discuten, explicando efectos no esperados así como razonando los sí esperados o sus posibles desviaciones.

A partir de la discusión de las pruebas se realizan conclusiones sobre el sistema, abriendo nuevos caminos a posibles ampliaciones futuras, mostrando qué aspectos, funcionalidades o soluciones que exceden del ámbito de este sistema serían interesantes integrar.

2 Descripción del objetivo del proyecto

En este capítulo se describirá en detalle el objetivo del proyecto. El usuario marca una serie de funcionalidades que desea que tenga el producto, definiendo sus requisitos y restricciones, los cuales marcarán los requisitos del sistema.

2.1 Descripción

Este proyecto tiene como objetivo diseñar una herramienta especialmente pensada para medir el desarrollo de fuerzas provocadas sobre el cuerpo humano.

Los acelerómetros electrónicos son fabricados para medir la fuerza de aceleración e inclinación en una, dos o tres dimensiones o ejes.

Los acelerómetros también puede ser empleados para determinar la posición de un cuerpo, pues al conocerse su aceleración en todo momento, es posible calcular los desplazamientos que tuvo (considerando que se conocen la posición y velocidad original del cuerpo bajo análisis, y sumando los desplazamientos medidos se determina la posición).

2.2 Especificaciones

2.2.1 Requisitos de usuario

(Prioridad: 0 = Máxima, 5 = Mínima)

Identificador	REQ-USU-1 Mediciones triaxiales		
Descripción	El sistema debe tener capacidad para medir las fuerzas sobre cualquiera de los tres ejes (X, Y, Z)		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-USU-2 Autonomía		
Descripción	El sistema debe tener suficiente autonomía para garantizar al usuario un tiempo de funcionamiento preestablecido		
Importancia	Esencial	Prioridad	0
Estabilidad	Sujeto a cambios		

Identificador	REQ-USU-3 Salida de datos		
Descripción	El sistema debe poder comunicarse con algún medio externo, que permita al usuario ver, guardar y manipular las medidas durante su actividad física		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-USU-4 Indicadores		
Descripción	Deberán existir indicadores lumínicos para detallar en todo momento		

Dispositivo acelerómetro microcontrolado

	al usuario, qué proceso se está ejecutando		
Importancia	Negociable	Prioridad	3
Estabilidad	Total		

Identificador	REQ-USU-5 Persistencia de datos		
Descripción	El sistema debe mantener los datos recogidos de manera permanente, independientemente de posibles fallos de alimentación		
Importancia	Esencial	Prioridad	0
Estabilidad	Sujeto a cambios		

Identificador	REQ-USU-6 Dimensiones		
Descripción	El sistema debe poder ser fijado a extremidades físicas como brazos, piernas, cintura... así como también a otras superficies tales como plásticos		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-USU-7 Peso		
Descripción	El peso total del sistema nunca deberá ser un impedimento para el desarrollo normal de la actividad en cuestión		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-USU-8 Seguridad		
Descripción	Las tensiones de alimentación y corrientes que circulen por el sistema nunca deben representar un peligro para la integridad física del usuario. Debiendo por ello estar perfectamente aislado. Además no deben existir aristas, puntas o bordes cortantes		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-USU-9 Coste		
Descripción	El sistema producido en serie no debe tener un coste superior al de productos de funcionalidades semejantes existentes en el mercado		
Importancia	Negociable	Prioridad	2
Estabilidad	Sujeto a cambios		

2.2.2 Requisitos de sistema

Identificador	REQ-SIS-1 Alimentación		
Descripción	El sistema estará alimentado por tensiones comprendidas entre los 6 voltios y los 9 voltios		
Importancia	Esencial	Prioridad	0

Dispositivo acelerómetro microcontrolado

Estabilidad	Total
--------------------	-------

Identificador	REQ-SIS-2 Conector comunicación		
Descripción	El sistema dispondrá de conectores adecuados para poder comunicarse con medios externos, como por ejemplo ordenadores		
Importancia	Esencial	Prioridad	0
Estabilidad	Sujeto a cambios		

Identificador	REQ-SIS-3 Conector programación/test		
Descripción	El sistema dispondrá de conectores adecuados para poder programar el microcontrolador en el propio circuito y realizar test		
Importancia	Negociable	Prioridad	5
Estabilidad	Sujeto a cambios		

Identificador	REQ-SIS-4 Funcionalidad única		
Descripción	El sistema al activar un tipo de trabajo, debe desactivar cualquier otro		
Importancia	Esencial	Prioridad	1
Estabilidad	Sujeto a cambios		

Identificador	REQ-SIS-5 Inmunidad		
Descripción	El sistema no debe verse perturbado en su funcionalidad por radiaciones electromagnéticas o ruidos eléctricos		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-SIS-6 Robustez		
Descripción	El sistema debe estar preparado para asumir y resolver posibles fallos en la lectura o procesamiento de datos, ante operaciones para las que no ha sido diseñado o programado		
Importancia	Esencial	Prioridad	0
Estabilidad	Total		

Identificador	REQ-SIS-7 Eficiencia software		
Descripción	El sistema no deberá realizar operaciones inútiles para cumplir con su funcionalidad durante su funcionamiento normal		
Importancia	Negociable	Prioridad	4
Estabilidad	Total		

Identificador	REQ-SIS-8 Eficiencia hardware		
Descripción	El sistema no deberá realizar uso indebido o innecesario de la fuente de alimentación, que suponga picos de consumo injustificados		
Importancia	Negociable	Prioridad	4
Estabilidad	Total		

Identificador	REQ-SIS-9 Actualización		
Descripción	El sistema debe ser capaz de actualizarse a nivel de software en		

	caliente, es decir, sobre el propio circuito		
Importancia	Negociable	Prioridad	5
Estabilidad	Sujeto a cambios		

Identificador	REQ-SIS-10 Software guardar datos		
Descripción	El software encargado de la comunicación entre el circuito y el ordenador debe ser capaz de salvar los datos transmitidos, en distintos formatos. Por ejemplo .txt, .mdb...		
Importancia	Negociable	Prioridad	3
Estabilidad	Sujeto a cambios		

Identificador	REQ-SIS-11 Software construcción de gráficas		
Descripción	El software encargado de la comunicación entre el circuito y el ordenador construirá gráficas con la información recibida		
Importancia	Negociable	Prioridad	3
Estabilidad	Sujeto a cambios		

Identificador	REQ-SIS-12 Software multiplataforma		
Descripción	El software encargado de la comunicación entre el circuito y el ordenador debe desarrollarse en una tecnología multiplataforma		
Importancia	Negociable	Prioridad	5
Estabilidad	Sujeto a cambios		

2.3 Ámbito

El sistema no tiene como objetivos:

- Ofrecer, en tiempo real, información y medidas de la actividad que se esté llevando a cabo por el usuario.
- Definirse como una herramienta complementaria para el cuidado de la salud. Es decir, el usuario no debe tomar como referencia la información del sistema, ya que ésta será meramente informativa.
- Ser una herramienta de alta precisión, ya que sólo se pretende obtener información de carácter orientativo.

3 Solución propuesta

En este capítulo se describe detalladamente la solución propuesta. Por tratarse de un sistema de software embebido, se divide la exposición del sistema en dos partes: hardware y software.

En la primera se muestra el hardware del sistema, componentes, características y decisiones tomadas a partir de los requisitos del sistema.

En la segunda se describe el software embebido en el hardware, así como también el software externo al propio sistema encargado de la comunicación entre él mismo y un ordenador. El diseño a partir de los requisitos del usuario, cómo maneja los recursos y las características especiales de los distintos componentes del hardware, y posibles problemas en la implementación.

3.1 Descripción

Se propone un sistema de software embebido, por el cual se controla el hardware de una determinada manera para cumplir con los requisitos del sistema y añadir, si procede, funcionalidades útiles para el cumplimiento de objetivos.

El hardware se basa en un dispositivo programable central que controlará el funcionamiento del sistema, y será el responsable de adquirir y procesar la información generada por el acelerómetro.

El dispositivo poseerá entradas de control para el usuario, y por otra parte, entradas de datos de otros elementos como el acelerómetro, que empleará líneas de convertidores analógicos-digitales.

Las salidas del dispositivo central serán indicadores lumínicos y el puerto RS232 (señales Rx y Tx), que permitirá establecer una comunicación activa con el software externo al sistema de recepción e interpretación de datos.

3.2 Hardware

Los requisitos del sistema se desglosan en todos aquellos conceptos de hardware que son propios de los requerimientos anteriormente descritos. Estudiando para cada uno de ellos las prioridades, dependencias y efectos sobre el sistema global.

El principal objetivo de este sistema es medir las distintas fuerzas o aceleraciones que aparecen sobre la ejecución de una actividad física. De tal modo que el primer problema a resolver es cómo capturar esas medidas.

3.2.1 Medida de las fuerzas sobre tres ejes

Hoy en día no existen muchas posibilidades a la hora de pensar en cómo resolver el problema de medir fuerzas actuantes sobre ejes. Por eso, rápidamente se ha decidido emplear un acelerómetro.

Dispositivo acelerómetro microcontrolado

El acelerómetro es uno de los transductores más versátiles, siendo el más común el piezoeléctrico por compresión. Este se basa en el principio de que cuando se comprime un retículo cristalino piezoeléctrico, se produce una carga eléctrica proporcional a la fuerza aplicada.

Los elementos piezoeléctricos están hechos normalmente de circonato de plomo. Los elementos piezoeléctricos se encuentran comprimidos por una masa, sujeta al otro lado por un muelle y todo el conjunto dentro de una caja metálica. Cuando el conjunto es sometido a vibración, el disco piezoeléctrico se ve sometido a una fuerza variable, proporcional a la aceleración de la masa. Debido al efecto piezoeléctrico se desarrolla un potencial variable que será proporcional a la aceleración. Dicho potencial variable se puede registrar sobre un osciloscopio o voltímetro.

Este dispositivo junto con los circuitos eléctricos asociados se puede usar para la medida de velocidad y desplazamiento además de la determinación de formas de onda y frecuencia. Una de las ventajas principales de este tipo de transductor es que se puede hacer tan pequeño que su influencia sea despreciable sobre el dispositivo vibrador. El intervalo de frecuencia típica es de 2 Hz a 10 KHz.

Se utiliza en instrumentación, para detectar defectos en maquina rotativas, prediciendo por ejemplo el mal estado de un rodamiento o cojinete antes de que se provoque la avería. En bombas impulsoras de líquidos detectan los fenómenos de cavitación que pulsan a unas frecuencias características.

Los acelerómetros electrónicos son fabricados para medir la aceleración en una, dos o tres dimensiones. De manera que sea posible medir la aceleración en cada eje. Esta característica permite medir la inclinación de un cuerpo, puesto que es posible determinar con el acelerómetro la componente de la aceleración provocada por la gravedad que actúa sobre el cuerpo. Siendo un caso típico de aplicación hoy en día el de los teléfonos móviles o PDAs, entre otros.

Un acelerómetro también es usado para determinar la posición de un cuerpo, pues al conocerse su aceleración en todo momento, es posible calcular los desplazamientos que tuvo. Considerando que se conocen la posición y velocidad original del cuerpo bajo análisis, y sumando los desplazamientos medidos se determina la posición.

Los principios de funcionamiento que utiliza se basan en la unión de una masa m , a un dinamómetro cuyo eje está en la misma dirección que la aceleración. Por la ley fundamental de la Dinámica o Segunda Ley de Newton, se sabe que:

$$\vec{F} = m\vec{a}$$

donde \vec{F} representa las fuerzas que actúan sobre la masa m y \vec{a} , es la aceleración.

Dado que el dinamómetro indica el modulo de \vec{F} , y el valor de m es una característica del instrumento, se puede conocer el módulo de la aceleración \vec{a} :

Dispositivo acelerómetro microcontrolado

$$\vec{a} = \frac{\vec{F}}{m}$$

$$a = \frac{\partial v}{\partial t} = \frac{\partial^2 x}{\partial t^2}$$

La aceleración se define como la variación de la velocidad respecto del tiempo. Las unidades de medida son metros/segundos². Sin embargo también se emplea la unidad *g* que equivale a 9.8 m/s², la fuerza de gravedad que la tierra ejerce sobre los cuerpos a nivel del mar.

En el mercado existen diferentes modelos de acelerómetros, que pueden ser muy distintos unos de otros según la aplicación a la cual van destinados y las condiciones en las que han de trabajar. Hay dos parámetros principales a la hora de escoger el medidor adecuado, los rangos de funcionamiento de temperatura y la frecuencia. A continuación estudiaremos algunos de los más comunes, en base a sus principios de funcionamiento:

Electromecánicos: basan su funcionamiento en la retroalimentación. En esos instrumentos, una masa sensible a la aceleración, es mantenida muy cerca de una posición neutral o de cero desplazamiento, a través de la medición y realimentación de su movimiento. Una fuerza magnética proporcional es generada, que se opone al movimiento de la masa desplazada del punto neutral.

Este método cuenta con la ventaja de tener buena linealidad y eliminar el efecto de histéresis (cuando se le compara con los acelerómetros que emplean resortes).

Piezoeléctricos: son ampliamente utilizados para medición (de propósito general) de aceleración, impacto y vibración. Son básicamente transductores de movimiento con señales de salida de gran amplitud y tamaño pequeño.

El dispositivo emplea una masa en contacto directo con un dispositivo piezoeléctrico, o cristal. Cuando un movimiento variable es aplicado al acelerómetro, el cristal experimenta una fuerza de excitación variable ($F = m \cdot a$).

Piezoresistivos: son esencialmente galgas extensiométricas semiconductoras con gran sensibilidad. Una mejor sensibilidad es crítica en la medición de vibración, ya que permite la miniaturización del acelerómetro. La gran mayoría emplean dos o cuatro galgas activas, conectadas en un puente de Wheatstone. Se emplean resistencias de ultra-precisión en la circuitería de control de la sensibilidad, balanceo y compensación de los efectos de la temperatura. Estos dispositivos son especialmente utilizados para detectar vibraciones de baja frecuencia.

De capacitancia diferencial: están basados en el principio de cambio de la capacitancia en proporción a la aceleración aplicada. Vienen en varias formas y tamaños. En uno de los tipos, la masa sísmica del acelerómetro constituye la parte móvil (condensador variable) de un oscilador eléctrico. El sistema se caracteriza por tener una frecuencia nominal cuando no está sometido a perturbación. Si el instrumento es acelerado, la frecuencia varía del punto nominal, dependiendo de la dirección de la aceleración.

Dispositivo acelerómetro microcontrolado

De galga extensiométrica: están basados en las propiedades resistivas de los conductores eléctricos. Si un conductor es comprimido, su resistencia se altera debido a cambios dimensionales y de sus propiedades piezoresistivas. Esto indica que la resistividad del conductor depende de la presión mecánica a la que es sometido.

Sísmicos: hacen uso de una masa sísmica que esta suspendida por un resorte o una leva dentro de un marco rígido. El marco que sostiene la masa sísmica esta firmemente unido a la fuente de vibración que quiere ser medida. Cuando el sistema vibra, la masa tiende a permanecer en una posición fija de tal manera que el movimiento puede ser medido con el desplazamiento relativo entre la masa y el marco de soporte. Este desplazamiento es medido por un transductor apropiado para su posterior procesamiento. Aunque es imposible que la masa sísmica permanezca absolutamente estática, para determinadas frecuencias puede ser satisfactoriamente considerado como una referencia de posición. Seleccionando apropiadamente la masa, el resorte y el amortiguador, es posible la utilización de este dispositivo para medición de aceleración o desplazamiento. En general grandes masas son aconsejables para medición de desplazamiento mientras que las pequeñas son más convenientes para vibración.

Inerciales: la fuerza requerida para limitar el movimiento de la masa en presencia de vibración, es provista por un sistema inercial como por ejemplo un giro péndulo, rotor de vibración o cuerda vibratorio entre otros.

De realimentación de fuerza electrostática: están basados en la ley de Coulomb entre dos electrodos cargados. Ellos miden el voltaje en términos de la fuerza requerida para sostener un electrodo móvil dentro de un área, masa y separación conocida de un electrodo móvil.

Microacelerómetros: están basados en tecnología integrada de procesamiento (circuito integrado) que es usada para fabricar estructuras electromecánicas tridimensionales. El primer acelerómetro de este tipo fue desarrollado en 1979.

3.2.1.1 Acelerómetro triaxial MMA7260QT

Atendiendo a las necesidades técnicas que se plantean en este proyecto, encontramos el acelerómetro MMA7260QT de la compañía Freescale. Es un acelerómetro triaxial, con sensibilidades configurables (1.5g/2g/4g/6g), alta sensibilidad (800 mV/g @ 1.5g), reducidas dimensiones y bajo consumo.

Principio de operación:

Es un circuito integrado de montaje superficial. El dispositivo consta de dos celdas sensibles capacitivas, y una señal de acondicionamiento.

El sensor está sellado herméticamente en el nivel de wafer, utilizando una gruesa capa. La célula es una estructura mecánica formada por materiales semiconductores (polisilicio), utilizando para ello los procesos de semiconductores (enmascaramiento y grabado). Puede ser modelado como un conjunto de vigas junto a una masa móvil central que se mueve entre ellas, al someter el sistema a una aceleración.

Dispositivo acelerómetro microcontrolado

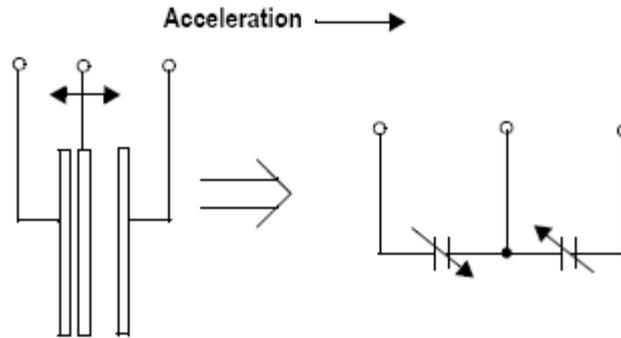


Figura 1. Simplificación del modelo de transductor

La masa central se mueve, aumentando la distancia entre ella y una de las vigas fijas, y disminuyendo en la misma cantidad respecto a la viga del otro lado. El cambio en la distancia es una medida de la aceleración.

En la célula, las vigas forman dos condensadores. Cuando el haz central se mueve con la aceleración, la distancia entre las vigas varía y por lo tanto el valor de los condensadores también, siendo el valor del cambio, $(C = A\epsilon / D)$. Donde A es el área de la viga, ϵ es la constante dieléctrica, y D es la distancia entre las vigas.

El ASIC utiliza técnicas de condensadores de conmutación para medir el valor de los condensadores de la célula y extraer datos de la aceleración. Siendo la aceleración la diferencia entre los dos condensadores.

Existe la posibilidad de seleccionar entre 4 tipos de sensibilidades presentes en el dispositivo. Dependiendo de la lógica entrada en los pines 1 y 2, la ganancia interna del dispositivo cambiará, pudiendo funcionar a 1.5g, 2 g, 4g, o 6 g.

Tabla 1. Descripción de los pines g-Select

g-Select2	g-Select1	g-Range	Sensitivity
0	0	1.5g	800 mV/g
0	1	2g	600 mV/g
1	0	4g	300 mV/g
1	1	6g	200 mV/g

Esta característica es ideal para cuando un producto tiene requerimientos de funcionamiento a diferentes sensibilidades para un óptimo rendimiento. La sensibilidad se puede cambiar en cualquier momento durante la operación del producto.

3.2.2 Microcontroladores

Podemos decir que un microcontrolador es un pequeño computador construido dentro de un circuito integrado, gracias a los avances tecnológicos de la electrónica que ha permitido implementar todos los componentes de un computador dentro de un chip de silicio.

Los microcontroladores son de tamaño reducido y pueden estar situados dentro de la misma aplicación a la que gobiernan. Siendo por ello de gran uso en casi todos los sectores de la vida cotidiana, como por ejemplo: comunicaciones, electrodomésticos, domótica, climatización, automoción...

Características comunes a los microcontroladores:

La estructura fundamental y las características básicas de todos los microcontroladores son muy parecidas ya que todos deben disponer de unos bloques esenciales:

La CPU: es el elemento más importante de los microcontroladores y determina sus características principales. La Unidad Central de Proceso en cuanto a su arquitectura y funcionalidad puede hacerse con tres orientaciones distintas:

- Del tipo CISC que significa Computadores de Juego de Instrucciones Complejo, esto es, disponen de más de 80 instrucciones en su repertorio, algunas necesitando muchos ciclos para ejecutarse.
- Del tipo RISC que significa Computadores de Juego de Instrucciones Reducido, con instrucciones muy simples y que se ejecutan casi todas en un ciclo.
- Del tipo SISC que significa Computadores de Juegos de Instrucciones Específico, esto es, instrucciones de un número reducido y que se adaptan a las necesidades de la aplicación prevista.

El procesador de los modernos microcontroladores responde a la arquitectura RISC (Computadores de Juego de Instrucciones Reducido). Otra aportación frecuente que aumenta el rendimiento del computador es el fomento del paralelismo implícito, que consiste en la segmentación del procesador (pipe-line), descomponiéndolo en etapas para poder procesar una instrucción diferente en cada una de ellas y trabajar con varias a la vez.

La Memoria: está integrada en el propio chip y constará de un tipo de memoria no volátil (tipo ROM) destinada a contener el programa de instrucciones y otra volátil (tipo RAM) destinada a guardar variables y datos.

Hay 5 versiones de memoria no volátil disponible en los microcontroladores:

- ROM con máscara, que se graba durante la fabricación del chip. Se utilizan cuando se necesitan cantidades superiores a varios miles de unidades.
- OTP, que es de sólo lectura y programable una sola vez (One Time Programmable) por el usuario. Es recomendable cuando se construyen prototipos y series muy pequeñas. Se pueden encriptar mediante fusibles.

Dispositivo acelerómetro microcontrolado

- EPROM, que pueden grabarse y borrarse muchas veces (Erasable Programmable Read Only Memory) por el usuario. Para borrarse disponen de una ventana por donde con rayos ultravioleta pueden borrarse.
 - EEPROM, que son de solo lectura pero que pueden programarse y borrarse eléctricamente (Electrical Erasable Programmable Read Only Memory) por el usuario. Los microcontroladores que llevan memoria EEPROM pueden grabarse y borrarse cuantas veces se quiera sin ser retirados del circuito.
 - FLASH, que son no volátiles y de bajo consumo, se pueden escribir y borrar. Funcionan como una ROM y una RAM juntas pero con menos consumo y más pequeña. Pero a diferencia de la ROM es programable en el propio circuito.
- Está recomendada frente a la EEPROM cuando se necesite una gran cantidad de memoria de programa no volátil.

Los Puertos de E/S: se destinan a proporcionar el soporte a las señales de entrada, salida y control, que comunican al computador interno con los periféricos exteriores.

El Reloj: será un circuito oscilador que generará una onda cuadrada de alta frecuencia para configurar los impulsos de reloj necesarios para sincronizar todas las operaciones del sistema.

En cuanto a la arquitectura interna la hay de dos tipos:

- Arquitectura de Von Newman: lleva la memoria de instrucciones y la memoria de datos en un solo chip y por tanto se accede a ella a través de un sistema de buses único.

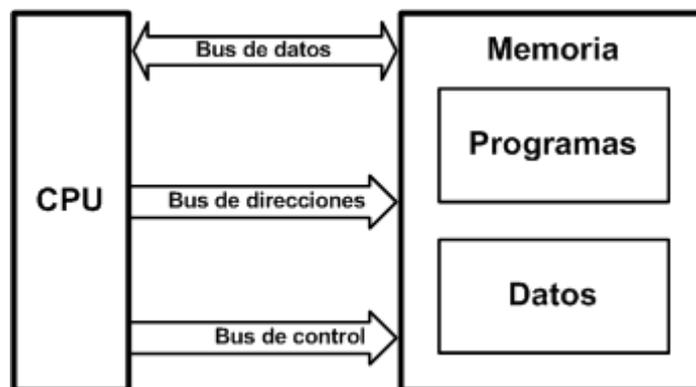


Figura 2. Diagrama de bloques Von Newman

- Arquitectura Harvard: las memorias son independientes, memoria de instrucciones por un lado y memoria de datos por otro. Cada una dispone de sus propios sistemas de buses y por tanto será posible realizar operaciones de lectura o escritura en ambas simultáneamente.

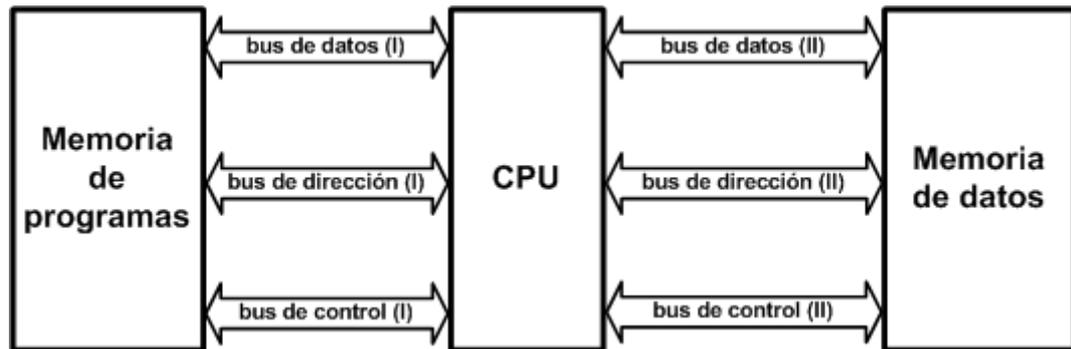


Figura 3. Diagrama de bloques Harvard

Líneas de Entrada/Salida

A excepción de dos patitas destinadas a recibir la alimentación, otras dos para el cristal de cuarzo, que regula la frecuencia de trabajo, y una más para provocar el Reset, las restantes patitas de un microcontrolador sirven para soportar su comunicación con los periféricos externos que controla. Las líneas de E/S que se adaptan con los periféricos manejan información en paralelo y se agrupan en conjuntos de ocho, que reciben el nombre de Puertas. Hay modelos con líneas que soportan la comunicación en serie; Otros disponen de conjuntos de líneas que implementan puertas de comunicación para diversos protocolos, como el I2C, RS232...

Recursos especiales de los microcontroladores:

Los principales recursos específicos que pueden incorporar los microcontroladores son:

Los temporizadores o Timers, para controlar períodos de tiempo o poder llevar la cuenta de sucesos exteriores (contadores).

El perro guardián o Watchdog, que es un registro que hará que el computador no se bloquee por un fallo de software o de otra causa.

Protección ante fallo de alimentación o Brownout, que se trata de un circuito que resetea al microcontrolador cuando la alimentación baja de un voltaje único.

Estado de reposo o bajo consumo, que se consigue mediante una instrucción especial (SEP) y en el cual los requerimientos de potencia son mínimos.

Convertor A/D, para poder procesar señales analógicas. Disponen de un multiplexor que permite aplicar a la entrada del convertor diversas señales analógicas desde diferentes pines del microcontrolador.

Convertor D/A, que transformará los datos digitales contenidos en el proceso del computador en señales analógicas que saldrán al exterior por alguno de los pines del microcontrolador.

Comparador analógico, capaz de comparar una señal fija de referencia y otro variable que se aplique en alguno de los pines del microcontrolador, y de proporcionar una salida con nivel lógico "1" o "0" según cual sea mayor.

Dispositivo acelerómetro microcontrolado

Modulador de anchura de impulsos PWM, que es una serie de circuitos capaces de proporcionar en su salida impulsos de anchura variable que saldrán al exterior por alguno de los pines.

3.2.2.1 Microcontrolador PIC 16F873

Dentro de todos los microcontroladores existentes en el mercado, nos encontramos con el microcontrolador PIC 16F873. Las características que nos ofrece el microprocesador se ajustan a las necesidades del proyecto.

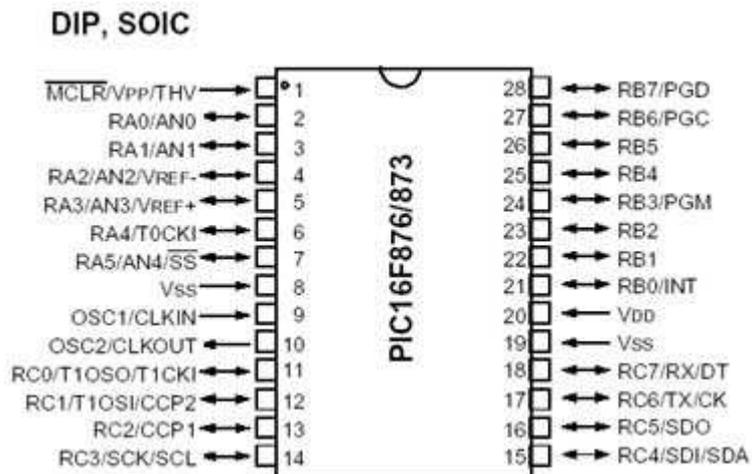


Figura 4. Esquemático del patillaje PIC16F876/873

Las características principales del micro son:

- Memoria de programa: 7,2 kB
- Cantidad de palabras de instrucción: 4096
- Datos SRAM (bytes): 192
- EEPROM (bytes): 128
- Entradas / Salidas: 22
- Canales A/D: 5
- PWM: 2
- SPI: Si
- I2C maestro: Si
- USART: Si
- Comparadores: Si

El juego de instrucciones de este micro es reducido, 35 instrucciones, con lo que facilita la programación del mismo. A continuación se detallan las instrucciones que admite el micro:

Instrucciones Orientadas a bits

Este grupo de instrucciones actúa directamente sobre un bit específico de cualquier registro. Con ellas es posible escribir o leer un bit para luego tomar decisiones de acuerdo al resultado.

Tabla 2. Resumen de instrucciones

Sintaxis	Descripción	Ciclos de Instrucción
BCF f,b	Pone en "0" el bit b del registro f	1
BSF f,b	Pone en "1" el bit b del registro f	1
BTFSS f,b	Lee el bit b del registro f y salta la instrucción siguiente si está en "1"	1 (2 si salta)
BTFSC f,b	Lee el bit b del registro f y salta la instrucción siguiente si está en "0"	1 (2 si salta)

Instrucciones Orientadas a Registros

Este tipo de instrucciones maneja registros directamente, es decir, con estas instrucciones vamos a poder asignar valores a cada registro, operar matemáticamente con ellos, operar con lógica booleana entre ellos...

Tabla 3. Resumen de instrucciones

Sintaxis	Descripción	Ciclos de Instrucción
ADDWF f,d	Suma el contenido del registro W con el de f	1
ANDWF f,d	Realiza la operación lógica AND entre el contenido del registro W con el de f	1
CLRF f	Borra el contenido del registro f	1
CLRW	Borra el contenido del registro W	1
COMF f,d	Niega el valor del registro f	1
DECF f,d	Decrementa en una unidad el contenido del registro f	1
DECFSZ f,d	Decrementa en una unidad el contenido del registro f y salta la instrucción siguiente si es cero.	1 (2 si salta)
INCF f,d	Incrementa en una unidad el contenido del registro f	1
INCFSZ f,d	Incrementa en una unidad el contenido del registro f y salta la instrucción siguiente si es cero.	1 (2 si salta)
IORWF f,d	Realiza la operación lógica OR entre el contenido del registro W con el de f	1
MOVF f,d	Mueve el contenido del registro f	1
MOVWF f	Mueve el contenido del registro W al registro f	1
NOP	No realiza ninguna operación	1
RLF f,d	Rota el contenido del registro f hacia la izquierda a través del carry bit	1
RRF f,d	Rota el contenido del registro f hacia la derecha a través del carry bit	1
SUBWF f,d	Resta el contenido del registro W al registro f	1
SWAPF f,d	Intercambia los cuatro primeros bits con los cuatro últimos del registro f	1
XORWF f,d	Realiza la operación lógica XOR entre el contenido del registro W con el de f	1

- Todas las operaciones se guardan en:
 - El registro **W** si d=0
 - El registro **f** si d=1
- Para el caso de **RLF** y **RRF** el bit puesto al rotado se pone en 0

Instrucciones Orientadas a Control

Este juego de instrucciones nos permite controlar algunos registros y funciones específicas. También estas instrucciones se utilizan para guardar algún valor determinado en un registro o hacer operaciones matemáticas y lógicas entre un registro y un valor cualquiera que no necesariamente esté guardado en un registro, es decir, un valor literal.

Tabla 4. Resumen de instrucciones

Sintaxis	Descripción	Ciclos de Instrucción
ADDLW k	Suma el valor k al registro W	1
ANDLW k	Realiza la operación lógica AND entre el valor k y el registro W	1
CALL k	Llamada a la subrutina k	2
CLRWDT	Limpia el WD-Timer	1
GOTO k	Salto incondicional a la etiqueta k	2
IORLOW k	Realiza la operación lógica OR entre el valor k y el registro W	1
MOVLW k	Guarda en el registro W el valor k	1
RETFIE	Retorno de una interrupción	2
RETLW	Retorno de una subrutina con el valor k en el registro W	2
RETURN	Retorno de una subrutina	2
SLEEP	Se pone el PIC en estado de reposo (dormido)	1
SUBLW k	Resta el valor k al registro W	1
XORLW	Realiza la operación lógica XOR entre el valor k y el registro W	1

El circuito externo necesario para que el micro sea capaz de leer el programa grabado en él solo necesita dos aspectos fundamentales: el reloj y la alimentación, lo demás es a medida que necesitemos entradas y/o salidas adecuando cada una de ellas con circuitería externa.

El reloj se usa para darle una base de tiempo al micro, se puede usar una resistencia y un condensador o algo más fiable como un cristal de cuarzo piezoeléctrico. Usando cristales se puede asegurar la frecuencia de oscilación del micro, lo cual es útil para calcular tiempos de ejecuciones de las instrucciones, temporizaciones precisas...

El Reloj

El PIC 16F873 disponen de un ciclo de instrucción igual a cuatro ciclos del reloj principal.

La máxima frecuencia del reloj externo es de 20 MHz. es decir que el tiempo mínimo necesario para leer y ejecutar cada instrucción es de 200 ns. ($20\text{MHz} / 4 = 5\text{MHz}$; $1/5\text{MHz} = 200\text{nS}$)

Dispositivo acelerómetro microcontrolado

Un ciclo de instrucción equivale a 4 ciclos de reloj. Esto es porque con el primer ciclo de reloj el micro busca en su memoria a la instrucción a leer, en el segundo ciclo se carga en la memoria principal, el tercer ciclo es el encargado de ejecutarla propiamente dicho y el último y cuarto ciclo limpia la memoria para volver a buscar la siguiente instrucción y así sucesivamente.

La alimentación

En cuanto a la alimentación, no debe superar los $5\text{ V} \pm 5\%$ y se debe tener en cuenta que existe un terminal llamado MCLR (master-clear o reset), que debe estar con valor lógico 1 para que el micro pueda leer el programa.

Aunque se debe alimentar con 5 V, con 3,5 V también funciona consumiendo un poco más de corriente. La corriente máxima que puede circular por el terminal VDD (donde se conecta el positivo de la fuente de 5 V) es de 250 mA y por cada salida no se debe hacer circular más de 25 mA.

Dado que la capacidad de entregar corriente del micro es bastante reducida, se suelen usar transistores o relés para comandar circuitos o artefactos de potencia. La potencia que puede disipar como máximo es de 1 W y se calcula mediante la siguiente fórmula:

$$P_{disipada} = V_{DD}(I_{DD} - \Sigma I_{OH}) + \Sigma[(V_{DD} - V_{OH})I_{OH}] + \Sigma(V_{OL}I_{OL})$$

donde:

- V_{DD} es la tensión suministrada por la fuente de alimentación.
- I_{OH} es la corriente suministrada por las salidas del PIC en estado alto.
- V_{OL} es la corriente absorbida por las salidas del PIC en estado bajo.
- V_{OH} es la tensión entregada por los terminales en estado alto.
- V_{OH} es la tensión presente en los terminales en estado bajo.

Circuito de Reset Externo

Tiene un terminal llamado MCLR o master-clear el cual debe estar en “1” lógico para que se lean y ejecuten las instrucciones. En esta parte describiremos como resetear el micro, es decir hacer que empiece a leer el programa grabado en el desde el principio.

Registros Internos

Cuenta con registros internos (la mayoría de estos son de escritura-lectura) los cuales permiten configurar cada bloque para que funcione de acuerdo a las necesidades.

Para el 16F873 estos registros son de 8 bits y podemos acceder a cada uno de ellos, o bien a todo el registro. También existen algunos registros particulares de 16 bits pero físicamente son dos registros de 8 unidos.

La Memoria de registros es la parte de la memoria se encuentran los registros que manejan a cada modulo del micro y también existen registros no definidos o bien a definirse por el programador para cualquier uso (variables, constantes, acumuladores

Dispositivo acelerómetro microcontrolado

auxiliares...) estos registros de propósito general también son de 8 bits y se definen asignándoles un nombre y una dirección.

Para el 16F873 la memoria de registros está dividida en 4 bancos.

El registro W

El registro auxiliar de trabajo W (working register) es un registro de 8 bits con un espacio físico en la memoria principal del micro. Este registro se puede escribir desde el exterior y se puede leer en cualquier momento ya que se puede acceder a él desde cualquier banco de memoria. Por medio de este registro se cargan valores a los distintos registros del micro, es decir, primero se pone un valor en el registro W y luego se copia el valor del registro W al registro que se quiera. No se puede cargar un valor directamente a otro registro que no sea W.

El registro STATUS

Existen registros de control en los 4 bancos disponibles para el 16F873, por lo tanto hay que tener claro que para modificarlos o asignarles algún valor hay que seleccionar el banco correspondiente. Generalmente se trabaja en el banco 0, pero no por eso se puede decir que los otros no se utilizan. Además el banco 0 es el predeterminado cuando se enciende el micro.

El registro de estado (STATUS) proporciona continuamente información mediante los bits del 0 al 2 de los resultados de operaciones matemáticas, desbordes de contadores... Seteando los bits 5 y 6 podemos seleccionar el banco de memoria correspondiente para trabajar con los registros necesarios y el bit 7 se usa para establecer que parte de cada banco está seleccionado.

Palabra de Configuración

Es un registro donde se eligen algunas condiciones de funcionamiento iniciales, para que el propio PIC sepa como será usado. Su cantidad de bits es fija, 14 bits, pero solamente se tiene acceso a algunos de ellos. Para el 16F873 existen 11 bits configurables en este registro.

Conversores Analógicos-Digital

Son de 10 bits y su funcionamiento es bastante simple comparados con los conversores A/D R2R, los de resistencia ponderada...

El PIC 16F873 viene equipado con 5 entradas para hacer conversiones A/D. La técnica que utiliza el micro para la conversión es la de incremento y comparación, la cual consiste en usar un registro auxiliar, compararlo con la entrada analógica y si es menor incrementarlo, volver a comparar y así hasta que el valor del registro sea lo más aproximado posible (pero sin pasarse) a la entrada analógica.

El rango de conversión es de 0 a 5 V, pero si hubiera que hacer alguna conversión de más voltaje bastará con poner a la entrada del conversor un divisor de tensión correctamente calculado o bien trabajar con alguna tensión de referencia externa al micro.

Dispositivo acelerómetro microcontrolado

La resolución que tiene cada bit de la conversión tiene un valor que es función de la tensión de referencia externa (en caso que la hubiere) y viene dada por:

$$Res = \frac{V_{ref+} - V_{ref-}}{1024}$$

3.2.3 Memorias

Las memorias son los circuitos que permiten almacenar y recuperar información. Aunque parezcan conceptualmente sencillas presentan tal vez la más amplia diversidad de tipos, tecnología, estructura, prestaciones y costo, entre todos los componentes electrónicos.

Jerarquía de memoria

El diseño de una memoria se puede resumir en tres cuestiones: capacidad, velocidad y precio. Existe un compromiso entre estas tres características y para realizar los sistemas de memoria se emplean diversas tecnologías, que cumplen las siguientes relaciones:

- A menor tiempo de acceso, mayor coste por bit.
- A mayor capacidad, menor coste por bit.
- A mayor capacidad, mayor tiempo de acceso.

Tipos de memorias:

La más común es la denominada memoria de acceso aleatorio (RAM, Random-access Memory). Una característica principal es la posibilidad de leer y escribir rápidamente datos en ella mediante señales eléctricas. Otra característica distintiva de una RAM es que es volátil, es decir debe estar continuamente alimentada, ya que si se interrumpe la alimentación, se pierden los datos. Por lo tanto estas memorias pueden utilizarse sólo como almacenamiento temporal.

Las tecnologías RAM se dividen en dos variantes: estáticas y dinámicas. Las RAM dinámicas están hechas con celdas, que almacenan los datos como cargas en condensadores.

La presencia o ausencia de cargas en un condensador se interpretan como el 1 o el 0 binarios. Ya que los condensadores tienen una tendencia natural a descargarse, las RAM dinámicas requieren refrescos periódicos para mantener memorizados los datos.

En una RAM estática, los valores binarios se almacenan utilizando configuraciones de puertas que forman biestables (flip-flops).

En claro contraste con las RAM están las memorias de sólo lectura (ROM, Read-Only Memory). Como su nombre sugiere, una ROM contiene de manera permanente datos que no pueden alterarse (no es volátil). Aunque es posible leer de una ROM, no se pueden escribir nuevos datos en ella.

Una ROM se construye como cualquier otro chip de circuito integrado, con los datos cableados en el chip durante el proceso de fabricación.

Existen también memorias ROM programables (PROM), que al igual que las ROM, son no volátiles y pueden grabarse sólo una vez. Para la PROM el proceso de escritura se lleva a cabo eléctricamente y puede realizarlo el fabricante o el cliente con

Dispositivo acelerómetro microcontrolado

posterioridad a la fabricación de chip original. Se requiere un equipo especial para el proceso de escritura.

Otra variante de la memoria de sólo lectura es la memoria Read-Mostly, que es útil para aplicaciones en las que las operaciones de lectura son bastante más frecuentes que las de escritura, pero para las que se requiere un almacenamiento no volátil. Hay tres formas comunes de memoria Read-Mostly: EPROM, EEPROM y memorias Flash.

EPROM (Erasable Programmable Read-Only Memory): es una memoria de sólo lectura programable y borrable ópticamente. Se lee y escribe eléctricamente como la PROM, sin embargo, antes de la operación de escritura todas las celdas de almacenamiento deben ser borradas a la vez, mediante la exposición del chip a radiación ultravioleta. Este proceso de borrado puede realizarse repetidas veces.

EEPROM (Electrically Erasable Programmable Read-Only Memory): es una memoria de sólo lectura programable y borrable eléctricamente. Es una memoria en la que se puede escribir en cualquier momento sin borrar su contenido anterior; sólo se actualiza el byte o bytes direccionados. La EEPROM combina la ventaja de no ser volátil con la flexibilidad de ser actualizable in situ, utilizando las líneas de datos, de direcciones y de control de un bus ordinario.

Memorias Flash: se denominan así por la velocidad a la que pueden reprogramarse. Estas memorias se encuentran, en coste y funcionalidad, entre las EPROM y las EEPROM. Al igual que las EEPROM utilizan una tecnología de borrado eléctrico. En una memoria Flash es posible borrar bloques concretos de memoria, en lugar de todo el chip. Sin embargo, las memorias flash no permiten borrar a nivel del byte.

3.2.3.1 Memoria EEPROM 24LC256

Una vez se han descrito los principales tipos de memorias existentes hoy en día, necesitamos seleccionar una que se ajuste a las necesidades del proyecto. Y ya que uno de los principales requerimientos que se especificó al principio del proyecto fue la posibilidad de almacenar toda la información capturada por el acelerómetro, optamos por una memoria EEPROM. En concreto la memoria 24LC256.

La memoria 24LC256 es una PROM eléctricamente borrable de 32K x 8 (256 Kbit), capaz de funcionar en un amplio rango de voltaje (1.7V V a 5,5 V). Ha sido desarrollada para aplicaciones de baja potencia tales como las comunicaciones personales o de adquisición de datos. Este dispositivo también tiene la capacidad de escribir una página de hasta 64 bytes de datos. Es capaz realizar lecturas aleatorias y secuenciales de hasta 256 K.

Permite hasta ocho dispositivos en el mismo bus, para un máximo de 2 Mbit de espacio de direccionamiento.

3.3 Software del PIC

Para realizar la programación del microcontrolador, se pueden optar entre varias opciones, de las que destacan Ensamblador y C. Dado que en la actualidad existen potentes librerías de funciones C, se opta por la programación en este lenguaje.

Dispositivo acelerómetro microcontrolado

En C, para poder generar los ficheros hexadecimales que posteriormente serán grabados en el microcontrolador, es necesario un compilador de C. El compilador de C que se va a utilizar es el PCW de la casa CCS Inc. que está integrado en un entorno de desarrollo integrado (IDE).

El compilador PCW traduce las instrucciones contenidas en un fichero .C al lenguaje máquina del microcontrolador, generando por lo tanto un fichero .HEX.

(Código del programa en Anexo 1)

3.4 Software externo

Uno de los requerimientos de este proyecto es que sea posible leer y manejar los datos que se capturen con el dispositivo acelerómetro.

Actualmente existen diversos lenguajes de programación que disponen de potentes y diversas características. Cabe destacar, entre otros muchos, lenguajes, Java y .NET. A continuación se detallan tratará de resolver las siguientes cuestiones:

- Introducción a Java.
- Introducción a .NET.
- Comparar lenguajes de programación.
- Comparar la portabilidad en ambos casos.

Java:

Java nace como un lenguaje de programación fácil de utilizar a partir de algunos de los programadores de Sun Microsystems que cansados de usar con C++, obtienen la autorización de sus jefes para desarrollar un lenguaje de programación que sea muy sencillo de utilizar. Aparece así un primer producto, OAK3, que un tiempo después, con la aparición de WWW, se convierte en Java 4. Es un lenguaje de programación orientado a objetos, que además incluye una máquina virtual, y una serie de desarrollos básicos que pueden ser empleados por los programadores para simplificar sus nuevos desarrollos; además permite la inclusión de porciones de código ejecutables en las páginas que se publican en internet, a través de WWW. Sun permite también que otros fabricantes de software tomen a Java como el centro de sus nuevas herramientas de software, es decir, desarrollen entornos de programación considerando a Java como su corazón; de manera que en un tiempo relativamente corto IBM, Borland, Oracle y muchos otros (incluyendo a Microsoft) están desarrollando herramientas que les permitan ofrecer productos portables entre diferentes máquinas y sistemas operativos, pues todos compilan para la misma máquina: la máquina virtual de Java. Hoy en día, Java cuenta con un API amplio, en el cual pueden encontrarse puntos de partida para crear los más diversos tipos de programas, tiene una serie de estándares para programación reconocidos y aceptados por la mayoría de sus programadores, existen las versiones de su máquina virtual para casi todas las plataformas comerciales vigentes en este momento y está en capacidad de hacer interfaz con casi todas las bases de datos existentes en el mercado. Además, Java tiene otra gran ventaja: Microsoft le declaró la guerra hace algún tiempo y muchas de las personas involucradas con el negocio del software que no aceptan a esta empresa ven a Java como su aliado. A partir de lo presentado hasta el momento se puede creer que Java es un lenguaje de programación bastante popular, y en lo que a internet se refiere esto es cierto, pues tiene buena parte de los servidores conectados a esta red. Sin embargo no todo es bueno, ya que en la parte de aplicaciones comerciales para las empresas no se ha logrado tener el mercado

ni la aceptación que se quisiera, y las razones son varias: Java no ofrece un entorno de programación completo, además el producto final no es muy veloz en tiempo de ejecución, entre otras cosas.

.NET:

Prácticamente desde la aparición de Java, Microsoft ha querido ser su competencia, y así aparece .NET. Hay que tratar de revisar qué es .NET, inicialmente se podría pensar que es un lenguaje de programación orientado a objetos, o un conjunto de lenguajes de programación todos ellos orientados a objetos, y se pueden citar C#.NET, C++.NET y VisualBasic.NET; pero a esta última definición le hace falta incluir lo referente al acceso a las bases de datos, es decir ADO.NET, las herramientas para desarrollo en internet, ASP.NET y el conjunto de facilidades para construir Servicios Web, además del hecho de que todo viene integrado dentro de un entorno completo denominado VisualStudio.NET. Para lograr elaborar estas herramientas y tener la certeza de que se integrarán bien, se definieron los siguientes elementos:

- Un lenguaje común de ejecución: CLR (Common Language Runtime).
- Un conjunto de tipos de datos básicos: CTS (Common Type System), el cual incluye, además de todos los tipos de datos básicos, las clases Object y String. Un CLS (Common Language Specification), que es el conjunto de reglas que especifica lo referente a la implementación de las características de la POO y a otras estructuras sintácticas.
- Un MSIL (Microsoft Intermediate Language), o lenguaje intermedio común. Este lenguaje es el equivalente al bytecode de Java.
- Un compilador capaz de traducir del MSIL a lenguaje binario, comúnmente denominado JIT (Just In Time).

Así pues, se puede decir que .NET es más que un conjunto de herramientas: con .NET se rompe la filosofía tradicional de Microsoft, pues se ha hecho un esfuerzo para que .NET sea abierto y estándar; para lo cual sometieron sus especificaciones del lenguaje y su tipo común de datos a la revisión de organismos internacionales dedicados a la regulación y estandarización de las plataformas de programación. De hecho Microsoft garantiza que si otro productor de software construye un lenguaje que al compilar lleve al tipo común de datos y respete las especificaciones que ellos han dado, ese lenguaje podrá ser incorporado a ambientes .NET, y las clases creadas a través de él podrán interactuar con los elementos de .NET sin ningún problema.

Java vs. .NET

Tanto .NET como Java, son lenguajes de programación orientados a objetos, para lo cual se partirá de ver la forma en la cual cada uno de ellos implementa las características de la POO, cómo es el manejo de los objetos como tales, el encapsulamiento, la herencia y el polimorfismo.

Tipos de estructuras que el lenguaje le permite crear a un programador:

Java: Clases e interfaces.

.NET: Clases, interfaces, struct y enum.

Elementos que pueden definirse dentro de una clase:

Dispositivo acelerómetro microcontrolado

Java: Atributos, métodos y clases internas.

.NET: Atributos, métodos, clases internas, propiedades, eventos y delegates.

Niveles de encapsulamiento:

Java: public, private, protected y visibilidad de paquete, este último se asume cuando se omite.

.NET: public, private, internal, protected y la combinación de estos dos últimos. En caso de omisión se asume private.

Herencia:

Java: No se permite la herencia múltiple, se puede simular a través del uso de interfaces. Por omisión se hereda de Object.

.NET: No se permite la herencia múltiple, se puede simular a través del uso de interfaces. Por omisión se hereda de Object.

Polimorfismo:

Java: Se permite que una clase sobrecargue o sobrescriba métodos definidos por su clase padre, a menos que la clase padre lo impida mediante la palabra reservada final, en el encabezado del método. Si un objeto de una clase hija es referenciado a través de una referencia a su clase padre, su comportamiento, al invocar un método sobrescrito, será el que definió la clase a la cual él pertenece.

.NET: Se permite que una clase sobrecargue o sobrescriba métodos definidos por su clase padre, a menos que la clase padre lo impida empleando la palabra reservada sealed, en el encabezado del método. Si un objeto de una clase hija es referenciado a través de una referencia a su clase padre, su comportamiento, al invocar un método sobrescrito, dependerá de los permisos establecidos por la clase padre, y de la decisión tomada por quien definió la clase. Por omisión se comportará como lo definió la clase padre.

Sobrecarga de operadores para una clase:

Java: No permite la sobrecarga de ninguno de los operadores básicos.

.NET: Se permite la sobrecarga de algunos de los operadores básicos:

Unitarios: +, -, !, ~, ++, --, true, false

Binarios: +, -, *, /, %, &, |, ^, <<, >>, ==, !=, >, <, >=, <=

Portabilidad en ambos casos

La portabilidad es presentada, por los defensores de cada una de las plataformas, como una fortaleza suya y una debilidad del otro. La verdad es que ambas plataformas ofrecen portabilidad, pero en dos ámbitos completamente distintos.

Java ofrece portabilidad entre máquinas, es decir no depende ni del Hardware ni del sistema operativo. Y eso es una gran ventaja; sin embargo si un programador que trabaja en Java se encuentra con que quiere incluir en su código algunas funcionalidades elaboradas en C, por ejemplo, el proceso es complicado. Lograr que el código elaborado en Java pueda ser incluido en desarrollos elaborados en otros lenguajes tampoco es un trabajo sencillo.

.NET no tiene esta dificultad, la interacción entre código elaborado en cualquiera de los lenguajes que incluye esta herramienta es inmediata, como lo será con cualquier lenguaje que sea elaborado cumpliendo con las especificaciones del CLS y que lleve sus datos al CTS, al menos en la teoría. Pero esta plataforma, que ofrece una gran

portabilidad en lo que a lenguajes de programación se refiere, está bien asociada a un tipo de sistema operativo específico: el que produzca Microsoft, al menos en el momento actual.

Cada una de ellas ofrece un tipo de portabilidad del cual la otra carece y es esa diferencia la que se intenta presentar como ventaja o desventaja en cada caso; cuando el análisis real para identificar la portabilidad de cada una depende de la situación específica y del alcance que se espera tener con el software a desarrollar.

CONCLUSIONES

Es indudable que Java cuenta con mayor madurez, con más programadores, con el respaldo de muchas casas de software, entre otras cosas que pueden considerarse ventajosas; pero no es menos cierto que .NET ha sido desarrollado por algunas de las personas que más conocen de Java, que cuenta con el respaldo de una inmensa casa de software, y con el apoyo de muchas personas en el mundo, además de tener una gran aceptación comercial. Ambas tecnologías, "frameworks" tienen puntos verdaderamente asombrosos y ninguna de las dos tiene todo lo que tiene la otra.

Por todo esto cualquiera de las dos tecnologías podrían satisfacer las necesidades planteadas en este proyecto. Pero finalmente optaremos por la tecnología .NET debido a su sencillez a la hora de programar, diseñar y definir los requerimientos.

Los principales aspectos que debe cubrir la aplicación serán:

- Comunicación con alguno de los puertos del ordenador en el que resida la aplicación.
- Flexibilidad a la hora de poder definir los parámetros de comunicación entre el microcontrolador y la aplicación.
- Recuperar toda la información almacenada en las memorias EEPROM de la placa.
- Generar graficas que representen la información adquirida.
- Guardar la información adquirida en algún tipo de fichero.

(Código del programa en Anexo 2)

4 Simulación del prototipo

Antes de proceder al montaje final de la placa, vamos a realizar una simulación a nivel de hardware y software del microcontrolador. De este modo verificaremos que el diseño y funcionalidades esperadas del sistema son correctos.

Para ello empleamos el software Proteus 7 Professional, un software especializado en simulaciones electrónicas.

Una vez abierto un nuevo proyecto dentro de Proteus, agregamos los componentes del sistema y establecemos las conexiones eléctricas pertinentes.

Ya que el acelerómetro entrega a sus salidas (X, Y, Z) tensiones en función de las fuerzas G's medidas, usamos unos voltímetros digitales para poder llevar a cabo su simulación.

En cuanto a la simulación del microcontrolador PIC 16F873, Proteus permite programar el microcontrolador directamente con código C, tal y como lo haremos con el chip real. El esquema final quedará de la siguiente manera:

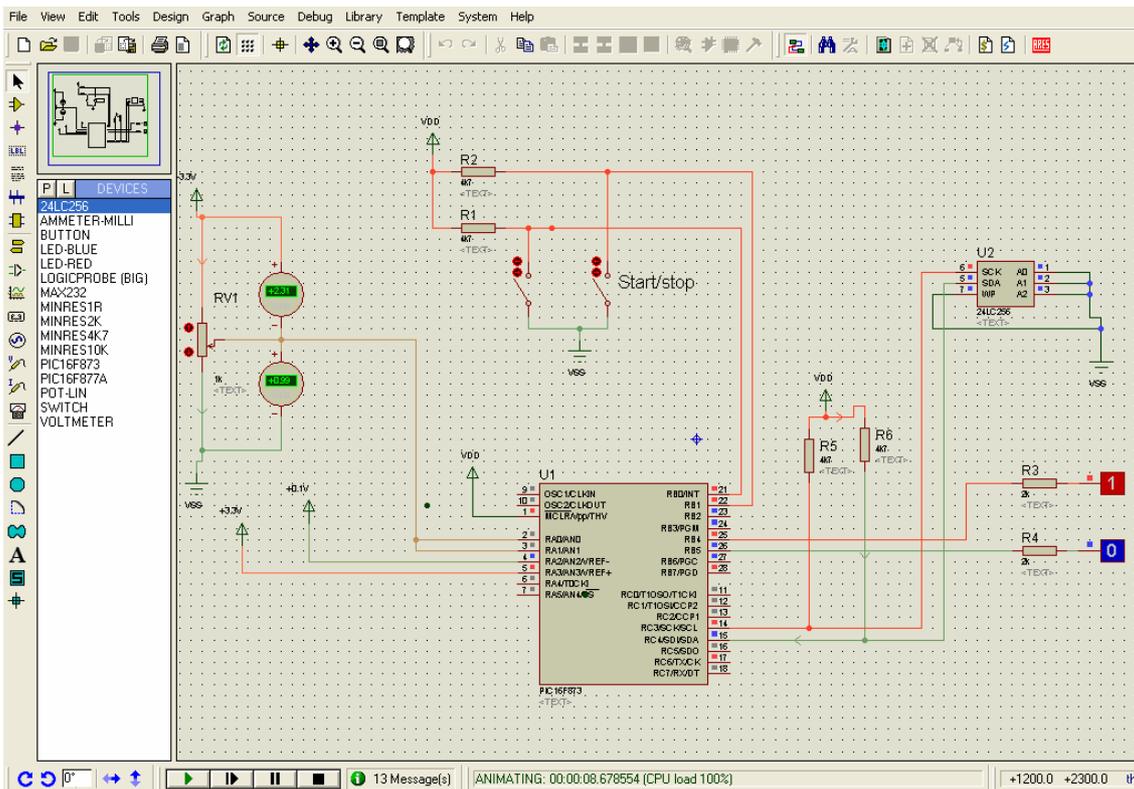


Figura 5. Esquema de simulación en Proteus

Dispositivo acelerómetro microcontrolado

Una vez hecho el esquema eléctrico ya podemos cerrar la ventana del dibujo esquemático (Schematic). Momento en el cual accederemos a la siguiente ventana:

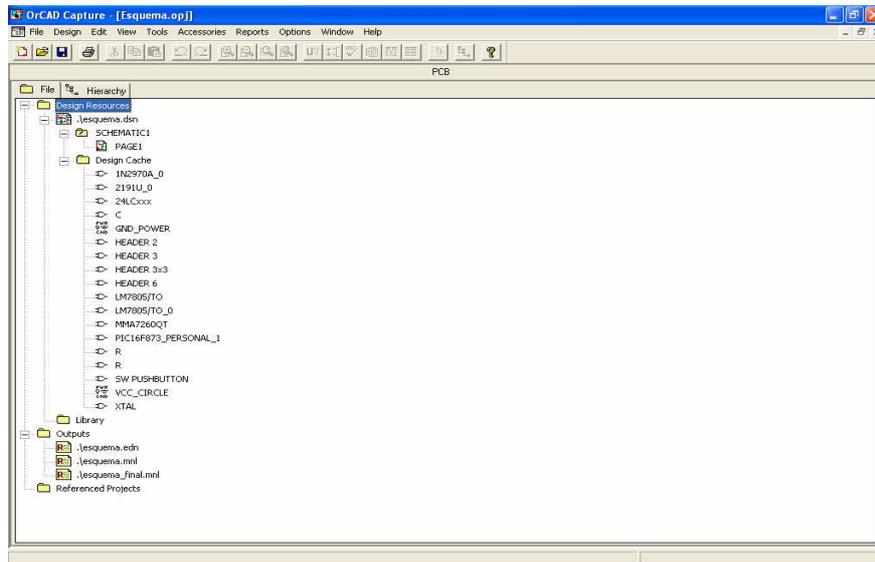


Figura 8. Ventana de diálogo Orcad

En esta pantalla tenemos todos los archivos que se crean y que hemos creado al realizar el proyecto. El archivo llamado PAGE1, es el archivo que contiene el esquema eléctrico; este archivo muestra el esquema eléctrico haciendo doble clic en él. Teniendo ya todo lo necesario en la parte esquemática, debemos prepararlo para llevarlo al proceso del Layout. Es en esta parte donde se hará el ruteado de las pistas del circuito impreso, así como también la disposición de los componentes y los taladros. Para pasar el esquema eléctrico al Layout, seguimos los siguientes pasos:

Tools -> Create Netlist

Seleccionamos la pestaña correspondiente a Layout y colocamos las opciones como se muestra en la figura:

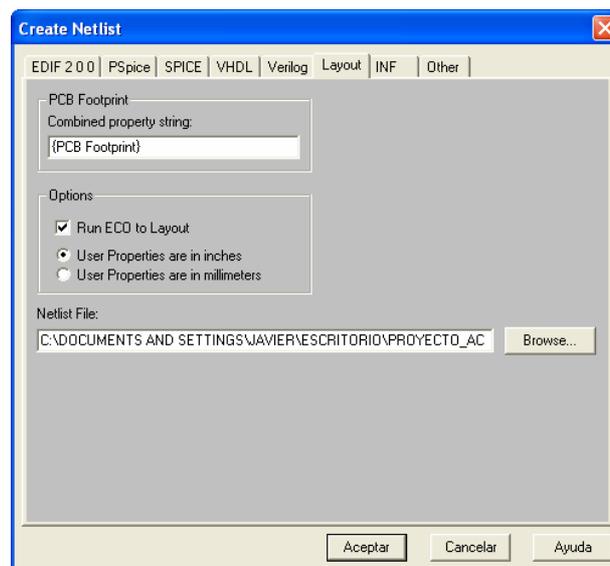


Figura 9. Ventana de diálogo Orcad

Dispositivo acelerómetro microcontrolado

Una vez hallamos realizado esto, pulsaremos el botón Aceptar, y seguidamente el programa OrCAD creará un fichero de extensión MNL, que podemos comprobar en el árbol principal del menú de OrCAD. Este archivo contiene toda la información referente a cómo han de estar conectadas las líneas del esquema eléctrico.

Ahora pasamos al diseño layout. Para ello abrimos Layout Plus y seguimos el paso:

File -> New -> Default

Y seguidamente, abriremos el archivo de extensión MNL creado en el OrCAD Capture. Llegados a este punto, OrCAD necesita pasar de formato MNL a formato MAX, con lo que nos pedirá un nombre para guardar el archivo MNL que por defecto es el mismo nombre que tiene en MNL.

Una vez hecho esto, ya entramos dentro de la parte de diseño de placas de OrCAD Layout.

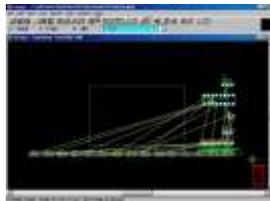


Figura 10. Esquema del circuito

En principio, los componentes están situados de manera arbitraria, pero nosotros podemos colocarlos de la forma que deseemos moviéndolos.

Podemos establecer también los límites de la placa, esto se consigue de la siguiente forma:

Tool -> Obstacle -> New

Y aquí daremos por coordenadas los puntos donde queremos que sean límites de placa.

Después de colocar los límites de placa y los componentes de forma que se aproveche al máximo la capacidad de la placa, iremos a rutear.

Antes debemos decirle que capas queremos que rutee la placa, pudiendo ser o bien una, o bien todas las que tiene el OrCAD.

También estableceremos la distancia entre pistas, su anchura máxima, su anchura mínima y su anchura media.

Todo esto se hace en las pestañas de *Tool* y *Options*.

Después de preparar la placa, ya podemos rutearla; para hacer esto debemos:

Auto -> Autoroute -> Board

Y la placa se ruteará sola, quedando si las hubiere, pistas que no se han podido rutear automáticamente, y de las que se deberán rutear a mano o colocar un cable soldado.

Dispositivo acelerómetro microcontrolado

El resultado final sería:



Figura 11. Esquema del circuito Final

De forma que tendremos todas las pistas ruteadas y se puede imprimir en fotolito o exportar (export Layout to DXF) a AutoCAD para imprimirla o retocarla.

6 Posibles mejoras

Pensando en posibles mejoras aplicables en el futuro, nos encontramos con dos claros puntos.

Por una parte conseguir miniaturizar la parte física de la placa. Para ello se debería pensar en aplicar componentes con encapsulado SMD, así como también realizar el diseño del circuito en tecnología multicapa.

En cuanto a la alimentación, una buena opción sería enfocarlo hacia baterías de litio, debido a su pequeño tamaño, escaso peso y larga autonomía.

Por otra parte, el otro punto a mejorar sería el software que gobierna el microcontrolador y el encargado de recibir la información en el ordenador oportuno. Se podría llegar a desarrollar un algoritmo de interpretación y comunicación que incrementase la velocidad de transferencia y la exactitud de los datos.

7 Documentación anexa

7.1 Anexo1. Código programa PIC

```
#include <C:\Program Files\PICC\Devices\16F873.h>
#define EEPROM_SDA PIN_C4 //pines comunicación eeprom
#define EEPROM_SCL PIN_C3
#include <C:\Program Files\PICC\Drivers\24256.c> //libreria eeprom externa 24LC256
#use delay(clock=4000000) //reloj 4Mhz
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8)
#fuses XT,NOWDT,NOPROTECT
#use fast_io(B)

void main()
{
    long direccion=0;
    int i;
    int eje[3];

    set_tris_b(0b00000011); //B7,B6,B5,B4,B3,B2,B1,B0; 1->ENTRADA, 0->SALIDA
    setup_adc_ports(AN0_AN1_AN2_AN4_VSS_VREF); //A0,A1,A2,A4 entradas
                                                //analógicas. A3 Vreferencia
    setup_adc(ADC_CLOCK_INTERNAL); //reloj para el conversor
    delay_ms(1);
    init_ext_eeprom(); //inicializa eeprom
    delay_ms(1);

    output_bit(PIN_B4,1); //enciende leds
    output_bit(PIN_B5,1);
    output_bit(PIN_B2,1); //desactiva acelerometro (SLEEP)

    while(true) //bucle infinito
    {
        output_bit(PIN_B2,0); //activa acelerometro (SLEEP)
        set_adc_channel(0); //eje X
        delay_us(20); //espera estabilizacion del canal
        eje[0]=read_adc();
        printf("%u",eje[0]); //envia byte al puerto rs232
        delay_ms(10);
        set_adc_channel(1); //eje Y
        delay_us(20); //espera estabilizacion del canal
        eje[0]=read_adc();
        printf("%u",eje[0]); //envia byte al puerto rs232
        delay_ms(10);
        set_adc_channel(2); //eje Z
        delay_us(20); //espera estabilizacion del canal
        eje[0]=read_adc();
        printf("%u",eje[0]); //envia byte al puerto rs232
        delay_ms(10);
    }
}
```

7.2 Anexo2. Código programa VB

```
Imports System.IO

Public Class main

    Dim ejex(3) As Integer
    Dim ejey_ant(3) As Integer
    Dim cont_eje As Integer = -1
    Dim dif As Integer
    Dim mitad As Integer

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
        End
    End Sub

    Private Sub main_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim i As Integer
        For i = 0 To My.Computer.Ports.SerialPortNames.Count - 1
            ComboBox1.Items.Add(My.Computer.Ports.SerialPortNames.Item(i).ToString
)
        Next i
        ComboBox1.SelectedIndex = 0
        mitad = pb.Height / 2
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
        If Button1.Text = "Activar" Then
            Button1.Text = "Desactivar"
            Button2.Enabled = False
            ComboBox1.Enabled = False
            TextBox1.Enabled = False
            TextBox2.Enabled = False
            TextBox3.Enabled = False
            ejex(0) = 0
            ejex(1) = 0
            ejex(2) = 0
            ejey_ant(0) = 0
            ejey_ant(1) = 0
            ejey_ant(2) = 0
            pb.Image = pb.InitialImage
            My.Application.DoEvents()
            Try
                With SerialPort1
                    .PortName = ComboBox1.Text
                    .BaudRate = TextBox1.Text
                    .DataBits = TextBox2.Text
                    .ParityReplace = TextBox3.Text
                    .Open()
                End With
            Catch ex As Exception
                MsgBox("No se ha podido abrir el puerto de
comunicacion " & ComboBox1.Text & ". Compruebe la configuracion.",
MsgBoxStyle.Critical, "Acelerometro")
                Button1.Text = "Activar"
                Button2.Enabled = True
                ComboBox1.Enabled = True
                TextBox1.Enabled = True
            End Catch
        End If
    End Sub
End Class
```

Dispositivo acelerómetro microcontrolado

```
        TextBox2.Enabled = True
        TextBox3.Enabled = True
    End Try
Else
    SerialPort1.Close()
    Button1.Text = "Activar"
    Button2.Enabled = True
    ComboBox1.Enabled = True
    TextBox1.Enabled = True
    TextBox2.Enabled = True
    TextBox3.Enabled = True
End If
End Sub

Private Sub SerialPort1_DataReceived(ByVal sender As Object, ByVal
e As System.IO.Ports.SerialDataReceivedEventArgs) Handles
SerialPort1.DataReceived
    If cont_eje = 2 Then
        cont_eje = 0
    Else
        cont_eje = cont_eje + 1
    End If
    funcion(SerialPort1.ReadExisting.ToString, cont_eje)
End Sub

Private Sub funcion(ByVal txt As String, ByVal cnt As Integer)
    Dim dibujo As Graphics = pb.CreateGraphics
    Dim lapiz As New Pen(Color.Red)
    Dim cero As Integer

    If cnt = 0 Then
        lapiz.Color = Color.Red 'X
        cero = NumericUpDown1.Value
    ElseIf cnt = 1 Then
        lapiz.Color = Color.Blue 'Y
        cero = NumericUpDown2.Value
    Else
        lapiz.Color = Color.Green 'Z
        cero = NumericUpDown3.Value
    End If

    Dim p1 As New Point
    Dim p2 As New Point
    Dim aux1 As Integer

    Try
        aux1 = Convert.ToInt32(txt)

        dif = aux1 - cero
        If dif <> 0 Then
            dif = dif * NumericUpDown5.Value
        End If
        p1.X = ejex(cnt) - 1
        p1.Y = ejey_ant(cnt)
        p2.X = ejex(cnt)
        p2.Y = mitad - dif
        dibujo.DrawLine(lapiz, p1, p2)
        If ejex(cnt) = pb.Width Then
            ejex(cnt) = 0
            pb.Image = pb.InitialImage
        Else
```

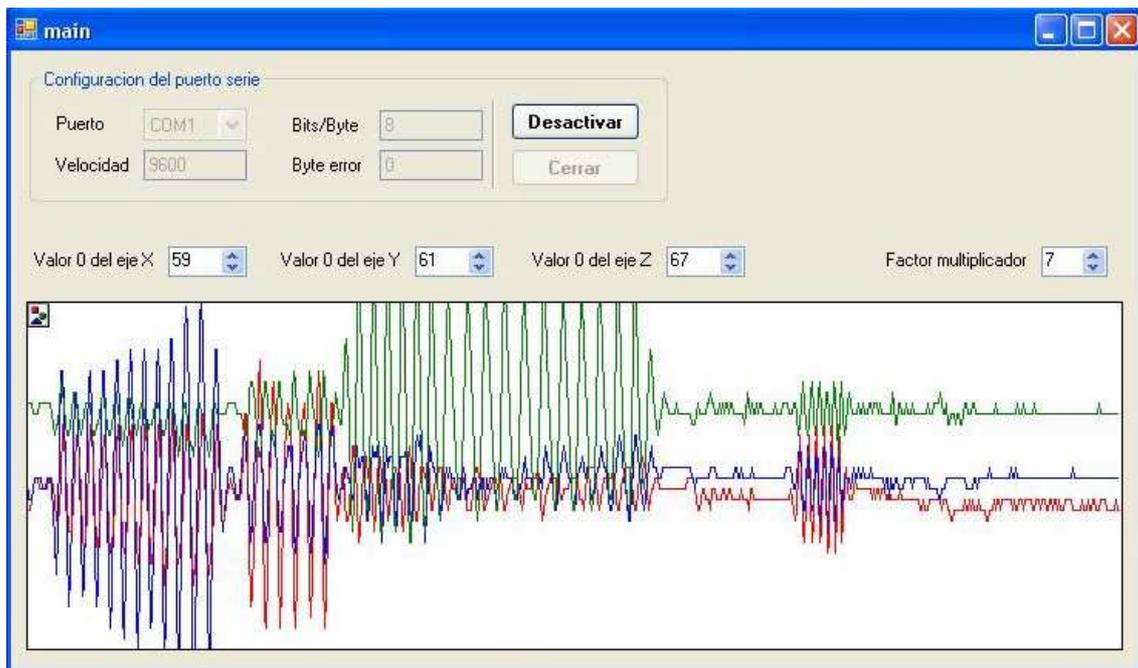
Dispositivo acelerómetro microcontrolado

```
        ejex(cnt) = ejex(cnt) + 1
    End If
    ejey_ant(cnt) = mitad - dif

    Catch ex As Exception

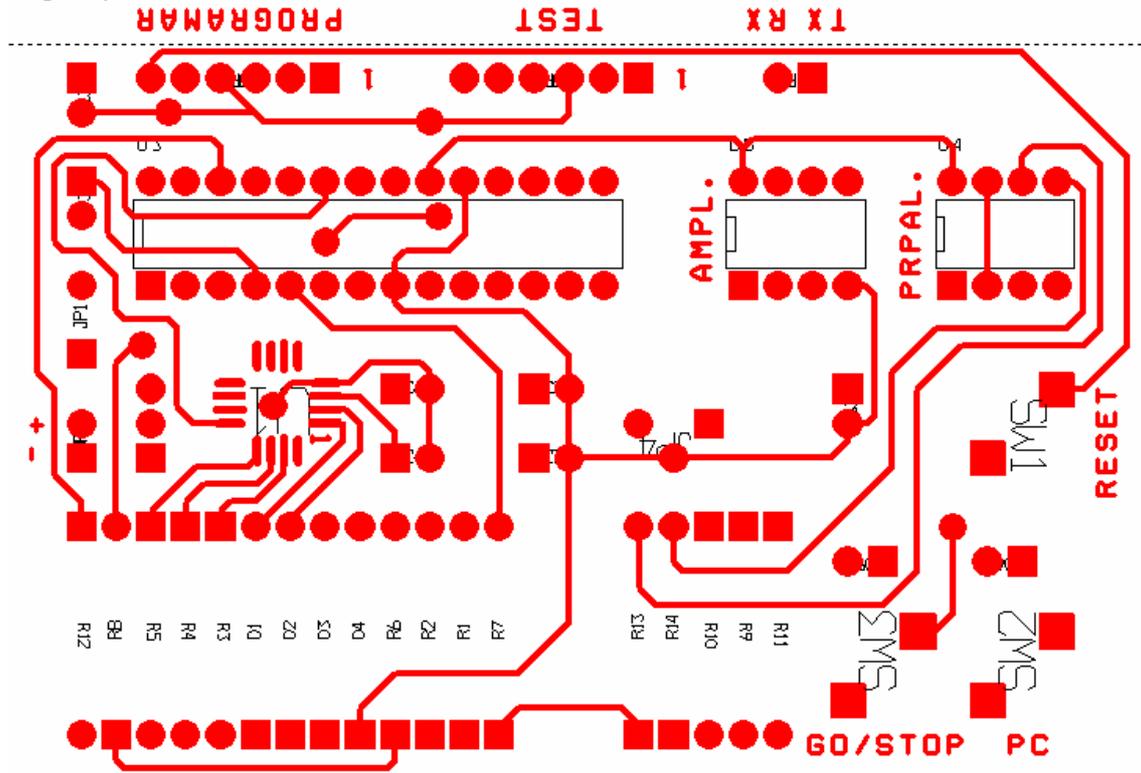
    End Try

    End Sub
End Class
```



7.3 Anexo3. Capas del circuito

Top Layer



Bottom Layer

