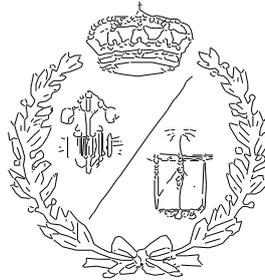


**ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN**

UNIVERSIDAD DE CANTABRIA



Proyecto Fin de Carrera

**ATLAS DE ACELERACIONES DEL CUERPO
HUMANO EN MOVIMIENTO DE CARRERA Y
TRABAJO EN ESCALERAS**

**(Acceleration Atlas of the Human Body in Running and
Stair Climbing Movements)**

Para acceder al Título de

INGENIERO INDUSTRIAL

Autor: Eduardo Bayona Blanco

Julio - 2012

A mis padres, porque ellos me han enseñado a valorar las cosas importantes en esta vida y luchar por ellas, siempre me han apoyado incondicionalmente y han valorado mis esfuerzos. Sin ellos no soy absolutamente nada. Gracias por vuestra confianza y por estar siempre presentes en cada paso que he dado en mi vida.

A mi hermano Juan José, porque desde lejos ha estado orgulloso de mí y me daba fuerzas para seguir adelante.

A mis amigos de la Universidad de Cantabria por compartir conmigo este largo camino y especialmente a aquellos con los que he conseguido ver el final del camino y han hecho de esto algo mucho más especial.

A Mario, por darme la oportunidad de trabajar con él enseñándome mucho más que conocimientos, que han hecho que este último año haya sido increíble.

A todos mis amigos, con los que he tantos momentos he compartido durante esta etapa y que son inolvidables.

Índice general

I	MOTIVACIONES	10
II	OBJETIVOS	12
III	ANTECEDENTES Y ESTADO DEL ARTE	14
2.	Energy Harvesting	15
2.1.	Concepto y Base Teórica	15
2.2.	Orígenes	16
2.3.	Tipos de Fuentes	17
2.3.1.	Vibración (Energía Cinética)	18
2.3.2.	Luz (Energía Fotovoltaica)	20
2.3.3.	Calor (Energía Térmica)	22
2.3.4.	Radiofrecuencia (Energía Inductiva)	25
2.4.	Rendimiento y Viabilidad	26
2.5.	Actualidad en Energy harvesting	27
3.	Generadores Lineales	29

3.1. Principio de Funcionamiento y Base Teórica	29
3.2. Usos y aplicaciones	31
4. Estudio de Aceleraciones	32
4.1. Concepto y Orígenes	32
4.2. Antecedentes y Aplicaciones	33
4.3. Principio de Funcionamiento	35
4.4. Tipos de Acelerómetros	36
4.4.1. Acelerómetros para Medidas Estáticas	37
4.4.2. Acelerómetros para Medidas Dinámicas	37
4.4.3. Acelerómetros para Medidas en varias dimensiones	38
4.4.4. Acelerómetros según Impedancia de Salida	40
4.4.5. Tecnologías de Fabricación	40
 IV MEMORIA DESCRIPTIVA	 49
5. Planteamiento del Problema	50
5.1. Condiciones del Instrumental	51
5.1.1. Consideraciones Constructivas	51
5.1.2. Rango de Medición	52
5.1.3. Frecuencia de Trabajo	52
5.1.4. Condiciones Ambientales	53
5.2. Caracterización del Ensayo	53

5.2.1. Elección del Acelerómetro	53
5.2.2. ADXL345	55
5.2.3. Condiciones del Ensayo	68
6. Campaña de Medidas	71
7. Herramienta desarrollada	74
7.1. Adquisición de Datos	77
7.2. Procesado	78
7.3. Representación Gráfica	80
8. Resultados	83
8.1. CARRERA a 3 km/h	85
8.2. CARRERA a 5 km/h	88
8.3. CARRERA a 10 km/h	91
8.4. ESCALERAS A 33 spm	94
8.5. ESCALERAS A 86 spm	97

V	Conclusiones	100
VI	Líneas Futuras	104
VII	Conclusión Personal	108
VIII	Anexos	111
8.6.	ANEXO A	112
8.7.	ANEXO B	121
8.8.	ANEXO C	122
8.9.	ANEXO D	123
8.10.	ANEXO E	129

Índice de figuras

2.1. Generador Solenoidal Inercial	18
2.2. Esquema de Funcionamiento de una Célula Fotovoltaica	21
2.3. Módulo Termoeléctrico	24
3.1. Modelo de Alternador Eléctrico	30
3.2. Ejemplo de Generador Lineal	31
4.1. Esquema básico interno de un acelerómetro con masa sísmica	36
4.2. Ejemplo esquematizado de una galga extensiométrica convencional	42
4.3. Constitución interna de un acelerómetro de galgas extensiométricas	43
4.4. Acelerómetro Piezoeléctrico	43
4.5. Esquema del Modelo Interno de un Acelerómetro Piezoeléctrico	44
4.6. Acelerómetro Capacitivo	46
4.7. Acelerómetro Térmico	47
4.8. Acelerómetro Micromecánico MEMS	48
5.1. Dimensiones del ADXL345	56
5.2. Placa de Desarrollo del ADXL345	57

5.3.	Activación del Modo de Descarga en Serie para reprogramar la placa . . .	59
5.4.	Disposición de la Placa de Desarrollo del ADXL345	59
5.5.	Esquema Electrónico de la Placa de Desarrollo del ADXL345	62
5.6.	Ejes asociados a la sensibilidad del acelerómetro	63
5.7.	Señales de respuesta según orientación respecto a la gravedad	64
5.8.	Formato de Construcción de los Datos	65
8.1.	Resultados para Carrera a 3 km/h	85
8.2.	Resultados para Carrera a 5 km/h	88
8.3.	Resultados para Carrera a 10 km/h	91
8.4.	Resultados para trabajo de subida de escaleras a 33 spm	94
8.5.	Resultados para trabajo de subida de escaleras a 86 spm	97

Índice de cuadros

2.1. Consumo del ADXL345	27
5.1. Consumo del ADXL345	60
5.2. Consumo en Modo de Baja Potencia	61
5.3. Formato de Salida de los Datos del ADXL345	66

Parte I

MOTIVACIONES

Este proyecto se realiza en el Departamento de Ingeniería Eléctrica y Energética de la Universidad de Cantabria bajo la dirección del Profesor Mario Mañana Canteli como Proyecto de Fin de Carrera de la titulación de Ingeniería Industrial.

La motivación para realizar este proyecto tiene su origen en un trabajo previo realizado por Raúl González Barros con el Profesor Mario Mañana Canteli titulado “Acelerómetro Microcontrolado Portátil”. A partir del concepto de un acelerómetro inalámbrico capaz de realizar mediciones dinámicas sin necesidad de un terminal que recoja sus resultados en tiempo real, surge la idea de llevar a cabo un estudio de las aceleraciones que se producen en el cuerpo humano – en alguna de sus partes - en el transcurso de diferentes actividades cotidianas, como correr, subir escaleras, etc..., y , a partir de los datos ergométricos obtenidos, determinar su potencial dinámico y energético y , mediante la utilización de un generador lineal, colocado en la posición idónea en cada caso, obtener energía suficiente para alimentar pequeñas cargas.

Este aprovechamiento energético que se busca en el cuerpo humano se enmarca en el concepto de “Energy Harvesting” o “recolección de energía”, proceso de obtención de energía, captura y almacenamiento de energía a partir de fuentes externas y que, en este proyecto, pretende, con la obtención de pequeñas cantidades del movimiento cotidiano del cuerpo humano, mantener la alimentación de pequeñas cargas sin conexión a la red, u otra fuente o batería; y, como aplicación práctica, conseguir el funcionamiento autónomo de pequeños dispositivos inalámbricos, convirtiéndolos en autosuficientes.

Para optimizar la recolección de energía a través del generador lineal, es imprescindible conocer en qué partes del cuerpo, situaciones y actividad corporal se genera mayor producción energética; en definitiva, realizar un atlas de aceleraciones del cuerpo humano que, en este proyecto, concretamos en dos ámbitos de esfuerzo: en movimiento de carrera y en trabajo en escaleras. La confección del atlas completo sobrepasa las expectativas de un proyecto fin de carrera como el presente y, por su complejidad, amplio recorrido y potencialidad, quizás sea merecedora de un proyecto de investigación más completo.

Parte II

OBJETIVOS

El principal objetivo de este proyecto es realizar un estudio de aceleraciones de los movimientos de un sujeto mediante ensayos en entornos conocidos y bajo condiciones particulares con el fin de conocer la posición del cuerpo humano durante la realización de dichos movimientos donde será mayor la obtención de energía a través de un generador lineal.

Con este fin se han llevado a cabo diversos ensayos para realizar una recogida de datos que mediante un estudio estadístico y físico de los mismos permita concluir en que caso será más apropiado el uso de un generador para alimentar pequeñas cargas. Los experimentos se han realizado usando tanto las instalaciones como el instrumental disponible en el Departamento de Ingeniería Eléctrica y Energética de la Universidad de Cantabria.

Para el estudio y representación de los resultados obtenidos se ha utilizado una herramienta de software desarrollada de forma exclusiva para este caso y que ha facilitado el tratamiento de los datos por el usuario final de la misma. Esta función se realizará utilizando el software MatLab, software que, por la gran variedad de posibilidades que ofrece tanto en cálculo como en las representaciones, ha resultado particularmente adecuado a las características y necesidades del proyecto.

Durante el desarrollo del estudio se ha pasado por diferentes fases. En primer lugar se han definido las condiciones para el escenario de realización de los ensayos, así como la configuración de todo el equipo que se utilizará durante todo el proceso.

Seguidamente se ha realizado una campaña de medidas en diferentes posiciones y velocidades, para realizar la caracterización de los movimientos que se pretenden estudiar.

Por último, a la luz de los datos obtenidos se ha estudiado la productividad dinámica de distintas partes del cuerpo y la posición más apropiada para colocar un generador lineal sobre las mismas a fin de obtener un óptimo aprovechamiento energético de sus movimientos ordinarios.

Parte III

ANTECEDENTES Y ESTADO DEL ARTE

Capítulo 2

Energy Harvesting

La disposición de un generador lineal que sea accionado por el movimiento humano se engloba dentro del concepto de “*Energy Harvesting*” (Recolección de Energía). En este caso, a partir del estudio de aceleraciones que se pretende realizar, se trata de conseguir encontrar un punto en el se pueda optimizar en mayor o menor grado el aprovechamiento de su colocación para la obtención de energía que alimente pequeñas cargas.

2.1. Concepto y Base Teórica

Aplicando este tipo de tecnología a sistemas y pequeños dispositivos dependientes de una batería para su funcionamiento autónomo, conseguiríamos dos grandes ventajas: Por un lado, conferiríamos al dispositivo una autonomía prácticamente ilimitada, y por otro, se eliminaría el uso de las baterías convencionales, con sus consiguientes problemas de mantenimiento y tratamiento posterior como residuo de efectos nocivos.

Por todas estas características, este proyecto se sitúa dentro de la “*Energy Harvesting*” (Recolección de Energía), anteriormente mencionado, y como un ejemplo más de

aprovechamiento de energías renovables limpias, seguras y ligadas al entorno.

La utilización de energías renovables está presente desde antiguo en la historia de la humanidad, y en las últimas décadas, debido al aumento de la preocupación internacional por los riesgos de la energía nuclear y por la ecología, han tenido un notable incremento industrial e investigador: han primado las líneas de investigación dirigidas a la búsqueda y desarrollo de energías alternativas dirigidas a la generación de la energía eléctrica que demanda la sociedad. En pequeña escala, este proyecto pertenece a esa amplia vía de investigaciones en curso sobre la captación de energía presente en el entorno para alimentar dispositivos electrónicos de baja potencia.

2.2. Orígenes

Los orígenes del concepto de *Energy Harvesting* se remontan al uso de molinos de viento y de agua, como una de las primeras formas de aprovechamiento y transformación de la energía presente en el ámbito del mecanismo para su funcionamiento. A lo largo de la Historia, y con la aparición de nuevas formas de energía, se continuó con la búsqueda de nuevas vías para la explotación de la energía del entorno, que se adaptasen a las necesidades y escenarios en cada caso.

La aparición de la Electrónica, terminó de definir la idea de *Energy Harvesting* al servir de empujón definitivo para la investigación de dispositivos capaces de alimentar sensores dispositivos móviles sin el uso de baterías.

En las últimas décadas, abordar la cuestión del cambio climático y el calentamiento global del planeta, ha servido también como motivación para conseguir avances en este campo, ya que el uso de tecnología que funcione de forma autónoma y limpia, evita el uso de combustibles fósiles u otro tipo de energías no renovables, y por consiguiente ayuda a reducir las emisiones de gases de efecto invernadero.

2.3. Tipos de Fuentes

Las fuentes ambientales de energía se suelen clasificar según la naturaleza de la fuente de origen. Dentro de esta clasificación, las de principal relevancia en un típico escenario al aire libre son las siguientes:

- Vibración (Energía Cinética)
- Luz (Energía Fotovoltaica)
- Calor (Energía Térmica)
- Radio Frecuencia (Energía Inductiva)

Existen a su vez, otras energías alternativas, de menor importancia actualmente, que están siendo objeto de investigación y desarrollo. Un ejemplo de estas variantes sería la Emisión Radiactiva (Energía Nuclear).

Durante el proceso de selección del sistema de obtención de energía del dispositivo a alimentar, se debe tener en cuenta las características particulares de cada uno de los tipos de fuentes, de forma que podamos adaptar el diseño del sistema, a las condiciones de uso para las que está siendo diseñado. En muchos casos, sin embargo, es necesario utilizar otro tipo de fuente que o bien haya incorporado nuevos sistemas o bien que haya adaptado el ya existente. Estas situaciones se suelen dar cuando se cuenta con una única fuente de suministro energético que por circunstancias diversas- económicas, geoestratégicas o de producción limitada – no ofrece garantías de disponibilidad permanente y continua. En estos casos, es procedente la diversificación, la combinación de dos o más sistemas energéticos especialmente adaptados a las condiciones del entorno.

El diseño del dispositivo deberá también ajustarse en la medida de lo posible al escenario en el que se situará su funcionamiento durante su vida útil, ya que puede ayudar

a facilitar la labor de alimentación y reducir en ocasiones de manera sustancial el coste del mismo, algo que un mercado en crecimiento, supone una gran ventaja ante posibles competidores.

2.3.1. Vibración (Energía Cinética)

Normalmente la captación de energía cinética se produce a partir de sistemas comerciales que aprovechan la energía generada por una masa oscilante (Vibración), aunque también clasificaremos como tal la energía electrostática recolectada por células piezoeléctricas (Piezoelectricidad).

Principio de funcionamiento

El aprovechamiento del movimiento vibratorio de una masa oscilante para la obtención de energía se realiza mediante la conversión de la energía mecánica generada, en energía eléctrica utilizando un generador solenoidal inercial. Para ello se magnetiza la masa oscilante, que al desplazarse a través de una bobina genera un campo magnético induciendo una intensidad de corriente en el solenoide, según el principio de inducción electromagnética. Esta energía puede ser almacenada y utilizada para alimentar dispositivos.

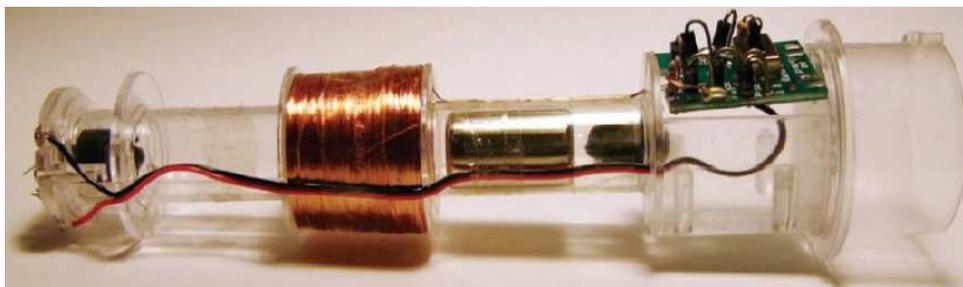


Figura 2.1: Generador Solenoidal Inercial

La masa magnética al desplazarse a través de la bobina genera una intensidad de corriente en ella[13]

La piezoelectricidad es una propiedad de un tipo particular de material cristalino, el cual, cuando se les aplica una presión mecánica, produce un voltaje proporcional a la presión aplicada. De la misma manera, al aplicar un voltaje entre dos puntos del material, éste se estirará o encojerá en la dirección del voltaje aplicado. El efecto es completamente reversible y no se produce deformación.

Por tanto al someter a un material piezoelectrico a una vibración, se generan tensiones en la estructura cristalina, que generan una diferencia de potencial en forma de onda senoidal, con una amplitud que será proporcional a la fuerza de la vibración.

Usos y aplicaciones

Los generadores solenoidales inerciales nos permiten alimentar circuitos y dispositivos en tiempo real - tales como la iluminación de LED o de otros dispositivos - en tiempo real y de forma instantánea, de modo que, al cesar la vibración o el movimiento oscilatorio, se interrumpe la alimentación de forma instantánea. Otra alternativa es el almacenamiento de la energía durante el funcionamiento del generador con vistas a su uso posterior en un periodo de tiempo distinto al de la obtención de la energía. Este tipo de sistemas son muy efectivos, pero cuentan con la desventaja de su peso y tamaño. Para las especificaciones de muchos sistemas móviles, estas características son inapropiadas.

Otras líneas de desarrollo emplean los materiales piezoelectricos para convertir las vibraciones en electricidad. En estos casos una masa comprime el material y genera la tensión necesaria entre los bornes. Un aplicación muy efectiva y que sirve de ejemplo de aplicación, es el aprovechamiento de la pisada de un individuo sobre unos dispositivos piezoelectricos dispuestos en la suela del zapato, consiguiendo hasta 7 W. por zapato en una persona de 70 kg. con una pisada de 1 Hz. de frecuencia [3].

Tipos de Vibraciones

Se pueden diferenciar 3 tipos de vibraciones en función de su origen:

- **Industria:** En estas aplicaciones pueden aprovecharse de la “frecuencia conocida” de las vibraciones que se producen, ya que los motores AC producen vibraciones derivadas de armónicos de la frecuencia de corriente. Incluimos en este caso: Motores, compresores, turbinas, bombas, transformadores, etc.
- **Estructuras:** Tienden a tener vibraciones de baja amplitud. Suelen ser las más difíciles de utilizar para alimentar baterías y regular alimentación de dispositivos. Se incluyen: Puentes, carreteras, sistemas de ventilación y calefacción, etc.
- **Vehículos:** En este tipo de aplicaciones tienden a tener vibraciones aleatorias, más complejas a la hora de realizar un aprovechamiento de las mismas. Se incluyen: Trenes, automóviles, aeronaves, etc.

2.3.2. Luz (Energía Fotovoltaica)

La energía fotovoltaica es una obtenida a través de la incidencia de una radiación electromagnética sobre un dispositivo con la capacidad de realizar un efecto fotovoltaico, siendo normalmente una lámina metálica semiconductor llamada célula fotovoltaica, o una disposición de metales sobre un sustrato llamada capa fina. Actualmente están en fase de investigación métodos orgánicos que produzcan el mismo efecto.

Es una de las energías más desarrolladas y utilizadas en los últimos años. Su rendimiento y la forma de ser utilizada son cualidades bien caracterizadas de la energía fotovoltaica. Al aire libre, el Sol puede proporcionar cerca de $100mW/cm^2$ de potencia óptica, siendo esta potencia en un día nublado de alrededor de $10mW/cm^2$. En una habitación bien iluminada (interior) la incidencia en la mayoría de superficies rondará los $0,5mW/cm^2$. Las

células solares estándar tienen un rango de eficiencia entre el 5 % y el 20 % en condiciones estándar aumentando la eficiencia según aumenten los niveles de iluminación [1].

Se comprueba como el valor obtenido es sustancialmente menor para operaciones en interior; sin embargo, incluso en interiores, la recolección de energía fotovoltaica puede proporcionar suficiente potencia para alimentar dispositivos de baja potencia como pueden ser sensores inalámbricos.

Principio de funcionamiento

El efecto fotovoltaico deriva directamente del efecto fotoeléctrico, que consiste en la capacidad de un material que se expone a suficiente energía lumínica despiden electrones. En el caso de la energía fotovoltaica, un semiconductor expuesto a la luz, arranca un electrón creando al pasar un hueco. Normalmente, el electrón encuentra rápidamente un hueco para volver a llenarlo disipando energía.

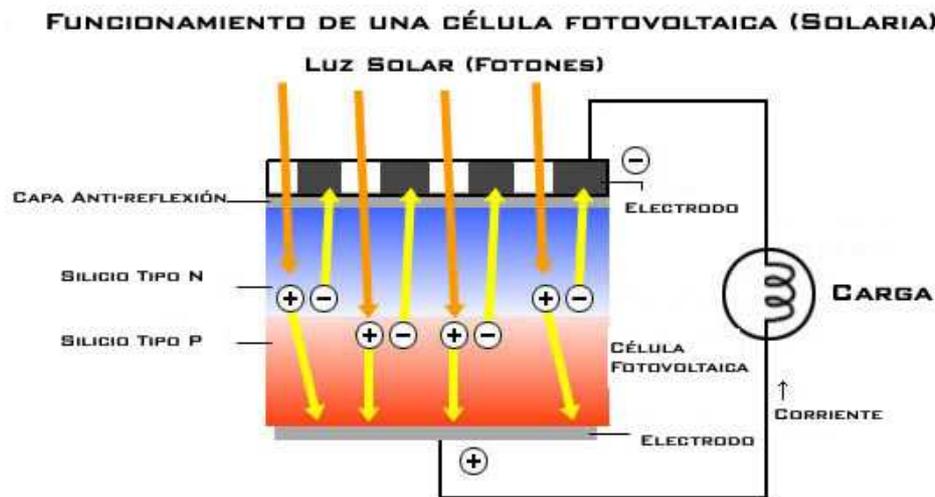


Figura 2.2: Esquema de Funcionamiento de una Célula Fotovoltaica [18]

El principio de una célula fotovoltaica es obligar a los electrones y a los huecos a avanzar hacia el lado opuesto del material en lugar de simplemente recombinarse en él,

de esta manera, se producirá una diferencia de potencial y por lo tanto, corriente entre las dos partes del material, como ocurre en una pila. Para ello, se crea un campo eléctrico permanente, a través de una unión P-N, entre dos capas dopadas.

Usos y aplicaciones

Las aplicaciones de la energía solar son habituales hoy en día. Cada vez es más frecuente ver en tejados de viviendas unifamiliares, de empresas y de hospitales y edificios públicos grandes placas solares proporcionando electricidad o energía calorífica para calentar agua. Es también habitual su uso en pequeños dispositivos electrónicos (calculadores, relojes, etc...), pero existen otros usos menos populares pero no por ello menos importantes.

En el caso particular del *“Energy Harvesting”* existen casos en los que pequeñas placas portátiles pueden servir para alimentar pequeños dispositivos que necesiten de alimentación continua en terrenos de difícil acceso a la alimentación. Otro ejemplo de los usos de la recolección de energía solar (Solar Energy Harvesting) es la refrigeración de determinados ambientes, como recipientes o estancias, en los que no puede suministrarse alimentación desde la red principal, pero para lo que es primordial que se asegure determinadas condiciones de temperatura por la conservación del material almacenado o transportado.

2.3.3. Calor (Energía Térmica)

La energía térmica puede encontrarse en prácticamente todos los ambientes y escenarios, y en muchas ocasiones de forma estable y constante. La mayoría de ellas se presenta en forma de disipación de energía calorífica en un sistema. Esta idea se ajusta perfectamente al concepto que manejamos de *“Energy Harvesting”*, y nos ofrecerá un amplio

rango de escenarios en los que realizar una recolección de energía térmica.

Los dispositivos termoeléctricos, que son aquellos capaces de convertir calor en electricidad, tienen un gran potencial en la recolección de energía térmica. Estos dispositivos no sólo implican una gran ayuda en el campo de la eficiencia energética, sino que también suponen un impacto para el medio ambiente reduciendo el consumo de combustibles fósiles usados en la obtención de energía.

Principio de funcionamiento

El fenómeno termoeléctrico describe la interacción y conversión entre calor y electricidad en sólidos, que está representada por tres efectos termoeléctricos: El efecto Seebeck, el efecto Peltier y el efecto Thomson.

- **El Efecto Seebeck:** Establece que en presencia de un gradiente de temperatura entre las uniones de dos metales o semiconductores diferentes se genera un cierto potencial eléctrico. A esta unión de dos metales se le llama termopar.
- **El Efecto Peltier:** Si en vez de aplicar una diferencia de temperatura entre los bordes de un material, aplicamos un voltaje en los mismos, la corriente circulante por el material resultará en una absorción de energía en uno de los bordes y una disipación de energía en el otro, debido al transporte de calor realizado por los electrones en movimiento, volviéndose un extremo frío y el otro caliente.
- **El efecto Thomson:** Ambos efectos, tanto Peltier como Seebeck, pueden observarse sólo en el caso de un sistema que consista en al menos dos materiales distintos. Sin embargo, la absorción o disipación de calor a lo largo de un solo material, puede darse cuando el material es sometido a una diferencia de temperatura y a una corriente eléctrica de forma simultánea.

Un generador termoeléctrico se basa en la unión de varios termopares en serie y en paralelo. La potencia generada dependerá del tamaño del generador termo-eléctrico, de la temperatura ambiente y (en el caso de recolectar el calor de un ser humano) del nivel de actividad del metabolismo.

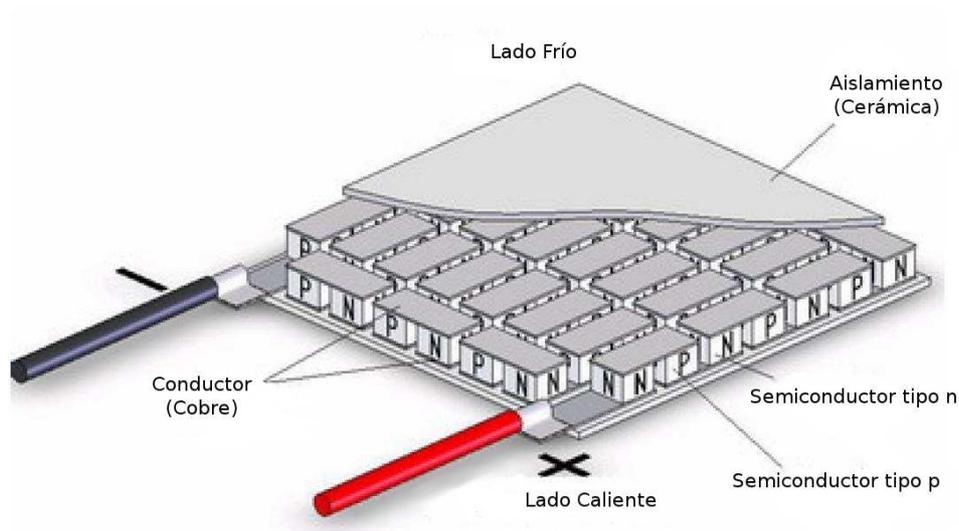


Figura 2.3: Módulo Termoeléctrico [19]

Usos y aplicaciones

Existen infinidad de usos para esta forma de “*Energy Harvesting*”, aunque las investigaciones más recientes apuestan por el aprovechamiento del calor disipado durante el funcionamiento de automóviles y cualquier tipo de vehículo, mejorando así el rendimiento del combustible utilizado.

Existen también aplicaciones novedosas como la posibilidad de captar energía a partir de la luz y la diferencia de temperatura que se produce en fuentes de luz artificiales, mejorando igualmente su rendimiento y abriendo un gran potencial en aplicaciones biomédicas y meteorológicas.

En cualquier caso, las aplicaciones más usuales continuarán siendo aquellas que permiten captar energía del ambiente, particularizando para cada escenario las posibilidades y su viabilidad, según el uso y necesidades de cada caso.

2.3.4. Radiofrecuencia (Energía Inductiva)

La radiofrecuencia es cualquier tipo de onda electromagnética dentro del rango entre los 3 kHz y los 300 GHz. Con el avance de las telecomunicaciones, nos encontramos cada vez más rodeados de radiofrecuencia generada por las estaciones emisoras de señales de televisión, radio, satélite, redes móviles, Wi-Fi, etc.

Con la cantidad actual de estaciones radioemisoras distribuidas por todo el territorio, se puede considerar la radiación que provocan como una fuente de energía uniforme y estable, sobre todo en ambientes urbanos. Por otro lado los sistemas electrónicos que recogen esta energía no pueden proporcionar una potencia muy elevada, a no ser que se encuentre muy cerca del emisor o involucren un área muy grande.

Principio de funcionamiento

Llamamos onda electromagnética a la forma de propagación de la radiación electromagnética. Esta radiación se produce al aplicarle una tensión variable a una antena. El cambio de voltaje y la circulación de corriente producen un campo electromagnético que se propaga por el aire transportando la energía. Esta energía se capta con otra antena, generando corriente aparentemente de la nada. La radiofrecuencia se propaga a la velocidad de la luz, de la misma manera que cualquier radiación electromagnética, pero su intensidad se reduce con el cuadrado de la distancia que recorre. Es por ésto que la eficiencia de estos sistemas suele ser bastante baja.

Modos de Funcionamiento

El modo de extraer energía eléctrica a partir de la radiofrecuencia del ambiente lo podemos enfocar de 3 maneras diferentes:

- A través de una antena de banda ancha para todo tipo de radiofrecuencia, intentando captar toda la energía posible en todo el espectro de frecuencias.
- Con una antena diseñada para aprovechar una frecuencia específica que se sabe que se encuentra en el entorno.
- Mediante un sistema que genere potencia en una frecuencia conocida y con una antena sintonizada.

Usos y aplicaciones

Las aplicaciones de este tipo de captación de energía son prácticamente infinitas, e involucran a cualquier dispositivo portátil que tenga necesidad de alimentación. Los usos en ámbitos civiles, médicos, militares, etc., son abundantes, aunque pocos son los que se hallan en una fase lo suficientemente avanzada como para ser operativos.

La radiación de RF puede ser una buena solución si tenemos la fuente de energía cerca del circuito que empleará dicha energía y si además se debe repartir la energía a más de un dispositivo en el circuito global.

2.4. Rendimiento y Viabilidad

Es indudable que se puede captar energía del ambiente. Es una práctica que se lleva realizando desde hace tiempo en muchos ámbitos de la industria. Dependiendo de la

fuente y tipo de energía utilizada, la tecnología actual permite absorberla en mayor o menor grado. A medida que ésta avanza, la eficiencia de los sistemas es mayor, y por lo tanto se mejoran los resultados. Conviene tener una idea aproximada de la capacidad de obtención de cada una de las fuentes y tecnologías, así como un registro comparativo entre ellas que muestre la diferencia en las magnitudes con las que se llega a trabajar en cada caso.

A día de hoy, los valores típicos que se obtienen para las distintas tecnologías son los que se muestran recopilados en la siguiente tabla.

Cuadro 2.1: Consumo del ADXL345

	<i>Conditions</i>	<i>Power Density</i>	<i>Area of Volume</i>	<i>Energy/Day</i>
Vibration	$1m/s^2$	$100 \mu W/cm^3$	$1cm^3$	$8,64J$ (assuming continuous vibration)
Solar	Outdoors	$7.500 \mu W/cm^2$	$1cm^2$	$324J$ (assuming light is available 50 % of the time)
Solar	Indoors	$100 \mu W/cm^2$	$1cm^2$	$4,32J$ (assuming light is available 50 % of the time)
Thermal	$\Delta T = 5^\circ C$	$60 \mu W/cm^2$	$1cm^2$	$2,59J$ (assuming heat is available 50 % of the time)

2.5. Actualidad en Energy harvesting

La industria está mostrando mucho interés en esta forma de aprovechamiento de energía dadas las numerosas aplicaciones que se pueden desprender de estas técnicas. En muchas aplicaciones la tecnología está suficientemente madura para poder implementar sistemas de alimentación basados en la “recolección energética” (Energy Harvesting). Esto, a nivel de diseño, evita tener que incorporar baterías, diseñar sistemas de recarga o de reemplazo. Pero a nivel de usuario simplemente permite a este olvidarse por completo del mantenimiento. El reto que se plantea por otro lado está en la exigencia de mínimo consumo que se impone dentro de estos diseños para hacerlos viables.

Otras aplicaciones incorporan sistemas de *Energy Harvesting* para aumentar la autonomía de los dispositivos, siendo esta técnica un apoyo al sistema de alimentación tradicional. Así, aún teniendo las tradicionales baterías alimentando los dispositivos, podemos extender su funcionamiento más allá de lo que exclusivamente la carga de la batería es capaz de mantener.

La unión de la tecnología wireless junto con la de *Energy Harvesting* da lugar a nuevas aplicaciones para monitorización de procesos con sensores autónomos y distribuidos. Estas redes de sensores, una vez instalados no requieren de atención por parte del usuario en términos de mantenimiento. Esta línea de desarrollo es una de las que más interés despierta entre los fabricantes y actualmente existen varios proyectos de desarrollo de productos que pretenden llevarlo en un corto espacio de tiempo al mercado.

Capítulo 3

Generadores Lineales

Un generador eléctrico es todo dispositivo capaz de mantener una diferencia de potencial entre sus bornes. Reciben el nombre de *alternadores*, aquellos dispositivos capaces de transformar energía mecánica en eléctrica, generando, mediante fenómenos de inducción, una corriente alterna. Aunque la corriente generada es corriente alterna, puede ser rectificadas para obtener una corriente continua. El proceso inverso sería el realizado por un motor eléctrico, que transforma energía eléctrica en mecánica.

3.1. Principio de Funcionamiento y Base Teórica

Un alternador genera electricidad usando el mismo principio que los generadores de corriente continua. Una variación en el campo magnético que rodea a un conductor, induce una corriente eléctrica en dicho conductor. A partir de este principio, se desarrolla el funcionamiento de un alternador.

Comúnmente, un imán giratorio (rótora), gira dentro de un conjunto de conductores fijos enrollados en bobinas sobre un núcleo de hierro (Estátor). Una entrada mecánica

producirá el giro del rotor y por tanto del imán permanente. El campo magnético generado por el imán, cruza los conductores, generando una fuerza electromotriz inducida (FEM) y en consecuencia, induciendo un voltaje de corriente alterna en los bobinados del estátor.

El campo magnético del rotor puede ser producido por inducción, por medio de imanes permanentes o por un rotor bobinado alimentado con corriente continua a través de anillos colectores o escobillas.

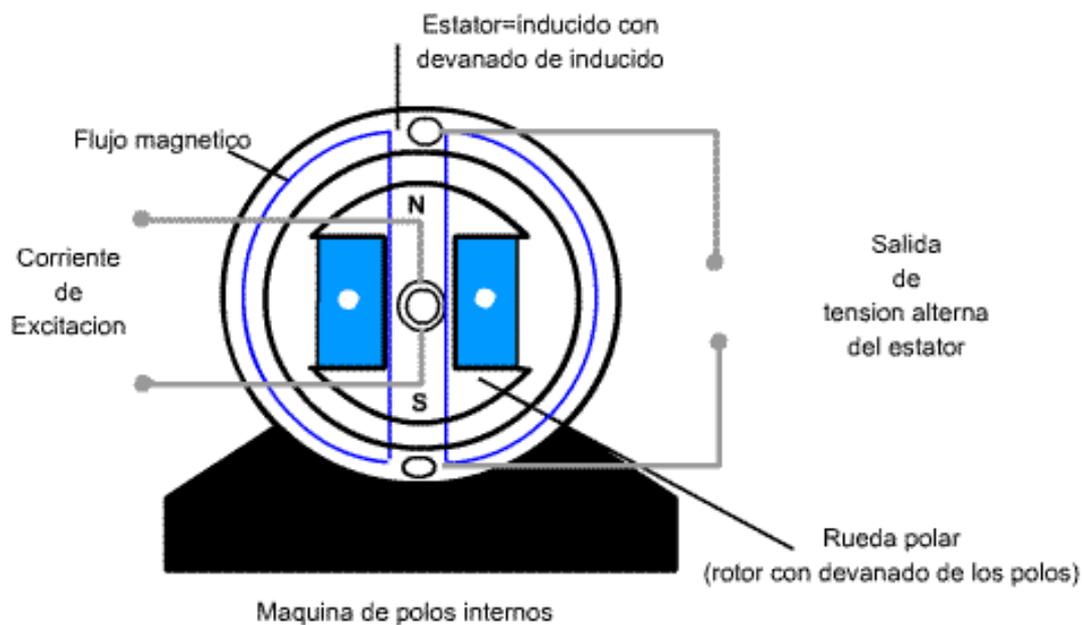


Figura 3.1: Modelo de Alternador Eléctrico

Un alternador lineal o generador lineal es en esencia un motor lineal con un funcionamiento como generador eléctrico. El alternador lineal trabaja bajo el principio de inducción electromagnética. Comunmente los alternadores trabajan con un movimiento de rotación, sin embargo, los alternadores lineales, trabajan con movimiento lineal en dirección única.

Es el más comunmente utilizado para convertir movimiento un movimiento armónico simple directamente en energía eléctrica. De esta manera se evita la integración de manivelas o soluciones que permitan convertir el movimiento lineal en movimiento de rotación para que éste fuera compatible con un generador rotativo.

3.2. Usos y aplicaciones

El tipo más simple de alternador lineal, y que todos podemos tener en mente es la *Linterna Faraday*. Ésta es una linterna que contiene una bobina y un imán permanente. Al sacudirse el dispositivo hacia atrás y adelante, el imán oscila a través de la bobina e induce una corriente eléctrica. Esta corriente se utiliza para cargar un condensador, con el fin de almacenar la energía obtenida para su uso posterior. El dispositivo puede entonces producir luz, por lo general de un diodo emisor, hasta que la energía del condensador se descarga. A continuación se puede recargar repitiendo la agitación anterior.



Figura 3.2: Ejemplo de Generador Lineal

Como se observa en la figura, que sirve de ejemplo de aplicación para un generador lineal, éste consta de un bobinado con un imán que oscila en su interior induciendo la corriente eléctrica que hará funcionar la iluminación de la linterna

Cabe destacar que existen diferentes dispositivos en desarrollo para el uso de alternadores lineales para la generación de electricidad.

Capítulo 4

Estudio de Aceleraciones

El estudio de aceleraciones (en inglés “*Accelerometry*”), se define como la determinación cuantitativa de la aceleración y deceleración del conjunto del cuerpo humano o de parte de él durante la realización de una tarea.

4.1. Concepto y Orígenes

La aceleración es una magnitud que indica la variación de velocidad por unidad de tiempo a la que está sometido un cuerpo en movimiento. La aceleración, en medidas dinámicas, es capaz de dar información sobre la magnitud de un impacto (choque) o sobre la amplitud y frecuencia de una vibración. En medidas estáticas, la aceleración puede dar información sobre fuerzas centrífugas, deceleraciones o incluso inclinaciones aprovechando la medida de aceleración de la gravedad.

La gravedad terrestre es la aceleración provocada por la atracción gravitatoria de la Tierra. Se considera una constante, aunque en realidad el valor de la gravedad local no es exactamente el mismo en cualquier posición de la Tierra. Ésta varía según la altitud y

la latitud debido, esta última, a que la Tierra tiene una ligera forma de esferoide.

Generalmente, para realizar cambios de unidades de fuerza, se suele aproximar el valor de la aceleración de la gravedad por $9,80665\text{m/s}^2$, lo cual significaría que un cuerpo en caída libre en el vacío pasa de 0 a $9,80665\text{m/s}$ en 1 segundo.

El acelerómetro es un transductor electromecánico que capta la aceleración a la que está sometido físicamente y proporciona a su salida una variación de magnitud eléctrica proporcional. Esta magnitud eléctrica depende de la tecnología del elemento sensor y del acondicionamiento de señal que incorpore el dispositivo.

El estudio de aceleraciones es una herramienta adecuada para la monitorización de movimientos humanos de una forma objetiva y fiable, aplicable en la vida diaria de los sujetos sin implicar grandes costes. Los acelerómetros están siendo utilizados en la monitorización de diferentes movimientos. Se pueden obtener una amplio abanico de medidas como: clasificación de movimientos, valoración del nivel de actividad física, estimación del gasto de energía metabólica, medida del equilibrio, ritmo de marcha y control al levantarse-sentarse. Combinando la este análisis de aceleraciones con giróscopos y magnetómetros se podrá añadir información relacionada con la orientación y los cambios de posición

4.2. Antecedentes y Aplicaciones

La aceleración es una cantidad física fundamental, manifestada de muchas maneras gravedad, vibración, actividad sísmica, etc. La medición de la aceleración de manera continua, exacta y a bajo coste, abre numerosas aplicaciones para los acelerómetros.

El acelerómetro es capaz de captar las aceleraciones producidas en un choque o una vibración dentro de un rango de frecuencias y amplitud limitado. En ocasiones, el ace-

lerómetro también es capaz de captar la inclinación de un plano de superficie midiendo la aceleración de la gravedad en su eje de actuación, en cuyo caso al acelerómetro se le suele llamar inclinómetro o clinómetro.

Los acelerómetros están siendo cada vez más atraídos hacia mercados diferentes del automotriz y de la aviación, donde pueden ser usados para medir la inercia. Ellos también miden la inclinación, característica que es usada principalmente para transportes, perforación, telemetría, navegación de ciegos u otras aplicaciones médicas o choques, usado también para mediciones sísmicas, monitoreo del estado de las máquinas.

Los fabricantes de ordenadores portátiles buscan formas para hacer sus productos más seguros. Los elementos con más facilidades para dañarse son los dispositivos de almacenamiento masivo (con la consecuente pérdida de información almacenada), discos duros particularmente. El delicado mecanismo que lee y escribe información a los discos, flota sobre los discos; un movimiento repentino puede provocar fácilmente un problema, destruyéndose la información. Un acelerómetro puede detectar el “ataque del daño potencial”, contrarrestar los choques y evitar que se dañe el disco.

Las aplicaciones militares incluyen ingeniosos sistemas de detonación para misiles y bombas. En este caso un acelerómetro forma parte del sistema difuso, la detección de impacto por la rápida desaceleración asociada. La continua variación de salida del acelerómetro sería rápidamente analizado, estableciendo el instante preciso en que la carga explosiva debe ser detonada produciendo el daño máximo sobre el objetivo.

También puede ser utilizado para monitorizar máquinas de salud, maquinas de rotación para mostrar las características de vibración; grietas o fatigas de las máquinas, monitorizando continuamente la vibración de una máquina, es posible avisar de algún fallo inminente.

Estos son algunos ejemplos de usos actuales de estos dispositivos. En nuestro caso nos centraremos en el uso del *Estudio de Aceleraciones*.

Durante los últimos años, los estudios de aceleraciones basados en el uso de acelerómetros para determinar patrones de movimiento en el ser humano han pasado a ser cada vez más frecuentes entre los investigadores. Las ventajas del uso de acelerómetros en contraposición al uso de instrumentos más tradicionales de análisis del modo de caminar humano, incluyen no sólo el bajo coste del ensayo, dado que los acelerómetros no restringen los experimentos al entorno del laboratorio, sino también el reducido tamaño que hoy en día alcanzan estos dispositivos, reduciendo los errores asociados durante el desplazamiento (no restringen el movimiento del sujeto) y por tanto a la velocidad asociada.

4.3. Principio de Funcionamiento

A pesar de las diferencias encontradas entre las distintas tecnologías utilizadas en la fabricación de acelerómetros, todas ellas se basan en la segunda ley de Newton que describe la ecuación fundamental de la dinámica.

$$\bar{F} = m \cdot \bar{a} \quad (4.1)$$

La estructura básica interna de un acelerómetro se ilustra en la figura. En ella se observa cómo se dispone una masa llamada “masa sísmica” o “masa inercial” fijada sobre un sensor de fuerza. Dicho sensor de fuerza se encuentra anclado a la carcasa por un extremo y a la masa sísmica por el otro.

De manera básica, el funcionamiento es el siguiente:

Al aplicar una aceleración en el eje de actuación del acelerómetro, la masa inercial ejerce una fuerza de tracción o compresión sobre el sensor de fuerza. Este sensor capta dicha fuerza que será directamente proporcional a la aceleración y al valor de la masa inercial. Conociendo el valor de dicha masa, es posible conocer el valor absoluto de aceleración

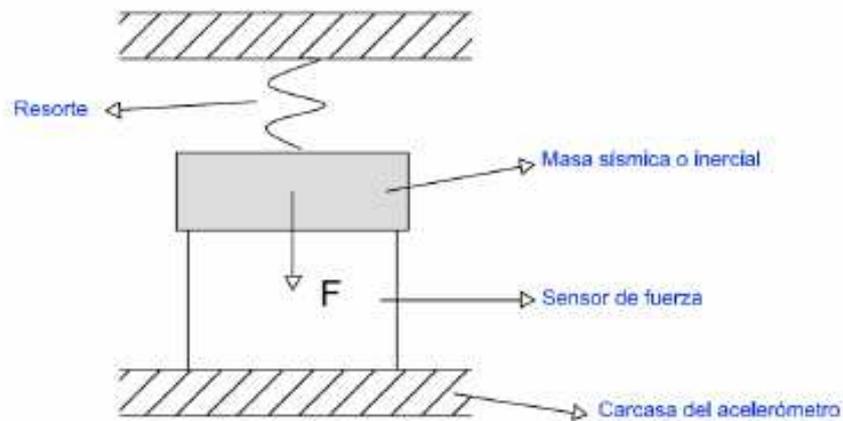


Figura 4.1: Esquema básico interno de un acelerómetro con masa sísmica

aplicando la primera ley de Newton.

La fuerza ejercida por la masa sísmica respecto a la aceleración del acelerómetro se modela por una función de transferencia de segundo orden con bajo coeficiente de amortiguamiento. Esto explica el hecho de que el sistema siempre posea una frecuencia de resonancia con un pico abrupto de amplitud.

4.4. Tipos de Acelerómetros

A la hora de escoger el acelerómetro adecuado para el ensayo es necesario tener en cuenta aspectos como el rango, sensibilidad, respuesta en frecuencia o el tipo de señal de salida. Todos estos parámetros dan información del tipo de aceleración que es capaz de medir el acelerómetro, ya sea una aceleración estática, una vibración de mayor o menor frecuencia o un impacto.

4.4.1. Acelerómetros para Medidas Estáticas

Son aquellos que ofrecen respuesta ante aceleraciones que no varían en el tiempo o ante aceleraciones de muy baja frecuencia o frecuencia nula. Un ejemplo de este tipo de aceleraciones es la aceleración de la gravedad.

En ensayos de automoción este tipo de aceleraciones suelen ser de valor de amplitud relativamente bajo, desde décimas de g hasta aceleraciones del orden de los 15g de forma continuada en las pruebas más severas.

4.4.2. Acelerómetros para Medidas Dinámicas

Los acelerómetros para medidas dinámicas están preparados para medir señales de aceleración que varían rápidamente en el tiempo. Se pueden diferenciar dos tipos de medidas dinámicas, las persistentes (también llamadas vibraciones) y las transitorias.

Aceleraciones Dinámicas Persistentes

Son el tipo de aceleraciones dinámicas que se prolongan en el tiempo. Este tipo de vibraciones se puede descomponer en términos de Fourier y su estudio se centra en un ancho de banda que puede ir desde los 20Hz a los 2000Hz. Dependiendo del tipo de ensayo de vibraciones, las amplitudes máximas que se pueden llegar a obtener están en torno a los 40g.

Aceleraciones Dinámicas Transitorias

Las aceleraciones dinámicas transitorias son las que suceden en un espacio de tiempo muy breve y suelen venir provocadas por impactos. La amplitud de este tipo de aceleración

suele ser mayor que la provocada por una vibración, pudiendo llegar a sobrepasar los 1000g.

Un impacto puede tener una duración desde menos de 1ms en ensayos de choque en la carrocería, hasta del orden de 10 ó 20 *ms* en otro tipo de ensayo de impactos menos severos.

4.4.3. Acelerómetros para Medidas en varias dimensiones

En ocasiones es preciso medir un vector aceleración cuya dirección no es completamente conocida o varía en el tiempo de manera arbitraria. En estos casos, el uso de un acelerómetro constituido por un único eje de medición no es la solución correcta. La forma correcta de medir la aceleración resultante es mediante acelerómetros biaxiales o triaxiales.

Acelerómetros Uniaxiales o Monoaxiales

Un acelerómetro uniaxial posee un elemento sensor que es capaz de medir la aceleración paralela a su eje de actuación. Dicho eje es fijado por el fabricante y la sensibilidad que nos proporciona está directamente ligada a ese eje de actuación.

Si el eje del acelerómetro no se coloca en paralelo con el vector de aceleración a medir, no se obtendrá la sensibilidad adecuada y por tanto los resultados obtenidos no serán los deseados. Dicho eje coincide siempre con la perpendicular a la superficie de anclaje del acelerómetro.

Los acelerómetros uniaxiales son los más utilizados en el campo de la instrumentación debido a que, por lo general, los impactos que se llevan a cabo en los ensayos son totalmente controlados y la dirección del vector aceleración que se desea medir es conocida en

el momento de producirse.

Acelerómetros Biaxiales

Los acelerómetros biaxiales resultan útiles en aplicaciones que requieran medir aceleraciones que suceden en un plano. Como su nombre indica, un acelerómetro biaxial posee dos sensores dispuestos de manera perpendicular, lo cual permite que sea capaz de medir la aceleración en dos ejes de coordenadas.

Acelerómetros Triaxiales

Los acelerómetros triaxiales permiten medir la aceleración en tres dimensiones. Poseen un total de tres elementos sensores, uno para cada eje (X, Y, Z). Cada sensor proporciona una señal eléctrica en función de la aceleración del eje en el que está orientado internamente. Para conocer el módulo, dirección y sentido de la aceleración resultante basta con realizar la suma vectorial de las aceleraciones medidas por cada eje. Lo único que hay que tener en cuenta es la posición de referencia del acelerómetro para conocer la dirección real de la aceleración.

Se utilizan para medir aceleraciones que se producen en ejes distintos al de la superficie de anclaje del acelerómetro, o para captar aceleraciones (módulo y sentido) cuya dirección varía en el tiempo, como por ejemplo un objeto determinado durante un ensayo de choque.

Los acelerómetros triaxiales (así como los biaxiales) disponen de una superficie plana para cada eje que sirve como base preparada para su calibración.

4.4.4. Acelerómetros según Impedancia de Salida

Los acelerómetros convierten la aceleración de gravedad o de movimiento, en una señal eléctrica analógica proporcional a la fuerza aplicada al sistema, o mecanismo sometido a vibración o aceleración. Esta señal analógica indica en tiempo real, la aceleración instantánea del objeto sobre el cual el acelerómetro está montado.

Acelerómetros de Alta Impedancia

La salida de carga del cristal tiene una impedancia de salida muy alta y se puede obtener fácilmente. Se pueden emplear técnicas especiales para obtener la señal del sensor. La alta impedancia resultante del acelerómetro es útil donde las temperaturas exceden los 120°C, prohibiendo el uso de sistemas microelectrónicos dentro del sensor.

Acelerómetros de Baja Impedancia

En un acelerómetro de baja impedancia se deben emplear sistemas microelectrónicos ubicados dentro de la carcasa del sensor, para detectar la carga generada por el cristal piezoeléctrico.

4.4.5. Tecnologías de Fabricación

Existen distintas tecnologías en la fabricación del elemento sensor del acelerómetro. Las propiedades y características del sensor vienen en gran parte condicionadas por la tecnología, por lo que para la correcta elección del acelerómetro hay que conocer las diferencias entre una tecnología y otra.

Acelerómetros Piezorresistivos

Un sensor piezorresistivo es un transductor que transforma un esfuerzo mecánico experimentado por el elemento sensor en una variación de su resistencia eléctrica. El efecto piezorresistivo se produce cuando un filamento metálico o semiconductor es traccionado o comprimido, variando su longitud y, por tanto, su resistencia eléctrica. Esto es debido a la propiedad de todo conductor, cuya resistencia eléctrica es siempre inversamente proporcional a su longitud y directamente proporcional a su sección.

$$R = \frac{\rho \cdot L}{A} \quad (4.2)$$

Siendo ρ la Resistividad del conductor, L la longitud del conductor y A la sección del conductor.

En el caso del filamento o sustrato semiconductor, la variación de resistencia con la deformación es causada por una variación en la concentración de portadores. La **galga extensiométrica** se basa en dicho efecto y constituye el elemento sensor de transductores piezorresistivos.

La extensimetría se basa en la ley de elasticidad de Hooke la cual explica la relación entre la tensión y la deformación de un cuerpo sometido a esfuerzos mecánicos. Esta ley establece que todo cuerpo homogéneo sometido a un esfuerzo mecánico, sufre un alargamiento unitario proporcional a la fuerza aplicada dentro del rango elástico del material. Siendo la expresión para el caso unidimensional la siguiente:

$$\sigma = E \cdot \epsilon \quad (4.3)$$

El filamento conductor o semiconductor de la galga, también llamado hilo activo, es

arrollado y encolado sobre un fino soporte aislante. Un ejemplo esquematizado se ilustra en la siguiente figura.

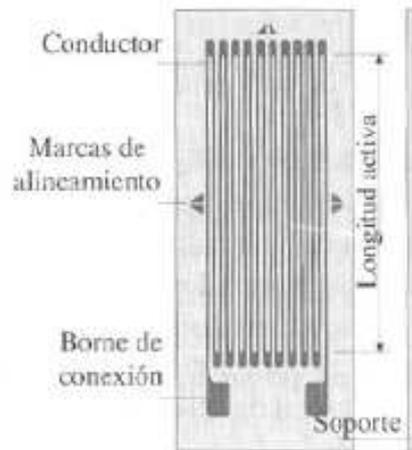


Figura 4.2: Ejemplo esquematizado de una galga extensiométrica convencional

El soporte de la galga se adhiere sobre la superficie en la cual se desea medir un esfuerzo. Partiendo de la hipótesis de que la galga sufrirá la misma deformación que el material al cual va adherida, al provocar un esfuerzo sobre la superficie en el eje de actuación de la galga, el hilo activo experimenta una ligera variación en su longitud y sección provocada por la deformación elástica del material sobre el que se está midiendo.

Los acelerómetros del tipo piezoresistivo son los de uso más extendido en el campo de la instrumentación. Esto es debido a que son los más económicos y sus características se adaptan a casi cualquier aplicación. El elemento sensor está constituido por un conjunto de galgas extensiométricas que miden la fuerza ejercida por la masa sísmica cuando el acelerómetro es sometido a una aceleración.

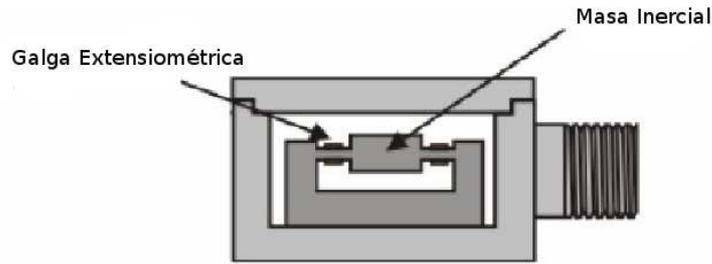


Figura 4.3: Constitución interna de un acelerómetro de galgas extensiométricas
[16]

Acelerómetros Piezoeléctricos

Los transductores piezoeléctricos son transductores activos que se basan en el fenómeno de la piezoelectricidad descubierto en 1881 por Pierre y Jacques Curie. Dicho fenómeno se da en algunos cristales (generalmente de manera natural) como el cuarzo o el rubidio y consiste en un desplazamiento de cargas eléctricas en el cristal que se produce al someterlo a un esfuerzo mecánico.

Este fenómeno hace que aparezcan cargas eléctricas de sentido opuesto en ambas caras del cristal. La cantidad de carga eléctrica generada será proporcional al esfuerzo mecánico sometido.



Figura 4.4: Acelerómetro Piezoeléctrico
[20]

En el momento en que el esfuerzo cesa o se mantiene constante, también cesa el movimiento de cargas, por lo que la carga eléctrica se irá disipando hasta restablecerse el equilibrio inicial. De aquí se deduce que los transductores piezoeléctricos se utilizan para medidas dinámicas, es decir, medidas de magnitudes físicas que varían en el tiempo, como por ejemplo, una vibración o un impacto.

Al igual que en el caso de los acelerómetros piezorresistivos, el elemento sensor (en este caso el cristal piezoeléctrico) se ancla firmemente a la masa sísmica para ser sometido a una fuerza proporcional a la aceleración.

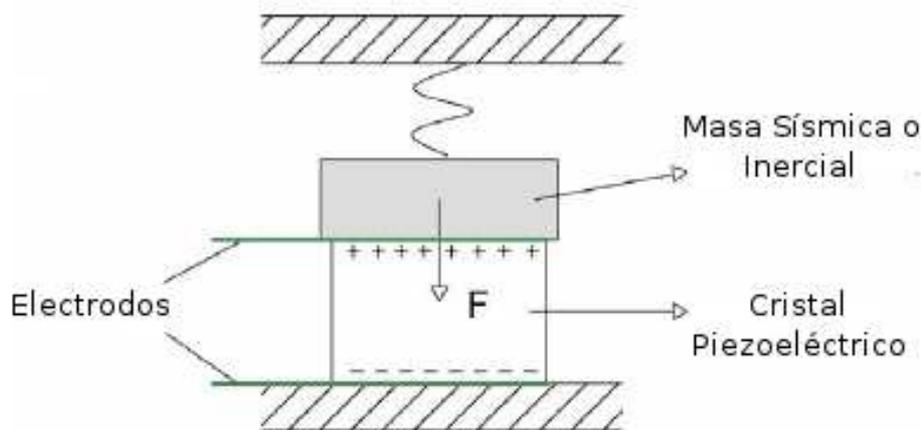


Figura 4.5: Esquema del Modelo Interno de un Acelerómetro Piezoeléctrico

El uso de acelerómetros piezoeléctricos es muy extendido en el campo de la instrumentación industrial a pesar de ser más caros que otro tipo de acelerómetros como los piezorresistivos. Esto es porque dicha tecnología proporciona una serie de características al acelerómetro que lo hacen ser el idóneo para según qué aplicaciones como vibraciones de alta frecuencia, impactos de corta duración en los que se requiera mucha precisión, estabilidad y repetibilidad, o en aplicaciones en las que las condiciones ambientales son muy adversas.

Los acelerómetros piezoeléctricos con electrónica incorporada son acelerómetros cuya tecnología está también basada en el transductor piezoeléctrico pero con la diferencia de

que el acelerómetro incorpora internamente la electrónica que hace las veces de amplificador de carga.

Acelerómetros Capacitivos

Un transductor capacitivo se basa en el efecto condensador para transformar una magnitud física en una variación de la capacidad de un condensador que es traducida en una variación de tensión eléctrica.

Un condensador consiste básicamente en dos placas conductoras separadas por un dieléctrico. La polarización de ambas placas crea un campo magnético entre ellas que hace que éstas almacenen una carga eléctrica que varía según sus áreas, la separación entre ellas y el dieléctrico utilizado.

$$C = \frac{K \cdot A}{4\pi \cdot d} \quad (4.4)$$

Siendo K la constante del dieléctrico, A el área del mismo y d la distancia de separación entre placas.

Variando la distancia entre las placas o el área de influencia del campo magnético se consigue variar proporcionalmente la capacidad del condensador.

Un acelerómetro capacitivo está constituido por uno o más condensadores que varían su capacidad por efecto de la aceleración por la variación de la separación entre placas.

El dispositivo realmente trabaja en un lazo de control electrónico de fuerza/balanceo. Este lazo de control evita el movimiento de la masa en aceleración, por la aplicación de una fuerza igual pero opuesta creada por la aplicación de un voltaje en las placas del condensador. Este voltaje aplicado es directamente proporcional a la aceleración.

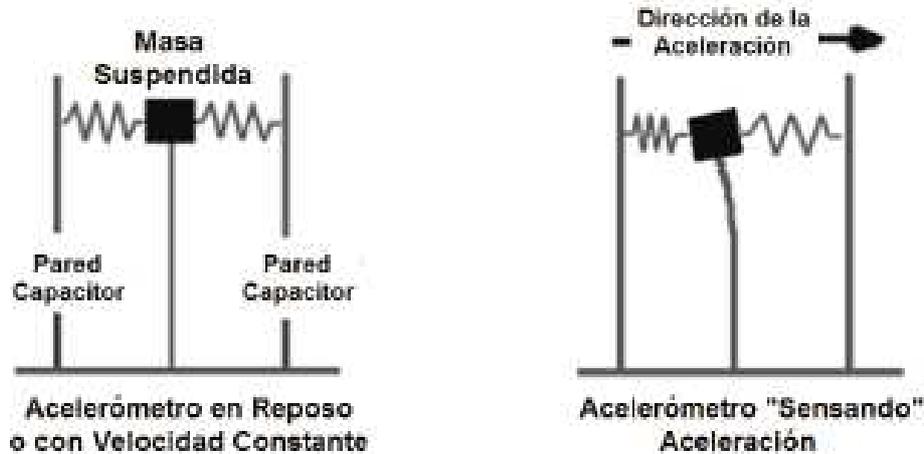


Figura 4.6: Acelerómetro Capacitivo
[16]

Acelerómetros Térmicos

Se trata de un nuevo acelerómetro basado en la convección termal. Este tipo de acelerómetro posee un diseño de tecnología MENS muy simple y práctico al mismo tiempo; simplemente utilizando un sustrato de silicio en el cual se hace un hueco para meter una pequeña resistencia que hace de calentador, con dos termopares⁴ en los extremos. Con esta estructura conseguimos que se forme una cavidad de aire caliente, llamamos burbuja, sobre los termopares.

La principal característica de estos dispositivos es que tienen sólo un elemento móvil, la burbuja diminuta de aire caliente, herméticamente sellado dentro de una cavidad existente en el encapsulado del sensor. Cuando una fuerza externa como el movimiento, la inclinación, o la vibración es aplicada, la burbuja de aire caliente se mueve de una forma análoga al mismo. El cambio de estado dentro de la cavidad del integrado, produce un voltaje que es función de la diferencia de temperatura y que tras ser amplificado, condicionado, se proporciona como salida el valor de un voltaje absoluto.

Para el diseño de estos acelerómetros debemos crear una zanja en la superficie del

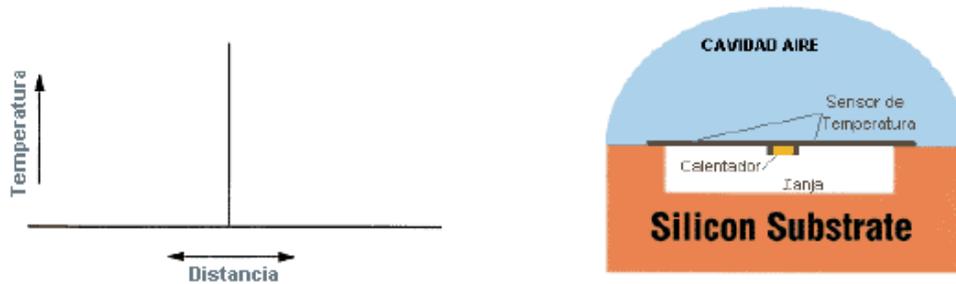


Figura 4.7: Acelerómetro Térmico [16]

silicio que conforma el sustrato del sensor. Colocamos un calentador, resistencia de silicio, suspendida en el centro de la zanja generada. Colocamos dos termopares a ambos lados del calentador de forma que queden simétricos respecto a este, teniendo como resultado una configuración muy similar a la que presenta el puente de Wheatstone.

Es necesario además, tener en cuenta al encapsular que debemos dejar una cavidad de aire, o burbuja, sobre la que se producirá la variación de las condiciones de temperatura al producirse movimiento. Este cambio de temperatura entre los termopares creará una señal diferencial que será amplificada y condicionada según las aplicaciones para las que este diseñado el acelerómetro, obteniéndose como salida del mismo.

Acelerómetros Micromecánicos (MEMS)

Los avances en tecnología electromecánica micro de los sistemas (MEMS) han permitido la detección del movimiento o los sensores de inercia, conocidos como acelerómetros, para ser puesto en ejecución en muchos usos en un buen número de industrias.

Con relación a la tecnología básica, distinguimos tres categorías principales de acelerómetros de MEMS: el capacitivo de silicio, el piezorresistivo y, finalmente, los acelerómetros térmicos.



Figura 4.8: Acelerómetro Micromecánico MEMS
[16]

Parte IV

MEMORIA DESCRIPTIVA

Capítulo 5

Planteamiento del Problema

Como primer paso, se debían decidir las posiciones del cuerpo humano sobre las que se realizaría el ensayo. A fin de que quedara representada de manera general toda la anatomía humana en un atlas de aceleraciones, se decidió reducir los ensayos a las partes mas representativas del cuerpo como son tronco y extremidades. De manera particular se decidió ensayar sobre **Brazo, Cadera y Pierna**. Se descartó el ensayo en la cabeza por razones de operatividad real, y tras consultar publicaciones [6] sobre el movimiento de esta parte del cuerpo, se estimo que las aceleraciones que se iban a medir durante el ensayo no serían representativas en comparación con el resto de resultado, y que complicarían el estudio de manera innecesaria.

Ante toda esta información, en primer lugar debemos de conocer las herramientas con las que contamos a la hora de realizar el ensayo de análisis de aceleraciones y la posterior obtención de energía a partir del generador lineal, así como los resultados que se esperan obtener y el formato en el que se obtendrán de manera que podamos obtener conclusiones a partir de ellos. Se definen por tanto dos etapas consecutivas e interrelacionadas.

Inicialmente en una primera etapa, se deben definir las condiciones particulares de los ensayos, así como conocer en detalle las características del instrumental con el que

contamos, de manera que podamos conocer todos los aspectos del experimento a realizar.

En una segunda etapa se desarrollarán las herramientas necesarias para la adquisición y procesamiento de los datos recogidos, así como su posterior representación e interpretación.

5.1. Condiciones del Instrumental

En el mercado existen muchas posibilidades de sensores para medir la aceleración, la elección de uno de ellos depende de las características del sensor: los márgenes de valores de la aceleración que admite, capacidad para medir en continua o sólo en alterna, la máxima frecuencia a la que puede trabajar, los parámetros característicos del sensor...

En este proyecto, la elección del acelerómetro más adecuado se hará en función de los siguientes criterios:

- Consideraciones constructivas (El espacio disponible, cableado, etc.)
- Los valores máximos y mínimos del nivel de la señal que esperamos (Rango).
- Frecuencia de trabajo o margen de frecuencias de uso.
- Condiciones ambientales (Temperatura de trabajo, aspectos ambientales, etc.)

5.1.1. Consideraciones Constructivas

Como punto de partida para la elección del material a utilizar debemos tener claro qué es lo que se desea medir. En este caso se partía del propósito de medir la aceleración presente en el movimiento natural humano durante las actividades de correr y subir escaleras, por lo que con la intención de no condicionar el ensayo durante su realización

se precisó la necesidad de utilizar un acelerómetro que no precise de conexión en tiempo real con el terminal de recogida de datos, así como la intención de reducir el tamaño del mismo lo posible de forma que el sujeto actúe con la mayor naturalidad posible.

5.1.2. Rango de Medición

Es importante disponer de instrumental que trabaje en el rango de medición adecuado, de forma que los ensayos no pierdan detalle por un excesivo rango que no permita ver detalle en los resultados, así como se pierda información en los resultados debido a un rango demasiado corto que no permita ver los picos de aceleración durante el ensayo. Para ello se consultaron diversas publicaciones y estudios previos relacionados con el tema en los que se aproximan los valores máximos que caben esperar en los movimientos sobre los que se va a medir. Ajustándonos a las publicaciones consultadas, se estima aceptable considerar para los movimientos de andar y correr los valores aproximados de $\pm 2,5g$ y $\pm 10g$ respectivamente. De esta manera se dispuso que el acelerómetro utilizado midiera con holgura un rango de $\pm 10g$.

De la misma manera se estimó interesante utilizar un acelerómetro que dispusiera de la posibilidad de ser configurado para modificar su rango de medida a elección del usuario, de manera que se pudiera ajustar lo máximo posible según el ensayo que se fuera a realizar y obtener resultados mucho más representativos y útiles.

5.1.3. Frecuencia de Trabajo

El rango de frecuencias o frecuencia de funcionamiento que se desea para el acelerómetro, aunque no se consideró un valor particular, se estimó necesario que fuera de valores altos, de manera que el muestreo que se realice sea suficientemente grande como para conseguir unos datos de alta exactitud y visualizar fácilmente los detalles de los mis-

mos. Se consideró importante la posibilidad de modificar esta frecuencia de trabajo de manera que se pudiera adaptar el valor utilizado según las necesidades en cada momento.

5.1.4. Condiciones Ambientales

Las condiciones ambientales de uso del acelerómetro son importantes en caso de que se desee realizar ensayos en situaciones desfavorables, como en los que se impliquen temperaturas extremas o ambientes húmedo o acuáticos. En este proyecto no se han contemplado estos escenarios adversos, ya que las condiciones en las que se trabaja son las ambientales y perfectamente soportables por la gran mayoría de instrumental y electrónica utilizados en este tipo de experimentos.

5.2. Caracterización del Ensayo

El proyecto ha sido realizado en el Departamento de Ingeniería Eléctrica y Energética de la Universidad de Cantabria, de ahora en adelante, el departamento, usando tanto sus instalaciones y laboratorios como todo instrumental necesario durante los ensayos.

5.2.1. Elección del Acelerómetro

En un primer lugar se planteó el ensayo del estudio de aceleraciones, usando el acelerómetro desarrollado en un Proyecto previo en el departamento, aunque finalmente se descartó ya que tras varios ensayos preliminares se comprobaba que el tamaño del mismo no era apropiado para los mismos y que perturbaba el movimiento durante el experimento, así como la dificultad de fijación en algunas partes del cuerpo. Esta situación provocaba unos resultados equívocos que no se asemejaban a los que cabía esperar dada la naturaleza del ensayo.

Posteriormente se acotaron con exactitud los criterios que eran necesarios para garantizar un ensayo fiable. A modo de resumen, se buscaba un acelerómetro de tamaño reducido con un rango de medición superior a los y la posibilidad de configuración del rango por el usuario. Se buscaba flexibilidad en la frecuencia de trabajo y funcionamiento garantizado en condiciones ambientales de normalidad.

Tras una primera búsqueda de los dispositivos más apropiados, se decidió reducir las posibilidades a tres acelerómetros. Todos ellos basados en tecnología MEMS, triaxiales y de baja impedancia.

- **ADXL345** Ofrece un rango máximo de $\pm 16g$, pero modificable por usuario a $\pm 2g, \pm 4g$ o $\pm 8g$. Su frecuencia de trabajo es seleccionable por el usuario con un rango desde 1 Hz a 5 kHz. Tiene un tamaño reducido y ofrece numerosas posibilidades de montaje y recepción de datos. Es capaz de trabajar con normalidad bajo condiciones mucho más agresivas que las supuestas en el ensayo. Ha sido desarrollado por *Analog Devices*.
- **MMA7341** Ofrece un rango máximo de $\pm 11g$, pero modificable por usuario hasta $\pm 3g$. Su frecuencia de trabajo no es seleccionable por el usuario (11 kHz), aunque mediante la adaptación de la salida se pueden realizar muestreos alternativos. Tiene un tamaño reducido y facilidad de implementación. Es capaz de trabajar con normalidad bajo condiciones y temperaturas mucho más extremas que las supuestas en el ensayo. Ha sido desarrollado por *FreeScale Semiconductor*.
- **MMA7331L** Ofrece un rango máximo de $\pm 12g$, pero modificable por usuario hasta $\pm 4g$. Su frecuencia de trabajo no es seleccionable por el usuario (11 kHz), aunque mediante la adaptación de la salida se pueden realizar muestreos alternativos. De nuevo nos encontramos con un dispositivo de tamaño reducido y facilidad de implementación, capaz de trabajar con normalidad bajo condiciones y temperaturas mucho más extremas que las supuestas en el ensayo. Ha sido desarrollado por *FreeScale Semiconductor*.

Finalmente tras analizar las 3 opciones y valorar sus ventajas e inconvenientes, se decidió elegir el **ADXL345** de *Analog Devices*.

Se valoró muy positivamente la facilidad con la que se podía modificar la frecuencia de trabajo desde el propio usuario, así como el amplio abanico de rangos seleccionables, muy útil en caso de realizar ensayos muy particulares en los que la precisión podría ser primordial. También se estimó importante la existencia de soluciones comerciales del producto, como Placas de Evaluación, que ahorraban el trabajo de conectar y gestionar las señales del acelerómetro para ser almacenadas.

5.2.2. ADXL345

El ADXL345 es un acelerómetro, pequeño, delgado, triaxial y de muy baja potencia. Tiene alta resolución (13-bits) de medida hasta $\pm 16g$. La salida digital del dispositivo está formateada como un complemento a dos de 16 bits y es accesible a través de una interfaz SPI o I2C digital. El ADXL345 es muy apropiado en aplicaciones para dispositivos móviles. Es capaz de medir aceleraciones estáticas, como la gravedad, así como aceleraciones dinámicas resultantes de movimiento o impacto. Su alta resolución de 3.9 mg/LSB (Siendo *LSB* bit menos significativo) permite también medidas de cambios en la inclinación de menos de 1° , aunque en nuestro caso no será necesario su aplicación en medida de inclinaciones.

Las dimensiones del mismo se recogen en su hoja de características, siendo éstas las siguientes:

Descripción Constructiva (Evaluation Board)

De forma que se facilite la implementación del ADXL345 en una placa de evaluación para su uso en la toma de medidas, se encuentra una solución comercial ofrecida por

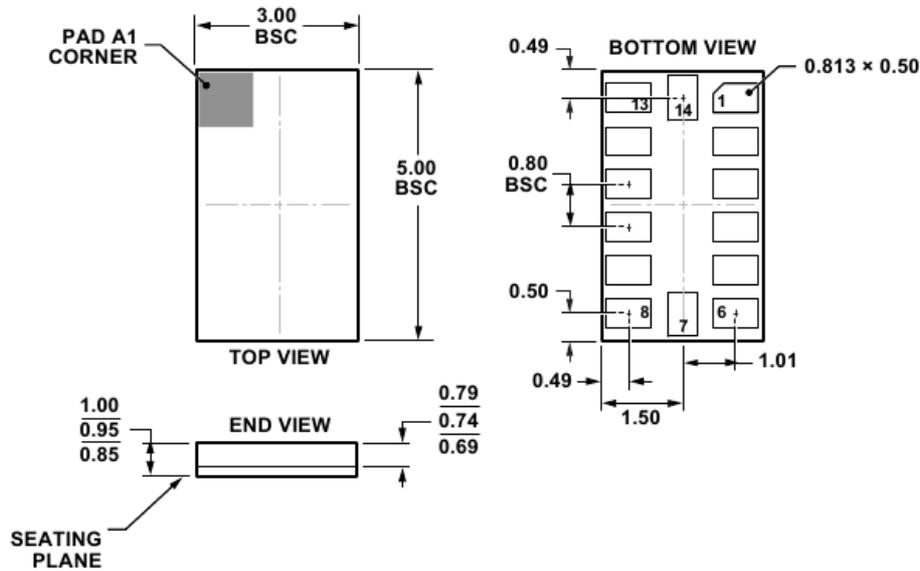


Figura 5.1: Dimensiones del ADXL345
[17]

el fabricante que consiste en una placa de desarrollo que incluye todo lo necesario para gestionar la señal obtenida y obtener un *output* digital que pueda ser procesado desde un ordenador, evitando trabajo en términos de diseño de hardware.

Este dispositivo recibe el nombre de *iMEMS® ADXL345 Sensor Datalogger and Development Board*, y es fabricado, al igual que el ADXL345, por *Analog Devices*. Es una herramienta de uso directo, que facilita el diseño de prototipos y ensayos, trabajando como un centro de recogida de datos, a través de un firmware inicial, sobre el que realizar cambios desde el usuario, de manera que se configure su uso según la necesidad particular de cada caso.

Dos baterías AAA alimentan la placa, de manera que se pueda integrar en aplicaciones inalámbricas tal y como se necesita para el caso que nos ocupa. Las comunicaciones y conexiones son gestionadas por un micro procesador ADuC7024, que puede ser reprogramable en caso de necesidad desde la interfaz que ofrece el fabricante. En este caso no se realizarán cambios por no ser necesario modificar la forma de procesar la señal, siendo

apropiada la predeterminada.

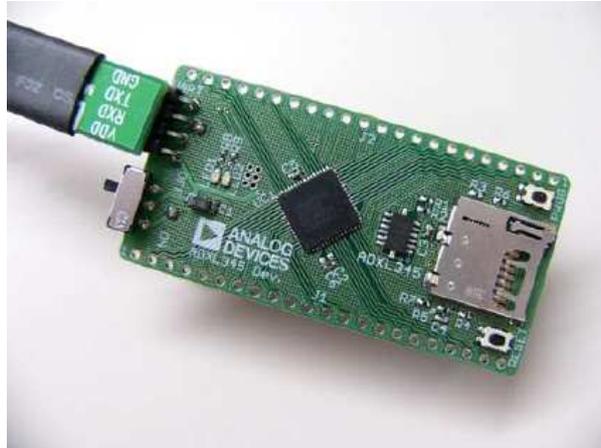


Figura 5.2: Placa de Desarrollo del ADXL345
[17]

La recogida de datos se realiza a través de una tarjeta de memoria MicroSD, ofreciendo prácticamente una capacidad ilimitada de memoria de almacenamiento y versatilidad a la hora de integrar su uso dentro del sistema de operación. Los datos son recogidos en un archivo de texto (.txt), por lo que no es necesario ningún tipo de instalación de nuevo software para realizar la lectura de las medidas y por tanto para el uso de la placa. Si se desea reprogramar la placa para variar las condiciones de uso, el fabricante ofrece la posibilidad de obtener software para su uso.

Constructivamente la placa de desarrollo consta de las siguiente características:

- Una placa de circuito impreso de doble capa de dimensiones 1.20 x 2.30 pulgadas de factor de forma.
- Dos baterías AAA para alimentación.
- Botones de Reinicio y Descarga de archivos (RESET/PROG).
- Puerto UART de 4 *pins* para cable de conexión con interfaz RS-232.
- Indicadores LED.

- Puerto de Salida y Entrada para acceso al microcontrolador.
- Firmware de prueba para toma de datos a 100 Hz.

Alimentación: Dos pilas AAA alimentan la placa. El soporte para las baterías se encuentra en la parte posterior de la placa. Un interruptor *on/off* en la parte inferior izquierda de la parte delantera controla la alimentación de la misma. El voltaje de la batería no está regulado, pero se evita el acoplamiento con la señal a través de un condensador de $10\mu F$ a nivel global y con un condensador adicional de $1\mu F$ en los *pins* del dispositivo del suministro a tierra.

Conexión con la Interfaz: El microprocesador ADuC7024 se conecta con la interfaz de usuario RS-232 mediante el cable UART. Este cable permite la conexión directa con un ordenador a través de un puerto serie. Se ha asegurado que la conexión se realiza correctamente, coincidiendo las conexiones de tierra (GND) y de tensión (VDD) respectivamente tanto en la placa como en el cable.

Botones RESET/PROG Existe un botón de RESET que permite que el usuario restablezca manualmente el dispositivo. A su vez, los botones sirven para facilitar al usuario el uso del modo de descarga. En la placa de desarrollo, el *Modo de Descarga en Serie* se inicia sencillamente manteniendo el botón de descarga en serie (PROG) pulsado, mientras que se pulsa y se suelta el botón de reinicio (RESET), tal y como se ilustra en la figura.

Indicador LED: Se dispone de dos LED en la placa. Un LED rojo (LED1) y un LED verde conectados al ADuC7024. Ambos LED pueden ser reprogramados para diferentes usos que los predeterminados, a través del firmware.

Firmware: Junto con la placa de evaluación se proporciona un firmware a través de la página web del ADXL345 en el sitio web del fabricante. Permite la descarga de un proyecto de firmware predeterminado, que puede ser modificado según la necesidad del usuario.

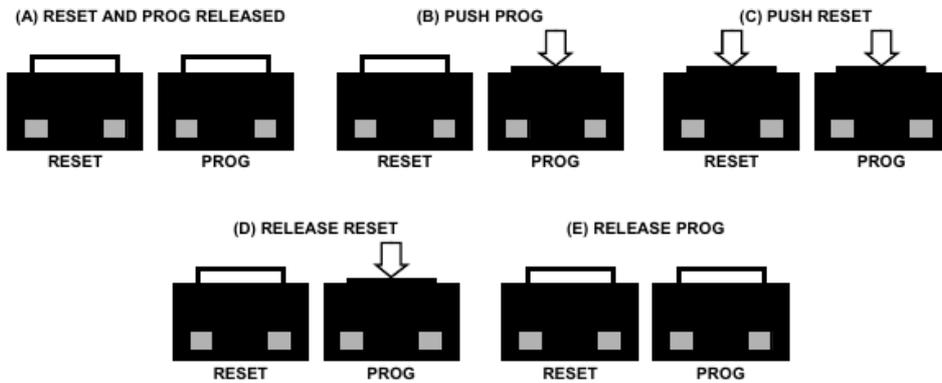


Figura 5.3: Activación del Modo de Descarga en Serie para reprogramar la placa [17]

La disposición de la placa de evaluación según expresa su hoja de características es la siguiente:

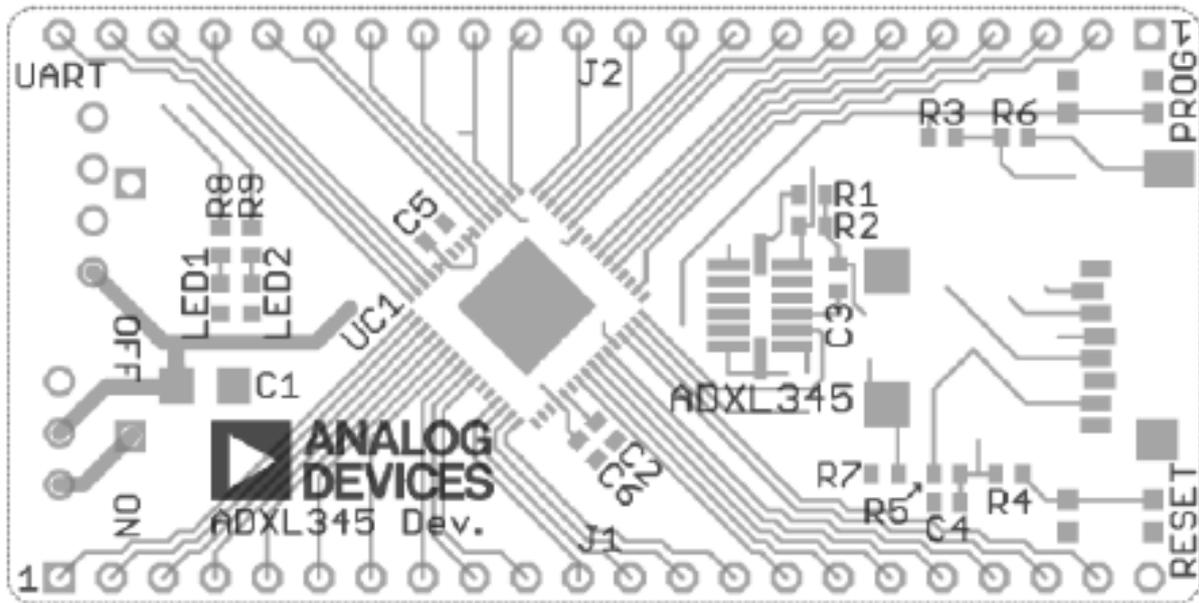


Figura 5.4: Disposición de la Placa de Desarrollo del ADXL345 [17]

Descripción Electrónica y Energética

El ADXL345 modula automáticamente su consumo energético proporcionalmente con su frecuencia de muestreo de la salida. Existe un modo de bajo consumo disponible, pero su uso no es recomendable a todas las frecuencias, por lo que se descarto para su uso en este proyecto.

De manera que se pueda comprobar los consumos de corriente del dispositivo, por si fuera necesaria una estimación de la duración de la autonomía para un ensayo, éstos quedan expresados en el cuadro 5.1, estimados para una tensión $V_s = 2,5V$.

Cuadro 5.1: Consumo del ADXL345

Frecuencia de Salida (Hz)	Ancho de Banda (Hz)	Código del Registro	$I_{DD}(\mu A)$
3200	1600	1111	140
1600	800	1110	90
800	400	1101	140
400	200	1100	140
200	100	1011	140
100	50	1010	140
50	25	1001	90
25	12.5	1000	60
12.5	6.25	0111	50
6.25	3.13	0110	45
3.13	1.56	0101	40
1.56	0.78	0100	34
0.78	0.39	0011	23
0.39	0.20	0010	23
0.20	0.10	0001	23
0.10	0.05	0000	23

[17]

En cualquier caso la estimación de los consumos no es relevante en este caso, ya que los ensayos a realizar no necesitarán de largos periodos de autonomía de la placa, y en caso de que la alimentación se agote, se podrán sustituir o recargar las pilas utilizadas y continuar trabajando sin retrasos ni problemas relevantes.

Si se quieren ahorros de energía adicionales a los conseguidos con el cambio de la frecuencia de muestreo, está disponible la posibilidad de programar un modo de baja potencia. Bajo estas condiciones, se reduce la frecuencia de muestreo interna del ADXL345, permitiendo conseguir una reducción del consumo en el rango de frecuencias de trabajo entre los $12,5Hz$ y los $400Hz$ a expensas de que el ruido producido sea ligeramente mayor.

Para introducir el modo de baja potencia, se debe establecer el bit *LOW POWER* (Bit 4) en el registro *BW RATE* de dirección $0x2C$. El consumo para este modo se reflejan en la siguiente tabla (número), para casos en los que suponga una ventaja significativa. El uso del modo de baja potencia para frecuencias que no se reflejen en la tabla, no supondrá ninguna ventaja en términos de ahorro, sobre la misma frecuencia en modo normal de potencia. Se recomienda por lo tanto ceñirse a su uso para estos valores de frecuencia. De nuevo los valores representados en el cuadro 5.2 están estimados para una tensión $V_s = 2,5V$.

Cuadro 5.2: Consumo en Modo de Baja Potencia

	Frecuencia de Salida (Hz)	Ancho de Banda (Hz)	Código del Registro	$I_{DD}(\mu A)$
	400	200	1100	90
[17]	200	100	1011	60
	100	50	1010	50
	50	25	1001	45
	25	12.5	1000	40
	12.5	6.25	0111	34

En caso de que se quiera modificar, interpretar o manipular la placa de alguna manera electrónicamente, bien por una posible avería, o bien porque se detectan comportamientos que se quieren modificar, se podrá disponer de un esquema de la distribución electrónica de la placa de evaluación del ADXL345.

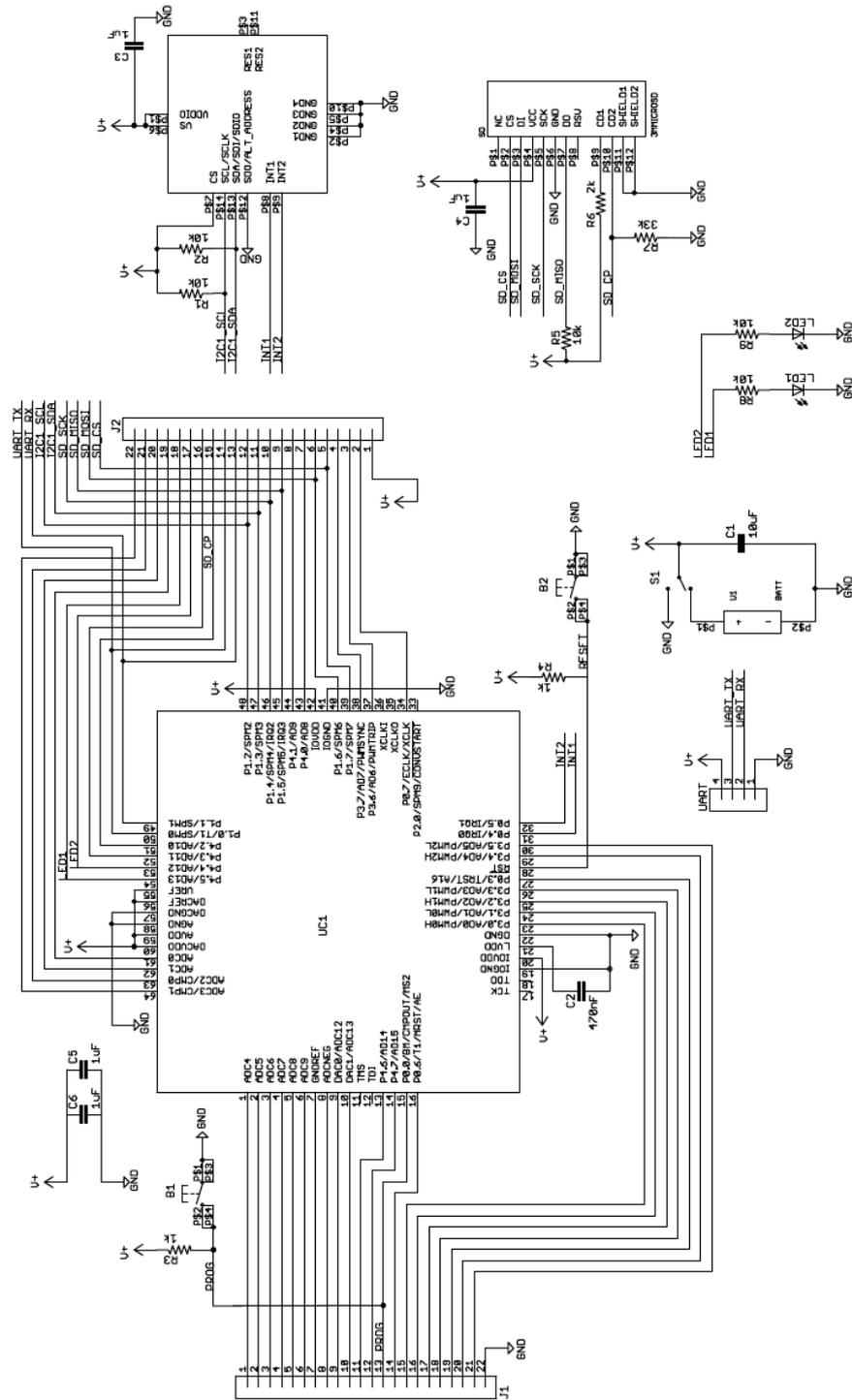


Figura 5.5: Esquema Electrónico de la Placa de Desarrollo del ADXL345 [17]

Funcionamiento

El ADXL345 es un sistema completo triaxial de medida de aceleraciones. Mide tanto aceleraciones dinámicas (movimiento e impacto), como estáticas (gravedad). El dispositivo está formado por una estructura de superficie micromecanizada de polisilicona, construida sobre una oblea de material semiconductor. La estructura se suspende sobre la oblea con un resorte de polisilicio y proporciona resistencia contra las fuerzas debidas a la aceleración aplicada.

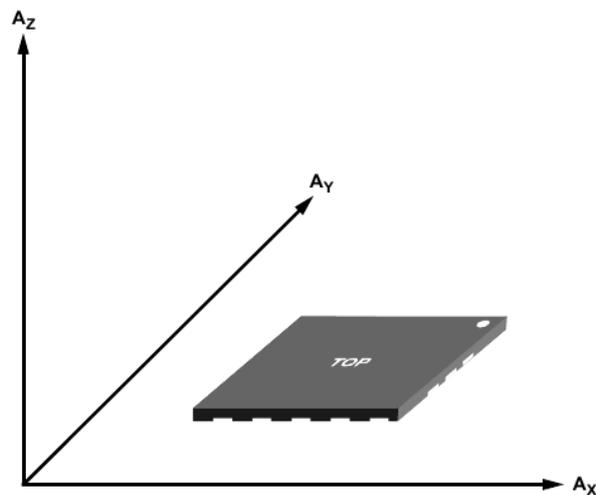


Figura 5.6: Ejes asociados a la sensibilidad del acelerómetro [17]

La desviación de la estructura se mide utilizando condensadores diferenciales que consisten en placas fijas independientes y placas unidas a la masa en movimiento. La aceleración deforma la masa de prueba y desequilibra el condensador diferencial, lo que produce una salida en el sensor, cuya amplitud es proporcional a la aceleración. A partir de esta salida se puede determinar la magnitud y polaridad de la aceleración.

Durante el uso del ADXL345 ha de tenerse en cuenta la geometría del montaje del acelerómetro en la placa de evaluación, así como la propia orientación del conjunto del dispositivo durante el ensayo.

A la hora de interpretar los posibles resultados que ofrece el acelerómetro, en caso de ser necesario, se deberá tener en cuenta las señales de salida que se registran con el dispositivo el reposo, respecto a la orientación del mismo con respecto a la dirección de la gravedad. Estas señales de salida quedan registradas en la siguiente imagen.

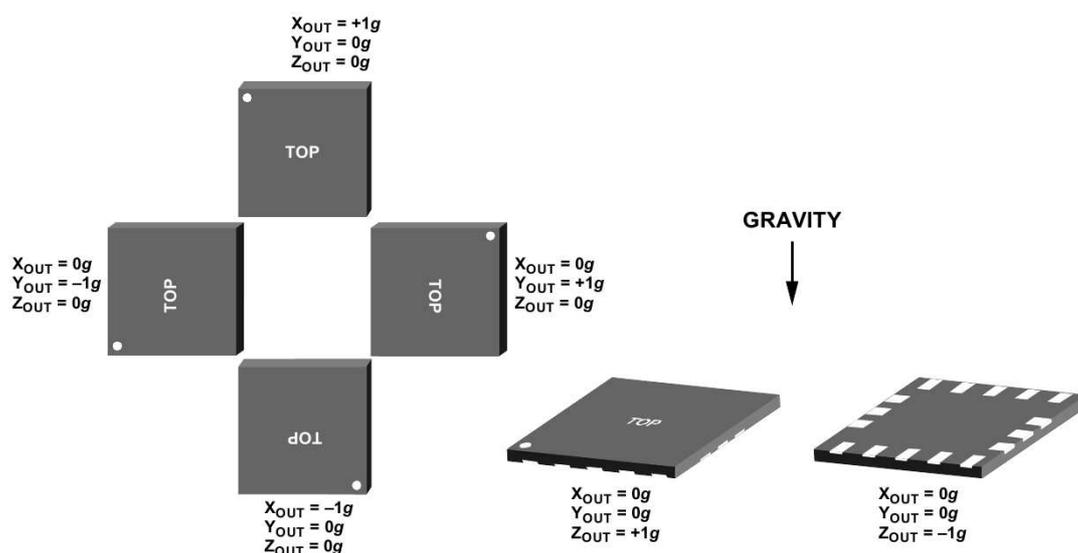


Figura 5.7: Señales de respuesta según orientación respecto a la gravedad [17]

En este caso particular no es necesario tener en cuenta las consecuencias de la gravedad, ya que esta aceleración actuará en superposición con el resto de las registradas y no supone ninguna distorsión en los resultados de las señales que se obtienen.

La escritura de las lecturas del acelerómetro en el archivo *.txt* se realiza a partir de los datos guardados en los registros de control. La lectura de estos datos de salida para ser incluidos en los registros se realizan mediante la señal de interrupción *DATA READY* que indicará cuando serán actualizados. Esta señal se conmuta a “*high*” cuando los nuevos datos están preparados en los registros. Si se desea controlar la rutina de interrupción, existe la posibilidad de hacerlo modificando valores del registro *DATA FORMAT*, aunque en este caso no se realizará ya que los valores iniciales funcionan perfectamente para el uso que se quiere hacer del ADXL345. Los datos a escribir en el archivo se leen desde los registros *DATA X0*, *DATA X1*, *DATA Y0*, *DATA Y1*, *DATA Z0*, y los registros de *DATA Z1*.

Para poder garantizar la coherencia de los datos durante la escritura, se debe realizar la recuperación de los datos desde el dispositivo, mediante lecturas *multi-byte*. El propio código inicial realiza la lectura correctamente y se decidió no actuar sobre él para evitar posibles consecuencias en otros aspectos. La organización de los bits en cada uno de los datos se refleja en la siguiente imagen.

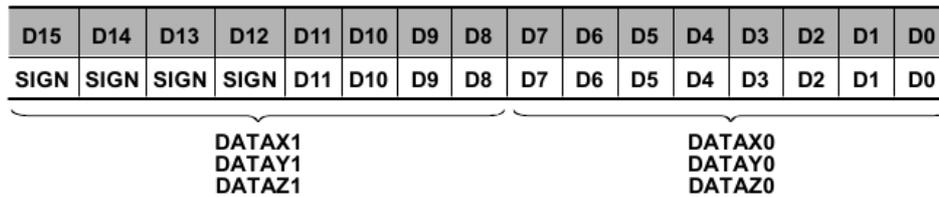


Figura 5.8: Formato de Construcción de los Datos
[17]

El formato de los datos del ADXL345 es de 16 bits. Una vez que los datos de aceleraciones son adquiridos por los registros de control de los datos, la reconstrucción de los mismos depende directamente de la configuración del dispositivo que realice el usuario. *DATAX0* es el byte inferior del registro que gestiona las aceleraciones del eje *X* y *DATAX1* es el byte superior, formando así los 16 bits usados en cada dato obtenido. Esta misma estructura se sigue para el resto de los ejes con los que trabaja el acelerómetro. Se puede comprobar el formato de salida que corresponderá a cada dato en el cuadro 5.3.

En este caso se ha utilizado el modo de 13 bits, donde los 4 bits superiores son bits de signo 5.3. Existe un modo de 10 bits, que está disponible configurando el registro *DATA FORMAT*. En este modo la equivalencia de los datos reflejados en el archivo de salida es diferente que en el modo de 13 bits, y se especifica con más detalle en el anexo A.

El ADXL345 usa un formato para los datos de complemento a dos, por lo que funcionando en modo de 13 bits, *1LSB* representa *3,9mg*.

Cuadro 5.3: Formato de Salida de los Datos del ADXL345

[17]

Código 16-bit (Hexadecimal)	Representación en Complemento a dos (Decimal)	Aceleración (<i>mg</i>)
0FFF	4095	+1599
...
0002	+2	+7.8
0001	+1	+3.9
0000	0	0
FFFF	-1	-3.9
FFFE	-2	-7.8
...
F000	-4095	-1600

Configuración del Dispositivo

El ADXL345 tiene la capacidad de ser modificado desde el usuario de forma que su toma de datos se ajuste a las características que se desean según el ensayo. El fabricante ofrece ejemplos de firmware ya desarrollados de forma que puedan ser directamente cargados sin necesidad de reprogramar el mismo, pero para el ensayo realizado se necesitan unas condiciones particulares no recogidas en los ejemplos que se ofrecían.

El firmware de la placa es completamente reprogramable y modificable. Está escrito en C y se compila para ser cargado en el microprocesador ADuC7024. Una vez cargado el archivo, el funcionamiento del acelerómetro pasará a ser el indicado en código compilado.

En el caso de este proyecto se requería un rango de $\pm 16g$ y una frecuencia de muestreo de $400Hz$.

Para conseguir que el dispositivo modifique las medidas predeterminadas por el código inicialmente cargado, debemos modificar los registros de control correspondientes en el código C a los valores que se indican en la hoja de características, realizando estos cambios en un lugar adecuado del código de manera que sean anteriores al proceso de toma de

datos y realice la escritura de los mismos de manera correcta.

Se deben modificar los registros, de nombre *DATA FORMAT* y *BW RATE*, modificando los valores que el código asocia para los bits que definen el comportamiento del dispositivo a través del microprocesador. La hoja de características contiene los datos acerca de los registros a modificar, y tras un profundo estudio del código al completo, se decidió la posición y la sintaxis que se utilizaría para realizar el cambio.

```
buffer[0] = XL345_DATA_FORMAT;  
buffer[1] = XL345_RANGE_16G | XL345_FULL_RESOLUTION;  
xl345Write(2, buffer);
```

Mediante este código se fija el rango de medida a $\pm 16g$ así como al modo de 13 bits modificado a través de la variable *XL345 FULL RESOLUTION* que realiza el cambio del bit correspondiente del registro para que adopte el modo de funcionamiento.

El uso del modo de 13 bits es el más propicio en este caso, ya que ofrece la posibilidad de que la conversión final de los resultados a unidades de aceleración no varíe según el rango de medida elegido, algo que simplifica las modificaciones a realizar en la herramienta de software, ya que el algoritmo de conversión permanecerá igual y no será necesario modificarle con cada modificación de rango, haciéndola más operativa para el usuario. Si se usará el modo de 10 bits la proporción cambiaría con el rango, y por tanto se debería modificar con mayor frecuencia el código, haciendo el proceso de representación más complejo.

Para el cambio de la frecuencia de trabajo del acelerómetro el código aporta una estructura de código predefinida para ayudar al cambio al usuario, de forma que un único cambio de un valor, produce los cambios pertinentes automáticamente para que se realice la nueva escritura en el registro y se modifique como se pretende el valor de la frecuencia.

```
#define RATE_400
```

Mediante esta línea se define la frecuencia de muestreo a 400 Hz, ejecutándose las líneas de código asociadas al valor que acabamos de introducir para definir posteriormente en el código los registros de manera que se modifiquen los bits pertinentes para conseguir la frecuencia deseada, así como el modo energético más adecuado en cada caso.

```
#define XL345_RATE XL345_RATE_400
#define SENSORBUFFERSIZE 600
#define XL345_PWR_MODE XL345_LOW_NOISE
```

La elección de esta frecuencia se debe a tanto a motivos de representatividad de los datos recogidos, como el mejor comportamiento del propio dispositivo trabajando a valores elevados de frecuencia pero por debajo de los 800 Hz. Con estas modificaciones el acelerómetro queda configurado según va a ser usado para el ensayo.

5.2.3. Condiciones del Ensayo

Tras configurar el acelerómetro de la manera deseada, resta definir las condiciones tanto ambientales como técnicas bajo las que se realizará el ensayo pretendido.

Se realizará un ensayo de *40 segundos* para cada una de las partes del cuerpo que se pretende estudiar, que es el tiempo y muestras estimadas para obtener conclusiones apropiadas con exactitud:

- Brazo
- Cintura
- Pierna

El acelerómetro se fija al cuerpo mediante una sujeción elástica que fija el dispositivo al cuerpo de manera solidaria, por lo que quedan reflejados los movimientos de la zona

donde se fija, de manera exacta, descartando posibles errores por movimiento relativo del acelerómetro sobre el cuerpo, que pudieran falsear los resultados.

Corriendo

Cada uno de estos triples ensayos se realizarán para distintas velocidades. Para garantizar la exactitud, se utilizará una cinta de correr, de manera que se pueda concretar la velocidad a la que se quiere correr y hacerlo de manera constante en todo momento. Las velocidades utilizadas son:

- 3 km/h
- 5 km/h
- 10 km/h

Se obtendrá una única señal, en la que se puede distinguir cada uno de los tres ensayos, de manera que bajo la misma representación podamos comparar los resultados concluyendo mediante posterior análisis que caso se considera más apropiado.

Escaleras

De nuevo se realizará un triple ensayo en cada una de las partes del cuerpo humano a estudiar. En este caso se utiliza una máquina de simulación de subida de escaleras, muy común en ámbitos de realización de ejercicio físico como gimnasios o centros de rehabilitación o tecnificación. Este aparato nos permitirá fijar de nuevo la cadencia de subida de escalones, de manera que podamos determinar con exactitud la señal que obtendremos en el acelerómetro. Las cadencia de subida utilizadas son:

· 33 smp

· 85 spm

Bajo estas premisas, queda definir las condiciones del ambiente en que se realizará el ensayo. El entorno para el experimento se buscó que fuera cotidiano, de manera que las simulaciones reflejaran lo más fielmente posible los movimientos más habituales que se producen en el cuerpo humano. Se utilizó una estancia cubierta adecuada para la realización de trabajo físico, en la que no existieran obstáculos o dificultades que modificaran el movimiento durante el ensayo, ya que se buscaba realizar un ensayo limpio en el que el movimiento fuera completamente libre y natural.

El sujeto será un varón de 25 años, completamente sano de 1,90 metros de estatura y 80kg de peso. Tiene motricidad normal sin ningún tipo de discapacidad o de dificultad para el movimiento. El propio sujeto realizará las tareas de encendido y apagado del dispositivo así como los cambios de posición requeridos para tomar medidas en las diferentes partes estipuladas.

Capítulo 6

Campaña de Medidas

Tal y como se han comentado las medidas se han organizado según tres parámetros: Parte del cuerpo, tipo de movimiento y Velocidad. Para conocer como realizar estas medidas, debemos primero describir como se produce la adquisición de datos por parte del acelerómetro.

La recogida de datos por la placa de evaluación del ADXL345 sigue un proceso particular que termina con la obtención de un archivo de texto (*.txt*) en el que aparecen recogidos los valores finales de aceleración obtenidos. En primer lugar hay que alimentar el dispositivo, en este caso se alimentará con dos pilas AAA. Se interta también la tarjeta SD en la ranura dispuesta en la placa. La tarjeta SD ha de estar formateada con un sistema de archivos FAT32, formato con el suelen venir la mayoría de tarjetas SD de forma predeterminada. En este momento ya se puede accionar el interruptor de encendido, comenzando la toma de datos en el momento que el indicador LED se encienda en color verde. Cuando se termine con el ensayo, se debe mover el interruptor de nuevo para apagar la placa. En ese momento ya se puede recoger la tarjeta SD con los datos dentro en forma de archivo *.txt*. Para procesar los datos debemos introducir la tarjeta en un lector, que posteriormente conectaremos en un puerto USB a un ordenador, donde

volcaremos el archivo de texto.

Una vez conocemos por completo el proceso de toma de datos, se debe definir una rutina de procedimiento para los experimentos, que garantice la futura comparación de resultados de forma exacta y evitando posibles errores por la diferenciación en los ensayos. De esta manera podremos generar una herramienta de software que realice el análisis de las señales independientemente de los parámetros bajo los que se produjo el ensayo, teniendo en cuenta solamente la rutina de tiempos y muestras bajo la que se rige cada experimento.

Primero se realizará la colocación del dispositivo en la parte del cuerpo elegida para comenzar. Se elegirá la velocidad particular para ese ensayo y se comenzará a realizar el movimiento elegido. Posteriormente se realizará el encendido del ADXL345 y se comenzará a tomar datos durante los 40 segundos estimados como representativos. A continuación, sin apagar el dispositivo, se realizará el cambio de posición de la sujeción elástica del acelerómetro a la siguiente posición corporal a estudiar, durando este cambio entre 5 y 10 segundos que producirán una rotura en la señal que nos servirá para diferenciar cada uno de los ensayos. Se continuará con el ensayo, y se repetirá el proceso hasta que los datos de los movimientos en las 3 partes del cuerpo hayan quedado registrados. Llegados a este punto se apagará el dispositivo para que la toma de datos finalice.

En este instante, ya se dispondrá de un archivo *.txt* dentro de la tarjeta SD. Una vez transferido a un ordenador, comprobaremos la estructura en la que se han grabado los datos en el archivo, siendo un ejemplo de representación el siguiente:

ndx	HR:MN:SC:1/128s	x	y	z
0	00:00:00:118	78	17	273
1	00:00:00:119	83	14	311
2	00:00:00:119	80	19	309
3	00:00:00:119	80	20	308
4	00:00:00:120	78	17	303
5	00:00:00:120	82	17	308

Siendo,

ndx: número de muestra dentro de los límites del ancho de banda.

HR:MN:SC:1/128s: Tiempo transcurrido en el momento de realizar el muestreo.

x,y,z: Eje del acelerómetro sobre el cuál se han recogido los datos.

En nuestro caso los valores interesantes a nivel de análisis serán exclusivamente aquellos que hacen referencia a la aceleración producida sobre los ejes coordenados. Sobre estos datos se debe actuar procesándoles de manera que podamos realizar un estudio más exhaustivo de sus características. Esta necesidad de profundizar en el estudio de los datos en crudo que se reciben del acelerómetro se traduce en la creación de una herramienta de software que permita automatizar, dentro de las condiciones especificadas, el procesado de los datos y su posterior representación para su lectura e interpretación. El uso de esta herramienta no se restringiría exclusivamente a este proyecto, sino que podría ampliarse a cualquier caso en que las condiciones coincidan con las expresadas, o realizando una sencilla modificación, adaptar la propia herramienta a otras condiciones similares, permitiendo una profundidad de estudio más elevada.

Capítulo 7

Herramienta desarrollada

El principal propósito de esta herramienta tal y como se ha introducido previamente, es proporcionar una forma de realizar el tratamiento de los datos almacenados en la tarjeta SD. En este caso es apropiado el desarrollo de una única función que englobe todo el proceso de representación de los datos, de manera que quede simplificado en gran medida y tras ejecutar el nuevo software, un único paso sea necesario para trasladar la información al ordenador, realizar los ajustes pertinentes, y finalmente realizar su representación. Para ello se tratará de aprovechar al máximo la gran variedad de herramientas de representación que nos ofrece la plataforma *MATLAB*.

Las representaciones gráficas que se incluirán consistirán en tres gráficos distintos, realizados para cada uno de los ejes registrados, para cada uno de los ensayos realizados.

Señal Completa. Se representa la señal recogida por el acelerómetro al completo, incluyendo en un mismo gráfico la que se obtiene para cada una de las partes del cuerpo a considerar, de forma que su comparación sera directa y fácil.

Diagrama de Dispersión de los picos. Se busca que aparezcan únicamente los picos de señal obtenidos, de forma que podamos identificar los puntos en los que la

dirección de la aceleración cambia de sentido. Mediante este diagrama podemos conocer los valores máximos (tanto positivos como negativos) de la señal, así como estimar el valor medio que tendrán a lo largo de la función estos puntos interesantes para el caso que se estudia. Se podrán utilizar para concluir en que puntos se producirán y con que intensidad los mayores cambios de sentido de la aceleración y por tanto velocidad, extrapolarlo los resultados al caso de un generador lineal en la misma situación y condiciones.

Función de Densidad de Probabilidad y Diagrama de Cajas. La *función de densidad de probabilidad* de una variable aleatoria continua es una función, usualmente denominada $f(x)$ que describe la densidad de la probabilidad en cada punto del espacio de tal manera que la probabilidad de que la variable aleatoria tome un valor dentro de un determinado conjunto sea la integral de la función de densidad sobre dicho conjunto.

Siendo X una variable aleatoria real, la función de densidad de probabilidad es una función tal que:

$$P[a \leq X \leq b] = \int_b^a f(x)dx \quad (7.1)$$

De modo que si F es la función de distribución de X , entonces:

$$F(x) = \int_{-\infty}^x f(u)du \quad (7.2)$$

Siendo por tanto:

$$f(x) = \frac{d}{dx}F(x) \quad (7.3)$$

De esta manera se puede conocer en cada caso la probabilidad de que dentro del ensayo

se consiga un valor particular, estimando a su vez cuales son los valores más repetidos, pudiendo conocer cual es el suceso más común. Es este caso se realizará una aproximación a la distribución *normal*.

El *Diagrama de Cajas* es un gráfico, basado en cuartiles, mediante el cual se visualiza un conjunto de datos. Está compuesto por un rectángulo, la "caja", y dos brazos, los "bigotes". Suministra información sobre los valores mínimo y máximo, los cuartiles Q1, Q2 o mediana y Q3, y sobre la existencia de valores atípicos. De la misma manera, nos permite conocer la simetría de la distribución, indicándonos la estabilidad de los valores obtenidos y poder ver si muchos de ellos no son representativos en el conjunto.

Conocidos los gráficos a usar para la interpretación de datos, es necesario definir una estructura para construir el código.

En primer lugar se realizará la adquisición del archivo de datos y su almacenado para su posterior uso a lo largo de la función.

Seguidamente es necesario continuar con la función para realizar los cálculos que conviertan los datos y poder trabajar con valores de aceleración en "G" así como la aplicación de herramientas matemáticas para interpretar de la mejor manera posible el comportamiento durante el ensayo.

Por último, terminaremos la función realizando la representación gráfica de los resultados obtenidos en la segunda parte de la función. Esta representación tratará de aportar la mayor cantidad de información en un solo gráfico para que sea accesible de un solo vistazo y la interpretación y comparación del mismo se simplifique y al mismo tiempo sea más clara. Se realizarán gráficos para cada uno de los ensayos, particularizando también los resultados para cada uno de los ejes sobre los que se recogen datos.

Podemos dividir la función desarrollada en tres partes diferenciadas:

- Adquisición de Datos
- Procesado
- Representación Gráfica

A continuación se explicará el funcionamiento de las partes que integran la función; y en el apartado de anexos, se dispondrá del código completo de la misma.

7.1. Adquisición de Datos

Tal y como están dispuestos los datos en el archivo *.txt*, la forma más apropiada de almacenado de los mismos es en una matriz de tantas columnas como ejes trabaja el dispositivo (3 en este caso), siendo el número de filas el total de muestras tomadas por el acelerómetro durante el ensayo, desechando el resto de datos por no tener influencia en este proyecto. Usando una matriz se facilitan mucho las operaciones posteriores por ser un entorno común y ampliamente conocido.

Sirviendo como ejemplo el caso anterior de muestra del formato del archivo *.txt*, la matriz que obtendríamos tendría 3 columnas, al ser un acelerómetro triaxial, y 6 filas, que hacen referencia a las 6 muestras que aparecen representadas. (matriz 6×3)

Existen dos posibilidades según se quiera indicar en código el archivo *.txt* que se desea analizar, o bien dejar a elección del usuario por pantalla una vez ejecutada la función la elección del archivo.

En el primer caso los comandos a ejecutar serán:

```
A=importdata('./resul/carrera_01.txt',' ',' ',1);
```

En el segundo caso se debera ejecutar:

```
[filename, pathname]=uigetfile('*.txt','Datos a Cargar');
archivo=strcat(pathname,filename);
A=importdata(archivo, ',', 1);
```

Se debe seleccionar únicamente los datos utilizables en este caso, desechando tanto la cabecera utilizada durante la grabación de datos por el acelerómetro, como los datos de muestras y tiempos, para ello utilizamos una función que nos permitirá centrarnos en los datos numéricos y almacenar sólo los datos deseados,

```
A=A.data;
```

La función "*importdata*" obtiene los datos a partir del archivo *.txt* con formato ASCII, por lo que para trabajar con ellos en el espacio de trabajo de MATLAB se debe llamar a la función auxiliar *.data* para que almacene en la matriz elegida exclusivamente el valor numérico de cada uno de los caracteres en formato ASCII.

Como resultado obtenemos la matriz buscada, con cada uno de los datos recogidos para cada uno de los ejes y diferenciados por columnas.

7.2. Procesado

Estos valores tal y como se recogen del archivo no nos sirven para sacar ninguna conclusión, ya que aparecen en valores LSB, pero tal y como se indicó anteriormente, la equivalencia directa de 3.9 mg/LSB hace los valores proporcionales y fácilmente expresables en "*G*".

A fin de facilitar la representación gráfica posterior, se plantea un bucle que repita el proceso desarrollado en la función para cada una de las columnas (ejes) de la matriz obtenida. En este caso para resumir y evitar repeticiones innecesarias se centrará la atención en la primera columna. Se recoge en un vector al que llamaremos *x*, en relación al

eje al que representa, la primera columna de la matriz de datos A . Una vez tenemos separados los valores exclusivos para el eje correspondiente sólo queda realizar la conversión proporcional mediante una sencilla operación de proporcionalidad, generalizando para cada uno de los valores.

```
x=A(:,1);  
x=(x*3.9/1000);
```

Se obtiene así un vector con los valores recogidos anteriormente pero convertidos a G .

Para realizar el diagrama de dispersión de los picos de señal necesitamos primero localizar los valores que suponen un pico en el conjunto de la señal. Para ello realizamos un estudio de los valores y almacenamos en un vector tanto los valores de dichos picos, como las muestras en las que se encontraron.

```
for j=c:d  
  
    if (x(j)>x(j-1)) && (x(j)>x(j+1)) && x(j)>desvmedmas  
        picos10=[picos10;x(j)];  
        vector10=[vector10;j];  
    end  
  
    if (x(j)<x(j-1)) && (x(j)<x(j+1)) && x(j)<desvmedmenos  
        picosneg10=[picosneg10;x(j)];  
        vectorneg10=[vectorneg10;j];  
    end  
  
end
```

Siendo c y d el número de las muestras elegidas como inicio y final del ensayo particular para cada una de las partes del cuerpo.

Se representa también el valor medio de todos estos picos, que sirve como punto de comparación más exacto entre cada uno de los casos particulares. Para ellos se usan funciones intrínsecas de *MatLab*.

```
mediapicospos=mean(picos10);  
mediapicosneg=mean(picosneg10);
```

Para la función de densidad, no es necesario tratamiento previo de los datos, y se usa directamente el vector columna obtenido con cada una de las muestras obtenidas.

7.3. Representación Gráfica

La gráfica de la señal completa resultará de una simple representación del vector con los resultados de las muestras, sobre un vector formado con el número total de muestras que se han usado en el ensayo, o un intervalo elegido por el usuario.

```
x1=x(c:d);  
t=c:d;
```

Siendo c y d el número de las muestras elegidas como inicio y final de la señal.

Dibujamos la señal

```
plot(t,x1)
```

En el segundo caso del diagrama de dispersión, se realiza una representación simple de los vectores que se consiguen tras el proceso de obtención de los picos. Mediante la función *scatter*, se automatiza el dibujo del diagrama. Para que quede mejor definido, se particulariza cada diagrama para las muestras que corresponden a la parte del ensayo de cada una de las partes del cuerpo. Estas muestras las decide el usuario tras una inspección previa de la señal completa. Por tanto se realizan 3 gráficos que compartirán el mismo eje de forma que se visualicen juntos. En código se realizará un bucle que recorra valores para las 3 zonas del cuerpo estudiadas, dentro del bucle que recorre cada columna de

la matriz con los datos de los ejes, representando los datos en tres momentos diferentes según se definan los límites con las muestras correspondientes. Los picos representados, serán aquellos que superen el valor de la desviación típica de toda la distribución de los datos, ya que se consideran aquellos menores de dicho valor como no representativos en la conclusión posterior.

```
hold on
scatter(vector10,picos10,3)
```

Si se desea representar los valores medio de las dispersiones, se deberá añadir código para ello

```
hold on
plot(t1,mediapicospos,'k')
plot(t1,mediapicosneg,'k')
```

Se debe recurrir a una función auxiliar en el caso la función de densidad de probabilidad, ya que de otra manera se complicaría mucho el código principal. Esta función auxiliar recibe el nombre de *fdp.m*. Recibe como *input* el vector columna *x1* con los valores del ensayo y una constante *p* que se refiere a cada uno de los procesos particulares para cada parte del cuerpo estudiada con valores entre 1 y 3, diferenciando cada caso con un color distinto. Se devolverá la representación de la función de densidad de probabilidad sobre los ejes indicados, y superpuesta con el resto de casos, para compararlos claramente. Realizaremos la llamada a la función:

```
ejex=x1;
h_(p)=fdp(ejex,p);
```

El cuerpo de la función $fdp(ejex,p)$ se encuentra representada en el Anexo C.

Posteriormente para añadir los Diagramas de cajas, se crean diferentes ejes en los que representar el gráfico y se ajustan posteriormente al gráfico. *MatLab* tiene una función

interna denominada *boxplot* que dibuja de forma automática el diagrama para los datos que contenga el vector que se indique.

```
ax1=gca;
hold on
set(ax1,'Box','off','XLim',[-6 10])
pause(0.1)

ej1=axes('Position',get(ax1,'Position'),'Color','none');
ej2=axes('Position',get(ax1,'Position'),'Color','none');
ej3=axes('Position',get(ax1,'Position'),'Color','none');

h1=boxplot(ej1,x,'widths',0.6);
h2=boxplot(ej2,x1,'widths',0.6);
h3=boxplot(ej3,x2,'widths',0.6);
```

Identificamos el eje principal donde se encuentran representados las funciones de densidad de probabilidad

Una vez se tiene la representación completa se realiza el proceso de etiquetado y adaptación del espacio de trabajo. El código completo de la función de adjunta como anexo al final del documento.

Capítulo 8

Resultados

Los gráficos obtenidos, permiten caracterizar detalladamente las señales obtenidas. Se utilizarán para comparar cada uno de los movimientos estudiados y en los casos que se estipularon, con profundidad, evitando posibles consideraciones que con una inspección básica de la señal completa llevaran a error, de forma que se puedan adaptar a las aplicaciones contempladas en este caso.

En cada uno de las gráficas que se adjuntan, y que tienen leyenda que permita identificar los resultados, se ha anotado un número que hace referencia a la parte del cuerpo donde se realizó la parte del ensayo que se asocia con el número.

1. Brazo
2. Cintura
3. Pierna

Tal y como indican los gráficos en particular, cada una de las columnas de la representación en conjunto, representa un eje del espacio tal y como se midió al realizar el

experimento con el acelerómetro, coincidiendo con el código de colores que se observa en la representación, correspondiendo el color azul, rojo y verde, con el eje ' x ', ' y ' y ' z ' respectivamente. En el caso de la función de densidad de probabilidad, el código de colores difiere del anterior, estando indicado en la leyenda del propio gráfico la correspondencia. Se ha realizado un etiquetado de los diagramas de cajas , que coinciden con la asociación que se realizó en las gráficas superiores de número y parte del cuerpo.

A través del estudio se todos los gráficos que se adjuntan con los resultados, se pueden sacar conclusiones específicas con representaciones detalladas, evitando posibles confusiones originadas por inspecciones preliminares de la figura de la señal sin tratamiento posterior.

8.1. CARRERA a 3 km/h

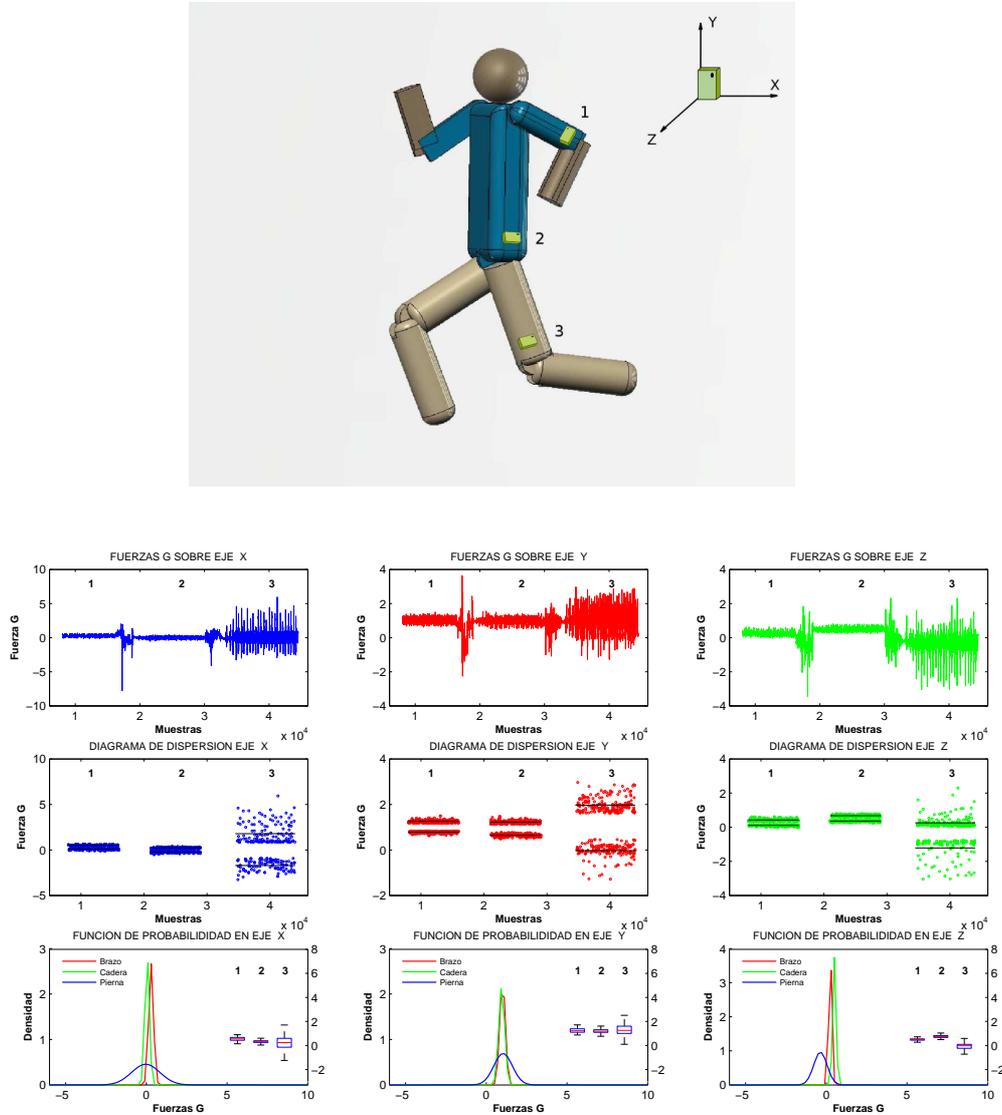


Figura 8.1: Resultados para Carrera a 3 km/h

En una primera aproximación al gráfico de resultados, observamos como en la señal de aceleraciones que obtenemos es clamorosa la diferencia de amplitud en los datos que se representan. El tramo de señal correspondiente a la pierna deja ver picos mucho mas grandes de fuerzas G generadas, mientras que los otros dos planos (brazo y cintura) no muestran una diferencia sustancial en sus magnitudes. Sólo en la gráfica del eje Z se llega

a distinguir una pequeña diferencia en el valor medio, siendo mas elevada para el segundo tramo (cintura), pero sin llegar a algo destacable.

Si se pasa a observar los diagramas de picos representados, se encuentran diferencias más grandes de las que poder obtener un argumento. En los tres ejes coordenados observamos como los picos de aceleración que se producen tienen valores similares, aunque es la posición del cuerpo donde encontramos más cambios en los gráficos. Destaca especialmente sobre las otras dos opciones, el gráfico de dispersión de los resultados obtenidos en la pierna. Los valores de pico, no sólo sugieren una mayor amplitud de las aceleraciones, sino que reflejan una suficiente simetría en los valores superiores e inferiores, que invitan a pensar en unos resultados constantes. Los valores medios de los picos superiores e inferiores, destacan significativamente sobre los del resto de distribuciones de datos, por lo que es claro el mayor interés que tiene la posición de la pierna si consideramos tanto la magnitud de los valores, como su continuidad durante el movimiento realizado. De cualquier manera, será necesario consultar esta condición con las funciones de densidad de probabilidad y los diagramas de cajas para confirmar las conclusiones que se perciben.

Las campanas obtenidas como resultados para los ejes X, Y , no dejan lugar a dudas sobre las características de cada distribución. Los gráficos para las posiciones del cuerpo más altas (brazo y cintura), devuelven una imagen delgada y elevada, lo que sugiere la repetición de los mismos valores durante todo el ensayo, pero de valores en G poco elevados. Por el contrario, aunque la función de los resultados en la pierna, tiene una repetición menos marcada, se obtienen valores más grandes de aceleración, siendo mucho más apropiado en el caso que se estudia. En todos los casos existe una simetría bastante marcada en el gráfico, lo que certifica su condición de ensayos constantes que a pesar de sus diferencias, indica que todos nos dan resultados considerablemente fiables.

En el caso del eje Z , los gráficos de densidad presentan resultados algo menos marcados, aunque reflejando también con claridad el resultado más recomendable. De nuevo los resultados en brazo y cadera repiten la misma tónica de delgadez y valor pequeño,

aunque el gráfico de la pierna refleja mayor cantidad de aceleraciones negativas que en otras direcciones del espacio. Esta situación en este caso particular no supone un problema ya que al generador lineal que se pretende utilizar sólo le atañen los movimientos en una dirección, pero no distinguen entre el sentido positivo y negativo. En cualquier caso, es una característica que cabe señalar.

Por último, inspeccionando los diagramas de cajas, en todos los casos encontramos gráficos muy simétricos que proporcionan confianza en los resultados de los ensayos y confirman la diferencia entre ensayos, comprobándose rápidamente en una primera inspección todo lo concluido anteriormente y confirmándose a través de los valores límite y la distribución simétrica de los mismos y de los diferentes cuartiles.

8.2. CARRERA a 5 km/h

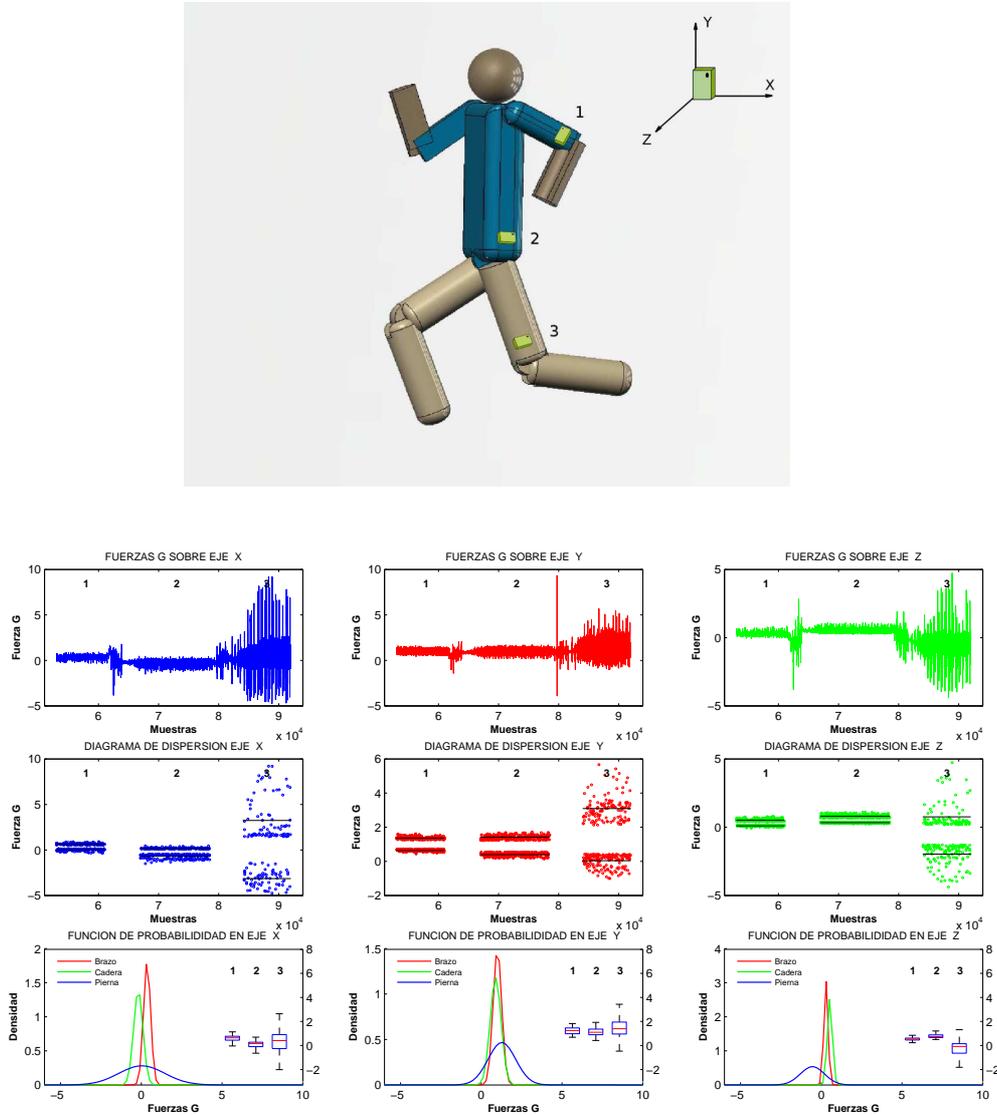


Figura 8.2: Resultados para Carrera a 5 km/h

El siguiente caso ensayado es el de un movimiento en carrera constante a 5 km/h. La señal completa deja ver claramente la diferencia en los valores de la aceleración según la posición. Destaca ampliamente los resultados obtenidos en la pierna, quedando en segundo plano los conseguidos en la cintura y el brazo. De nuevo en el eje Z se nota una pequeña diferencia en el valor medio de las señales, siendo el de la señal registrada

para la cintura algo superior a las demás. En cualquier caso no supone una diferencia suficientemente significativa como para sacar conclusión alguna de dicha diferencia.

Los diagramas de dispersión de los picos de las señales obtenidas, confirman en gran medida las primeras impresiones de las señales completas. Se observa muy claramente como los picos de aceleración durante el ensayo en la pierna, superan ampliamente los valores que se obtienen para las otras dos posiciones. La simetría obtenida es razonable, por lo que sin lugar a dudas, se puede razonar que la posición más favorable para el movimiento será esta extremidad. Si particularizamos la comparación a los ejes del espacio, destaca el eje Y como el que registra picos de menor valor, apenas registrando datos cerca de las $5G$, e incluso menores en la parte negativa de los resultados, contra los valores obtenidos en los ejes X, Z , que superan en ocasiones este valor (En el caso del eje X), y se nota una mayor concentración de puntos en valores más elevados que para el eje Y .

En la comparación entre los ejes X e Z , podemos en una primera ocasión pensar que los picos que se recogen en el eje X , al tener valores máximos mayores que en el eje Z , reflejan una mayor aceleración durante el ensayo, pero si se analiza con mayor detenimiento, se observa como, aun teniendo valores extremos de menor magnitud, la densidad de estos valores es mucho mayor que la que se registra para el eje Z , por lo que se concluye que aunque de menor valor, el eje Z posee mayor posibilidad de que se produzca una aceleración elevada durante el desarrollo del movimiento. En cualquier caso, es necesario confirmar estas observaciones con el análisis de las funciones de densidad de probabilidad y los diagramas de caja.

Se repite de nuevo en los gráficos de densidad el patrón observado en los resultados a 3 km/h, siendo de nuevo destacable la forma de la campana para los ensayos en la pierna sobre las representadas para los ensayos en brazo y cintura, reflejándose valores mas elevados de aceleración y con más probabilidad de suceder, siendo por lo tanto una situación mucho más deseable.

Por lo tanto se ha de centrar la atención en la comparación entre las diferentes direcciones del espacio. En este caso merece la pena prestar atención a la escala de los resultados, comprobando como tanto el eje Y como el Z , tienen los máximos de la campana de la función de densidad cerca del 0,5, estando el correspondiente al eje X sobre el 0,25, aproximadamente la mitad que en los otros dos casos.

La simetría de los diagramas de cajas en todos los ensayos es clara, pero se distingue una mejor distribución para el caso del ensayo en pierna en el eje X . Los valores extremos son más considerables y se ajusta mejor su simetría al valor "0", consiguiendo una densidad de probabilidad aceptable en un espectro bastante amplio de valores de aceleración, tal y como se ve en la campana de su función de densidad de probabilidad, achatada, pero amplia en la base. Esta característica lleva a pensar en la dirección del eje X como la más apropiada para la obtención de una mejor distribución de aceleraciones, y la pierna como la posición predilecta dentro del cuerpo humano.

8.3. CARRERA a 10 km/h

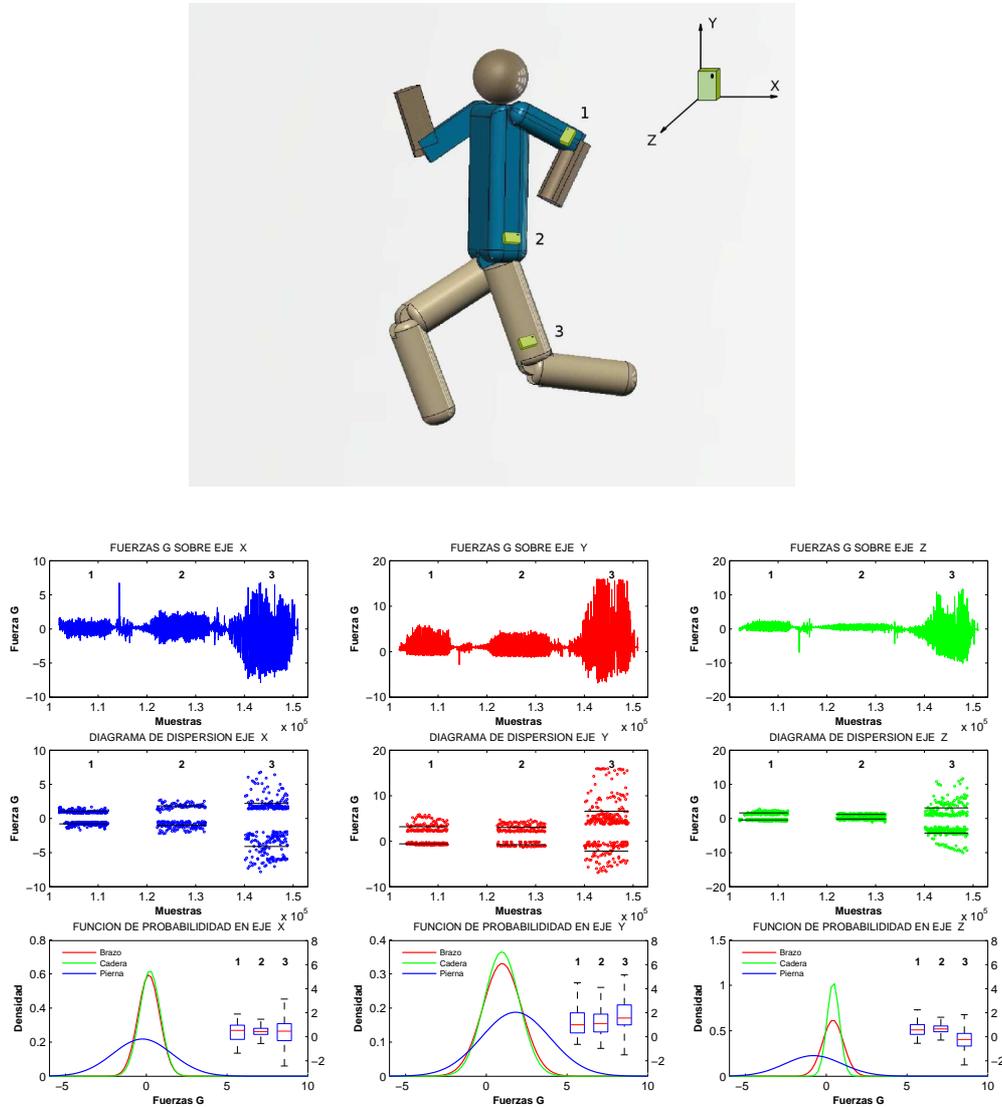


Figura 8.3: Resultados para Carrera a 10 km/h

En un primer examen de la señal completa del ensayo a 10 km/h en carrera, observamos de nuevo al igual que en los casos anteriores como destaca la señal obtenida en la pierna del sujeto que realiza el ensayo, en contraposición a las otras dos posiciones de brazo y cintura. En cualquier caso, merece la pena destacar cómo a simple vista las señales recogidas en este ensayo a la velocidad de 10 km/h reflejan una obtención de valores de una considerable mayor magnitud a los obtenidos en los estudios anteriores.

Para confirmar las primeras sensaciones, debemos contrastarlas con la información que nos ofrece las dispersiones representadas con los picos de aceleración adquiridos. De nuevo, tal y como se esperaba, destacan sobre el resto los picos que se registran para los ensayos realizados en la pierna del sujeto. La continuidad de estos resultados, lleva a concluir con fiabilidad que es la pierna la parte del cuerpo humano donde el movimiento de carrera registra mayores valores extremos y mantiene constante un valor considerable de aceleración a lo largo del ensayo, tal y como refleja la buena concentración de picos que contiene el gráfico.

En cualquier caso, son notables también los resultados obtenidos para los ensayos realizados en brazo y cintura, tal y como se comentó al inspeccionar la señal completa. Las distribuciones mantienen una concentración elevada de picos en valores representativos de aceleración a lo largo de todo el ensayo, siendo bastante superior a las obtenidas en las mismas posiciones bajo otras condiciones de velocidad menores. Para llegar a conclusiones más precisas sobre las aceleraciones que serán posibles registrar en este caso y sus posibles aplicaciones en estas condiciones, serían necesarios ensayos más particulares.

Las funciones de densidad de probabilidad en este caso ofrecen lecturas claras y muy parecidas de nuevo para los ejes X e Z , dejando clara una mayor concentración de aceleraciones a valores bajos en los ensayos de brazo y cintura, y reflejando para el ensayo en la pierna menor concentración en las aceleraciones obtenidas, pero consiguiendo valores extremos de mayor tamaño. Los diagramas de caja confirman lo planteado en ambos ejes, comprobando como su simetría sugiere una obtención de valores constantes, pero estudiando los "bigotes" y cuartiles de los mismos, se refleja un mayor valor para los valores extremos de los ensayos en la pierna. Merece la pena destacar también como los diagramas permiten distinguir con mayor claridad que en otros gráficos como los ensayos en el brazo ofrecen mayores valores extremos a lo largo de su distribución, pero aun siendo algo menores, los obtenidos en la cintura gozan de una concentración muy alta.

Es en este caso en el eje Y donde las diferencias se hacen menos notables en los

resultados obtenidos. Los diagramas de cajas tienen una similitud grande entre ellos, no pudiendo concluir nada concreto de ellos. Lo único destacable es la posible falta de simetría en el diagrama del ensayo en el brazo, que sugiere una bajas aceleraciones durante una parte del movimiento, que podrían no ser deseables según la aplicación que se considere. Existe una pequeña falta de simetría también en el ensayo de cintura, aunque menos destacable, siendo el ensayo en la pierna el que mantiene una simetría de garantías.

De nuevo la función de densidad revela como el ensayo para la pierna sufre aceleraciones extremas más elevadas, pero en este caso particular, las campanas obtenidas en los ensayos de brazo y cintura tienen formas que plantean una posible duda acerca de la distribución más deseable. Según la aplicación, sería preferible obtener aceleraciones lo más elevadas posibles, sacrificando una mayor constancia en los resultados obtenidos, o al contrario, puede que sea ventajoso obtener unos valores extremos más pequeños, pero garantizar la adquisición de unas aceleraciones fijas aun siendo de valores menores.

8.4. ESCALERAS A 33 spm

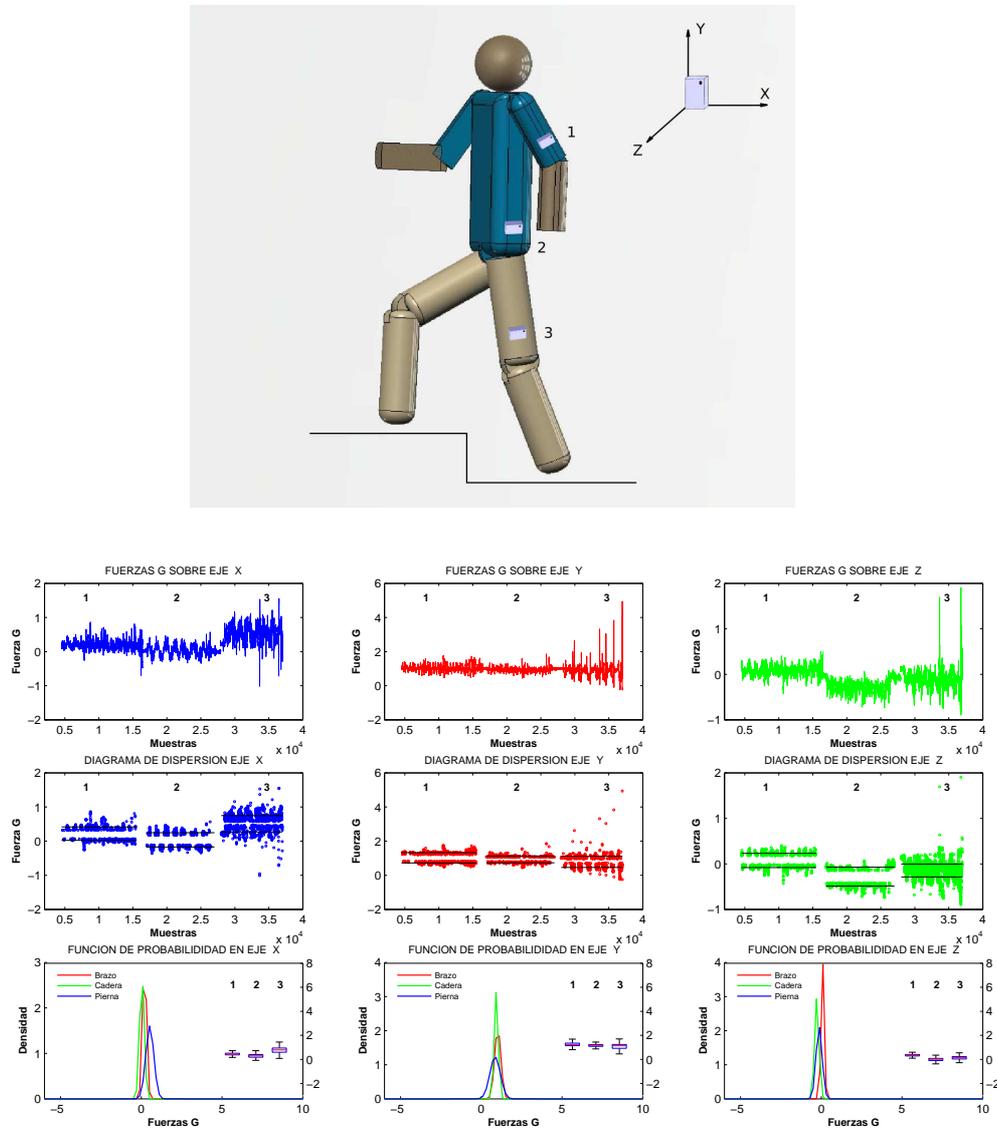


Figura 8.4: Resultados para trabajo de subida de escaleras a 33 spm

En el cotejo de las señales obtenidas en los distintos ensayos, se advierten algunas variantes significativas.

La señal obtenida de los movimientos de trabajo en escaleras presenta notables diferencias respecto a la conseguida para los trabajos de carrera. Se percibe nítidamente cómo los patrones que siguen los datos son distintos: quedan mejor definidos los ciclos

de aceleraciones que se producen durante el movimiento en escaleras que los obtenidos en el movimiento de carrera. Y, sin embargo, en ambos casos se aprecia que los valores de las aceleraciones obtenidas a la velocidad de 33 spm son muy bajos, incluso en el ensayo realizado sobre la pierna, extremidad que habitualmente proporciona valores de más elevados en otros casos.

La explicación de resultados tan bajos reside en que también la velocidad del movimiento es muy baja, y en esta característica podría asemejarse al movimiento que realizaría una persona de avanzada edad o con dificultades para realizar el movimiento correctamente. Y de este paralelismo se extrae deducciones de gran interés: Aunque los ensayos realizados a velocidades bajas, tanto para carrera como en escaleras, puedan parecer poco representativos, proporcionan datos muy valiosos, pues nos permite extrapolar los resultados al caso de personas de edad avanzada cuyos movimientos son más lentos de lo habitual, y su estudio, igualmente importante, sería difícil de realizar desde ensayos específicos con personas mayores.

Observando los diagramas de picos, se confirman las impresiones iniciales sobre la baja magnitud de los valores de aceleración, pero atendiendo a otras constantes, como la parte del cuerpo estudiada y las direcciones del espacio, los resultados que se obtienen permiten extraer también conclusiones singulares. De nuevo, al igual que en el ensayo de carrera continua, se observa que los resultados de los ensayos realizados en la pierna ofrecen valores más altos que el resto: Los picos obtenidos son mucho más pronunciados, y visualmente la densidad de los resultados es suficiente para asumir su representatividad.

En el caso de brazo y cintura, no se observan diferencias significativas entre resultados que puedan ser altamente destacables, aunque cabría señalar que para el eje X e Y , parece algo mayor en cuanto a valor absoluto de los picos, los resultados obtenidos para el brazo, algo perfectamente posible, ya que intuitivamente se puede pensar que el movimiento espontáneo del brazo durante el trabajo en escaleras es más brusco que en la cintura. Esta divergencia, sin embargo, no se confirma en el caso del eje Z ya que los resultados

muestran que el valor medio de las aceleraciones conseguidas en la cintura es mayor que en el brazo, e incluso parece que mayores que en la pierna (aunque para afirmar este extremo haría falta observar los resultados de densidad).

Centrando la atención en los diagramas de densidad de probabilidad, para el caso de los ejes X e Y , se observa claramente como, aunque ligeramente, la amplitud de la campana para el caso del ensayo en la pierna, es mayor que en los otros dos casos, concluyendo por tanto como el uso de esta parte del cuerpo para la colocación de un generador lineal se antoja más ventajoso. En cualquier caso cabe destacar que harían falta estudios posteriores para cuantificar en que grado sería ventajoso sobre el resto de opciones.

Más dudoso es el caso del eje Z . los diagramas de densidad reflejan una gran similitud entre los valores obtenidos en los distintos ensayos: las diferencias de valoración entre cintura, brazo y pierna son insignificantes. Sus resultados son simétricos y de valores tan próximos que ni siquiera recurriendo a los diagramas de cajas se puede llegar a una conclusión definitiva que lleve a decidir uno como preferente sobre los otros. Ambos son simétricos y de valores muy similares, pudiéndose llegar a pensar en la pierna como la situación de mayor aprovechamiento por sus valores extremos ligeramente mayores, aunque por contra su densidad de probabilidad en los valores de mayor magnitud es algo menor que para el caso de la cadera. Se necesitaría análisis más profundos para revelar una conclusión clara. En este caso se considerarán casos prácticamente similares.

Los diagramas de cajas de los casos anteriores de ejes X e Y , confirman las conclusiones obtenidas con antelación. Todos ellos guardan una simetría que les hace fiables y equilibrados en cuanto a los datos obtenidos. Los valores extremos en ambos casos, en los casos de ensayo en la pierna, son ligeramente mayores que los obtenidos en los ensayos de brazo y cadera. Esto concluye de nuevo la preferencia de la pierna para el uso deseado en estas dos direcciones del espacio.

8.5. ESCALERAS A 86 spm

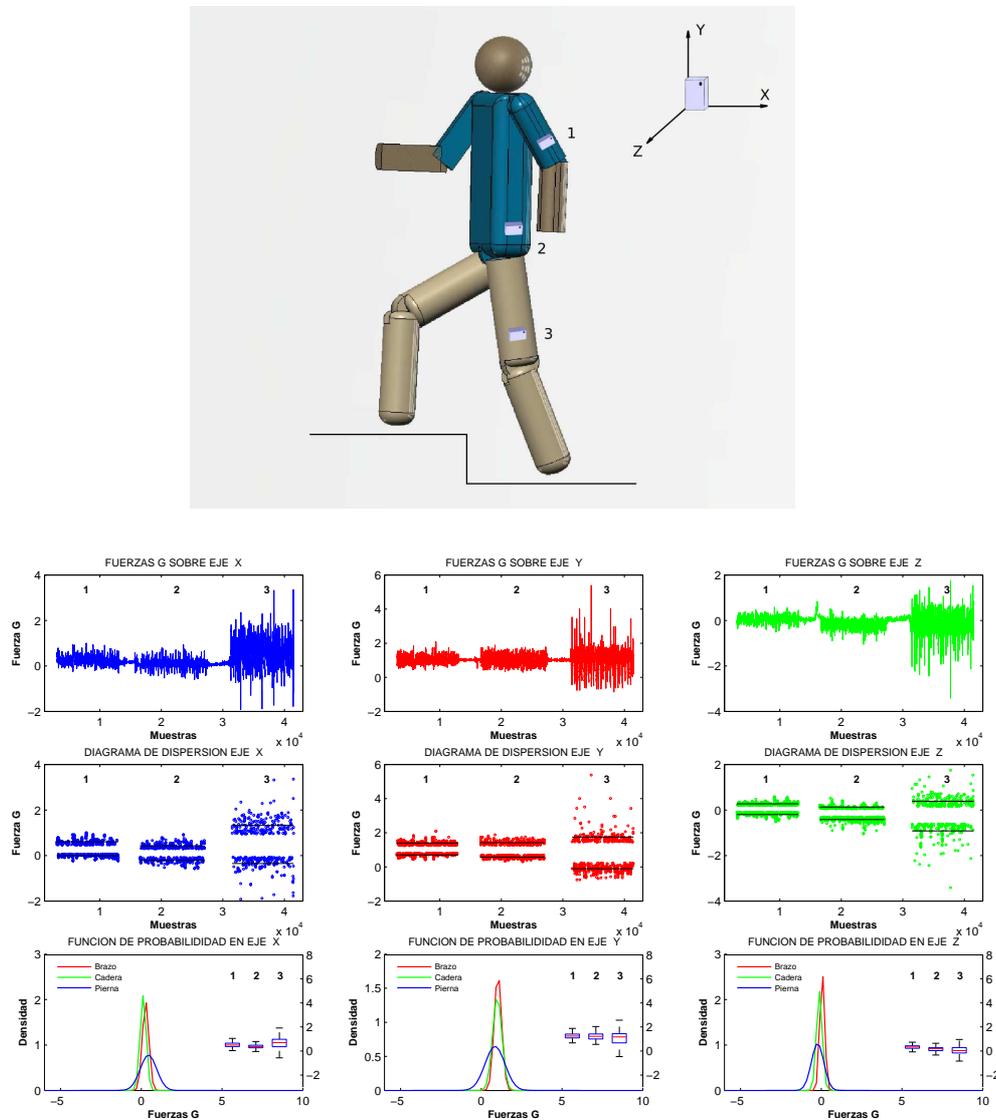


Figura 8.5: Resultados para trabajo de subida de escaleras a 86 spm

Una primera verificación muestra que la señal completa de los ensayos se asemeja en mayor medida a los ensayos realizados previamente durante carrera, lo que no debe presuponer que, tras analizar los datos, éstos revelen también tal similitud entre ellos. Observando los ejes, se comprueba que los valores son de mayor magnitud que los recogidos en el ensayo a 33spm, si bien estas diferencias cuantitativas no son muy elevadas ni particularmente significativas.

En ambos casos de ensayo en escalera, se comprueba como las aceleraciones obtenidas son mayores que en los casos más lentos de carrera a 3 y 5 km/h en todas las direcciones, no siendo así para carrera a 10 km/h , donde los resultados son mayores que durante el trabajo en escaleras.

La observación de los diagramas de picos, deja más claro las aplicaciones iniciales. Se comprueba como la mayor magnitud de los valores obtenidos en este ensayo a mayor velocidad de subida de escalones inapreciables y no significativas, aunque si queda patente que las aceleraciones producidas son mayores que en el caso anterior. Es en los ensayos sobre la pierna, donde se pueden distinguir diferencias más claras entre los valores extremos de ambos ensayos, siendo clara la superioridad de las aceleraciones obtenidas para el caso de 86 spm en la pierna sobre las obtenidas para 33 spm para la misma parte del cuerpo. En este caso las diferencias entre los resultados para cada una de las direcciones del espacio no son remarcables, y será simplemente el valor medio de los valores pico que se obtienen la razón para decantarse por uno u otro eje como el más apropiado para el uso del generador lineal.

Al analizar con detenimiento los diagramas de densidad de probabilidad, vuelven a obtenerse las mismas conclusiones que las observadas en el resto de ensayos realizados. La función que representa los ensayos realizados en la pierna, tiene una campana mucho más apropiada para el uso que se pretende, puesto que se muestra más ancha que las demás y con valores más altos de aceleración. Aunque la densidad de estos datos sea menor, es deseable obtener valores elevados aunque éstos sean más dispersos, adaptándose mucho mejor las distribuciones obtenidas en la pierna a esta situación. En este caso, de nuevo, no existen diferencias entre ejes, valiendo la misma conclusión para todos ellos al darse el mismo caso para las tres direcciones del espacio.

Igualmente se confirma todo lo comentado anteriormente al examinar los diagramas de cajas de los resultados. De nuevo destacan los resultado en la pierna sobre el resto de los ensayos, aunque en este caso, podríamos destacar los ejes X e Y sobre el eje Z ,

donde observamos que los valores extremos conseguidos son ligeramente mayores, aunque en cualquier caso cabe destacar la gran similitud entre resultados y la simetría de los diagramas que representan unas distribuciones de datos muy equilibrados y constantes.

A la luz de todo lo anteriormente expuesto, es conveniente destacar la prevalencia del eje X por sus resultados, no sólo superiores en magnitud, sino también más constantes en densidad y frecuencia, tal y como se puede comprobar en sus diagramas de picos. Dentro de los ensayos realizados sobre esta dirección, claramente es el ensayo de aceleraciones en la pierna el que destaca abiertamente por su mayor magnitud de resultados, resultados, que le hacen preferible sobre el resto para desarrollar más eficazmente la función energética objeto de estudio del proyecto.

Parte V

Conclusiones

A partir del trabajo realizado, pueden obtenerse conclusiones basadas en el análisis de los valores obtenidos en los diferentes experimentos o ensayos realizados, en los que se ha variado, para cada caso particular, las condiciones de los mismos. Podemos dividir estas conclusiones en dos partes; aquellas que hacen referencia a la herramienta de tratamiento y representación de datos que se ha desarrollado, y las referentes al estudio de los gráficos donde se representan la distribución de los datos de aceleración obtenidos durante los ensayos.

Herramienta de Software

Se ha conseguido un código muy compacto, que permite, con pequeños cambios en las variables que hacen referencia a los límites de los intervalos de muestras de cada ensayo, el tratamiento y representación de los datos obtenidos, con mucha rapidez y desde cualquier equipo con capacidad de ejecución de la función.

El código utilizado, puede ser modificado por el usuario del mismo, bien manteniendo la esencia de la función modificando solamente aspectos estéticos y operativos, o bien cambiando parte de la misma, modificando parcialmente o por completo la fisionomía de los gráficos, así como la información que se recibiría de ellos. Esto permite ampliar las posibilidades del trabajo realizado permitiendo ser utilizado y modificado por cualquier usuario para adaptarse lo mejor posible a las aplicaciones concretas que se requieran.

El hecho de realizar tanto el tratamiento como la representación de los datos obtenidos con MatLab, permite integrar en un sólo software toda la herramienta al completo, evitando así exportar e importar datos entre funciones, utilizando únicamente el archivo generado por el ADXL345 a través de la tarjeta SD, evitando posibles problemas de compatibilidad de formatos o del tiempo requerido en la ejecución.

La posibilidad de trasladar los datos en una tarjeta SD, permite utilizar la función desde cualquier equipo capaz de ejecutarla y en el momento que se desee. Esto la convierte

en una herramienta muy versátil que puede ser de gran utilidad en labores de campo, pero que permite también su uso en laboratorio para estudios posteriores, pudiendo comprobar los resultados en el mismo momento de su obtención durante los trabajos de campo de manera que se pueda comprobar previamente que han sido tomados de manera correcta, así como realizar más tarde análisis más precisos y de mayor profundidad.

Estudio de Resultados

El estudio realizado, revela de forma objetiva las aceleraciones que registra un cuerpo humano bajo condiciones controladas y un entorno conocido, dejando a un lado las ideas preconcebidas intuitivamente sobre cuáles debían ser los resultados a obtener.

Las representaciones gráficas de los datos obtenidos en el acelerómetro permiten cuantificar de manera exacta las diferencias entre cada uno de los ensayos, dando a través de estadísticas una idea no sólo cualitativa, sino cuantitativa de las aceleraciones que se obtienen en cada uno de los casos estudiados.

La pierna destaca claramente como la zona estudiada donde conseguir valores de aceleración media y extremos de mayor magnitud tanto en ensayos de carrera como durante el trabajo en escaleras. Los resultados no dejan lugar a dudas sobre dónde son mayores las aceleraciones en todo momento y, en consecuencia, donde será más interesante colocar un generador lineal que permita alimentar pequeñas cargas. Entre los resultados en brazo y cadera, es difícil determinar con claridad a afirmar cuál de las dos partes del cuerpo resulta más interesante como punto de colocación para un generador lineal, tanto en carrera como en trabajo en escaleras, y a cualquiera de las velocidades, por lo que para confirmar una posible mejor viabilidad de una sobre otra, serían necesarios realizar experimentos más precisos y específicos que permitan cuantificar de manera específica las ventajas de una posición sobre la otra, en cada una de las condiciones particulares estudiadas.

La dirección del espacio donde podemos encontrar una mejor distribución de aceleraciones y de valores más elevados dependerá del tipo de ensayo y la velocidad a la que se realice. Para carrera a velocidades bajas (3 y 5 km/h), encontramos que el eje X es donde se consiguen los resultados más interesantes, pero si atendemos a los resultados a 10 km/h , encontramos como el eje Y se convierte en el más conveniente, y el eje X pasa a ser el menos recomendable.

Por otro lado, el estudio en trabajo de escaleras revela como, a bajas velocidades, es la fuerza de la gravedad la que ayuda a decantarse al eje Y como el de condiciones más favorables, aunque el conjunto de las distribuciones permite concluir como las aceleraciones registradas durante trabajo de escaleras a bajas velocidades tienen valores muy cercanos a cero. Por esta circunstancia, es una situación muy desfavorable para su uso en labores de generación de energía mediante un generador lineal. En el caso de velocidades más elevadas en escaleras, se nos presenta la misma situación que en bajas velocidades, siendo el eje Y el más favorable a raíz de ser el eje sobre el cuál la fuerza de la gravedad ejerce una mayor influencia aumentando así los valores de aceleraciones que se consiguen. Omitiendo esta particularidad, sería muy difícil decantarse por un eje por delante del resto, siendo necesarios ensayos exhaustivos que dieran valores exactos de obtención de energía y clarificaran la elección.

El trabajo realizado, por último, aporta valiosa información general sobre las diferentes situaciones elegidas, si bien, tanto por la amplitud del tema como por las posibilidades de aplicación tecnológica e industrial que ofrece, se hace imprescindible completar su desarrollo en estudios posteriores.

Parte VI

Líneas Futuras

A partir de la actividad realizada, este proyecto abre un gran campo de posibilidades para líneas futuras de trabajo. Muchas de ellas quedan fuera del estudio realizado en este proyecto, pero forman parte de un planteamiento de mayor alcance para el que este proyecto fue concebido. Son muy variadas las aplicaciones prácticas en las que el trabajo realizado ofrece potencial.

- Profundizar en la misma línea de investigación ampliando el estudio a nuevos puntos de la anatomía humana, pudiendo no sólo analizar nuevas situaciones, sino también aumentar la exactitud de esos análisis eligiendo posiciones mucho más específicas sobre una parte del cuerpo aislada.
- Añadir diferentes movimientos al estudio, de manera que podamos distinguir para cada caso particular la posición más adecuada de colocación de un generador lineal. Podría trabajarse con movimientos más específicos como andar en bicicleta, la natación, o añadir inclinación a los movimientos ya estudiados, jugando así con diferentes condiciones que amplíen el campo de estudio.
- Estudiar la posibilidad a tenor de los resultados, de modificar el tipo de generador a utilizar. Las condiciones y medidas del ensayo serían probablemente objeto de cambio también, así como la posterior interpretación de los resultados que se obtendrían.
- Trabajar con diferentes frecuencias de muestreo, valorando la validez en cada caso de los resultados que se arrojarían, y si se ajustaría mejor al tipo de actividad que se esté realizando durante el estudio, dando lugar a datos más apropiados y que correspondan mejor con la realidad.
- Se puede trabajar con la herramienta, consiguiendo diferentes configuraciones para los gráficos de los ensayos, en caso de que se quiera obtener cualquier otra forma de expresarles porque se necesite estudiar otros aspectos de la señal. Asimismo, no sólo se podría plantear una modificación, sino la ocasión de implementar nuevos

módulos sobre los ya desarrollados ampliando el campo de formas de expresar e interpretar los datos, y posiblemente ampliando el campo de posibles aplicaciones prácticas.

- Es posible incorporar nuevas maneras de procesar los datos, sustituyendo o incorporando a la función, nuevos algoritmos de tratamientos de señales que me permitan alcanzar nuevas conclusiones a partir de los mismos datos obtenidos. Se puede plantear, como ejemplo, la integración de la señal, de manera que se pudiera visualizar las velocidades de las zonas ensayadas durante el experimento consiguiendo nuevas posibles interpretaciones.
- Se podría desarrollar un interfaz para la herramienta de software, en la que mediante botones y cuadros para inserción de datos y su tratamiento, se pudiera controlar la misma y realizar representaciones según la intención del usuario. Ésto facilitaría el manejo de la misma y haría posible su uso sin tener conocimientos previos de programación en MatLab. Podría ser muy útil en determinadas aplicaciones concretas, aunque en estudios muy concretos, seguirá siendo recomendable poder actuar directamente sobre el código ya que así se permitirá el control total de la labor realizada.
- El diseño de un nuevo generador para aplicar los resultados obtenidos, de manera que pueda adaptarse a las condiciones de alimentación según se requieran, con mayor exactitud y optimizar su funcionamiento y la energía obtenida.
- De manera que los diseños y estudios realizados tengan una aplicación más práctica y por tanto más comercial, se puede estudiar la posibilidad de reducir al máximo el tamaño del conjunto del dispositivo a utilizar, con el fin de lograr que aumenten sus posibles aplicaciones durante los diferentes momentos de una rutina diaria.
- Un análisis adicional que podría ser de utilidad, sería la realización de un trabajo de recopilación de las posibles cargas electrónicas de baja potencia que podrían ser alimentadas bajo las condiciones estipuladas, de manera que se tenga una idea

general de las posibilidades que ofrece el dispositivo, bien para aumentar el número de ellas, o bien para incidir en su investigación.

Parte VII

Conclusión Personal

La intención de realizar un proyecto de investigación como "Proyecto de Fin de Carrera" suponía familiarizarse con una nueva metodología de trabajo y con el incentivo de llegar a una conclusión nueva que tuviera posibles aplicaciones y recorrido futuro. A nivel personal suponía una oportunidad de iniciarse en el campo de la investigación y de trabajar en un tema en desarrollo que me permitiera aportar algo diferente y buscar un horizonte para ese aporte.

Los comienzos sirvieron para estudiar la actualidad del tema a tratar y realizar una labor de lectura e investigación de textos que ayudaran a contextualizar la labor que se iba a realizar posteriormente así como informar sobre estudios anteriores que aportaran datos y literatura sobre la materia. En mi experiencia personal, me enriqueció enormemente ya que aprendí muchas cosas totalmente nuevas para mí, en las que no se había llegado al detalle durante el resto de la carrera.

En la toma de decisiones sobre aspectos de los experimentos a realizar y el equipo a utilizar, me he sentido valorado por mi director, ya que se me permitió tener libertad de criterio y de decisión, siempre bajo supervisión, sobre cómo quería proceder. Durante estas cuestiones, el hecho de tener un abanico de posibilidades y la oportunidad de poder optar por la solución que consideras adecuada ha significado una experiencia de responsabilidad y de madurez muy valiosa para mí, porque el margen de movimientos que se te ofrece y la confianza depositada en tu decisión te obliga a reflexionar mucho más sobre la elección del camino a seguir.

Durante el curso de la labor realizada, se presentan multitud de dificultades, que retrasan de alguna manera el avance de la misma. Sirva como ejemplo la gran cantidad de tiempo dedicado a la configuración del acelerómetro, ya que aunque a nivel de usuario era totalmente modificable, se necesitaba estar familiarizado con el lenguaje del código de programación, así como de la estructura y las subrutinas usadas, para cambiar los aspectos deseados. Fue una tarea que llevó mucho tiempo, pero que se hizo correctamente.

Considero importante destacar que en todo momento ha existido una muy buena comunicación con el Director del proyecto. Esta circunstancia es algo que personalmente le agradezco profundamente y valoro sobremanera porque he sido testigo del poco tiempo que le dejaban sus labores académicas e investigadoras y, sin embargo, encontraba el momento para atenderme. Por la confianza que me ha dado durante el desarrollo de la investigación y la redacción de proyecto y, particularmente, al concederme la oportunidad de trabajar con él le reitero mi gratitud y el reconocimiento público en estas líneas de que estos meses de trabajo han constituido una grata experiencia de permanente recuerdo en mi vida.

Como conjunto la valoración del proyecto ha sido muy positiva. Tanto el tema como el proceso de trabajo han sido muy interesantes y originales, dando frescura y consiguiendo despertar siempre mi interés. En un futuro espero tener la oportunidad de poder continuar con el trabajo que se comenzó y aportar cosas nuevas a lo reflejado en este proyecto.

Parte VIII

Anexos

8.6. ANEXO A



3-Axis, $\pm 2\text{ g}/\pm 4\text{ g}/\pm 8\text{ g}/\pm 16\text{ g}$ Digital Accelerometer

ADXL345

FEATURES

Ultralow power: as low as $23\ \mu\text{A}$ in measurement mode and $0.1\ \mu\text{A}$ in standby mode at $V_S = 2.5\ \text{V}$ (typical)

Power consumption scales automatically with bandwidth
User-selectable resolution

Fixed 10-bit resolution

Full resolution, where resolution increases with g range, up to 13-bit resolution at $\pm 16\text{ g}$ (maintaining 4 mg/LSB scale factor in all g ranges)

Patent pending, embedded memory management system with FIFO technology minimizes host processor load

Single tap/double tap detection

Activity/inactivity monitoring

Free-fall detection

Supply voltage range: $2.0\ \text{V}$ to $3.6\ \text{V}$

I/O voltage range: $1.7\ \text{V}$ to V_S

SPI (3- and 4-wire) and I²C digital interfaces

Flexible interrupt modes mappable to either interrupt pin

Measurement ranges selectable via serial command

Bandwidth selectable via serial command

Wide temperature range (-40°C to $+85^\circ\text{C}$)

10,000 g shock survival

Pb free/RoHS compliant

Small and thin: $3\ \text{mm} \times 5\ \text{mm} \times 1\ \text{mm}$ LGA package

APPLICATIONS

Handsets

Medical instrumentation

Gaming and pointing devices

Industrial instrumentation

Personal navigation devices

Hard disk drive (HDD) protection

GENERAL DESCRIPTION

The ADXL345 is a small, thin, ultralow power, 3-axis accelerometer with high resolution (13-bit) measurement at up to $\pm 16\text{ g}$. Digital output data is formatted as 16-bit two's complement and is accessible through either a SPI (3- or 4-wire) or I²C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (3.9 mg/LSB) enables measurement of inclination changes less than 1.0° .

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion by comparing the acceleration on any axis with user-set thresholds. Tap sensing detects single and double taps in any direction. Free-fall sensing detects if the device is falling. These functions can be mapped individually to either of two interrupt output pins. An integrated, patent pending memory management system with a 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor activity and lower overall system power consumption.

Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

The ADXL345 is supplied in a small, thin, $3\ \text{mm} \times 5\ \text{mm} \times 1\ \text{mm}$, 14-lead, plastic package.

FUNCTIONAL BLOCK DIAGRAM

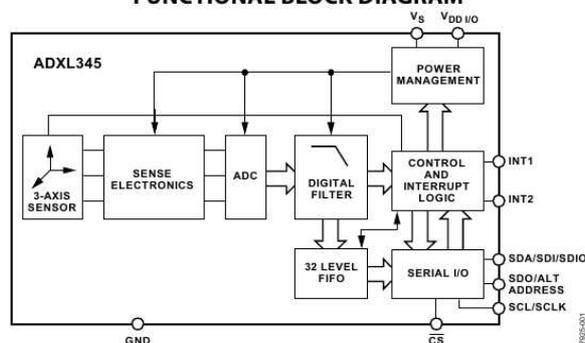


Figure 1.

SPECIFICATIONS

$T_A = 25^\circ\text{C}$, $V_S = 2.5\text{ V}$, $V_{DD I/O} = 1.8\text{ V}$, acceleration = 0 g, $C_S = 10\ \mu\text{F}$ tantalum, $C_{I/O} = 0.1\ \mu\text{F}$, output data rate (ODR) = 800 Hz, unless otherwise noted. All minimum and maximum specifications are guaranteed. Typical specifications are not guaranteed.

Table 1.

Parameter	Test Conditions	Min	Typ ¹	Max	Unit
SENSOR INPUT					
Measurement Range	Each axis User selectable		$\pm 2, \pm 4, \pm 8, \pm 16$		g
Nonlinearity	Percentage of full scale		± 0.5		%
Inter-Axis Alignment Error			± 0.1		Degrees
Cross-Axis Sensitivity ²			± 1		%
OUTPUT RESOLUTION					
All g Ranges	Each axis 10-bit resolution		10		Bits
$\pm 2\text{ g}$ Range	Full resolution		10		Bits
$\pm 4\text{ g}$ Range	Full resolution		11		Bits
$\pm 8\text{ g}$ Range	Full resolution		12		Bits
$\pm 16\text{ g}$ Range	Full resolution		13		Bits
SENSITIVITY					
Sensitivity at $X_{OUT}, Y_{OUT}, Z_{OUT}$	Each axis All g-ranges, full resolution	230	256	282	LSB/g
	$\pm 2\text{ g}$, 10-bit resolution	230	256	282	LSB/g
	$\pm 4\text{ g}$, 10-bit resolution	115	128	141	LSB/g
	$\pm 8\text{ g}$, 10-bit resolution	57	64	71	LSB/g
	$\pm 16\text{ g}$, 10-bit resolution	29	32	35	LSB/g
Sensitivity Deviation from Ideal	All g-ranges		± 1.0		%
Scale Factor at $X_{OUT}, Y_{OUT}, Z_{OUT}$	All g-ranges, full resolution	3.5	3.9	4.3	mg/LSB
	$\pm 2\text{ g}$, 10-bit resolution	3.5	3.9	4.3	mg/LSB
	$\pm 4\text{ g}$, 10-bit resolution	7.1	7.8	8.7	mg/LSB
	$\pm 8\text{ g}$, 10-bit resolution	14.1	15.6	17.5	mg/LSB
	$\pm 16\text{ g}$, 10-bit resolution	28.6	31.2	34.5	mg/LSB
Sensitivity Change Due to Temperature			± 0.01		%/ $^\circ\text{C}$
0 g OFFSET					
0 g Output for X_{OUT}, Y_{OUT}	Each axis	-150	0	+150	mg
0 g Output for Z_{OUT}		-250	0	+250	mg
0 g Output Deviation from Ideal, X_{OUT}, Y_{OUT}			± 35		mg
0 g Output Deviation from Ideal, Z_{OUT}			± 40		mg
0 g Offset vs. Temperature for X-, Y-Axes			± 0.4		mg/ $^\circ\text{C}$
0 g Offset vs. Temperature for Z-Axis			± 1.2		mg/ $^\circ\text{C}$
NOISE					
X-, Y-Axes	ODR = 100 Hz for $\pm 2\text{ g}$, 10-bit resolution or all g-ranges, full resolution		0.75		LSB rms
Z-Axis	ODR = 100 Hz for $\pm 2\text{ g}$, 10-bit resolution or all g-ranges, full resolution		1.1		LSB rms
OUTPUT DATA RATE AND BANDWIDTH					
Output Data Rate (ODR) ^{3, 4, 5}	User selectable	0.1		3200	Hz
SELF-TEST⁶					
Output Change in X-Axis		0.20		2.10	g
Output Change in Y-Axis		-2.10		-0.20	g
Output Change in Z-Axis		0.30		3.40	g
POWER SUPPLY					
Operating Voltage Range (V_S)		2.0	2.5	3.6	V
Interface Voltage Range ($V_{DD I/O}$)		1.7	1.8	V_S	V
Supply Current	ODR $\geq 100\text{ Hz}$		140		μA
	ODR $< 10\text{ Hz}$		30		μA
Standby Mode Leakage Current			0.1		μA
Turn-On and Wake-Up Time ⁷	ODR = 3200 Hz		1.4		ms

Parameter	Test Conditions	Min	Typ ¹	Max	Unit
TEMPERATURE					
Operating Temperature Range		-40		+85	°C
WEIGHT					
Device Weight			30		mg

¹ The typical specifications shown are for at least 68% of the population of parts and are based on the worst case of mean $\pm 1\sigma$, except for 0 g output and sensitivity, which represents the target value. For 0 g offset and sensitivity, the deviation from the ideal describes the worst case of mean $\pm 1\sigma$.

² Cross-axis sensitivity is defined as coupling between any two axes.

³ Bandwidth is the -3 dB frequency and is half the output data rate, bandwidth = ODR/2.

⁴ The output format for the 3200 Hz and 1600 Hz ODRs is different than the output format for the remaining ODRs. This difference is described in the Data Formatting of Upper Data Rates section.

⁵ Output data rates below 6.25 Hz exhibit additional offset shift with increased temperature, depending on selected output data rate. Refer to the Offset Performance at Lowest Data Rates section for details.

⁶ Self-test change is defined as the output (g) when the SELF_TEST bit = 1 (in the DATA_FORMAT register, Address 0x31) minus the output (g) when the SELF_TEST bit = 0. Due to device filtering, the output reaches its final value after $4 \times \tau$ when enabling or disabling self-test, where $\tau = 1/(\text{data rate})$. The part must be in normal power operation (LOW_POWER bit = 0 in the BW_RATE register, Address 0x2C) for self-test to operate correctly.

⁷ Turn-on and wake-up times are determined by the user-defined bandwidth. At a 100 Hz data rate, the turn-on and wake-up times are each approximately 11.1 ms. For other data rates, the turn-on and wake-up times are each approximately $\tau + 1.1$ in milliseconds, where $\tau = 1/(\text{data rate})$.

ABSOLUTE MAXIMUM RATINGS

Table 2.

Parameter	Rating
Acceleration	
Any Axis, Unpowered	10,000 g
Any Axis, Powered	10,000 g
V _S	-0.3 V to +3.9 V
V _{DD I/O}	-0.3 V to +3.9 V
Digital Pins	-0.3 V to V _{DD I/O} + 0.3 V or 3.9 V, whichever is less
All Other Pins	-0.3 V to +3.9 V
Output Short-Circuit Duration (Any Pin to Ground)	Indefinite
Temperature Range	
Powered	-40°C to +105°C
Storage	-40°C to +105°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

THERMAL RESISTANCE

Table 3. Package Characteristics

Package Type	θ_{JA}	θ_{JC}	Device Weight
14-Terminal LGA	150°C/W	85°C/W	30 mg

PACKAGE INFORMATION

The information in Figure 2 and Table 4 provide details about the package branding for the ADXL345. For a complete listing of product availability, see the Ordering Guide section.

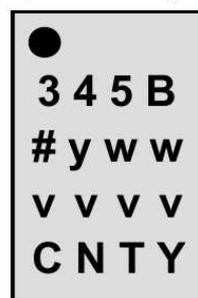


Figure 2. Product Information on Package (Top View)

Table 4. Package Branding Information

Branding Key	Field Description
345B	Part identifier for ADXL345
#	RoHS-compliant designation
yww	Date code
vvvv	Factory lot code
CNTY	Country of origin

ESD CAUTION



ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

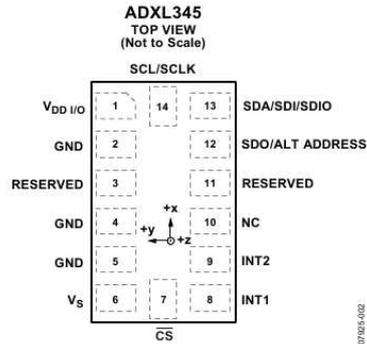


Figure 3. Pin Configuration (Top View)

Table 5. Pin Function Descriptions

Pin No.	Mnemonic	Description
1	V _{DD I/O}	Digital Interface Supply Voltage.
2	GND	This pin must be connected to ground.
3	RESERVED	Reserved. This pin must be connected to V _S or left open.
4	GND	This pin must be connected to ground.
5	GND	This pin must be connected to ground.
6	V _S	Supply Voltage.
7	\overline{CS}	Chip Select.
8	INT1	Interrupt 1 Output.
9	INT2	Interrupt 2 Output.
10	NC	Not Internally Connected.
11	RESERVED	Reserved. This pin must be connected to ground or left open.
12	SDO/ALT ADDRESS	Serial Data Output (SPI 4-Wire)/Alternate I ² C Address Select (I ² C).
13	SDA/SDI/SDIO	Serial Data (I ² C)/Serial Data Input (SPI 4-Wire)/Serial Data Input and Output (SPI 3-Wire).
14	SCL/SCLK	Serial Communications Clock. SCL is the clock for I ² C, and SCLK is the clock for SPI.

REGISTER MAP

Table 19.

Address		Name	Type	Reset Value	Description
Hex	Dec				
0x00	0	DEVID	R	11100101	Device ID
0x01 to 0x1C	1 to 28	Reserved			Reserved; do not access
0x1D	29	THRESH_TAP	R/W	00000000	Tap threshold
0x1E	30	OFSX	R/W	00000000	X-axis offset
0x1F	31	OFSY	R/W	00000000	Y-axis offset
0x20	32	OFSZ	R/W	00000000	Z-axis offset
0x21	33	DUR	R/W	00000000	Tap duration
0x22	34	Latent	R/W	00000000	Tap latency
0x23	35	Window	R/W	00000000	Tap window
0x24	36	THRESH_ACT	R/W	00000000	Activity threshold
0x25	37	THRESH_INACT	R/W	00000000	Inactivity threshold
0x26	38	TIME_INACT	R/W	00000000	Inactivity time
0x27	39	ACT_INACT_CTL	R/W	00000000	Axis enable control for activity and inactivity detection
0x28	40	THRESH_FF	R/W	00000000	Free-fall threshold
0x29	41	TIME_FF	R/W	00000000	Free-fall time
0x2A	42	TAP_AXES	R/W	00000000	Axis control for single tap/double tap
0x2B	43	ACT_TAP_STATUS	R	00000000	Source of single tap/double tap
0x2C	44	BW_RATE	R/W	00001010	Data rate and power mode control
0x2D	45	POWER_CTL	R/W	00000000	Power-saving features control
0x2E	46	INT_ENABLE	R/W	00000000	Interrupt enable control
0x2F	47	INT_MAP	R/W	00000000	Interrupt mapping control
0x30	48	INT_SOURCE	R	00000010	Source of interrupts
0x31	49	DATA_FORMAT	R/W	00000000	Data format control
0x32	50	DATA_X0	R	00000000	X-Axis Data 0
0x33	51	DATA_X1	R	00000000	X-Axis Data 1
0x34	52	DATA_Y0	R	00000000	Y-Axis Data 0
0x35	53	DATA_Y1	R	00000000	Y-Axis Data 1
0x36	54	DATA_Z0	R	00000000	Z-Axis Data 0
0x37	55	DATA_Z1	R	00000000	Z-Axis Data 1
0x38	56	FIFO_CTL	R/W	00000000	FIFO control
0x39	57	FIFO_STATUS	R	00000000	FIFO status

REGISTER DEFINITIONS

Register 0x00—DEVID (Read Only)

D7	D6	D5	D4	D3	D2	D1	D0
1	1	1	0	0	1	0	1

The DEVID register holds a fixed device ID code of 0xE5 (345 octal).

Register 0x1D—THRESH_TAP (Read/Write)

The THRESH_TAP register is eight bits and holds the threshold value for tap interrupts. The data format is unsigned, therefore, the magnitude of the tap event is compared with the value in THRESH_TAP for normal tap detection. The scale factor is 62.5 mg/LSB (that is, 0xFF = 16 g). A value of 0 may result in undesirable behavior if single tap/double tap interrupts are enabled.

Register 0x1E, Register 0x1F, Register 0x20—OFSX, OFSY, OFSZ (Read/Write)

The OFSX, OFSY, and OFSZ registers are each eight bits and offer user-set offset adjustments in twos complement format with a scale factor of 15.6 mg/LSB (that is, 0x7F = 2 g). The value stored in the offset registers is automatically added to the acceleration data, and the resulting value is stored in the output data registers. For additional information regarding offset calibration and the use of the offset registers, refer to the Offset Calibration section.

Register 0x21—DUR (Read/Write)

The DUR register is eight bits and contains an unsigned time value representing the maximum time that an event must be above the THRESH_TAP threshold to qualify as a tap event. The scale factor is 625 μ s/LSB. A value of 0 disables the single tap/double tap functions.

Register 0x22—Latent (Read/Write)

The latent register is eight bits and contains an unsigned time value representing the wait time from the detection of a tap event to the start of the time window (defined by the window register) during which a possible second tap event can be detected. The scale factor is 1.25 ms/LSB. A value of 0 disables the double tap function.

Register 0x23—Window (Read/Write)

The window register is eight bits and contains an unsigned time value representing the amount of time after the expiration of the latency time (determined by the latent register) during which a second valid tap can begin. The scale factor is 1.25 ms/LSB. A value of 0 disables the double tap function.

Register 0x24—THRESH_ACT (Read/Write)

The THRESH_ACT register is eight bits and holds the threshold value for detecting activity. The data format is unsigned, so the magnitude of the activity event is compared with the value in the THRESH_ACT register. The scale factor is 62.5 mg/LSB. A value of 0 may result in undesirable behavior if the activity interrupt is enabled.

Register 0x25—THRESH_INACT (Read/Write)

The THRESH_INACT register is eight bits and holds the threshold value for detecting inactivity. The data format is unsigned, so the magnitude of the inactivity event is compared with the value in the THRESH_INACT register. The scale factor is 62.5 mg/LSB. A value of 0 may result in undesirable behavior if the inactivity interrupt is enabled.

Register 0x26—TIME_INACT (Read/Write)

The TIME_INACT register is eight bits and contains an unsigned time value representing the amount of time that acceleration must be less than the value in the THRESH_INACT register for inactivity to be declared. The scale factor is 1 sec/LSB. Unlike the other interrupt functions, which use unfiltered data (see the Threshold section), the inactivity function uses filtered output data. At least one output sample must be generated for the inactivity interrupt to be triggered. This results in the function appearing unresponsive if the TIME_INACT register is set to a value less than the time constant of the output data rate. A value of 0 results in an interrupt when the output data is less than the value in the THRESH_INACT register.

Register 0x27—ACT_INACT_CTL (Read/Write)

D7	D6	D5	D4
ACT ac/dc	ACT_X enable	ACT_Y enable	ACT_Z enable
D3	D2	D1	D0
INACT ac/dc	INACT_X enable	INACT_Y enable	INACT_Z enable

ACT AC/DC and INACT AC/DC Bits

A setting of 0 selects dc-coupled operation, and a setting of 1 enables ac-coupled operation. In dc-coupled operation, the current acceleration magnitude is compared directly with THRESH_ACT and THRESH_INACT to determine whether activity or inactivity is detected.

In ac-coupled operation for activity detection, the acceleration value at the start of activity detection is taken as a reference value. New samples of acceleration are then compared to this reference value, and if the magnitude of the difference exceeds the THRESH_ACT value, the device triggers an activity interrupt.

Similarly, in ac-coupled operation for inactivity detection, a reference value is used for comparison and is updated whenever the device exceeds the inactivity threshold. After the reference value is selected, the device compares the magnitude of the difference between the reference value and the current acceleration with THRESH_INACT. If the difference is less than the value in THRESH_INACT for the time in TIME_INACT, the device is considered inactive and the inactivity interrupt is triggered.

ACT_x Enable Bits and INACT_x Enable Bits

A setting of 1 enables x-, y-, or z-axis participation in detecting activity or inactivity. A setting of 0 excludes the selected axis from participation. If all axes are excluded, the function is disabled. For activity detection, all participating axes are logically ORed, causing the activity function to trigger when any of the participating axes exceeds the threshold. For inactivity detection, all participating axes are logically ANDed, causing the inactivity function to trigger only if all participating axes are below the threshold for the specified time.

Register 0x28—THRESH_FF (Read/Write)

The THRESH_FF register is eight bits and holds the threshold value, in unsigned format, for free-fall detection. The acceleration on all axes is compared with the value in THRESH_FF to determine if a free-fall event occurred. The scale factor is 62.5 mg/LSB. Note that a value of 0 mg may result in undesirable behavior if the free-fall interrupt is enabled. Values between 300 mg and 600 mg (0x05 to 0x09) are recommended.

Register 0x29—TIME_FF (Read/Write)

The TIME_FF register is eight bits and stores an unsigned time value representing the minimum time that the value of all axes must be less than THRESH_FF to generate a free-fall interrupt. The scale factor is 5 ms/LSB. A value of 0 may result in undesirable behavior if the free-fall interrupt is enabled. Values between 100 ms and 350 ms (0x14 to 0x46) are recommended.

Register 0x2A—TAP_AXES (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	Suppress	TAP_X enable	TAP_Y enable	TAP_Z enable

Suppress Bit

Setting the suppress bit suppresses double tap detection if acceleration greater than the value in THRESH_TAP is present between taps. See the Tap Detection section for more details.

TAP_x Enable Bits

A setting of 1 in the TAP_X enable, TAP_Y enable, or TAP_Z enable bit enables x-, y-, or z-axis participation in tap detection. A setting of 0 excludes the selected axis from participation in tap detection.

Register 0x2B—ACT_TAP_STATUS (Read Only)

D7	D6	D5	D4	D3	D2	D1	D0
0	ACT_X source	ACT_Y source	ACT_Z source	Asleep	TAP_X source	TAP_Y source	TAP_Z source

ACT_x Source and TAP_x Source Bits

These bits indicate the first axis involved in a tap or activity event. A setting of 1 corresponds to involvement in the event, and a setting of 0 corresponds to no involvement. When new data is available, these bits are not cleared but are overwritten by the new data. The ACT_TAP_STATUS register should be read before clearing the interrupt. Disabling an axis from participation clears the corresponding source bit when the next activity or single tap/double tap event occurs.

Asleep Bit

A setting of 1 in the asleep bit indicates that the part is asleep, and a setting of 0 indicates that the part is not asleep. This bit toggles only if the device is configured for auto sleep. See the AUTO_SLEEP Bit section for more information on autosleep mode.

Register 0x2C—BW_RATE (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	LOW_POWER	Rate			

LOW_POWER Bit

A setting of 0 in the LOW_POWER bit selects normal operation, and a setting of 1 selects reduced power operation, which has somewhat higher noise (see the Power Modes section for details).

Rate Bits

These bits select the device bandwidth and output data rate (see Table 7 and Table 8 for details). The default value is 0x0A, which translates to a 100 Hz output data rate. An output data rate should be selected that is appropriate for the communication protocol and frequency selected. Selecting too high of an output data rate with a low communication speed results in samples being discarded.

Register 0x2D—POWER_CTL (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
0	0	Link	AUTO_SLEEP	Measure	Sleep	Wakeup	

Link Bit

A setting of 1 in the link bit with both the activity and inactivity functions enabled delays the start of the activity function until inactivity is detected. After activity is detected, inactivity detection begins, preventing the detection of activity. This bit serially links the activity and inactivity functions. When this bit is set to 0, the inactivity and activity functions are concurrent. Additional information can be found in the Link Mode section.

When clearing the link bit, it is recommended that the part be placed into standby mode and then set back to measurement mode with a subsequent write. This is done to ensure that the device is properly biased if sleep mode is manually disabled; otherwise, the first few samples of data after the link bit is cleared may have additional noise, especially if the device was asleep when the bit was cleared.

AUTO_SLEEP Bit

If the link bit is set, a setting of 1 in the AUTO_SLEEP bit enables the auto-sleep functionality. In this mode, the ADXL345 automatically switches to sleep mode if the inactivity function is enabled and inactivity is detected (that is, when acceleration is below the THRESH_INACT value for at least the time indicated by TIME_INACT). If activity is also enabled, the ADXL345 automatically wakes up from sleep after detecting activity and returns to operation at the output data rate set in the BW_RATE register. A setting of 0 in the AUTO_SLEEP bit disables automatic switching to sleep mode. See the description of the Sleep Bit in this section for more information on sleep mode.

If the link bit is not set, the AUTO_SLEEP feature is disabled and setting the AUTO_SLEEP bit does not have an impact on device operation. Refer to the Link Bit section or the Link Mode section for more information on utilization of the link feature.

When clearing the AUTO_SLEEP bit, it is recommended that the part be placed into standby mode and then set back to measurement mode with a subsequent write. This is done to ensure that the device is properly biased if sleep mode is manually disabled; otherwise, the first few samples of data after the AUTO_SLEEP bit is cleared may have additional noise, especially if the device was asleep when the bit was cleared.

Measure Bit

A setting of 0 in the measure bit places the part into standby mode, and a setting of 1 places the part into measurement mode. The ADXL345 powers up in standby mode with minimum power consumption.

Sleep Bit

A setting of 0 in the sleep bit puts the part into the normal mode of operation, and a setting of 1 places the part into sleep mode. Sleep mode suppresses DATA_READY, stops transmission of data to FIFO, and switches the sampling rate to one specified by the wakeup bits. In sleep mode, only the activity function can be used. When the DATA_READY interrupt is suppressed, the output data registers (Register 0x32 to Register 0x37) are still updated at the sampling rate set by the wakeup bits (D1:D0).

When clearing the sleep bit, it is recommended that the part be placed into standby mode and then set back to measurement mode with a subsequent write. This is done to ensure that the device is properly biased if sleep mode is manually disabled; otherwise, the first few samples of data after the sleep bit is cleared may have additional noise, especially if the device was asleep when the bit was cleared.

Wakeup Bits

These bits control the frequency of readings in sleep mode as described in Table 20.

Table 20. Frequency of Readings in Sleep Mode

Setting		Frequency (Hz)
D1	D0	
0	0	8
0	1	4
1	0	2
1	1	1

Register 0x2E—INT_ENABLE (Read/Write)

D7	D6	D5	D4
DATA_READY	SINGLE_TAP	DOUBLE_TAP	Activity
D3	D2	D1	D0
Inactivity	FREE_FALL	Watermark	Overrun

Setting bits in this register to a value of 1 enables their respective functions to generate interrupts, whereas a value of 0 prevents the functions from generating interrupts. The DATA_READY, watermark, and overrun bits enable only the interrupt output; the functions are always enabled. It is recommended that interrupts be configured before enabling their outputs.

Register 0x2F—INT_MAP (R/W)

D7	D6	D5	D4
DATA_READY	SINGLE_TAP	DOUBLE_TAP	Activity
D3	D2	D1	D0
Inactivity	FREE_FALL	Watermark	Overrun

Any bits set to 0 in this register send their respective interrupts to the INT1 pin, whereas bits set to 1 send their respective interrupts to the INT2 pin. All selected interrupts for a given pin are ORed.

Register 0x30—INT_SOURCE (Read Only)

D7	D6	D5	D4
DATA_READY	SINGLE_TAP	DOUBLE_TAP	Activity
D3	D2	D1	D0
Inactivity	FREE_FALL	Watermark	Overrun

Bits set to 1 in this register indicate that their respective functions have triggered an event, whereas a value of 0 indicates that the corresponding event has not occurred. The DATA_READY, watermark, and overrun bits are always set if the corresponding events occur, regardless of the INT_ENABLE register settings, and are cleared by reading data from the DATA_X, DATA_Y, and DATA_Z registers. The DATA_READY and watermark bits may require multiple reads, as indicated in the FIFO mode descriptions in the FIFO section. Other bits, and the corresponding interrupts, are cleared by reading the INT_SOURCE register.

Register 0x31—DATA_FORMAT (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
SELF_TEST	SPI	INT_INVERT	0	FULL_RES	Justify	Range	

The DATA_FORMAT register controls the presentation of data to Register 0x32 through Register 0x37. All data, except that for the ± 16 g range, must be clipped to avoid rollover.

SELF_TEST Bit

A setting of 1 in the SELF_TEST bit applies a self-test force to the sensor, causing a shift in the output data. A value of 0 disables the self-test force.

SPI Bit

A value of 1 in the SPI bit sets the device to 3-wire SPI mode, and a value of 0 sets the device to 4-wire SPI mode.

INT_INVERT Bit

A value of 0 in the INT_INVERT bit sets the interrupts to active high, and a value of 1 sets the interrupts to active low.

FULL_RES Bit

When this bit is set to a value of 1, the device is in full resolution mode, where the output resolution increases with the *g* range set by the range bits to maintain a 4 mg/LSB scale factor. When the FULL_RES bit is set to 0, the device is in 10-bit mode, and the range bits determine the maximum *g* range and scale factor.

Justify Bit

A setting of 1 in the justify bit selects left-justified (MSB) mode, and a setting of 0 selects right-justified mode with sign extension.

Range Bits

These bits set the *g* range as described in Table 21.

Table 21. *g* Range Setting

Setting		<i>g</i> Range
D1	D0	
0	0	±2 <i>g</i>
0	1	±4 <i>g</i>
1	0	±8 <i>g</i>
1	1	±16 <i>g</i>

Register 0x32 to Register 0x37—DATA_{X0}, DATA_{X1}, DATA_{Y0}, DATA_{Y1}, DATA_{Z0}, DATA_{Z1} (Read Only)

These six bytes (Register 0x32 to Register 0x37) are eight bits each and hold the output data for each axis. Register 0x32 and Register 0x33 hold the output data for the x-axis, Register 0x34 and Register 0x35 hold the output data for the y-axis, and Register 0x36 and Register 0x37 hold the output data for the z-axis. The output data is two's complement, with DATA_{X0} as the least significant byte and DATA_{X1} as the most significant byte, where x represent X, Y, or Z. The DATA_FORMAT register (Address 0x31) controls the format of the data. It is recommended that a multiple-byte read of all registers be performed to prevent a change in data between reads of sequential registers.

Register 0x38—FIFO_CTL (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_MODE		Trigger		Samples			

FIFO_MODE Bits

These bits set the FIFO mode, as described in Table 22.

Table 22. FIFO Modes

Setting		Mode	Function
D7	D6		
0	0	Bypass	FIFO is bypassed.
0	1	FIFO	FIFO collects up to 32 values and then stops collecting data, collecting new data only when FIFO is not full.
1	0	Stream	FIFO holds the last 32 data values. When FIFO is full, the oldest data is overwritten with newer data.
1	1	Trigger	When triggered by the trigger bit, FIFO holds the last data samples before the trigger event and then continues to collect data until full. New data is collected only when FIFO is not full.

Trigger Bit

A value of 0 in the trigger bit links the trigger event of trigger mode to INT1, and a value of 1 links the trigger event to INT2.

Samples Bits

The function of these bits depends on the FIFO mode selected (see Table 23). Entering a value of 0 in the samples bits immediately sets the watermark status bit in the INT_SOURCE register, regardless of which FIFO mode is selected. Undesirable operation may occur if a value of 0 is used for the samples bits when trigger mode is used.

Table 23. Samples Bits Functions

FIFO Mode	Samples Bits Function
Bypass	None.
FIFO	Specifies how many FIFO entries are needed to trigger a watermark interrupt.
Stream	Specifies how many FIFO entries are needed to trigger a watermark interrupt.
Trigger	Specifies how many FIFO samples are retained in the FIFO buffer before a trigger event.

0x39—FIFO_STATUS (Read Only)

D7	D6	D5	D4	D3	D2	D1	D0
FIFO_TRIG		0		Entries			

FIFO_TRIG Bit

A 1 in the FIFO_TRIG bit corresponds to a trigger event occurring, and a 0 means that a FIFO trigger event has not occurred.

Entries Bits

These bits report how many data values are stored in FIFO. Access to collect the data from FIFO is provided through the DATA_X, DATA_Y, and DATA_Z registers. FIFO reads must be done in burst or multiple-byte mode because each FIFO level is cleared after any read (single- or multiple-byte) of FIFO. FIFO stores a maximum of 32 entries, which equates to a maximum of 33 entries available at any given time because an additional entry is available at the output filter of the device.

8.7. ANEXO B



Evaluation Board User Guide UG-065

One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

*i*MEMS ADXL345/ADXL346 Inertial Sensor Datalogger and Development Board

FEATURES

- Ultralow power **ADXL345/ADXL346** accelerometer
- Inertial sensor development board
- Datalogs onto MicroSD card
- Fully programmable via serial interface; firmware examples provided
- Battery-powered for portable applications

REQUIREMENTS

- 2 AAA batteries
- MicroSD card and card reader (for datalogging)
- Computer with serial port (for programming)

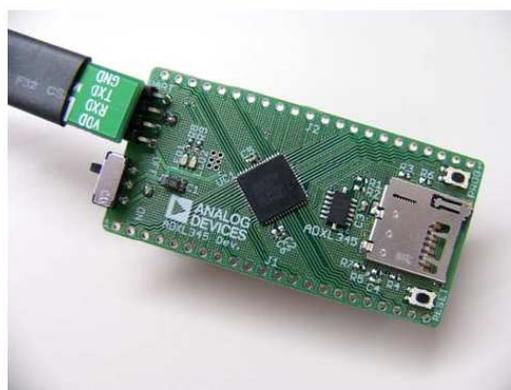


Figure 1. ADXL345 Inertial Sensor Development Board

GENERAL DESCRIPTION

It is often a timesaver in hardware development to make progress on the firmware and the hardware simultaneously. The challenge is that it proves difficult to develop firmware before the hardware exists. The *i*MEMS[®] ADXL345/ADXL346 development board is an easy-to-use tool that facilitates prototyping by providing a platform that can be duplicated in the final application. Additionally, the development board can be configured as a datalogger and can be used to gather data for refining algorithms, tuning thresholds, and generally familiarizing oneself with accelerometer data.

Two AAA batteries power the development board, and thus it integrates seamlessly into portable applications. Communications and processing are done by an ARM7-based **ADuC7024** microcontroller, and the interface provided is fully reprogrammable. Moreover, all ADuC7024 pins are broken out into headers to facilitate design of compatible expansion boards. Data is logged onto a MicroSD memory card, providing essentially unlimited memory capacity and operating system versatility. Data is stored in a text file; therefore, there is no need to install any software to operate the board or read data. Software is provided to assist with programming the board.

8.8. ANEXO C

FUNCIÓN DE REPRESENTACIÓN DE LAS FUNCIONES DE DENSIDAD DE PROBABILIDAD

```
r=0;g=0;b=0;
if p==1
    r=1;
elseif p==2
    g=1;
else
    b=1;
end

% Forzamos a las entradas a convertirse en vectores columna
ejex = ejex(:);

legh_ = []; legt_ = {};
ax_ = newplot;
set(ax_,'Box','on');
hold on;

t_ = ~isnan(ejex);
Data_ = ejex(t_);
[F_,X_] = ecdf(Data_,'Function','cdf');
Bin_.rule = 1;
[C_,E_] = dfswitchyard('dfhistbins',Data_,[],[],Bin_,F_,X_);
[N_,C_] = ecdfhist(F_,X_,'edges',E_);
h_ = bar(C_,N_,'hist');
set(h_,'FaceColor','none','EdgeColor',[r g b],'LineStyle','-','LineWidth',1,'Visible','off');
xlabel('Data');
ylabel('Density');

xlim_ = get(ax_,'XLim');
if all(isfinite(xlim_))
    xlim_ = xlim_ + [-1 1] * 0.01 * diff(xlim_);
    set(ax_,'XLim',xlim_)
end

x_ = linspace(-10,10,100);

t_ = ~isnan(ejex);
Data_ = ejex(t_);
pargs_ = cell(1,2);
[pargs_{:}] = normfit(Data_, 0.05);
p_ = [pargs_{:}];
y_ = normpdf(x_,p_(1), p_(2));
h_ = plot(x_,y_,'Color',[r g b],'LineStyle','-','LineWidth',1,'Marker','none','MarkerSize',6);
```

8.9. ANEXO D

HERRAMIENTA DE SOFTWARE PARA ADQUISICIÓN, TRATAMIENTO Y REPRESENTACIÓN DE LOS DATOS

Función totalmente configurable según el valor del intervalo de muestras en las que se esté interesado en realizar la representación de los datos.

```
functionCodigoFinal

clc,clear all
set(0,'defaultAxesFontName','arial')
set(0,'defaultTextFontName','arial');

% Inicio las variables estadísticas

h_=zeros(1,3);

% Cargamos el archivo con los datos en una matriz

%[filename,pathname]=uigetfile('*.txt','Elige el Archivo a Cargar');
%archivo = strcat (pathname,filename);
A=importdata ('./resul/carrera_02.txt',' ',1);
%A=importdata (archivo,' ',1);
A=A.data;

% Representamos los datos obtenidos. Han de quedar como las pruebas

[filas,columnas]=size(A);
B = ['b','r','g'];
C = ['X','Y','Z'];
h=figure('name','FUERZAS G OBTENIDAS EN EL ACELERÓMETRO ');
% 'Color','white',

    pause(0.2) % Pausa para dejar hacer al código y maximizar
    jf=get(h,'JavaFrame');
    set(jf,'Maximized',1);
    pause(0.2) % Tiempo para que asiente la ventana y comience a funcionar la función

for i=1:columnas

    % REPRESENTACION DE LA FUNCION COMPLETA

    c=102000;
    d=152000;

    x=A(:,i);
    x=(x*3.9/1000);

    x1=x(c:d);
    t=c:d;

    subplot(3,3,i)
    plot(t,x1,B(i))
    etiq=gca;
```

```

title(['FUERZAS G SOBRE EJE ',C(i)],'fontsize',9)
xlabel('Muestras','fontweight','b','fontsize',9)
ylabel('Fuerza G','fontweight','b','fontsize',9)
xlim([100000 154000])

% DIAGRAMAS DE DISPERSION Y FUNCION DE PROBABILIDAD SEGUN VELOCIDADES Y PARTE DEL CUERPO

% 10 km/h

for p=1:3

    if p==1
        c=104000; d=112500;

        picos10=[];
        picosneg10=[];
        vector10=[];
        vectorneg10=[];

        x1=x(c:d);

        med=mean(x1);
        desv=std(x1);

        hold on
        t1=[c,d];

        desvmedmas=med+desv;
        desvmedmenos=med-desv;

        mediavector=[med,med];
        desvector=[desvmedmas,desvmedmas];
        desvector2=[desvmedmenos,desvmedmenos];

        for j=c:d

            if (x(j)>x(j-1)) && (x(j)>x(j+1)) %&& x(j)>desvmedmas
                picos10=[picos10;x(j)];
                vector10=[vector10;j];
            end

            if (x(j)<x(j-1)) && (x(j)<x(j+1)) %&& x(j)<desvmedmenos
                picosneg10=[picosneg10;x(j)];
                vectorneg10=[vectorneg10;j];
            end

        end

        subplot(3,3,i+3)
        hold on
        scatter(vector10,picos10,3,B(i))
        etiq2=gca;
        box on
        xlim([100000 154000])
        hold on
        scatter(vectorneg10,picosneg10,3,B(i))

        mediapicospos=mean(picos10);
        mediapicospos=[mediapicospos,mediapicospos];

        mediapicosneg=mean(picosneg10);
        mediapicosneg=[mediapicosneg,mediapicosneg];

        plot(t1,mediapicospos,'k')
        plot(t1,mediapicosneg,'k')

```

```

title(['DIAGRAMA DE DISPERSION EJE ',C(i)],'fontsize',9)
xlabel('Muestras','fontweight','b','fontsize',9)
ylabel('Fuerza G','fontweight','b','fontsize',9)
hold on

subplot(3,3,i+6)
ejex=x1;
hold on
h_(p)=abc(ejex,p);

elseif p==2
    c=121000; d=132500;

picos10=[];
picosneg10=[];
vector10=[];
vectorneg10=[];

x2=x(c:d);

med=mean(x2);
desv=std(x2);

hold on
t1=[c,d];

desvmedmas=med+desv;
desvmedmenos=med-desv;

mediavector=[med,med];
desvector=[desvmedmas,desvmedmas];
desvector2=[desvmedmenos,desvmedmenos];

for j=c:d

    if (x(j)>x(j-1)) && (x(j)>x(j+1)) && x(j)>desvmedmas
        picos10=[picos10;x(j)];
        vector10=[vector10;j];
    end

    if (x(j)<x(j-1)) && (x(j)<x(j+1)) && x(j)<desvmedmenos
        picosneg10=[picosneg10;x(j)];
        vectorneg10=[vectorneg10;j];
    end

end
cambiar tamaño de fuente latex
subplot(3,3,i+3)
hold on
scatter(vector10,picos10,3,B(i))
hold on
scatter(vectorneg10,picosneg10,3,B(i))

mediapicospos=mean(picos10);
mediapicospos=[mediapicospos,mediapicospos];

mediapicosneg=mean(picosneg10);
mediapicosneg=[mediapicosneg,mediapicosneg];

plot(t1,mediapicospos,'k')
plot(t1,mediapicosneg,'k')

title(['DIAGRAMA DE DISPERSION EJE ',C(i)],'fontsize',8)
xlabel('Muestras','fontweight','b','fontsize',9)
ylabel('Fuerza G','fontweight','b','fontsize',9)
hold on

```

```

subplot(3,3,i+6)
xlabel('Muestras','fontweight','b','fontsize',9)
ylabel('Fuerza G','fontweight','b','fontsize',9)
ejex=x2;
hold on
h_(p)=abc(ejex,p);

else
    c=140000; d=149500;

picos10=[];
picosneg10=[];
vector10=[];
vectorneg10=[];

x3=x(c:d);

med=mean(x3);
desv=std(x3);

hold on
t1=[c,d];

desvmedmas=med+desv;
desvmedmenos=med-desv;

mediavector=[med,med];
desvector=[desvmedmas,desvmedmas];
desvector2=[desvmedmenos,desvmedmenos];

for j=c:d

    if (x(j)>x(j-1)) && (x(j)>x(j+1)) && x(j)>desvmedmas
        picos10=[picos10;x(j)];
        vector10=[vector10;j];
    end

    if (x(j)<x(j-1)) && (x(j)<x(j+1)) && x(j)<desvmedmenos
        picosneg10=[picosneg10;x(j)];
        vectorneg10=[vectorneg10;j];
    end

end

subplot(3,3,i+3)
hold on
scatter(vector10,picos10,3,B(i))
hold on
scatter(vectorneg10,picosneg10,3,B(i))

mediapicospos=mean(picos10);
mediapicospos=[mediapicospos,mediapicospos];

mediapicosneg=mean(picosneg10);
mediapicosneg=[mediapicosneg,mediapicosneg];

plot(t1,mediapicospos,'k')
plot(t1,mediapicosneg,'k')

title(['DIAGRAMA DE DISPERSION EJE ',C(i)],'fontsize',9)
xlabel('Muestras','fontweight','b','fontsize',9)
ylabel('Fuerza G','fontweight','b','fontsize',9)
hold on

subplot(3,3,i+6)

```

```

hold on
ejex=x3;
hold on
h_(p)=abc(ejex,p);

end

end

subplot(3,3,i+6);
title(['FUNCION DE PROBABILIDAD EN EJE ',C(i)],'fontsize',9)
ax1=gca;
hold on
set(ax1,'Box','off','XLim',[-6 10])
pause(0.1)

ej1=axes('Position',get(ax1,'Position'),'Color','none');
ej2=axes('Position',get(ax1,'Position'),'Color','none');
ej3=axes('Position',get(ax1,'Position'),'Color','none');

h1=boxplot(ej1,x1,'widths',0.6);
h2=boxplot(ej2,x2,'widths',0.6);
h3=boxplot(ej3,x3,'widths',0.6);

set(h1(7,:), 'Visible', 'off')
set(h2(7,:), 'Visible', 'off')
set(h3(7,:), 'Visible', 'off')

h1 = get(get(ej1,'child'),'child');
h2 = get(get(ej2,'child'),'child');
h3 = get(get(ej3,'child'),'child');
group_name_handle1 = findobj(h1,'type','text');
group_name_handle2 = findobj(h2,'type','text');
group_name_handle3 = findobj(h3,'type','text');

for j=1:length(group_name_handle1)
    text(1,0,get(group_name_handle1(j),'string'),'color','r','position',[j 48 0],'Visible','off');
end
delete(group_name_handle1)

for j=1:length(group_name_handle2)
    text(1,0,get(group_name_handle2(j),'string'),'color','r','position',[j 48 0],'Visible','off');
end
delete(group_name_handle2)

for j=1:length(group_name_handle3)
    text(1,0,get(group_name_handle3(j),'string'),'color','r','position',[j 48 0],'Visible','off');
end
delete(group_name_handle3)

set(ej1,'Box','off','XTick',[],'YTick',[],'Visible','off','XLim',[-7 4],'YLim',[-3 8])
set(ej2,'Box','off','XTick',[],'YTick',[],'Visible','off','XLim',[-8 3],'YLim',[-3 8])
set(ej3,'Box','off','XTick',[],'YTick',[],'Visible','off','XLim',[-9 2],'YLim',[-3 8])

leyenda=legend(h_,'String',{'Brazo','Cadera','Pierna'],'FontSize',7);
set(leyenda,'Box','off','Location','NorthWest')
text(-1.1,6.2,'1      2      3','FontWeight','bold','FontSize',9);

% Establezco los ejes coordenados de cada representación en un eje auxiliar

ylimts1 = get(ej1,'YLim');
ylimts2 = get(ej2,'YLim');
ylimts3 = get(ej3,'YLim');

```

```

limitmax=[ylimits1(2);ylimits2(2);ylimits3(2)];
limitmin=[ylimits1(1);ylimits2(1);ylimits3(1)];

limitemayor=max(limitmax);
limitemenor=min(limitmin);

ax=axes('Position',get(ax1,'Position'),'XAxisLocation','top','YAxisLocation','right','Color','none');
set(ax,'YLim',[limitemenor limitemayor],'Color','none','XTick',[]);
axis normal

xlabel(ax1,'Fuerzas G','fontWeight','b','fontSize',9)
ylabel(ax1,'Densidad','fontWeight','b','fontSize',9)

subplot(3,3,i)
axes('Position',get(etiq,'Position'),'Color','none','XTick',[],'YTick',[],'Visible','off');
text(0.15,0.9,'1','FontWeight','bold','FontSize',9);
text(0.5,0.9,'2','FontWeight','bold','FontSize',9);
text(0.85,0.9,'3','FontWeight','bold','FontSize',9);

subplot(3,3,i+3)
axes('Position',get(etiq2,'Position'),'Color','none','XTick',[],'YTick',[],'Visible','off');
text(0.15,0.9,'1','FontWeight','bold','FontSize',9);
text(0.5,0.9,'2','FontWeight','bold','FontSize',9);
text(0.85,0.9,'3','FontWeight','bold','FontSize',9);

end
refresh
end

```

8.10. ANEXO E

FIRMWARE DE CONFIGURACIÓN DEL ACELERÓMETRO MODIFICADO

Modificación personal del código del firmware de fábrica para lograr el funcionamiento en las condiciones deseadas

```
#include <string.h>
#include "XL345.h"
#include "xl345_io.h"
#include "dev_board.h"
#include "ADuC7024_RV.h"
#include "DAC.h"
#include <stdio.h>
#include <stdlib.h>

#include "debug_printf.h"
#include "efs.h"
#include "ls.h"

typedef int bool;
#define true 1
#define false 0

/*-----*/
DEFINES to be used to control various program options
/*-----*/

// #define FS_DEBUG_DEFINE // Define FS_DEBUG_DEFINE to include debug_printf's in error free FS main flow
#define DEBUG // Define DEBUG to include debug_printf's in the main project
// Note that DEBUG can also be defined in config.h for the efs12 library
// Project will need to be recompiled
// #define cyc_ctr_timer1 // Define cyc_ctr_timer1 to count cycles of an operation
// Note that this is instrumented code and you will need to enable timer1 before
// and disable after the event
#define one_min_timer2 // Use for HMS printf's and timing things in minute intervals.
// #define memory_test // Define to include memory/buffer test at program start
// #define batch_recording // Define to record to SD flash for 800Hz sensor rates (only with GNU)
// #define read_entrycount // Defining with increase interrupt time - do not define to keep interrupt as short as possible
// #define read_outrun_status // Define if you want to keep up with the sensor overrun status
// #define sleep_mode // Puts ADuC into sleep mode while waiting for next interrupt
#define power_monitor // Define if you want to trip on 2.79V and halt the processor - writes out message to flash
/*-----*/
DEFINE values
/*-----*/

#define DEFAULT_LOG_DIRECTORY "xl345"
#define DEFAULT_LOG_FILENAME "data"
// #define below - batch_recording_time // Define batch record time in seconds - defined in #if-elif below
// #define below - SENSORBUFFERSIZE // max is 848 ==> 848 * 6 Bytes ~ 6KBytes which is 3/4's of the SRAM!
// Must leave room for FS stack at top of SRAM
// At 992 data corruption happens because the FS stack (at the top of SRAM)
// writes into this buffer

#define SENSORDATASIZE 3
#define UINT_MAX 0xFFFFFFFF;

#define RATE_200 // Define only one rate from 6.25Hz to 800Hz

#if defined RATE_6_25
#define XL345_RATE XL345_RATE_6_25
#define #define SENSORBUFFERSIZE 62 // The smallest SENSORBUFFERSIZE is 2 * 31 = 62
#define XL345_PWR_MODE XL345_LOW_POWER // ADXL can be in low power mode from 6.25 to 400Hz
#define sleep_mode // ADuC put in sleep mode in Sensorloop (when data processing is complete)
#elif defined RATE_12_5
#define XL345_RATE XL345_RATE_12_5
#define #define SENSORBUFFERSIZE 62
#define XL345_PWR_MODE XL345_LOW_POWER
#define sleep_mode
#elif defined RATE_25
#define XL345_RATE XL345_RATE_25
#define #define SENSORBUFFERSIZE 62
```

```

        #define XL345_PWR_MODE XL345_LOW_POWER
        #define sleep_mode
    #elif defined RATE_50
        #define XL345_RATE XL345_RATE_50
        #define #define SENSORBUFFERSIZE 62
        #define XL345_PWR_MODE XL345_LOW_POWER
        #define sleep_mode
    #elif defined RATE_100
        #define XL345_RATE XL345_RATE_100
        #define SENSORBUFFERSIZE 200 // 100 samp/sec * .32msec --> must be >= 32 * 2 = 64
        #define XL345_PWR_MODE XL345_LOW_POWER
        #define sleep_mode
    #elif defined RATE_200
        #define XL345_RATE XL345_RATE_200
        #define SENSORBUFFERSIZE 400 // 200 samp/sec * .32msec --> must be >= 64 * 2 = 128
        #define XL345_PWR_MODE XL345_LOW_POWER
        #define sleep_mode
    #elif defined RATE_400
        #define XL345_RATE XL345_RATE_400
        #define SENSORBUFFERSIZE 600 // 400 samp/sec * .32msec --> must be >= 128 * 2 = 256
        #define XL345_PWR_MODE XL345_LOW_NOISE
        // #define sleep_mode // Don't define sleep mode - starting to lose data because the time
        // it takes to come out of sleep
    #elif defined RATE_800
        #define run_unmount 60 // Run the unmount() command every x seconds
        // (a 32-bit timer can do 1.7 minutes before rolling over)
        #define batch_recording_time_timer1 (41779200 * run_unmount) // Using the Cyc Counter to count seconds
        #define batch_recording // Batch recording must be used as 800Hz.
        #define batch_recording_time 59 // Set Batch recording time in minutes
        #define XL345_RATE XL345_RATE_800
        #define SENSORBUFFERSIZE 800 // 800 samp/sec * .32msec --> must be >= 256 * 2 = 512
        #define XL345_PWR_MODE XL345_LOW_NOISE
        #define fast_file_printf // The longer the printf line to the file, the slower the fwrite
        // Don't even convert to decimal! Minimize the printing.
        #undef DEBUG // Get rid of EVERYTHING that might eat up MIPs (all Debug printf's)
        #undef power_monitor // Can't even have the power monitor check - close to edge of not losing data.
        #undef one_min_timer2 // Compile with every optimization including "Link Time" optimization (removes debug capabilities),
        // You also want to compile in "thumb" mode with "Link Time" optimizations on
        // fwrite is the problem! I can reduce the amount of data that we write out and it helps!
    #endif

    #ifndef DEBUG
        #define DEBUG_PRINTF ; // That is the equivalent of a NOP in place of the printf
    #else
        #define DEBUG_PRINTF debug_printf
    #endif

    typedef struct _FilesystemState {
        EmbeddedFileSystem efs;
        EmbeddedFile logFile;
        char logFileName[100];
    } FilesystemState;

    typedef struct _SensorState {
        volatile int sensorDataIndex;
        // int sensorData[SENSORBUFFERSIZE * SENSORDATASIZE]; // old was an integer (wasted half of the SRAM space)
        volatile short sensorData[SENSORBUFFERSIZE * SENSORDATASIZE]; // new is a 16-bit value for each XYZ sensor value
        //int *sensorData;
    } SensorState;

    SensorState sensorState;

    int lastTime;
    char line[50] = {0};
    int i = 0;
    int t = 0;
    int result = 0;

    unsigned char interruptSource;
    //unsigned char seconds = 0;
    //unsigned char minutes = 0;
    //unsigned char hours = 0;
    //unsigned char hundredth_secs = 0;

    //float hundredth;
    //float mult=0.0078125;

    int Initialize();
    int InitFs(FilesystemState *fileSystem);
    int SensorLoop(FilesystemState *fileSystem);
    bool overrun_flag = 0;
    volatile bool state = 0;
    int num_of_int = 0;
    volatile int timer2_val;

    volatile void redLight(bool state);
    volatile void greenLight(bool state);

```

```

void i2cinit(void);

int main() {
    if (Initialize()) {
        FilesystemState fs;

// Define memory_test if you want to write something into sensorData to find a key location in SRAM
#ifdef memory_test
        sensorState.sensorData[SENSORBUFFERSIZE * SENSORDATASIZE - 1] = 0xDEADBEEF;
        DEBUG_PRINTF("You wrote: 0x%08X\n", sensorState.sensorData[SENSORBUFFERSIZE * SENSORDATASIZE - 1]);
        DEBUG_PRINTF("Line 153: 0x%08X, 0x%08X, 0x%08X\n", sensorState.sensorData[153*SENSORDATASIZE],
            sensorState.sensorData[153*SENSORDATASIZE+1], sensorState.sensorData[153*SENSORDATASIZE+2]);
        DEBUG_PRINTF("Line 154: 0x%08X, 0x%08X, 0x%08X\n", sensorState.sensorData[154*SENSORDATASIZE],
            sensorState.sensorData[154*SENSORDATASIZE+1], sensorState.sensorData[154*SENSORDATASIZE+2]);
        DEBUG_PRINTF("Line 155: 0x%08X, 0x%08X, 0x%08X\n", sensorState.sensorData[155*SENSORDATASIZE],
            sensorState.sensorData[155*SENSORDATASIZE+1], sensorState.sensorData[155*SENSORDATASIZE+2]);
#endif

        DEBUG_PRINTF(">>> Initialized System ok\n");
        if (InitFs(&fs)) {
            DEBUG_PRINTF(">>> Initialized Filesystem ok\n");
            SensorLoop(&fs); // Go to endless filesystem loop here
        } else {
            DEBUG_PRINTF("!!! Initialize Filesystem failed!\n");
            while (true);
        }

    } else {
        DEBUG_PRINTF("!!! Initialize failed!\n");
    }

    greenLight(false);
    redLight(true);
    while (true);
    return 0;
}

void IRQ_Handler(void) __irq
{
    unsigned char count = 0; // (JWS) old was an int
    unsigned char entryCount = 0;
    unsigned char databuf[8] = {0};
    int thisTime = T2VAL;
    short x1, y1, z1;

    num_of_int++; // count number of times the sensorloop is interrupted by this ISR
    /* Only define this code to count cycles of ISR
    TLLD = 0;
    TLCON = 0;
    TLCON = 0x000211c0; // Enable T1, internal clock, count up, periodic, clock/1, event 1 = timer 1
    */

#ifdef read_overrun_status
    xl345Read(1, XL345_INT_SOURCE, &interruptSource); // Note that overruns only seem to happen at startup
// due to not reading sensor after turning on interrupts
    if (interruptSource & XL345_OVERRUN) {
        DEBUG_PRINTF("Overrun @ %d\n", T2VAL);
        overrun_flag = 1;
    }

    if (interruptSource & XL345_WATERMARK) {
#endif

#ifdef read_entrycount
        xl345Read(1, XL345_FIFO_STATUS, &entryCount); // read the number of values in the FIFO of the ADXL sensor
        entryCount &= 0x3f; // get the lower 6 bits
#endif

        entryCount = 31; // fix the entry count to constant as reading from the sensor can always fall
        // between samples and give a +/- 1 sample error.

        for (count = 0; count < entryCount; count++) {
            //DEBUG_PRINTF("data[%d]=\n", sensorState.sensorDataIndex);
            xl345Read(6, XL345_DATA0, databuf);
            x1 = (short) (databuf[0] | databuf[1]<<8); // (JWS) old cast was an int
            y1 = (short) (databuf[2] | databuf[3]<<8);
            z1 = (short) (databuf[4] | databuf[5]<<8);
            //*****if (sensorState.sensorDataIndex <= SENSORBUFFERSIZE) { // Do inside of the sensorloop to save memory
            sensorState.sensorData[sensorState.sensorDataIndex * SENSORDATASIZE]

```

```

        = lastTime + count * (thisTime - lastTime) / (entryCount-1);
//      sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)]
        = count; // (JWS) remove time from buffer
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)] = x1;
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)+1] = y1;
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)+2] = z1;
//      DEBUG_PRINTF("sensorData[%d] = {t: 0x%X, x: %d, y: %d, z: %d}\n", // Never do a printf
        // inside the error free ISR!
        sensorState.sensorDataIndex,
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)],
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)+1],
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)+2],
        sensorState.sensorData[(sensorState.sensorDataIndex * SENSORDATASIZE)+3]); //
        sensorState.sensorDataIndex += 1;

        if (sensorState.sensorDataIndex >= SENSORBUFFERSIZE) {
            //DEBUG_PRINTF("%d >= %d, resetting to 0\n", sensorState.sensorDataIndex, SENSORBUFFERSIZE);
            sensorState.sensorDataIndex = 0; //
        }
        //} else {
        //      DEBUG_PRINTF("sensorState.sensorData = %x\n", sensorState.sensorData);
        //*****} //
        //DEBUG_PRINTF("entry[%d] = {%d, %d, %d}\n", count, x1, y1, z1);
    }
    lastTime = thisTime;
#endif read_outrun_status
    } else {
        DEBUG_PRINTF("Unknown interrupt source: %d\n", interruptSource);
    }
#endif

}

void SWI_Handler(void) __irq
{
    DEBUG_PRINTF("MYSWI\n");
    greenLight(false);
    redLight(true);
    while (true);
}

void FIQ_Handler(void) __irq
{
    DEBUG_PRINTF("MYFIQ\n");
    greenLight(false);
    redLight(true);
    while (true);
}

void Undef_Handler(void) __irq
{
    DEBUG_PRINTF("MYUNDEF\n");
    greenLight(false);
    redLight(true);
    while (true);
}

void DAbt_Handler(void) __irq
{
    DEBUG_PRINTF("MYDABORT\n");
    greenLight(false);
    redLight(true);
    while (true);
}

void PAbt_Handler(void) __irq
{
    DEBUG_PRINTF("MYPABORT\n");
    greenLight(false);
    redLight(true);
    while (true);
}

void delay (int length){
    while (length >=0)
        length--;
}

int Initialize() {
    unsigned char buffer[8];

//      IRQ = myisr; // Specify Interrupt Service Routine
//      FIQ = myfiq;
//      UNDEF = myundef;
//      DABORT = mydabort;
//      PABORT = mypabort;

```

```

GP3CON = GP3CONGPIOVAL; // configure GPIO pins
GP3PAR = ALLPUPOFF; // turn off pull ups
GP3DAT = GP3OFF; // set LED pins to output and PWM to GPIOOFF
GP4CON = GP4CONVAL;
// GP4DAT = GP4OUT;
GP4DAT = 0xFF000000;

i2cinit(); // set up the I2C bus

UART_Config();

/* soft reset for safety */
buffer[0] = XL345_RESERVED1; /* register address */
buffer[1] = XL345_SOFT_RESET;
xl345Write(2,buffer);

delay(2000); // allow delay after a soft reset

/*-----*/
TAP Configuration
/*-----*/

/* set up a buffer with all the initialization for tap */
buffer[0] = XL345_THRESH_TAP; /* register address */
buffer[1] = 80; /* THRESH_TAP = 5 Gee (1 lsb = 1/16 gee) */
xl345Write(2,buffer);

buffer[0] = XL345_DUR; /* register address */
buffer[1] = 13; /* DUR = 8ms 0.6125ms/lsb */
buffer[2] = 80; /* LATENT = 100 ms 1.25ms/lsb */
buffer[3] = 240; /* WINDOW 300ms 1.25ms/lsb */
xl345Write(4,buffer);

buffer[0] = XL345_TAP_AXES; /* register address */
buffer[1] = XL345_TAP_Z_ENABLE | XL345_TAP_Y_ENABLE
| XL345_TAP_X_ENABLE /*| XL345_TAP_SUPPRESS*/;
xl345Write(2,buffer);

buffer[0] = XL345_DATA_FORMAT;
buffer[1] = XL345_RANGE_16G | XL345_FULL_RESOLUTION;
xl345Write(2,buffer);

/*-----*/
activity - inactivity
/*-----*/

/* set up a buffer with all the initialization for activity and inactivity */
buffer[0] = XL345_THRESH_ACT; /* register address */
buffer[1] = 80; /* THRESH_ACT = 80/16 = 5 Gee (1 lsb = 1/16 gee) */
buffer[2] = 4; /* THRESH_INACT = 14/16 .25 Gee (1 lsb = 1/16 gee) */
buffer[3] = 5; /* TIME_INACT - 5 seconds 2 minutes*/
buffer[4] = XL345_ACT_DC | XL345_ACT_X_ENABLE | XL345_ACT_Y_ENABLE | XL345_ACT_Z_ENABLE
| XL345_INACT_AC | XL345_INACT_X_ENABLE
| XL345_INACT_Y_ENABLE | XL345_INACT_Z_ENABLE; /* ACT_INACT_CTL */
xl345Write(5,buffer);

/*-----*/
Power, bandwidth-rate, interrupt enabling
/*-----*/

/* set up a buffer with all the initialization for power*/
buffer[0] = XL345_BW_RATE; /* register address */
buffer[1] = XL345_RATE | XL345_PWR_MODE; /* BW rate and power mode */
buffer[2] = XL345_WAKEUP_8HZ | XL345_MEASURE; /* POWER_CTL */
xl345Write(3,buffer);

// set the FIFO control
buffer[0] = XL345_FIFO_CTL;
buffer[1] = XL345_FIFO_MODE_FIFO | 0 | 31; // set FIFO mode, link to INT1, number of samples
// buffer[1] = XL345_FIFO_MODE_STREAM | 0 | 31; // set for STREAM mode.
xl345Write(2, buffer);

// turn on the watermark interrupt and set the watermark interrupt to int1
buffer[0] = XL345_INT_ENABLE;
buffer[1] = XL345_WATERMARK | XL345_OVERRUN; // enable D1
buffer[2] = 0; // send all interrupts to INT1
xl345Write(3, buffer);

GP4CLR = 0xFF<<16;

// Setup timer 2 to use for timing
// T2LD = 0; // Timer 2 load register = 0
// T2CON = (1<<6) | (1<<7) | (1<<8) | (1<<10) | 0x4; // 32-bits Timer 2 control register
// //Enabled, internal osc., periodic mode, count up, Binary
// T2CON = 0x000007c0; // Enabled, core clock, count up, periodic (resets to zero at max), source clock/1
// T2CON = 0x000005cf; // Enabled, 32.768 osc, count up, periodic (resets to zero at max), clock/32768

PSMCON = 0x0002; // Enable Power monitor with trip at 2.79V

```

```

        return true;
    }

int data_log(FilesystemState *fileState, char *line) {
    int result = 0;
    result = file_write(&(fileState->logFile), strlen(line), line);
    return true;
}

/*
int batch_log(FilesystemState *fileState, int start, int end) {

//      T2LD = 0x0;
//      T2CON = 0x000007c0; // Enabled, core clock, count up, periodic (resets to zero at max), source clock/1

#ifdef batch_recording

    result = efs_init(&(fileState->efs), 0);

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
        return false;
    }

    result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")
                    = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
        return false;
    }

#endif

    for (i = start; i < end; i++) {

#ifdef batch_recording
        snprintf(line, 49, "%x,\t\t%d,\t\t%d,\t\t%d\n",
                snprintf(line, 49, "%d,\t%02d:%02d:%02d:%03d,\t\t%d,\t\t%d,\t\t%d\n",
                i, ((T2VAL>>24)&255, (T2VAL>>16)&255, (T2VAL>>8)&255, (T2VAL&255),
                T2VAL,
#endif

#ifdef batch_recording
        snprintf(line, 49, "%d,\t%02d:%02d:%02d:%03d,\t\t%d,\t\t%d,\t\t%d\n",
        i, (T2VAL>>24)&255, (T2VAL>>16)&255, (T2VAL>>8)&255, (T2VAL&255),
#endif

        sensorState.sensorData[i * SENSORDATASIZE],
        sensorState.sensorData[i * SENSORDATASIZE + 1],
        sensorState.sensorData[i * SENSORDATASIZE + 2]);
//      result = data_log(fileState, line);

        result = file_write(&(fileState->logFile), strlen(line), line);

        if (!result) {
            DEBUG_PRINTF(">>> Error in writing to data log from batch_log\n");
        }
    }

#ifdef batch_recording
    if (((T2VAL>>16)&255) >= batch_recording_time ) { // T2VAL is counting up in minutes

//      Don't reset timer back to zero
//      T2LD = 0; // Timer 2 load register = 0
//      T2CON = 0; // reset Timer 2 back to zero seconds
//      T2CON = 0x000005cf; // Enabled, 32.768 osc, count up, periodic (resets to zero at max), clock/3276
//      T2CON = 0x000005e8; // Enabled, 32.768 osc, count up, HR:MN:SC:HH, periodic (resets to zero at max), clock/256

        IRQCLR = (1<<15); // turn off ADXL interrupts for batch mode only

#endif

    result = file_fclose(&(fileState->logFile));

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on file_fclose(\"%s\")
                    = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
        return false;
    }
    result = fs_umount(&(fileState->efs.myFs));

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on fs_umount()
                    = %d\n", __FILE__, __LINE__, result);
        return false;
    }
}

```

```

        IRQEN = (1<<15);                // Reenable ADXL interrupts

#ifdef batch_recording

    result = efs_init(&(fileState->efs), 0);                // Reinit file system for next batch recording time

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
        return false;
    }

    result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")
                    = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
        return false;
    }

} // bracket for if statement above

#endif

return true;
}
*/

int InitFs(FilesystemState *fileState) {
    char logDirectory[100] = {0};
    int logCount = 0;
    int result = 0;

    // log directory
    snprintf(logDirectory,
             99,
             "%s",
             DEFAULT_LOG_DIRECTORY);

    // log filename
    snprintf(fileState->logFileName,
             99,
             "%s/%s%04d.txt",
             DEFAULT_LOG_DIRECTORY,
             DEFAULT_LOG_FILENAME,
             logCount);

    DEBUG_PRINTF("%s:%d InitFs(): going to log to %s\n", __FILE__, __LINE__, fileState->logFileName);

    // initialize filesystem
    result = efs_init(&(fileState->efs), 0);
    if (result) {
        DEBUG_PRINTF("%s:%d InitFs(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
        return false;
    }
    // make the directory if needed
    result = mkdir(&(fileState->efs.myFs), logDirectory);
    DEBUG_PRINTF("%s:%d InitFs(): result of mkdir(%s) = %d\n", __FILE__, __LINE__, logDirectory, result);
    // open the log file for writing
    result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');
    if (result) {
        DEBUG_PRINTF("%s:%d InitFs(): failed on file_fopen(\"%s\")
                    = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
        return false;
    }
}

// result = data_log(fileState, "t,x,y,z\n");

#ifdef fast_file_printf
    result = file_write(&(fileState->logFile), strlen("ndx:\tHR:MN:SC:1/128s:\tx:\ty:\tz:\n"),
                      "ndx:\tHR:MN:SC:1/128s:\tx:\ty:\tz:\n");
#endif

#ifdef fast_file_printf
    result = file_write(&(fileState->logFile), strlen("HR:MN:SC:\tx:\ty:\tz:\n"), "HR:MN:SC:\tx:\ty:\tz:\n");
#endif

if (!result) {
    DEBUG_PRINTF("%s:%d InitFs(): failed on data_log(\"%s\")
                = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}
DEBUG_PRINTF(">>> wrote header to log file\n");

result = file_fclose(&(fileState->logFile));
if (result) {
    DEBUG_PRINTF("%s:%d InitFs(): failed on file_fclose(\"%s\")
                = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}

```

```

    }

    result = fs_umount(&(fileState->efs.myFs));
    if (result) {
        DEBUG_PRINTF("%s:%d init_file_system(): failed on fs_umount()
                    = %d\n", __FILE__, __LINE__, result);
        return false;
    }

    DEBUG_PRINTF("%s:%d InitFs() done\n", __FILE__, __LINE__);
    return true;
}

int SensorLoop(FilesystemState *fileState) {

//      unsigned char buffer[8];
//      int half = 0;

#ifdef batch_recording
    int result = 0;
#endif

//      int t = 0;
//      int midway = SENSORBUFFERSIZE / 2;
//      DEBUG_PRINTF(">>> Starting Sensor loop\n");
//      sensorState.sensorDataIndex = 0;

    T2LD = 0; // Timer 2 load register = 0
    T2CON = 0;
//      T2CON = 0x000005cf; // Enabled, 32.768 osc, count up, periodic (resets to zero at max), clock/32768
//      T2CON = 0x000005e8; // Enabled, 32.768 osc, count up, HR:MN:SC:HH, periodic (resets to zero at max), clock/256

    x1345Read(1, XL345_INT_SOURCE, &interruptSource); // Read ADXL to clear first overrun

    IRQEN = (1<<15); // Enable Interrupt here for the ADXL = IRQ0 enabled

    greenLight(true); // Turn on green LED

    lastTime = T2VAL;

#ifdef batch_recording // Define if you want to capture x minutes of data at 800Hz which is then dumped to SD flash
// Note that it takes ~320ms for the FS to do an unmount and ADXL data will lost during this time

    result = efs_init(&(fileState->efs), 0);

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
        return false;
    }

    result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

    if (result) {
        DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")
                    = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
        return false;
    }

}

#endif

// #ifdef cyc_ctr_timer1
//      T1LD = 0;
//      T1CON = 0;
//      T1CON = 0x000211c0; // Enable T1, internal clock, count up, periodic, clock/1, event 1 = timer 1
// #endif

    while (true) { // Ping pong buffer while loop

//      IRQCLR = (1<<15); // BE CAREFUL HERE - can get in endless loop - ONLY uncomment to test the sleep mode
#ifdef sleep_mode
        POWKEY1 = 0x01; // Puts Micro into sleep mode
// only want to do this after everything is complete while we wait for next interrupt
        POWCON = 0x30;
        POWKEY2 = 0xF4;
#endif

#ifdef power_monitor
        if ((PSMCON & 0x0008) == 0) {
            IRQCLR = (1<<15); // turn off ADXL interrupts
            DEBUG_PRINTF("Battery below 2.79 volts!\n");

            efs_init(&(fileState->efs), 0);
            file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');
            file_write(&(fileState->logFile), strlen("Battery below 2.79 volts!\n"), "Battery below 2.79 volts!\n");
            file_fclose(&(fileState->logFile));
            fs_umount(&(fileState->efs.myFs));

            greenLight(false);
            redLight(true);
        }
#endif
    }
}

```

```

        POWKEY1 = 0x01;           // Puts Micro into stop mode - only want to do this if battery is really low
        POWCON = 0x40;           //
        POWKEY2 = 0xF4;

        while (true);           // Code will never get here!
    }
#endif

#ifdef one_min_timer2

    timer2_val = T2VAL;

    if (((timer2_val&0x00ff0000)>>16) >= 30) {           // set breakpoint at while(1) to halt after x minutes of data capture
                                                         // used to ensure no loss of data
        while (1);
        // line percentage for no loss of data = actual lines / (x * 60 * XL345_RATE_xxx) = .95 or 95% due to inaccurate timer
    }

#endif

#ifdef cyc_ctr_timer1

    if (T1VAL >= 208896000) {
        while (1);           // Use as a core cycle counter
                             // 208896000 = 5 sec = 5 * 41,779,200 Mhz
    }

#endif

    if ((half == 0) && (sensorState.sensorDataIndex >= midway)) {

        //      batch_log(fileState, 0, midway);           // where we write out lower half while the upper is filled

#ifdef batch_recording

        result = efs_init(&(fileState->efs), 0);

        if (result) {
            DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
            return false;
        }

        result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

        if (result) {
            DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")
                         = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
            return false;
        }

#endif

        #endif

        for (i = 0; i < midway; i++) {

#ifdef fast_file_printf
            snprintf(line, 49, "%02X%02X%02X, %04d,%04d,%04d\n",
                    (T2VAL>>24)&255, (T2VAL>>16)&255, (T2VAL>>8)&255,
                    #endif

#ifdef fast_file_printf
            snprintf(line, 49, "%d,\t%02d:%02d:%02d:%03d,\t%d,\t%d,\t%d\n",
                    i, (T2VAL>>24)&255, (T2VAL>>16)&255, (T2VAL>>8)&255, (T2VAL&255),
                    #endif
            sensorState.sensorData[i * SENSORDATASIZE],
            sensorState.sensorData[i * SENSORDATASIZE + 1],
            sensorState.sensorData[i * SENSORDATASIZE + 2]);

            result = file_write(&(fileState->logFile), strlen(line), line);

            if (!result) {
                DEBUG_PRINTF(">>> Error in writing to data log from batch_log\n");
            }
        }

#ifdef batch_recording
        //if (((T2VAL>>16)&255) >= batch_recording_time ) {           // T2VAL is counting up in minutes

        if (T1VAL >= batch_recording_time_timer1 ) {           // T1VAL is counting up to 1 minute
            /*
            // Reset timer back to zero
            T2LD = 0;           // Timer 2 load register = 0
            T2CON = 0;           // reset Timer 2 back to zero seconds
            //T2CON = 0x000005cf;
            // Enabled, 32.768 osc, count up, periodic (resets to zero at max), clock/3276
            T2CON = 0x000005e8;
            // Enabled, 32.768 osc, count up, HR:MN:SC:HH, periodic (resets to zero at max), clock/256
            */

            // Reset timer 1 back to zero
            T1LD = 0;

```

```

T1CON = 0;
T1CON = 0x000211c0; // Enable T1, internal clock, count up, periodic, clock/1, event 1 = timer 1

IRQCLR = (1<<15); // turn off ADXL interrupts for batch mode only

#endif

result = file_fclose(&(fileState->logFile));

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on file_fclose(\"%s\")
                = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}
result = fs_umount(&(fileState->efs.myFs));

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on fs_umount() = %d\n", __FILE__, __LINE__, result);
    return false;
}

IRQEN = (1<<15); // Reenable ADXL interrupts

#ifdef batch_recording

result = efs_init(&(fileState->efs), 0); // Reinit file system for next batch recording time

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
    return false;
}

result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")
                = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}

} // bracket for if statement above

#endif

//DEBUG_PRINTF("cycles = %d\n", T1VAL);
num_of_int = 0;
half = 1;
} else if ((half == 1) && (sensorState.sensorDataIndex < midway)) {

//batch_log(fileState, midway, SENSORBUFFERSIZE);
// where we write out the upper half while the lower is filled
#ifdef batch_recording

result = efs_init(&(fileState->efs), 0);

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
    return false;
}

result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")
                = %d\n", __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}

}

#endif

for (i = midway; i < SENSORBUFFERSIZE; i++) {

#ifdef fast_file_printf
    snprintf(line, 49, "%02X%02X%02X, %04d,%04d,%04d\n",
             (T2VAL>>24)&255, (T2VAL>>16)&255, (T2VAL>>8)&255,
             #endif

#ifdef fast_file_printf
    snprintf(line, 49, "%d,\t%02d:%02d:%02d:%03d,\t%d,\t%d,\t%d\n",
             i, (T2VAL>>24)&255, (T2VAL>>16)&255, (T2VAL>>8)&255, (T2VAL&255),
             #endif
             sensorState.sensorData[i * SENSORDATASIZE],
             sensorState.sensorData[i * SENSORDATASIZE + 1],
             sensorState.sensorData[i * SENSORDATASIZE + 2]);

    result = file_write(&(fileState->logFile), strlen(line), line);

```

```

        if (!result) {
            DEBUG_PRINTF(">>> Error in writing to data log from batch_log\n");
        }
    }

#ifdef batch_recording
//if (((T2VAL>>16)&255) >= batch_recording_time ) { // T2VAL is counting up in minutes
if (T1VAL >= batch_recording_time_timer1 ) { // T1VAL is counting up to 1 minute
    /*
    // Reset timer back to zero
    T2LD = 0; // Timer 2 load register = 0
    T2CON = 0; // reset Timer 2 back to zero seconds
    //T2CON = 0x000005cf;
    // Enabled, 32.768 osc, count up, periodic (resets to zero at max), clock/3276
    T2CON = 0x000005e8;
    // Enabled, 32.768 osc, count up, HR:MN:SC:HH, periodic (resets to zero at max), clock/256
    */

// Reset timer 1 back to zero
T1LD = 0;
T1CON = 0;
T1CON = 0x000211c0; // Enable T1, internal clock, count up, periodic, clock/1, event 1 = timer 1

IRQCLR = (1<<15); // turn off ADXL interrupts for batch mode only

#endif

result = file_fclose(&(fileState->logFile));

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on file_fclose(\"%s\")\n",
        __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}
result = fs_umount(&(fileState->efs.myFs));

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on fs_umount()\n",
        __FILE__, __LINE__, result);
    return false;
}

IRQEN = (1<<15); // Reenable ADXL interrupts

#ifdef batch_recording
result = efs_init(&(fileState->efs), 0); // Reinit file system for next batch recording time

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on efs_init() = %d\n", __FILE__, __LINE__, result);
    return false;
}

result = file_fopen(&(fileState->logFile), &(fileState->efs.myFs), fileState->logFileName, 'a');

if (result) {
    DEBUG_PRINTF("%s:%d batch_log(): failed on file_fopen(\"%s\")\n",
        __FILE__, __LINE__, fileState->logFileName, result);
    return false;
}

} // bracket for if statement above

#endif

num_of_int = 0;
half = 0;
}

} // end of while loop
return true;
} // end of sensorloop

volatile void redLight(bool state) {
    if (state) {
        GP4SET = 1<<21;
    } else {
        GP4CLR = 1<<21;
    }
}

volatile void greenLight(bool state) {
    if (state) {
        GP4SET = 1<<22;
    } else {
        GP4CLR = 1<<22;
    }
}

```

Bibliografía

- [1] STEPHEN BEEBY, NEIL WHITE, *Energy Harvesting for Autonomous Systems*, Artech House, Norwood MA, 2010
- [2] ALIREZA KHALIGH, OMER C. ONAR, *Energy Harvesting: Solar, Wind and Ocean Energy Conversion Systems*, Taylor & Francis Group.
- [3] COMMITTEE ON ELECTRIC POWER FOR THE DISMOUNTED SOLDIER, *Energy-Efficient Technologies for the Dismounted Soldier*, National Research Council, ISBN: 0-309-59127-9. National Academy Press, Washington, D.C. 1997
- [4] WALTER GUYTON CADY, *Piezoelectricity: An introduction to the Theory and Applications of Electromechanical Phenomena in Crystals (Vol.1)*, Dover Publications, 1964.
- [5] M.IZQUIERDO, A.MARTÍNEZ-RAMÍREZ, J.L.LARRIÓN, *Valoración de la capacidad funcional en el ámbito domiciliario y en la clínica. Nuevas posibilidades de aplicación de acelerometría para la valoración de la marcha, equilibrio y potencia muscular en personas mayores*, Centro de Estudios, Investigación y Medicina del Deporte, Pamplona 2008.
- [6] JUSTIN J.KAVANAGH, HYLTON B.MENZ, *Accelerometry: A technique for quantifying movement patterns during walking*. Gait & Posture. 2007.

- [7] DANIEL A. PERNIA-MÁRQUEZ, *Introducción a la Medición de Vibración*. Facultad de Ingeniería. Universidad de los Andes. Venezuela. 2004.
- [8] HYLTON B. MENZ, STEPHEN R. LORD, RICHARD C. FITZPATRICK, *Acceleration patterns of the head and pelvis when walking on level and irregular surfaces*. Gait & Posture. 2002.
- [9] MARIO A.LAFORTUNE, *Three-Dimensional Acceleration of the Tibia During Walking and Running*. Pergamon Press plc. 1991.
- [10] HENRY A.SODANO, DANIEL J.INMAN AND GYUHAE PARK, *Comparison of Piezoelectric Energy Harvesting Devices for Recharging Batteries*, Journal of Intelligent Material Systems and Structures, Vol. 16. 2005.
- [11] J.RABAEY, F.BURGHARDT, D. STEINGART, M.SEEMAN, P.WRIGHT, *Energy Harvesting: A System Perspective*, Berkeley Wireless Research Center, 2007.
- [12] JAVIER CATALÁN LÓPEZ, *TFM: Energy Harvesting*, Escuela de Ingeniería y Arquitectura. Universidad de Zaragoza. 2011.
- [13] JAVIER VALLE FERNÁNDEZ, *Estudio de un Sistema de Transmisión de Energía por RF*, Universitat Politècnica de Catalunya. 2006.
- [14] S. J. ROUNDY, *Energy Scavenging for Wireless Sensor Nodes with a Focus on Vibration to Electricity Conversion*.
- [15] LORETO MATEU SÁEZ, *Energy Harvesting from Human Power*. Phd Thesis Project. Enero 2004.
- [16] MARTA ARENAS MAS, *Diseño e Implementación de un Sistema de Adquisición de Aceleraciones con Procesamiento mediante microcontrolador*. Universidad de Sevilla. Junio 2008.
- [17] ANALOG DEVICES, INC, *ADXL345 3-axis Digital Accelerometer Data Sheet*. 2009-2011.

- [18] *www.report-fotografía.com/clima/clima02.htm*. Último acceso Junio 2012.
- [19] *www.kryotherm.ru/porto.phtml*. Último acceso Junio 2012.
- [20] *www.sinais.es/sensores/transductores-piezoelectricos.html*. Último acceso Junio 2012.