



UNIVERSIDAD DE LA REPÚBLICA
FACULTAD DE INGENIERÍA



Dispositivo de medida de posición y movimiento de una persona

MEMORIA DE PROYECTO PRESENTADA A LA FACULTAD DE
INGENIERÍA DE LA UNIVERSIDAD DE LA REPÚBLICA POR

Joaquín Facal, Nicolás Gammarano, Alex Gurevich

EN CUMPLIMIENTO PARCIAL DE LOS REQUERIMIENTOS
PARA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRICISTA.

TUTOR

Pedro Arzuaga..... Universidad de la República

TRIBUNAL

Pedro Arzuaga..... Universidad de la República

Sebastián Fernández..... Universidad de la República

Leonardo Steinfeld..... Universidad de la República

Montevideo
miércoles 16 de mayo, 2018

Dispositivo de medida de posición y movimiento de una persona, Joaquín Facal,
Nicolás Gammarano, Alex Gurevich.

Esta tesis fue preparada en L^AT_EX usando la clase iietesis (v1.1).
Contiene un total de 215 páginas.
Compilada el domingo 16 septiembre, 2018.
<http://iie.fing.edu.uy/>

Agradecimientos

Aprovechamos este espacio para agradecer a todas aquellas personas que nos ayudaron y apoyaron de distintas maneras durante el desarrollo de este proyecto. En particular, agradecemos

- a nuestras familias y amigos, a nuestro tutor Pedro Arzuaga, a la Facultad de Ingeniería de la Universidad de la República, y en particular al Instituto de Ingeniería Eléctrica y sus docentes por abrirnos sus puertas y dejarnos utilizar material y espacios como el Laboratorio de Software, el Laboratorio de Medidas, el Taller de Electrónica Libre y el Laboratorio de Proyectos,
- a *CCC Medical Devices* y su gente, en particular a Gabriel Barbat por guiarnos en el diseño del *layout* del prototipo y ayudarnos en la compra de componentes, el pedido de fabricación de los PCB y el *stencil* y población de las placas, y a Daniel Villamil por ayudarnos con los aspectos mecánicos, y darnos una caja y teclados de membrana para el dispositivo,
- a Leonardo Steinfeld por prestarnos un *100-pin socket board* con microcontrolador MSP430F5438 y un programador de MSP430 durante más de un año, y por prestarnos el amplificador logarítmico para analizar consumo,
- a Mariana Siniscalchi por dejarnos usar su caminadora para realizar pruebas.

Esta página ha sido intencionalmente dejada en blanco.

Joaquín desea dedicar este trabajo:
A mis padres Enrique y Mónica y a mi hermana Sofía.

Nicolás desea dedicar este trabajo:
*A mi familia José, Nicole, Rosina, Cecilia, Mari y
a mis compañeros de proyecto Joaquín y Alex por este año tan disfrutable.*

Alex desea dedicar este trabajo:
A su familia: Eduardo, Judith, Fabián y Karen; y a su novia: Mariel.

Esta página ha sido intencionalmente dejada en blanco.

Resumen

Este proyecto consiste en el diseño y fabricación de un dispositivo vestible capaz de detectar la posición del torso del portador y clasificar su movimiento en distintos rangos. La electrónica utilizada incluye un acelerómetro digital, un microcontrolador, circuitos de alimentación e interfaz con el usuario.

Por otra parte, el proyecto abarca también la redacción de dos documentos: *Requerimientos detallados* y *Protocolos de test y resultados*. En el documento de requerimientos se describe en detalle todos los requerimientos que el dispositivo debe cumplir, los cuales son luego verificados según el documento de protocolos de test.

El proyecto estuvo compuesto por varias etapas de desarrollo. Sobre los comienzos se eligieron los componentes y se diseñó un prototipo para desarrollo. Se diseñaron e implementaron métodos de calibración del acelerómetro y posicionamiento del mismo respecto al torso de la persona. Ya con la electrónica operativa, se pasó a una etapa de mediciones para analizar cómo son las aceleraciones del andar humano tanto en el tiempo como en frecuencia, de manera de definir la mejor manera de determinar posición y movimiento. Luego se comenzó una etapa de investigación de algoritmos para la detección del movimiento para elegir uno. Finalmente se diseñó un dispositivo final, de menor tamaño y de mejor apariencia para el usuario, testeando gran parte de los requerimientos.

Como resultado de este proyecto se logró diseñar un dispositivo que cumplió con gran parte de los requerimientos planteados en sus comienzos.

Esta página ha sido intencionalmente dejada en blanco.

Tabla de contenidos

Agradecimientos	I
Resumen	v
1. Introducción	1
2. Requerimientos	5
2.1. Requerimientos conceptuales	5
2.2. Requerimientos detallados	6
3. Diseño de la electrónica	7
3.1. Sistema general	7
3.2. Elección de los componentes	8
3.3. Descripción de la electrónica	10
3.3.1. Acelerómetro	12
3.3.2. Conversor USB-UART	12
3.3.3. Conector mini-USB-A	12
3.3.4. Conector JTAG (<i>Joint Test Action Group</i>)	13
3.3.5. Protección contra descargas electrostáticas	13
3.3.6. LEDs	15
3.3.7. Botones	15
3.3.8. Microcontrolador	15
3.4. Esquemático	16
4. Acelerómetro	17
4.1. Generalidades	17
4.1.1. Tipos de acelerómetro según el método de medición	17
4.1.2. Funcionamiento del acelerómetro seleccionado	18
4.2. Calibración del acelerómetro	23
4.2.1. Método de calibración de 6 puntos	23
4.2.2. Calibración realizada	29
4.2.3. Resultados obtenidos luego de la calibración	30
4.3. Posicionamiento del acelerómetro respecto a la persona	33
4.3.1. Método de posicionamiento propuesto	33
4.3.2. Resultados de posicionamientos	38

Tabla de contenidos

5. Detección de posición y movimiento	43
5.1. Frecuencia de muestreo de aceleraciones	47
5.2. Detección de posición	49
5.3. Detección de giros de 90°, levantadas y acostadas	50
5.4. Detección de movimiento	51
6. Firmware del dispositivo	57
6.1. Arquitectura	57
6.1.1. Modos de funcionamiento	58
6.2. Modularización	59
6.2.1. HAL (<i>Hardware Abstraction Layer</i>)	62
6.2.2. RTC (<i>Real Time Clock</i>)	63
6.3. Modo Measure	64
6.3.1. Tiempos de respuesta	66
6.4. Modo Standby	67
6.4.1. Intérprete de comandos	67
6.5. Uso de memoria	68
7. Adquisidor de datos	71
7.1. <i>Software</i> Adquisidor de Datos	71
7.2. Manual de usuario	72
8. Prototipo	79
8.1. Esquemático	79
8.1.1. <i>Step-up</i>	79
8.1.2. Regulador lineal	80
8.1.3. Acelerómetro	80
8.1.4. Conversor USB-UART	81
8.1.5. Conector mini-USB-A	81
8.1.6. Conector JTAG (<i>Joint Test Action Group</i>)	82
8.1.7. Protección contra descargas electrostáticas	82
8.1.8. LEDs	82
8.1.9. Botones	83
8.1.10. Conectores y <i>jumpers</i>	84
8.1.11. Microcontrolador	84
8.1.12. Resistencias de medida de consumo	86
8.2. <i>Layout</i>	86
8.3. Consumo del dispositivo	87
9. Diseño mecánico	89
10. Dispositivo final	93
10.1. Esquemático	93
10.2. <i>Layout</i>	94
10.3. Consumo y autonomía del dispositivo	94

11.Resultados	99
11.1. Verificación de los requerimientos	99
11.1.1. Resultados	99
11.2. Validación del diseño	100
11.2.1. Validación del algoritmo	100
11.2.2. Validación del dispositivo	101
12.Conclusiones	105
Apéndices	107
A. Requerimientos detallados	107
A.1. Propósito	107
A.2. Definiciones, abreviaturas y acrónimos	107
A.3. Convenciones	108
A.3.1. Requerimientos	108
A.3.2. Comentarios	108
A.3.3. Posiciones	108
A.3.4. Aceleración	109
A.4. Dispositivo	110
A.4.1. Requerimientos mecánicos y físicos	110
A.4.2. Interfaz de usuario	110
A.4.3. Requerimientos generales	111
A.4.4. Modo Standby	112
A.4.5. Modo Test	113
A.4.6. Modo Posicionamiento	113
A.4.7. Modo Measure	116
A.4.8. Comunicación con SAD	126
A.5. Sujeción	131
A.6. Adquisidor de datos	132
A.6.1. Posibles acciones	132
B. Protocolos de test y resultados	135
B.1. Propósito	136
B.2. Definiciones, abreviaturas y acrónimos	137
B.3. Convenciones	137
B.3.1. Formato de cada test	137
B.3.2. Comentarios	137
B.3.3. <i>Firmware</i> del dispositivo	137
B.4. Dispositivo	139
B.4.1. Requerimientos mecánicos y físicos	139
B.4.2. Interfaz de usuario	141
B.4.3. Requerimientos generales	142
B.4.4. Modo Standby	146
B.4.5. Modo Test	147
B.4.6. Modo Posicionamiento	148

Tabla de contenidos

B.4.7. Modo Measure	152
B.4.8. Comunicación con SAD	170
B.5. Sujeción	175
B.6. Adquisidor de datos	176
C. Estimación de memoria de microcontrolador	177
C.1. Memoria RAM	178
C.1.1. Datos estadísticos	178
C.1.2. Matrices de posicionamiento	178
C.1.3. <i>Buffers</i>	179
C.1.4. Variables auxiliares	179
C.1.5. Estimación final	179
C.2. Memoria FLASH	180
C.2.1. Datos estadísticos	180
C.2.2. Modo Test	180
C.2.3. Memoria de programa	180
C.2.4. Inicialización de datos y constantes	180
C.2.5. Estimación final	180
D. Resumen de acelerómetros considerados	181
E. Instalación de los drivers del dispositivo	183
F. Esquemático del prototipo	187
G. Esquemático del diseño final	191
Referencias	195
Índice de tablas	197
Índice de figuras	199

Capítulo 1

Introducción

Hoy en día los acelerómetros se utilizan en diferentes contextos, como por ejemplo:

- a) en celulares o *tablets*, para permitir voltear la imagen o realizar una cierta acción al agitarlos,
- b) en dispositivos médicos implantables o vestibles para diversas funciones vinculadas o no con la terapia a entregar,
- c) en automóviles, para determinar cuándo activar el *airbag*.

En particular, este proyecto se centrará en la segunda categoría de las mencionadas arriba.

El primer dispositivo médico implantable en que se monitoreó la actividad del paciente fue el marcapasos. La medida de actividad de pacientes por medio de acelerómetros se introdujo a fines de los años 70 y el objetivo es adaptar la frecuencia de estimulación cardíaca en función de la actividad del paciente. A mayor actividad mayor frecuencia de estimulación. Su uso en marcapasos continúa hasta nuestros días. Por su parte, los dispositivos médicos implantables para el tratamiento de apnea controlan la posición ya que usualmente sólo entregan terapia cuando el paciente está acostado y quieto. Otros dispositivos médicos implantables usan la medida de posición y movimiento con fines estadísticos. La cantidad de horas levantado y la actividad del paciente tienen relación con el bienestar del mismo y con el efecto del tratamiento.

El objetivo del proyecto es diseñar un dispositivo que sujeto al torso de una persona permita clasificar la posición y movimiento de la misma y que usando esta información almacene datos estadísticos diarios sobre su actividad.

La salida del proyecto debe ser un prototipo operativo y verificado que cumpla con los requerimientos planteados. Forman parte del alcance del proyecto los siguientes puntos:

- Definir la arquitectura del sistema y elegir los componentes.

Capítulo 1. Introducción

- Realizar un documento de requerimientos detallados (finales) a partir de los requerimientos conceptuales (iniciales).
- Idear un método de calibración del sensor e implementarlo.
- Idear un método para conocer la posición del paciente a partir de la posición del sensor e implementarlo.
- Poder almacenar y leer los datos a fin de validar la corrección de diseño, sin ser relevante la interfaz de usuario.
- Idear un algoritmo para clasificar posición (orientación) y movimiento de la persona a partir de los datos ya calibrados y posicionados.
- Idear la manera de colocar el dispositivo en el torso de la persona de manera que no se mueva respecto a la misma, y sin que le incomode durante las horas de sueño.
- Verificar el correcto funcionamiento del sistema. En particular verificar cada uno de los requerimientos finales y redactar protocolos de test para los más relevantes.

A continuación, se muestra un diagrama de bloques básico del sistema a implementar.

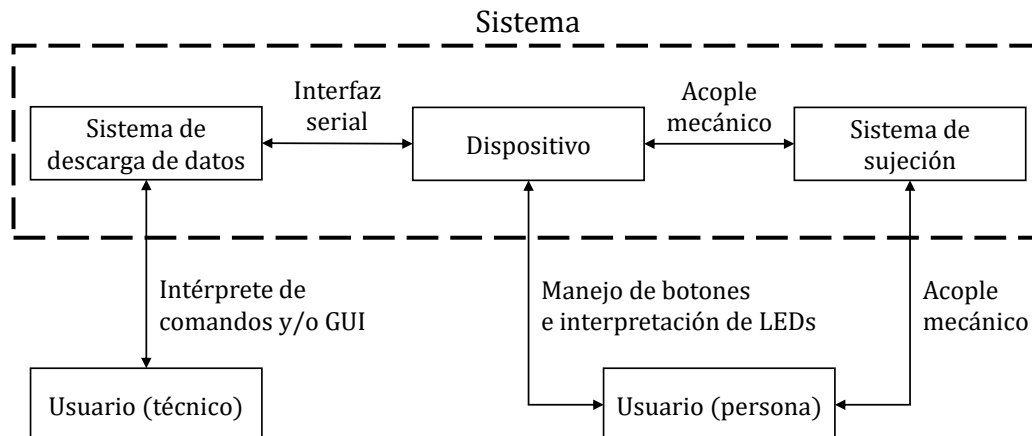


Figura 1.1: Diagrama de bloques del sistema

Como se observa en el diagrama de bloques, el sistema estará compuesto por el dispositivo, el sistema de descarga de datos y el sistema de sujeción.

- El **dispositivo** adquirirá los datos de aceleración, los procesará y almacenará los datos estadísticos diarios. Estará compuesto básicamente por un circuito de alimentación, un microcontrolador, un acelerómetro y una caja o carcasa en la cual estará contenido. A su vez, el usuario (persona) podrá comunicarse con el dispositivo a través de botones, interruptores, LEDs y/o *buzzers* (por

ejemplo interruptor de encendido/apagado, LED indicador de encendido, LED indicador de batería baja).

- El **sistema de descarga de datos** se comunicará con el dispositivo a través de una interfaz serial y permitirá la descarga de datos. Estará compuesto básicamente por una PC y una aplicación de *software* en la misma. A su vez, el usuario (técnico) podrá interactuar con el sistema de descarga de datos, y realizar ciertas acciones (entre ellas la extracción de los datos estadísticos, la consulta y seteo de la hora llevada por el dispositivo) a través de un intérprete de comandos y/o de una interfaz gráfica de usuario.
- El **sistema de sujeción** estará encargado de sujetar el dispositivo al torso del usuario (persona) de manera que quede quieto respecto al mismo y no le incomode durante actividades normales.

El resto de la documentación está organizado de la siguiente manera. El capítulo 2 presenta los requerimientos conceptuales que fueron el punto de partida del proyecto. En el capítulo 3, se describe cada componente del sistema, y en particular el diseño de la electrónica. En el capítulo 4, se entra en detalle en el acelerómetro, describiendo su método de medición, así como métodos de calibración y de posicionamiento. El algoritmo utilizado para clasificar posición y movimiento es descrito en el capítulo 5. La implementación de *firmware* se presenta en el capítulo 6 y el *software* adquisidor de datos en el capítulo 7. En el capítulo 8, se describe el diseño del prototipo fabricado con fines de desarrollo. El diseño mecánico es detallado en el capítulo 9. Se presenta el diseño del dispositivo final en el capítulo 10. Dicho dispositivo final se verifica y valida en el capítulo 11. Finalmente, se concluye el trabajo realizado en 12.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 2

Requerimientos

Los requerimientos conceptuales fueron el punto de partida del proyecto. A partir de los mismos, se definió la arquitectura del sistema, los componentes a utilizar y se redactaron los requerimientos detallados.

2.1. Requerimientos conceptuales

El sistema deberá estar compuesto por un dispositivo, un sistema de sujeción al torso de la persona y un adquisidor de datos (*software* en un PC). El uso esperado será el monitoreo de una persona, que por ejemplo podrá ser un paciente o un atleta.

El dispositivo deberá ser capaz de detectar la posición del usuario y clasificarla según parado, acostado con la cabeza hacia arriba, hacia abajo y hacia uno y otro costado. También deberá ser capaz de detectar el movimiento del usuario y clasificarlo en 3 niveles posibles.

En particular, el dispositivo deberá ser capaz de almacenar diariamente:

- total de tiempo en nivel de movimiento 0,
- total de tiempo en nivel de movimiento 1,
- total de tiempo en nivel de movimiento 2,
- total de tiempo en nivel de movimiento 3,
- total de tiempo parado,
- total de tiempo acostado,
- total de tiempo acostado boca arriba,
- total de tiempo acostado boca abajo,
- total de tiempo acostado sobre el hombro izquierdo,

Capítulo 2. Requerimientos

- total de tiempo acostado sobre el hombro derecho,
- cantidad de giros de 90 grados en posición horizontal, y
- hora de acostado y levantado.

El dispositivo deberá ser capaz de almacenar los datos anteriores para 3 usuarios distintos, para cada uno de los cuales hasta 7 días de datos.

El dispositivo deberá contar con un método de calibración y posicionamiento respecto a la persona. Además, deberá contar con una interfaz de usuario, por ejemplo LEDs, botones, *buzzers*, para que el usuario pueda interactuar con el dispositivo. En particular, para que el usuario pueda saber si el dispositivo está encendido o apagado o saber el estado de la batería.

El diseño de la electrónica deberá minimizar el consumo del dispositivo y la cantidad de componentes tanto como sea posible. En particular, el dispositivo deberá tener una autonomía mayor a una semana (requerimiento), con una meta de dos semanas.

El diseño mecánico deberá considerar que el sensor no debe moverse apreciablemente respecto al torso del usuario y que el uso del dispositivo no debe incomodar al usuario siempre que realice actividades normales, en particular durante las horas de sueño.

No hay requerimientos estéticos ni de protección anti-golpes o ingreso de líquidos.

2.2. Requerimientos detallados

A partir de los requerimientos conceptuales, se definieron los requerimientos detallados. Los requerimientos detallados son muy importantes en un proyecto, ya que definen de manera clara y sin ambigüedades qué se espera del mismo. Es importante resaltar que los requerimientos detallados conforman un documento en sí mismo, por lo que se agregó el mismo en el apéndice (ver apéndice A). Se recomienda fuertemente al lector leer dicho apéndice para comprender el resto de la documentación.

Capítulo 3

Diseño de la electrónica

3.1. Sistema general

Como se mencionó en el capítulo 1, el sistema estará conformado por un sistema de descarga de datos, un dispositivo y un sistema de sujeción.

A continuación, se listan en más detalle los principales componentes del dispositivo.

- Un microcontrolador, que será el que correrá el *firmware* diseñado, procesando adecuadamente los datos y un cristal externo para generar el reloj,
- un acelerómetro de 3 ejes, que tomará las muestras de aceleración de la persona,
- una fuente de alimentación autónoma que alimentará al circuito (2 pilas AA o AAA),
- un regulador lineal de voltaje, que regulará el voltaje del microcontrolador en 3,3 V,
- un *step-up*, ya que la idea es alimentar el circuito desde dos pilas AA o AAA, teniendo entonces 3 V de entrada. Para que el regulador lineal funcione correctamente, se necesitan 3,3 V más un cierto voltaje que depende del regulador,
- un conector USB para comunicarse con un PC y extraer los datos,
- un chip de protección contra descargas electrostáticas para el conector USB,
- un chip FTDI, que traducirá USB a UART y
- un conector JTAG, mediante el cual se programará el microcontrolador.

A continuación, se muestra un diagrama de bloques del sistema correspondiente a esta descripción más detallada.

Capítulo 3. Diseño de la electrónica

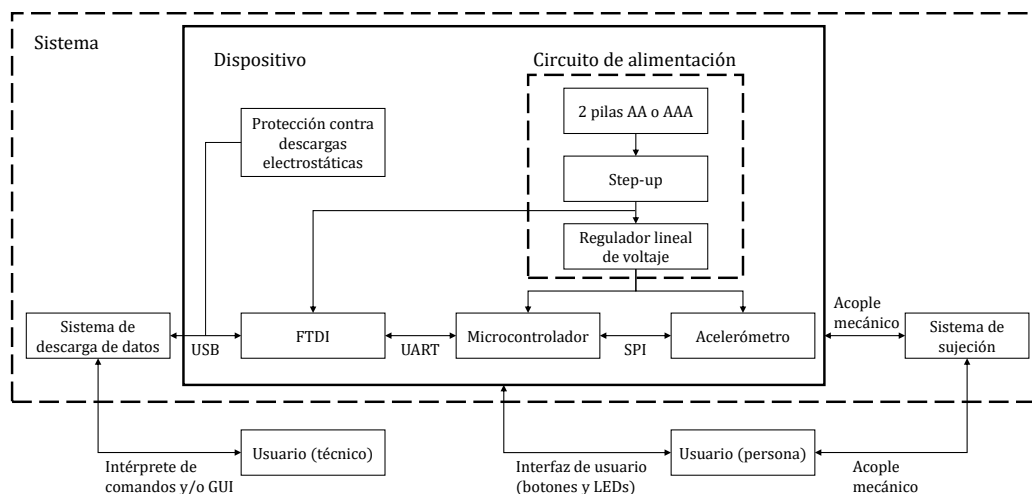


Figura 3.1: Diagrama de bloques del sistema

3.2. Elección de los componentes

En primer lugar, es importante resaltar que la empresa *CCC Medical Devices* proveyó los componentes, por lo que se intentó elegir componentes en *stock* en la misma.

Para la elección del microcontrolador a utilizar se realizó un cálculo aproximado de la memoria necesaria en la aplicación (ver apéndice C) y se concluyó que la aplicación requiere al menos 3,6 kB de memoria RAM y 50 kB de memoria FLASH.

Se decidió trabajar entonces con el microcontrolador MSP430F5342 [1] de *Texas Instruments*. Este microcontrolador cumple con los requisitos necesarios y a su vez pertenece a la familia MSP430x5xxx [2], la cual ya se había utilizado en el curso de Sistemas Embebidos para Tiempo Real. Se destacan las siguientes características del microcontrolador:

- 10 kB de memoria RAM,
- 128 kB de memoria FLASH,
- periférico USCI (*Universal Serial Communication Interface*), el cual soporta distintas comunicaciones seriales (UART, SPI e I²C),
- periférico RTC (*Real-Time Clock*), el cual resuelve la implementación de un reloj de tiempo real,
- periférico *timer* (A y B),
- periférico ADC de 12 bits, y
- 38 pines de entrada/salida.

3.2. Elección de los componentes

Al seleccionar el acelerómetro, se consideró principalmente la cantidad de ejes de medición, el consumo, el rango de medición de aceleraciones, el rango de alimentación, la resolución y la interfaz de comunicación con el microcontrolador.

Respecto a dichas características, se tuvieron las siguientes consideraciones:

- 3 ejes de medición para ser capaces de determinar si la persona está parada, acostada mirando hacia arriba o abajo, o acostada sobre su hombro derecho o izquierdo,
- el consumo (se busca el mínimo consumo posible, siendo preferible que tenga un modo de bajo consumo),
- el rango de medición de aceleraciones deberá ser capaz de registrar los movimientos cotidianos que realiza una persona, es suficiente con $\pm 4g$,
- el rango de alimentación (el acelerómetro deberá poder alimentarse con la misma alimentación del microcontrolador, que corresponde a $3,3V$),
- la resolución deberá ser tal que se pueda clasificar el esfuerzo dinámico de la persona, y
- la interfaz de comunicación deberá ser compatible con el microcontrolador elegido, por ejemplo salida analógica ya que cuenta con ADC, o salida digital a través de interfaces SPI o I²C.

El apéndice D resume los acelerómetros considerados y la comparación de sus funcionalidades. Se seleccionó el ADXL345 [3], el cual cumple las características mencionadas.

Las características del acelerómetro elegido son las siguientes:

- 3 ejes de medición: x, y, z,
- consumo de corriente: entre $23\mu A$ y $140\mu A$ en modo medición dependiendo de la frecuencia de datos de salida y $0,1\mu A$ en modo *standby*,
- rango de medición de aceleración seleccionable: $\pm 2g$, $\pm 4g$, $\pm 8g$ o $\pm 16g$,
- rango de alimentación: entre $2,0V$ y $3,6V$,
- resolución: $LSB = 3,9mg$,
- interfaz de comunicación: SPI o I²C.

Se eligieron el integrado de protección contra descargas electrostáticas TPD2E001DZDR [4] y el conversor USB-UART FT232RQ [5], dado que sus características son adecuadas para nuestro sistema.

Para la selección de los componentes de alimentación, se consideró que se cuenta con dos pilas AA o AAA, por lo que el voltaje será aproximadamente de $2 \times$

Capítulo 3. Diseño de la electrónica

$1,5\text{ V} = 3,0\text{ V}$ y caerá a medida que las pilas se descarguen, hasta aproximadamente $2 \times 0,75\text{ V} = 1,5\text{ V}$ cuando están totalmente descargadas.

Se requiere alimentar el microcontrolador (entre $1,8\text{ V}$ y $3,6\text{ V}$), el acelerómetro (entre $2,0\text{ V}$ y $3,6\text{ V}$) y el FT232R (entre $4,0\text{ V}$ y $5,25\text{ V}$).

Es entonces necesario aumentar el voltaje de entrada, ya que los $3,0\text{ V}$ no son suficientes para alimentar el FT232R. Además, a medida que las pilas se descargan, el voltaje bajará de $1,8\text{ V}$, por lo que en ese caso no es suficiente para alimentar ni el microcontrolador ni el acelerómetro. Es decir que si se quiere aprovechar al máximo la carga de las pilas, se deberá mantener un voltaje mayor a $1,8\text{ V}$ para alimentar al microcontrolador y mayor a $2,0\text{ V}$ para alimentar al acelerómetro. Se decidió utilizar un *step-up*, convertidor DC-DC de $4,5\text{ V}$ de salida. Dicho voltaje alimenta al FT232R.

Como el voltaje de salida de los convertidores DC-DC conmutados (por ejemplo el *step-up* mencionado) es ruidosa, se decidió utilizar un regulador lineal de voltaje a la salida del dicho convertidor, que regula el voltaje a $3,3\text{ V}$ para alimentar al microcontrolador y al acelerómetro. De esta manera, se evita introducir ruido eléctrico de conmutación a las medidas de aceleración. Se eligieron el regulador lineal LP2985AIM5-3.3 [6] y el *step-up* TPS61220DCK [7].

Los componentes como condensadores, resistencias e inductancias se eligieron en base a sugerencias de las respectivas hojas de datos de los integrados.

3.3. Descripción de la electrónica

3.3.0.1. *Step-up*

El *step-up* es un convertidor DC-DC que convierte un cierto voltaje de DC a otro voltaje de DC mayor (de ahí el nombre “*step-up*”). Dicho convertidor utiliza para la conversión una inductancia, un diodo y una llave.

El integrado elegido (TPS61220DCK) tiene muchos componentes integrados, por lo que únicamente se debe incluir la inductancia y los condensadores de desacople tanto a la entrada como a la salida. Además, el voltaje de DC a la salida es seleccionable dentro de un cierto rango a través de un voltaje en el pin FB (de *feedback*), por lo que se utiliza un divisor resistivo para fijar el valor en el pin FB de manera de tener a la salida el voltaje de DC deseado. Para que funcione correctamente, el voltaje de entrada V_{IN} debe estar entre $0,7\text{ V}$ y $5,5\text{ V}$.

Una aplicación típica se muestra en la figura 3.2 tomada de la hoja de datos del mismo [7].

Los valores de la inductancia y los condensadores de desacople elegidos son los recomendados en la hoja de datos [7]: $L_1 = 4,7\text{ }\mu\text{H}$, $C_1 = 10\text{ }\mu\text{F}$ y $C_2 = 10\text{ }\mu\text{F}$.

Para fijar $V_o = 4,5\text{ V}$, se utilizó un divisor resistivo. Según la hoja de datos [7], se debe cumplir $V_{FB} = 500\text{ mV}$ y $R_2 < 500\text{ k}\Omega$.

Se eligió $R_2 = 150\text{ k}\Omega < 500\text{ k}\Omega$. Del divisor resistivo, se tiene que $V_{FB} = V_{out} \left(\frac{R_2}{R_1 + R_2} \right)$. Despejando R_1 de la ecuación anterior, se tiene

3.3. Descripción de la electrónica

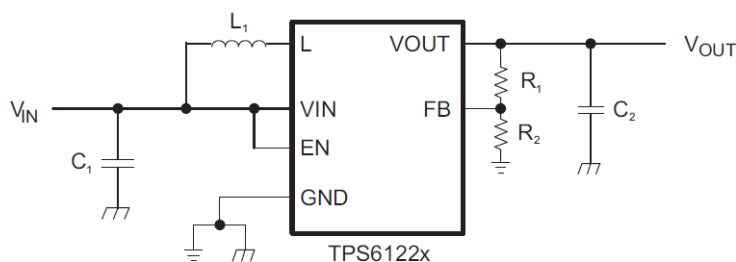


Figura 3.2: Aplicación típica del *step-up* TPS6122x

$$R_1 = R_2 \left(\frac{V_{out}}{V_{FB}} - 1 \right) = 150 \text{ k}\Omega \left(\frac{4,5 \text{ V}}{0,5 \text{ V}} - 1 \right) = 1200 \text{ k}\Omega = 1,2 \text{ M}\Omega$$

Se eligió entonces $R_1 = 1,2 \text{ M}\Omega$ y $R_2 = 150 \text{ k}\Omega$.

3.3.0.2. Regulador lineal

El regulador lineal mantiene fijo el voltaje de DC a la salida, siempre que el voltaje a la entrada sea mayor al de salida más un cierto voltaje.

El integrado que se eligió (el LP2985AIM5-3.3) tiene el regulador implementado internamente, por lo que solamente se deben incluir condensadores de desacople tanto a la entrada como a la salida.

Un esquemático de aplicación del regulador lineal LP2985 se muestra en la figura 3.3 tomada de la hoja de datos del mismo [6].

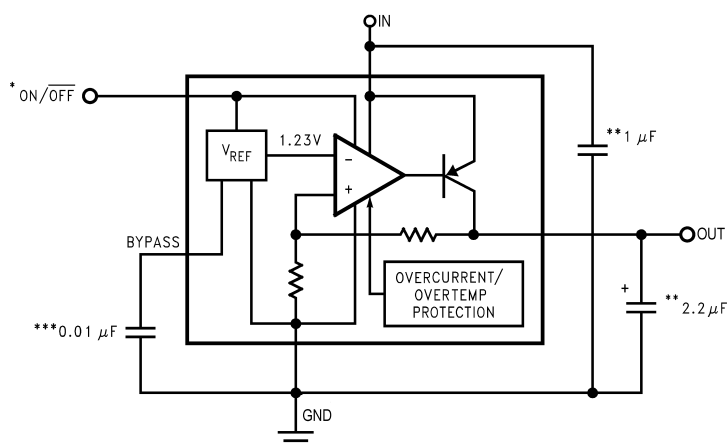


Figura 3.3: Aplicación del regulador lineal LP2985

Los valores de los condensadores de desacople elegidos siguen los requerimientos de la hoja de datos [6]: $C_{OUT} = 10 \mu\text{F} > 2,2 \mu\text{F}$ y $5 \text{ m}\Omega < ESR_{OUT} < 0,5 \Omega$, $C_{BYPASS} = 0,01 \mu\text{F}$ y $C_{IN} = 10 \mu\text{F} > 1 \mu\text{F}$.

3.3.1. Acelerómetro

El acelerómetro ADXL345 se comunicará con el microcontrolador a través de la interfaz SPI (*Serial Peripheral Interface*), por lo que se deben conectar las respectivas señales de dicha interfaz: SPI_MOSI (*Master Out Slave In*), SPI_MISO (*Master In Slave Out*), SPI_CLK (*Clock*) y SPI_CS (*Chip Select*) con el microcontrolador. Además, el acelerómetro dispone de dos pines de generación de interrupciones programables (INT1 e INT2), que se conectan a pines del microcontrolador que puedan usarse como fuente de interrupciones.

Un esquemático de aplicación del acelerómetro ADXL345 se muestra en la figura 3.4 tomada de la hoja de datos del mismo [3].

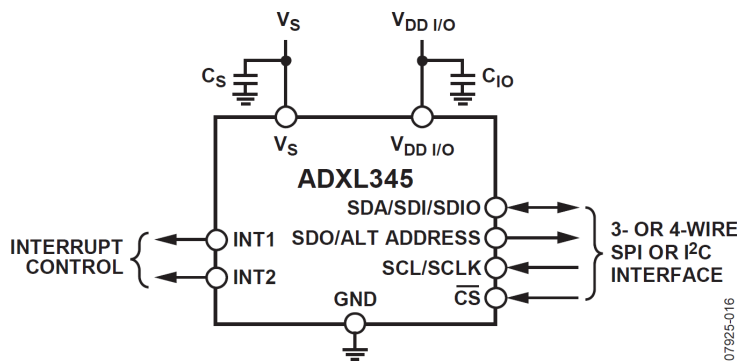


Figura 3.4: Aplicación del acelerómetro ADXL345

Los valores de los condensadores de desacople elegidos son los recomendados en la hoja de datos [3]: $C_{IO} = 0,1 \mu\text{F} = 100 \text{ nF}$ y $C_S = 1 \mu\text{F}$.

3.3.2. Conversor USB-UART

El integrado FT232R convierte USB (*Universal Serial Bus*) a UART (*Universal Asynchronous Receiver-Transmitter*), de manera que permite la comunicación entre el dispositivo (microcontrolador a través de la UART) y el PC (a través del conector USB).

Se deben conectar entonces las señales TXD y RXD a las interfaces de recepción y transmisión respectivamente de UART del microcontrolador; y las señales de la interfaz USB_DP (*Data Plus*) y USB_DM (*Data Minus*) al conector mini-USB-A.

Se utilizó la configuración llamada “*Self Powered Configuration*” [5] para alimentar el integrado FT232R desde los 4,5 V de salida del *step-up*. En la figura 3.5, tomada de [5], se presenta la configuración utilizada.

3.3.3. Conector mini-USB-A

El conector mini-USB-A se utiliza para comunicar la PC y el chip FT232R (conversor USB-UART). Las señales de la interfaz USB son VCC (fuente), GND (tierra), USB_DP (*Data Plus*) y USB_DM (*Data Minus*).

3.3. Descripción de la electrónica

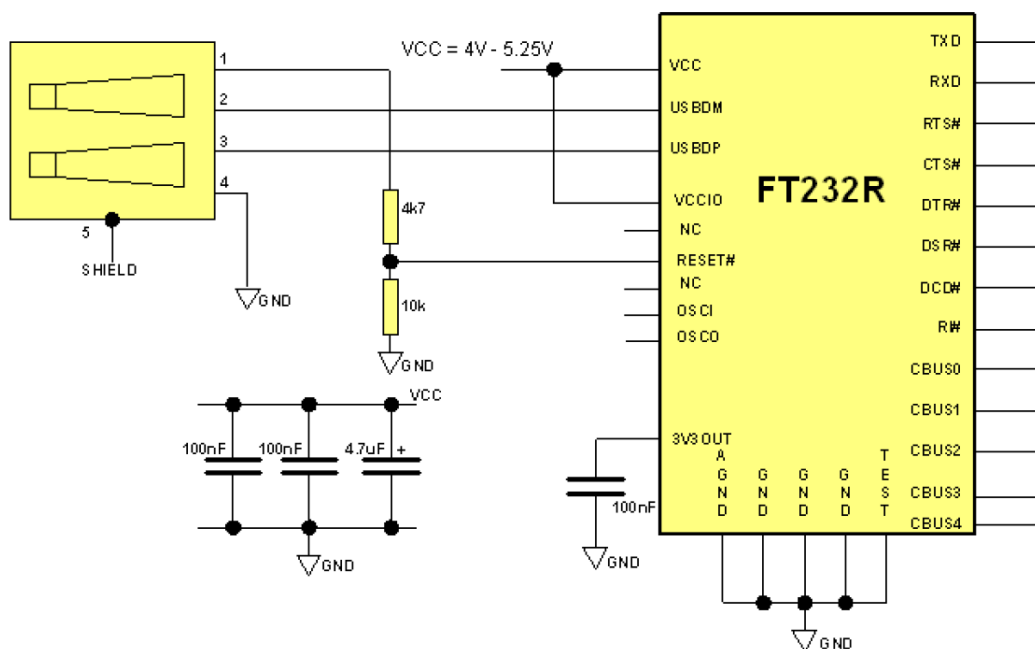


Figura 3.5: Aplicación del integrado FT232R en configuración *Self Powered*

3.3.4. Conector JTAG (*Joint Test Action Group*)

Existen programadores de microcontroladores que se conectan a una PC a través de un conector USB-B (con un cable USB-B a USB-A) y al microcontrolador a través del conector JTAG. El conector JTAG se utiliza para programar el microcontrolador mediante un FET (*Flash Emulation Tool*).

El estándar JTAG (*Joint Test Action Group*) es el nombre común para la norma *IEEE Std 1149.1 - IEEE Standard for Test Access Port and Boundary-Scan Architecture* [8].

Se utilizó la configuración *4-wire JTAG* [9]. Las señales del estándar JTAG a conectar son TDI (*Test Data In*), TDO (*Test Data Out*), TCK (*Test Clock*), TMS (*Test Mode Select*) y TRST (*Test Reset*).

En la figura 3.6, tomada de [9], se muestra el esquemático asociado a la conexión del conector JTAG con el microcontrolador.

3.3.5. Protección contra descargas electrostáticas

El conector USB está expuesto a varias fuentes de descargas electrostáticas (ESD del inglés *ElectroStatic Discharge*). Al conectar y desconectar el conector USB, se generan transitorios tanto de voltaje como de corriente. El contacto con el cuerpo humano puede generar descargas suficientes para dañar los circuitos integrados adyacentes al puerto USB a través del mismo [10]. En particular, en el dispositivo a diseñar el conector USB estará disponible al usuario. Por lo tanto, es importante que dicho conector esté protegido contra ESD.

El integrado que se eligió (el TPD2E001DZDR) provee dicha protección a

Capítulo 3. Diseño de la electrónica

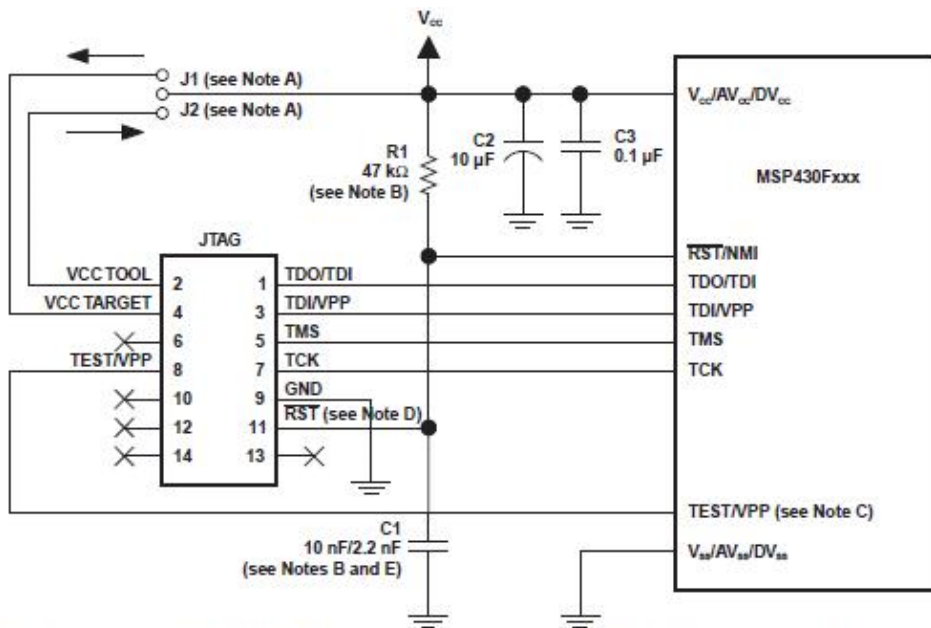


Figura 3.6: Esquemático asociado al conector JTAG

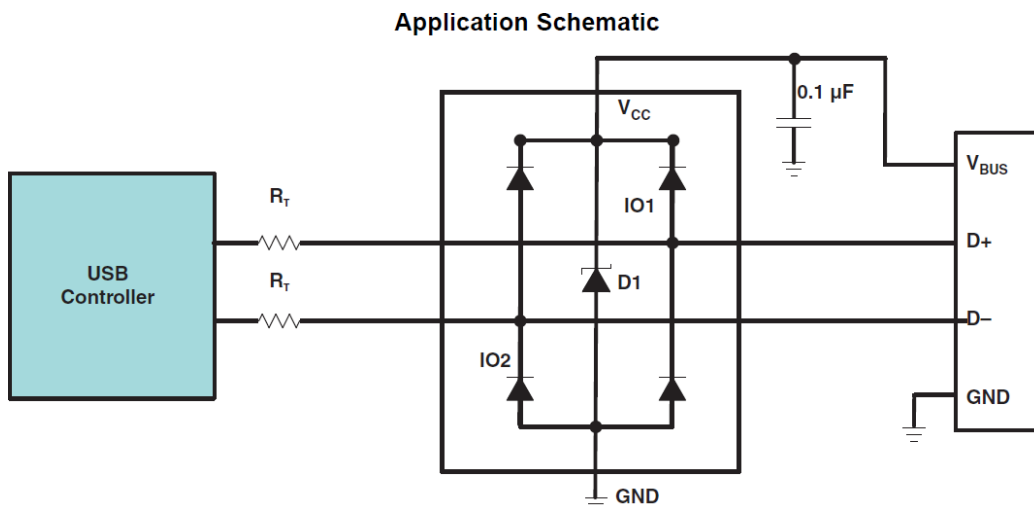


Figura 3.7: Aplicación típica del chip TPD2E001 de protección contra descargas electrostáticas

través de un arreglo de diodos, como se muestra en la figura 3.7 tomada de la hoja de datos del mismo [4]. La idea básica es que en caso de haber un voltaje mayor a $V_{CC} + V_{\gamma}$ o menor a $-V_{\gamma}$ (siendo V_{γ} el voltaje ánodo-cátodo de los diodos en conducción) en cualquiera de las líneas D_+ o D_- , el respectivo diodo entre en conducción eliminando el transitorio de voltaje.

Se utilizó la misma configuración que en la figura 3.7, y el valor del condensador elegido es el recomendado en la hoja de datos [4]: $C_{BUS} = 0,1 \mu\text{F} = 100 \text{ nF}$.

3.3.6. LEDs

Cada LED se conectará a un pin de salida del microcontrolador a través de una resistencia R . El valor de dicha resistencia se calcula simplemente utilizando la ley de Ohm:

$$R = \frac{V_R}{I_R} = \frac{V_{OH} - V_{LED}}{I_{LED}} = \frac{3,3\text{ V} - V_{LED}}{I_{LED}}$$

donde V_{OH} es el voltaje de nivel alto en el pin del microcontrolador (según su hoja de datos [1], dicho voltaje V_{OH} está entre $V_{CC} - 0,25\text{ V} = 3,05\text{ V}$ y $V_{CC} = 3,3\text{ V}$), I_{LED} es la corriente a través del LED y V_{LED} el voltaje en bornes del mismo. Para hallar R , se debe reemplazar I_{LED} por la corriente nominal del LED y V_{LED} por el voltaje para el cual se da dicha corriente nominal.

3.3.7. Botones

Cada botón se conecta a tierra (GND) y a un pin de entrada del microcontrolador (V_{BUT}). Son entonces activos por nivel bajo ya que al apretar el botón, se cortocircuita el voltaje entre sus bornes y entonces hay $V_{BUT} = 0\text{ V}$ en el pin de entrada del microcontrolador.

Para que cuando se suelte el botón haya $V_{BUT} = V_{CC} = 3,3\text{ V}$ en bornes del mismo, se conecta V_{BUT} al voltaje de alimentación V_{CC} a través de una resistencia de *pull up*. Dicha resistencia es necesaria para que no haya un cortocircuito entre fuente y tierra al presionar el botón. Para que la corriente entre V_{CC} y GND tome un cierto valor I_R menor a la máxima corriente que tolera el puerto del microcontrolador mientras el botón está pulsado, se dimensiona la resistencia R de la siguiente manera:

$$R = \frac{V_{CC}}{I_R}$$

Además, para eliminar rebotes, se agrega un capacitor C en paralelo al botón, de manera que al soltar el botón, el voltaje V_{BUT} no pase a valer instantáneamente V_{CC} , sino que tome forma exponencial ya que es un circuito RC, de constante de tiempo $\tau = RC$ y de valor final V_{CC} . Se elige C de manera que $\tau = RC$ sea suficientemente grande para eliminar los rebotes del botón (del orden de decenas de ms). Se dimensiona entonces el capacitor C de la siguiente manera:

$$C = \frac{\tau}{R}$$

3.3.8. Microcontrolador

El microcontrolador se conecta con gran parte de los componentes ya descritos:

- es alimentado por los 3,3 V del regulador lineal,

Capítulo 3. Diseño de la electrónica

- se comunica con el acelerómetro a través de una interfaz SPI, utilizando el periférico USCI B en modo SPI (las señales a conectar son SPI_MOSI, SPI_MISO, SPI_CLK y SPI_CS), y a través de 2 fuentes de interrupciones del acelerómetro INT1 e INT2,
- se comunica con el conversor USB-UART a través de una interfaz UART, utilizando el periférico USCI A en modo UART (las señales a conectar son UART_TX y UART_RX),
- se carga el programa del proyecto a través del conector JTAG en configuración *4-wire* JTAG (se deben conectar las señales TDI, TDO, TCK, TMS y TRST),
- se conecta a los LEDs y botones a través de pines GPIO (pines de entrada/salida de propósito general, del inglés “*General Purpose Input/Output*”),
- se conecta a un cristal externo de 32 768 Hz, y
- al voltaje de las pilas a través de un ADC para monitorearlo y dar un estimado del porcentaje de carga de las baterías.

3.4. Esquemático

Siguiendo la arquitectura descrita, se diseñó un dispositivo prototipo y un dispositivo final. Sus esquemáticos se encuentran en los apéndices F y G respectivamente. Si bien son similares, dichos dispositivos tienen propósitos diferentes por lo que cada esquemático tiene particularidades que son abordadas en los capítulos 8 y 10.

Capítulo 4

Acelerómetro

4.1. Generalidades

El acelerómetro es un transductor muy utilizado, que transforma una aceleración en otra magnitud física.

4.1.1. Tipos de acelerómetro según el método de medición

Existen distintos tipos de acelerómetros. A continuación se describen los más relevantes, clasificados según el método de medición utilizado.

- **Piezoeléctricos:** Estos dispositivos aprovechan la propiedad de piezoelectricidad de ciertos materiales que al deformarse producen una diferencia de potencial. Se obtienen entonces deformaciones y por lo tanto fuerzas, y aplicando la segunda ley de Newton ($\mathbf{F} = m\mathbf{a}$) se obtiene la aceleración. Para fabricar estos dispositivos, se coloca una masa conocida que somete al material piezoeléctrico a una cierta fuerza que será proporcional a la aceleración buscada.
- **Piezoresistivos:** De forma análoga, existen dispositivos que aprovechan propiedades piezoresistivas de ciertos materiales, que al ser sometidos a deformaciones mecánicas varían su resistencia eléctrica. Los materiales mencionados se colocan en una rama de un puente resistivo, y luego se reduce al caso anterior pues se obtiene un voltaje proporcional a la deformación mecánica.
- **Capacitivos:** Este tipo de acelerómetro acopla una masa (móvil) a una placa que se mueve entre otras dos (fijas), formando dos condensadores de placas paralelas.

La capacitancia de dos placas paralelas depende del área de las mismas, la distancia que las separa y la permitividad eléctrica del medio según $C = \frac{\epsilon A}{d}$. Al moverse la masa y por lo tanto la placa central, varía la separación entre las placas y por lo tanto el valor de las capacidades. Finalmente se utiliza el voltaje que se obtiene del divisor capacitivo para medir la aceleración.

4.1.2. Funcionamiento del acelerómetro seleccionado

El acelerómetro seleccionado es el ADXL345. Dicho sensor entra en la categoría de acelerómetros capacitivos por el método de medición que utiliza.

Además, dicho acelerómetro es lo que se llama un MEMS (sistema micro-electromecánico o en inglés *Micro-Electro-Mechanical System*) que corresponde a la tecnología electromecánica a escalas micrométricas con la que está fabricado.

La figura 4.1 muestra el principio básico de funcionamiento de un eje de los acelerómetros MEMS de la familia ADXL de *Analog Devices* [11].

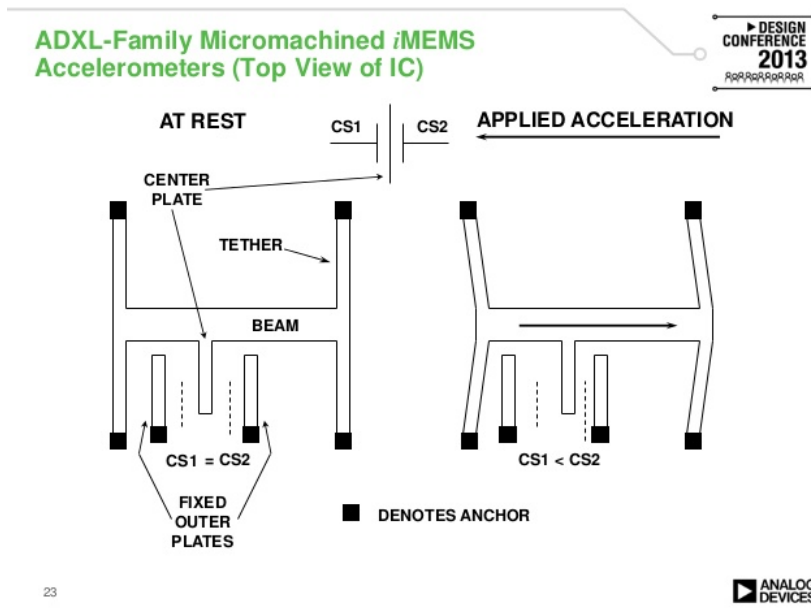


Figura 4.1: Vista superior del circuito integrado de los acelerómetros MEMS de la familia ADXL de *Analog Devices*

Como se mencionó, el ADXL345 es un acelerómetro capacitivo, formado por dos placas fijas (placas externas de los capacitores C_{s1} y C_{s2}) y una placa central que se mueve solidaria a la masa móvil (placa central de los capacitores C_{s1} y C_{s2}). A partir de la figura 4.1, se puede ver que en reposo, se tiene $C_{s1} = C_{s2} = \frac{\epsilon A}{x_0}$, mientras que con una cierta aceleración aplicada, se tiene $C_{s1} = \frac{\epsilon A}{x_0+x}$ y $C_{s2} = \frac{\epsilon A}{x_0-x}$. Se supone que en las placas externas fijas que forman los capacitores C_{s1} y C_{s2} hay un cierto voltaje $V_1 = V$ y $V_2 = -V$ respectivamente, y que el voltaje de la placa central es $v_o(t)$. Se tiene entonces, a partir de la ley de conservación de la carga:

$$Q_1(t) + Q_2(t) = 0$$

$$C_1(t) V_{C1}(t) + C_2(t) V_{C2}(t) = 0$$

$$\left(\frac{\epsilon A}{x_0 + x(t)} \right) (V - v_o(t)) + \left(\frac{\epsilon A}{x_0 - x(t)} \right) (-V - v_o(t)) = 0$$

4.1. Generalidades

Despejando de la expresión anterior, se obtiene

$$v_o(t) = -\frac{V}{x_0}x(t)$$

Modelando el movimiento de la masa como un resorte de constante k , despreciando los efectos de viscosidad y aplicando la segunda ley de Newton a la masa según el eje en el cual se mueve (eje horizontal en la figura 4.1), se tiene

$$ma_x(t) = \sum F_x(t) = -kx(t)$$

$$x(t) = -\frac{m}{k}a_x(t)$$

Finalmente se obtiene un voltaje de salida proporcional a la aceleración en el eje x :

$$v_o(t) = -\frac{V}{x_0}x(t) = \frac{Vm}{kx_0}a_x(t)$$

Es importante destacar ciertos aspectos de la aceleración medida por el ADXL345.

En primer lugar, la salida del acelerómetro se da en múltiplos de la aceleración gravitatoria “g” siendo esta la asociada al lugar donde se calibrará el acelerómetro y no en m/s^2 . Por lo tanto, la aceleración obtenida será relativa a la aceleración gravitatoria en el lugar donde se calibra el dispositivo, no será una aceleración absoluta. Es importante aclarar esto, ya que la aceleración gravitatoria no es constante en todos los puntos del planeta (varía principalmente debido a la latitud y la altura).

En segundo lugar, la aceleración medida no es la relativa a un sistema de coordenadas fijo con respecto a la Tierra, sino que es la aceleración del encapsulado del acelerómetro con respecto a la aceleración de la masa móvil mostrada en la figura 4.1 (lo que en inglés se denomina “*proper acceleration*”). De esta manera, la aceleración \mathbf{a}_{meas} medida (del inglés *measured*) por el acelerómetro es

$$\mathbf{a}_{\text{meas}} = \mathbf{a}_{\text{case}} - \mathbf{a}_{\text{mass}} = \mathbf{a}_{\text{case}} - \mathbf{a}_{\mathbf{g}} = \mathbf{a}_{\text{case}} - \mathbb{M} \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \mathbf{g}$$

siendo \mathbf{a}_{case} la aceleración del encapsulado del acelerómetro respecto a un sistema de coordenadas fijo, \mathbf{a}_{mass} la aceleración de la masa móvil respecto al sistema de coordenadas fijo, $\mathbf{a}_{\mathbf{g}}$ la aceleración gravitatoria, \mathbb{M} la matriz de giro asociada a la orientación de la aceleración gravitatoria y $\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \mathbf{g}$ la aceleración gravitatoria considerando $\mathbb{M} = \mathbb{I}$.

La tabla 4.1 resume el valor de los vectores y sus módulos para dos casos característicos: acelerómetro en reposo y acelerómetro en caída libre.

Por un lado, este fenómeno presenta la ventaja de que si el acelerómetro está en reposo, se puede determinar la dirección del vector de aceleración, y por lo tanto

Capítulo 4. Acelerómetro

Situación	\mathbf{a}_{case}	$\ \mathbf{a}_{\text{case}}\ _2$	\mathbf{a}_{meas}	$\ \mathbf{a}_{\text{meas}}\ _2$
reposo	$\vec{0} \text{ g}$	0 g	M $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ g}$	1 g
caída libre	M $\begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \text{ g}$	1 g	$\vec{0} \text{ g}$	0 g

Tabla 4.1: Salida del acelerómetro \mathbf{a}_{meas} para \mathbf{a}_{case} dado

la orientación de la persona.

Por otro lado, este fenómeno presenta la desventaja de que si se le aplica una fuerza al acelerómetro, es preciso desacoplar de alguna manera la aceleración debida al movimiento \mathbf{a}_{case} de la aceleración gravitatoria $\mathbf{a}_{\mathbf{g}}$, ya que en caso contrario no se podría clasificar correctamente el movimiento. Por ejemplo, si se mide $\mathbf{a}_{\text{meas}} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \text{ g}$, instantáneamente no se puede determinar cuál de los siguientes casos es en realidad el correcto:

- $\mathbf{a}_{\text{case}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ g}$ y $\mathbf{a}_{\mathbf{g}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ g}$ (persona quieta, con el torso vertical y la cabeza hacia abajo),
- $\mathbf{a}_{\text{case}} = \begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix} \text{ g}$ y $\mathbf{a}_{\mathbf{g}} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \text{ g}$ (persona en movimiento, por ejemplo agachándose, parada con el torso vertical),
- o cualquier otro par de vectores \mathbf{a}_{case} , $\mathbf{a}_{\mathbf{g}}$ que cumpla que $\|\mathbf{a}_{\mathbf{g}}\|_2 = 1 \text{ g}$, y que $\mathbf{a}_{\text{case}} - \mathbf{a}_{\mathbf{g}} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \text{ g}$, hay infinitas combinaciones.

En las secciones siguientes se muestran datos de aceleración en cada eje (a_x , a_y y a_z) para distintos casos.

4.1.2.1. Giros de 90°

En la figura 4.2, se observa que el acelerómetro está quieto entre giros de 90° , siendo la aceleración $\pm 1 \text{ g}$ en uno de los ejes y 0 g en los otros dos ejes y por lo tanto $\|\mathbf{a}_{\text{meas}}\|_2 = 1 \text{ g}$. Se puede suponer que entre los giros, el acelerómetro está quieto, por lo que $\mathbf{a}_{\text{case}} = \vec{0}$, y entonces $\mathbf{a}_{\text{meas}} = -\mathbf{a}_{\mathbf{g}}$, por lo que se puede determinar la orientación del acelerómetro:

- eje z hacia arriba (persona parada) entre 0 s y 5 s,
- eje x hacia arriba (persona acostada boca arriba) entre 5 s y 15 s,
- eje y hacia abajo (persona acostada sobre su hombro izquierdo) entre 15 s y 25 s,

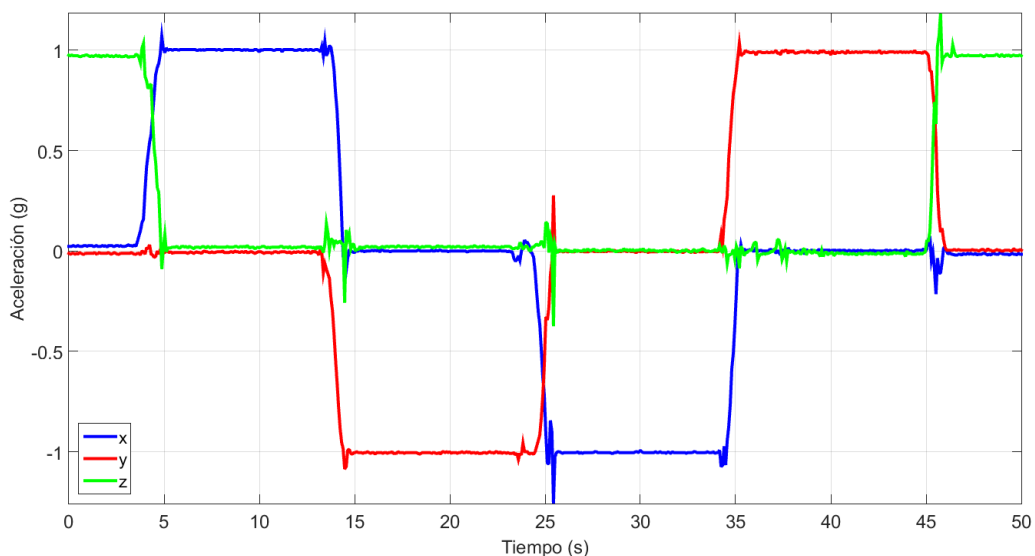


Figura 4.2: Giros de 90° del acelerómetro.

- eje x hacia abajo (persona acostada boca abajo) entre 25 s y 35 s,
- eje y hacia arriba (persona acostada sobre su hombro derecho) entre 35 s y 45 s,
- eje z hacia arriba (persona parada) entre 45 s y 50 s.

Por lo tanto, se puede concluir que hay una “acostada” en el segundo 5, 3 giros de 90° hacia la izquierda en los segundos 15, 25 y 35, y una “levantada” en el segundo 45. A su vez, la “acostada” duró 40 s, desde el segundo 5 hasta el segundo 45.

Este es el principio básico que se utilizará para determinar la orientación de la persona, el tiempo que la persona se encuentra en cada posición, la cantidad de giros de 90° hacia la izquierda y derecha, y las “acostadas” y “levantadas” y sus duraciones respectivas.

4.1.2.2. Movimiento en cada eje

En la figura 4.3, se observa que el acelerómetro se mueve hacia un lado y otro según un eje: primero el eje x, luego el eje y y finalmente el eje z. Se observa además que el valor medio de a_x y a_y es aproximadamente 0 g, mientras que el de a_z es aproximadamente 1 g, por lo que se puede suponer que la aceleración gravitatoria es $\mathbf{a}_g = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ g constante, es decir que la persona se encuentra parada durante los 14 s. Se tiene entonces que $\mathbf{a}_{\text{meas}} = \mathbf{a}_{\text{case}} - \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ g, y por lo tanto, la aceleración debida al movimiento de la persona es $\mathbf{a}_{\text{case}} = \mathbf{a}_{\text{meas}} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix}$ g. Esta aceleración \mathbf{a}_{case} es la que se debe utilizar para clasificar movimiento.

Capítulo 4. Acelerómetro

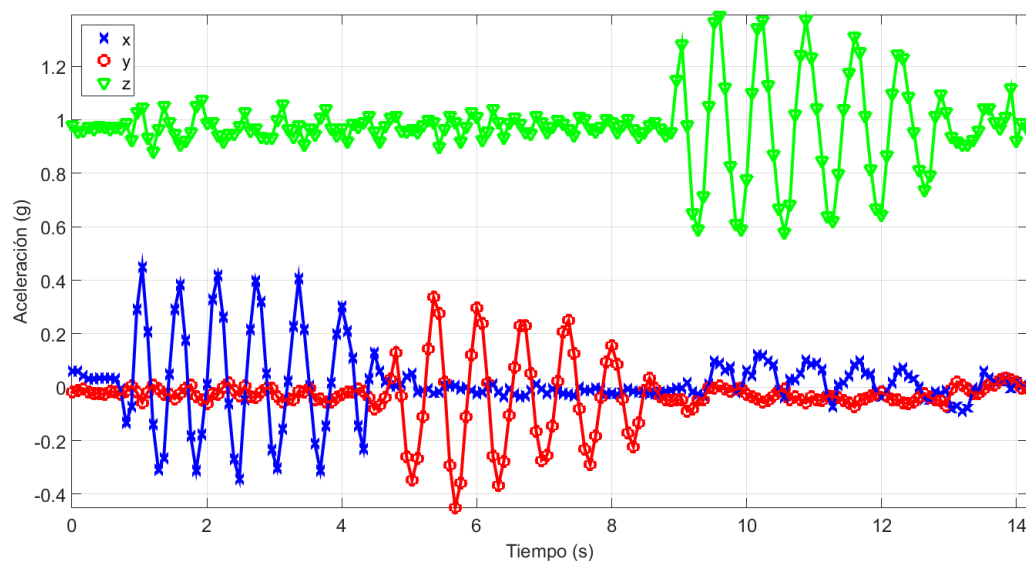


Figura 4.3: Movimiento del acelerómetro en cada eje.

Asumiendo que \mathbf{a}_{case} tiene valor medio $\vec{0}g$ y es de mayor frecuencia que \mathbf{a}_g , se puede observar que una manera intuitiva de separar \mathbf{a}_{case} de $-\mathbf{a}_g$ es a través de un filtro pasa-altos. $-\mathbf{a}_g$ correspondería a la componente de continua y frecuencias bajas de \mathbf{a}_{meas} , mientras que \mathbf{a}_{case} correspondería a la componente de frecuencias altas de \mathbf{a}_{meas} . Este es el principio básico que se utilizará para separar ambas aceleraciones, dando lugar a la determinación de la orientación de la persona cuando está en movimiento, así como la clasificación en rangos de movimiento.

4.1.2.3. Caída del acelerómetro

En la figura 4.4 se muestra la aceleración registrada con el acelerómetro en caída libre. Se ven dos caídas en la figura.

En primer lugar se dejó caer el acelerómetro. Se observa que el acelerómetro está quieto antes de dejarlo caer. Luego, se suelta el mismo (a los 0,3s), y mientras está en caída libre (entre 0,4s y 0,6s), se tiene $\mathbf{a}_{\text{meas}} = \vec{0}g$. Finalmente, cuando cae sufre un pico de aceleración en la dirección que cayó (eje z) debido a la fuerza de la superficie sobre la que cayó (entre 0,7s y 0,8s), estabilizándose nuevamente en 1g a partir de 0,75s.

Luego se tiró el acelerómetro hacia arriba. Se puede ver que el acelerómetro está quieto antes de tirarlo. Luego, al ser impulsado hacia arriba sufre una aceleración que se suma a la gravedad (a los 1,8s). Al estar en caída libre, se tiene $a_x = a_y = a_z = 0g$ (entre 2,2s y 2,6s). A partir de aquí se repite lo mismo que se observó en la primera caída.

4.2. Calibración del acelerómetro

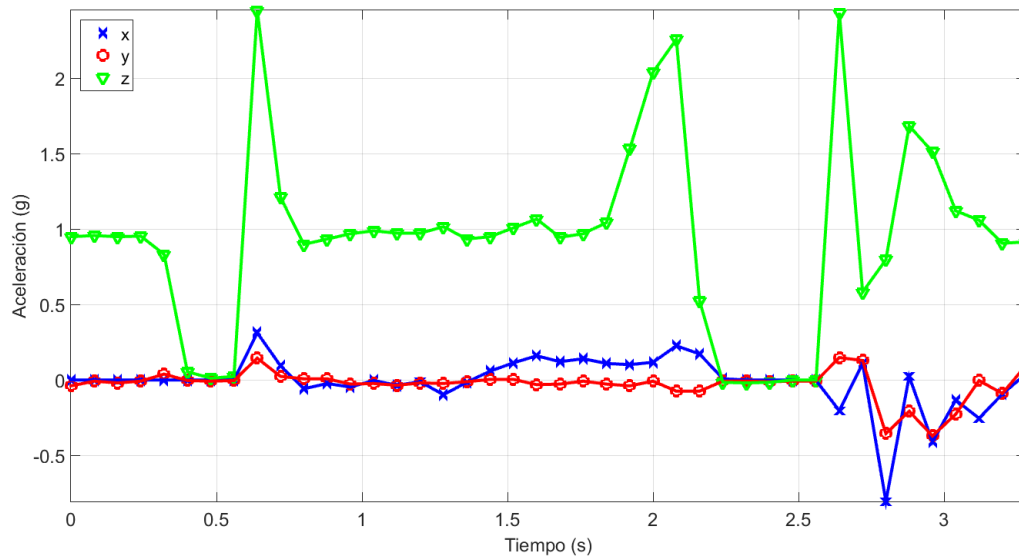


Figura 4.4: Caída del acelerómetro.

4.2. Calibración del acelerómetro

Según *Analog Devices*, fabricante del acelerómetro ADXL345 que se usará, las principales fuentes de error de los acelerómetros MEMS [12] son los siguientes.

- El “*zero g bias error*”, que corresponde a la desviación de 0g cuando el acelerómetro no sufre ningún estímulo. Los acelerómetros MEMS presentan al menos dos fuentes de “*zero g bias error*”: un error inicial y un error debido a la temperatura.
- El “*scale factor error*”, que corresponde a la desviación de la sensibilidad ideal del acelerómetro. Al igual que el “*zero g bias error*”, el “*scale factor error*” presenta una componente de error inicial y una de variación con la temperatura.
- El ruido Gaussiano de cierta densidad de ruido.

Según la hoja de datos del acelerómetro ADXL345 [3], el ruido según los ejes x e y es de 0,75 LSB rms y según el eje z es de 1,1 LSB rms a las frecuencias para las cuales se utilizará el acelerómetro. No se tendrá en cuenta el ruido en virtud de nuestra aplicación.

Se buscó en la literatura métodos de calibración de acelerómetros de 3 ejes, encontrando distintas versiones del mismo método: el método de calibración de 6 puntos [13–17].

4.2.1. Método de calibración de 6 puntos

Se tiene el siguiente modelo

Capítulo 4. Acelerómetro

$$\mathbf{a}_{\text{meas}} = \mathbb{G}\mathbf{a}_{\text{real}} + \mathbf{a}_{\text{offset}}$$

donde \mathbf{a}_{meas} es la aceleración medida (del inglés *measured*) por el acelerómetro en múltiplos de g , \mathbb{G} es la matriz de sensibilidad, \mathbf{a}_{real} es la aceleración real en múltiplos de g y $\mathbf{a}_{\text{offset}}$ es el *offset* de aceleración en múltiplos de g .

La idea es que el vector de *offset* de aceleración $\mathbf{a}_{\text{offset}}$ elimine el “*zero g bias error*”, mientras que la matriz de sensibilidad \mathbb{G} elimine el “*scale factor error*”.

- Si no tuviéramos “*zero g bias error*”, tendríamos $\mathbf{a}_{\text{offset}} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} g$, y entonces $\mathbf{a}_{\text{meas}} = \mathbb{G}\mathbf{a}_{\text{real}}$.
- Si no tuviéramos “*scale factor error*”, tendríamos $\mathbb{G} = \mathbb{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, y entonces $\mathbf{a}_{\text{meas}} = \mathbf{a}_{\text{real}} + \mathbf{a}_{\text{offset}}$.
- Si no tuviéramos ninguno de esos errores, tendríamos $\mathbf{a}_{\text{offset}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} g$ y $\mathbb{G} = \mathbb{I} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, y entonces $\mathbf{a}_{\text{meas}} = \mathbf{a}_{\text{real}}$.

Como ya se mencionó, tanto el *offset* de aceleración $\mathbf{a}_{\text{offset}}$ como la matriz de sensibilidad \mathbb{G} dependen de la temperatura. Es decir que $\mathbf{a}_{\text{offset}} = \mathbf{a}_{\text{offset}}(T)$ y $\mathbb{G} = \mathbb{G}(T)$. De la hoja de datos del acelerómetro ADXL345 [3], se obtuvo

- *0 g Offset vs. temperature for x-, y-Axes*: $\pm 0,4 \text{ mg}/^\circ\text{C}$
- *0 g Offset vs. temperature for z-axis*: $\pm 1,2 \text{ mg}/^\circ\text{C}$
- *Sensitivity change due to temperature*: $\pm 0,01 \text{ } \%/^\circ\text{C}$

Se utilizará el dispositivo en el rango de temperaturas entre -10°C y 40°C (ver requerimientos). Suponiendo que el dispositivo es calibrado a una temperatura ambiente de 15°C , entonces la máxima diferencia de temperatura será de 25°C tanto hacia un extremo (-10°C) como hacia el otro (40°C). Teniendo esto en cuenta, y utilizando los datos ya mencionados, se obtiene

- Máximo cambio en el *offset* para los ejes x e y

$$(\pm 0,4 \text{ mg}/^\circ\text{C}) (25^\circ\text{C}) = \pm 10 \text{ mg} = \pm 2,5 \text{ LSB}$$

- Máximo cambio en el *offset* para el eje z

$$(\pm 1,2 \text{ mg}/^\circ\text{C}) (25^\circ\text{C}) = \pm 30 \text{ mg} = \pm 7,5 \text{ LSB}$$

- Máximo cambio en la sensibilidad

$$(\pm 0,01 \text{ } \%/^\circ\text{C}) (25^\circ\text{C}) = \pm 0,25 \text{ } \% = \pm 0,64 \text{ LSB}$$

4.2. Calibración del acelerómetro

Se observa que el efecto de la temperatura es despreciable para el *offset* de los ejes x e y y para la sensibilidad y está en el límite de lo aceptable para el *offset* del eje z . Dicho efecto podría paliarse repitiendo la calibración cada cierto tiempo, por ejemplo mensualmente.

Se despreciará entonces el efecto de la temperatura en la calibración. Una posible forma de incluir la temperatura en la calibración (que no se implementó, quedó como trabajo futuro) es calibrar el acelerómetro a más de un temperatura (por ejemplo a T_1 y T_2) y luego interpolar linealmente los valores de calibración cada cierto tiempo para temperaturas T intermedias [15], obteniendo

$$\begin{cases} \mathbf{a}_{\text{offset}}(T) &= \left(\frac{T-T_2}{T_1-T_2}\right) \mathbf{a}_{\text{offset}}(T_1) + \left(\frac{T-T_1}{T_2-T_1}\right) \mathbf{a}_{\text{offset}}(T_2) \\ \mathbb{G}(T) &= \left(\frac{T-T_2}{T_1-T_2}\right) \mathbb{G}(T_1) + \left(\frac{T-T_1}{T_2-T_1}\right) \mathbb{G}(T_2) \end{cases}$$

4.2.1.1. Modelo con \mathbb{G} matriz diagonal

En una primera aproximación (modelo simple), se tiene que \mathbb{G} es una matriz diagonal y $\mathbf{a}_{\text{offset}}$ un vector de 3 valores, con un valor de *offset* en cada eje.

$$\begin{bmatrix} a_{\text{meas}x} \\ a_{\text{meas}y} \\ a_{\text{meas}z} \end{bmatrix} = \begin{bmatrix} G_x & 0 & 0 \\ 0 & G_y & 0 \\ 0 & 0 & G_z \end{bmatrix} \begin{bmatrix} a_{\text{real}x} \\ a_{\text{real}y} \\ a_{\text{real}z} \end{bmatrix} + \begin{bmatrix} a_{\text{offset}x} \\ a_{\text{offset}y} \\ a_{\text{offset}z} \end{bmatrix}$$

$$\begin{cases} a_{\text{meas}x} = G_x a_{\text{real}x} + a_{\text{offset}x} \\ a_{\text{meas}y} = G_y a_{\text{real}y} + a_{\text{offset}y} \\ a_{\text{meas}z} = G_z a_{\text{real}z} + a_{\text{offset}z} \end{cases}$$

El objetivo es determinar la matriz \mathbb{G} y el vector $\mathbf{a}_{\text{offset}}$, es decir determinar G_x , G_y , G_z , $a_{\text{offset}x}$, $a_{\text{offset}y}$ y $a_{\text{offset}z}$ para poder obtener los valores de aceleración calibrados (\mathbf{a}_{real}) a partir de los valores de aceleración medidos por el acelerómetro (\mathbf{a}_{meas}).

Para hallar dichos valores, se procede a tomar medidas \mathbf{a}_{meas} de posiciones conocidas. Asumiendo que el acelerómetro se encuentra en un paralelepípedo rectangular cuyas caras están alineadas con los ejes de dicho acelerómetro, está claro que lo más sencillo es medir \mathbf{a}_{meas} en cada una de las 6 caras de dicho paralelepípedo. Se usará la notación $\mathbf{a}_{\text{meas}}^{+j}$ para denotar al vector de aceleración medida que se obtiene al poner el acelerómetro con su eje $j \in \{x, y, z\}$ apuntando hacia arriba. Análogamente, se usará $\mathbf{a}_{\text{meas}}^{-j}$ para el caso del eje j apuntando hacia abajo.

Medición 1: se obtiene $\mathbf{a}_{\text{meas}}^{+x}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} g \implies \begin{cases} a_{\text{meas}x}^{+x} &= G_x (1g) + a_{\text{offset}x} \\ a_{\text{meas}y}^{+x} &= a_{\text{offset}y} \\ a_{\text{meas}z}^{+x} &= a_{\text{offset}z} \end{cases}$$

Capítulo 4. Acelerómetro

Medición 2: se obtiene $\mathbf{a}_{\text{meas}}^{-x}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \mathbf{g} \implies \begin{cases} a_{\text{meas}x}^{-x} = -G_x (1 \text{ g}) + a_{\text{offset}x} \\ a_{\text{meas}y}^{-x} = a_{\text{offset}y} \\ a_{\text{meas}z}^{-x} = a_{\text{offset}z} \end{cases}$$

Medición 3: se obtiene $\mathbf{a}_{\text{meas}}^{+y}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \mathbf{g} \implies \begin{cases} a_{\text{meas}x}^{+y} = a_{\text{offset}x} \\ a_{\text{meas}y}^{+y} = G_y (1 \text{ g}) + a_{\text{offset}y} \\ a_{\text{meas}z}^{+y} = a_{\text{offset}z} \end{cases}$$

Medición 4: se obtiene $\mathbf{a}_{\text{meas}}^{-y}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \mathbf{g} \implies \begin{cases} a_{\text{meas}x}^{-y} = a_{\text{offset}x} \\ a_{\text{meas}y}^{-y} = -G_y (1 \text{ g}) + a_{\text{offset}y} \\ a_{\text{meas}z}^{-y} = a_{\text{offset}z} \end{cases}$$

Medición 5: se obtiene $\mathbf{a}_{\text{meas}}^{+z}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{g} \implies \begin{cases} a_{\text{meas}x}^{+z} = a_{\text{offset}x} \\ a_{\text{meas}y}^{+z} = a_{\text{offset}y} \\ a_{\text{meas}z}^{+z} = G_z (1 \text{ g}) + a_{\text{offset}z} \end{cases}$$

Medición 6: se obtiene $\mathbf{a}_{\text{meas}}^{-z}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \mathbf{g} \implies \begin{cases} a_{\text{meas}x}^{-z} = a_{\text{offset}x} \\ a_{\text{meas}y}^{-z} = a_{\text{offset}y} \\ a_{\text{meas}z}^{-z} = -G_z (1 \text{ g}) + a_{\text{offset}z} \end{cases}$$

De las mediciones 1 y 2, se tiene el siguiente sistema de 2 ecuaciones y 2 incógnitas (G_x y $a_{\text{offset}x}$).

$$\begin{cases} a_{\text{meas}x}^{+x} = G_x (1 \text{ g}) + a_{\text{offset}x} \\ a_{\text{meas}x}^{-x} = -G_x (1 \text{ g}) + a_{\text{offset}x} \end{cases}$$

Resolviendo el sistema anterior, se tiene

$$\begin{cases} a_{\text{offset}x} = \frac{a_{\text{meas}x}^{+x} + a_{\text{meas}x}^{-x}}{2} \\ G_x = \frac{a_{\text{meas}x}^{+x} - a_{\text{meas}x}^{-x}}{2(1 \text{ g})} \end{cases}$$

4.2. Calibración del acelerómetro

Procediendo de manera análoga para los ejes y (mediciones 3 y 4) y z (mediciones 5 y 6), se tiene

$$\mathbb{G} = \begin{bmatrix} G_x & 0 & 0 \\ 0 & G_y & 0 \\ 0 & 0 & G_z \end{bmatrix} = \frac{1}{2(1g)} \begin{bmatrix} a_{measx}^{+x} - a_{measx}^{-x} & 0 & 0 \\ 0 & a_{measy}^{+y} - a_{measy}^{-y} & 0 \\ 0 & 0 & a_{measz}^{+z} - a_{measz}^{-z} \end{bmatrix}$$

$$\mathbf{a}_{offset} = \begin{bmatrix} a_{offsetx} \\ a_{offsety} \\ a_{offsetz} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} a_{measx}^{+x} + a_{measx}^{-x} \\ a_{measy}^{+y} + a_{measy}^{-y} \\ a_{measz}^{+z} + a_{measz}^{-z} \end{bmatrix}$$

Observación: Se busca hallar G_x , G_y , G_z , $a_{offsetx}$, $a_{offsety}$ y $a_{offsetz}$ (6 constantes). Siguiendo este método, sólo se utilizan a_{measx}^{+x} , a_{measx}^{-x} , a_{measy}^{+y} , a_{measy}^{-y} , a_{measz}^{+z} y a_{measz}^{-z} (6 constantes), dando lugar a un sistema lineal compatible determinado de 6 ecuaciones y 6 incógnitas. Notar que no se utilizan a_{measx}^{+y} , a_{measx}^{-y} , a_{measz}^{+x} , a_{measz}^{-x} , a_{measy}^{+z} , a_{measy}^{-z} , a_{measx}^{+x} , a_{measx}^{-x} , a_{measz}^{+y} ni a_{measz}^{-y} . Estas constantes sólo nos dan información de \mathbf{a}_{offset} y no de \mathbb{G} . Se podría entonces promediar, por ejemplo,

$$a_{offsetx} = \frac{1}{5} \left(a_{measx}^{+y} + a_{measx}^{-y} + a_{measz}^{+z} + a_{measz}^{-z} + \frac{a_{measx}^{+x} + a_{measx}^{-x}}{2} \right)$$

También se pueden promediar los valores tomados en el tiempo, por ejemplo tomar 100 medidas de aceleración consecutivas y luego promediarlas.

Finalmente, obtenidos $\mathbb{G} = \begin{bmatrix} G_x & 0 & 0 \\ 0 & G_y & 0 \\ 0 & 0 & G_z \end{bmatrix}$ y $\mathbf{a}_{offset} = \begin{bmatrix} a_{offsetx} \\ a_{offsety} \\ a_{offsetz} \end{bmatrix}$, se calcula \mathbf{a}_{real} .

$$\mathbf{a}_{real} = \mathbb{G}^{-1}(\mathbf{a}_{meas} - \mathbf{a}_{offset}) = \begin{bmatrix} G_x & 0 & 0 \\ 0 & G_y & 0 \\ 0 & 0 & G_z \end{bmatrix}^{-1} \begin{bmatrix} a_{measx} - a_{offsetx} \\ a_{measy} - a_{offsety} \\ a_{measz} - a_{offsetz} \end{bmatrix}$$

$$\mathbf{a}_{real} = \begin{bmatrix} \frac{1}{G_x} & 0 & 0 \\ 0 & \frac{1}{G_y} & 0 \\ 0 & 0 & \frac{1}{G_z} \end{bmatrix} \begin{bmatrix} a_{measx} - a_{offsetx} \\ a_{measy} - a_{offsety} \\ a_{measz} - a_{offsetz} \end{bmatrix}$$

4.2.1.2. Modelo con \mathbb{G} matriz genérica (no necesariamente matriz diagonal)

En este modelo, se tiene que \mathbb{G} es una matriz genérica, no necesariamente una matriz diagonal. Dicha matriz contemplaría la matriz de sensibilidad utilizada en el modelo anterior (matriz diagonal), pero además incluye el efecto de rotación del encapsulado del acelerómetro respecto al PCB y la interferencia entre ejes de medición.

$$\begin{bmatrix} a_{measx} \\ a_{measy} \\ a_{measz} \end{bmatrix} = \begin{bmatrix} G_{xx} & G_{yx} & G_{zx} \\ G_{xy} & G_{yy} & G_{zy} \\ G_{xz} & G_{yz} & G_{zz} \end{bmatrix} \begin{bmatrix} a_{realx} \\ a_{realy} \\ a_{realz} \end{bmatrix} + \begin{bmatrix} a_{offsetx} \\ a_{offsety} \\ a_{offsetz} \end{bmatrix}$$

Capítulo 4. Acelerómetro

El objetivo es determinar la matriz \mathbb{G} y el vector $\mathbf{a}_{\text{offset}}$, es decir determinar $G_{xx}, G_{yx}, G_{zx}, G_{xy}, G_{yy}, G_{zy}, G_{xz}, G_{yz}, G_{zz}, a_{\text{offset}x}, a_{\text{offset}y}$ y $a_{\text{offset}z}$. En este caso, tenemos 12 constantes a determinar y no 6 como en el caso anterior.

Las mediciones a realizar son las mismas 6 que en el caso anterior.

Medición 1: se obtiene $\mathbf{a}_{\text{meas}}^{+x}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \text{ g} \implies \begin{cases} a_{\text{meas}x}^{+x} = G_{xx} (1 \text{ g}) + a_{\text{offset}x} \\ a_{\text{meas}y}^{+x} = G_{xy} (1 \text{ g}) + a_{\text{offset}y} \\ a_{\text{meas}z}^{+x} = G_{xz} (1 \text{ g}) + a_{\text{offset}z} \end{cases}$$

Medición 2: se obtiene $\mathbf{a}_{\text{meas}}^{-x}$

$$\mathbf{a}_{\text{real}} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \text{ g} \implies \begin{cases} a_{\text{meas}x}^{-x} = -G_{xx} (1 \text{ g}) + a_{\text{offset}x} \\ a_{\text{meas}y}^{-x} = -G_{xy} (1 \text{ g}) + a_{\text{offset}y} \\ a_{\text{meas}z}^{-x} = -G_{xz} (1 \text{ g}) + a_{\text{offset}z} \end{cases}$$

De las mediciones 1 y 2, se tienen las siguientes ecuaciones.

$$\begin{cases} G_{xx} = \frac{a_{\text{meas}x}^{+x} - a_{\text{meas}x}^{-x}}{2(1 \text{ g})} \\ G_{xy} = \frac{a_{\text{meas}y}^{+x} - a_{\text{meas}y}^{-x}}{2(1 \text{ g})} \\ G_{xz} = \frac{a_{\text{meas}z}^{+x} - a_{\text{meas}z}^{-x}}{2(1 \text{ g})} \end{cases}$$

$$\begin{cases} a_{\text{offset}x} = \frac{a_{\text{meas}x}^{+x} + a_{\text{meas}x}^{-x}}{2} \\ a_{\text{offset}y} = \frac{a_{\text{meas}y}^{+x} + a_{\text{meas}y}^{-x}}{2} \\ a_{\text{offset}z} = \frac{a_{\text{meas}z}^{+x} + a_{\text{meas}z}^{-x}}{2} \end{cases}$$

Procediendo de manera análoga para los ejes y (mediciones 3 y 4) y z (mediciones 5 y 6), se tiene

$$\mathbb{G} = \begin{bmatrix} G_{xx} & G_{yx} & G_{zx} \\ G_{xy} & G_{yy} & G_{zy} \\ G_{xz} & G_{yz} & G_{zz} \end{bmatrix} = \frac{1}{2(1 \text{ g})} \left[\mathbf{a}_{\text{meas}}^{+x} - \mathbf{a}_{\text{meas}}^{-x} \quad \mathbf{a}_{\text{meas}}^{+y} - \mathbf{a}_{\text{meas}}^{-y} \quad \mathbf{a}_{\text{meas}}^{+z} - \mathbf{a}_{\text{meas}}^{-z} \right]$$

$$\begin{cases} a_{\text{offset}x} = \frac{a_{\text{meas}x}^{+x} + a_{\text{meas}x}^{-x}}{2} = \frac{a_{\text{meas}x}^{+y} + a_{\text{meas}x}^{-y}}{2} = \frac{a_{\text{meas}x}^{+z} + a_{\text{meas}x}^{-z}}{2} \\ a_{\text{offset}y} = \frac{a_{\text{meas}y}^{+x} + a_{\text{meas}y}^{-x}}{2} = \frac{a_{\text{meas}y}^{+y} + a_{\text{meas}y}^{-y}}{2} = \frac{a_{\text{meas}y}^{+z} + a_{\text{meas}y}^{-z}}{2} \\ a_{\text{offset}z} = \frac{a_{\text{meas}z}^{+x} + a_{\text{meas}z}^{-x}}{2} = \frac{a_{\text{meas}z}^{+y} + a_{\text{meas}z}^{-y}}{2} = \frac{a_{\text{meas}z}^{+z} + a_{\text{meas}z}^{-z}}{2} \end{cases}$$

4.2. Calibración del acelerómetro

Se puede ver que se tienen muchas ecuaciones para determinar $a_{offsetx}$, $a_{offsety}$ y $a_{offsetz}$. Se podría utilizar la ecuación correspondiente a su respectivo eje que naturalmente tendría menor error relativo, por ejemplo

$$\begin{cases} a_{offsetx} = \frac{a_{measx}^{+x} + a_{measx}^{-x}}{2} \\ a_{offsety} = \frac{a_{measy}^{+y} + a_{measy}^{-y}}{2} \\ a_{offsetz} = \frac{a_{measz}^{+z} + a_{measz}^{-z}}{2} \end{cases}$$

También se podría promediar, por ejemplo

$$\begin{cases} a_{offsetx} = \frac{1}{3} \left(\frac{a_{measx}^{+x} + a_{measx}^{-x}}{2} + \frac{a_{measx}^{+y} + a_{measx}^{-y}}{2} + \frac{a_{measx}^{+z} + a_{measx}^{-z}}{2} \right) \\ a_{offsety} = \frac{1}{3} \left(\frac{a_{measy}^{+x} + a_{measy}^{-x}}{2} + \frac{a_{measy}^{+y} + a_{measy}^{-y}}{2} + \frac{a_{measy}^{+z} + a_{measy}^{-z}}{2} \right) \\ a_{offsetz} = \frac{1}{3} \left(\frac{a_{measz}^{+x} + a_{measz}^{-x}}{2} + \frac{a_{measz}^{+y} + a_{measz}^{-y}}{2} + \frac{a_{measz}^{+z} + a_{measz}^{-z}}{2} \right) \end{cases}$$

También se pueden promediar los valores tomados en el tiempo, por ejemplo tomar 100 medidas de aceleración consecutivas y luego promediarlas.

Finalmente, obtenidos $\mathbb{G} = \begin{bmatrix} G_{xx} & G_{yx} & G_{zx} \\ G_{xy} & G_{yy} & G_{zy} \\ G_{xz} & G_{yz} & G_{zz} \end{bmatrix}$ y $\mathbf{a}_{offset} = \begin{bmatrix} a_{offsetx} \\ a_{offsety} \\ a_{offsetz} \end{bmatrix}$, se calcula \mathbf{a}_{real} .

$$\mathbf{a}_{real} = \mathbb{G}^{-1}(\mathbf{a}_{meas} - \mathbf{a}_{offset}) = \begin{bmatrix} G_{xx} & G_{yx} & G_{zx} \\ G_{xy} & G_{yy} & G_{zy} \\ G_{xz} & G_{yz} & G_{zz} \end{bmatrix}^{-1} \begin{bmatrix} a_{measx} - a_{offsetx} \\ a_{measy} - a_{offsety} \\ a_{measz} - a_{offsetz} \end{bmatrix}$$

4.2.2. Calibración realizada

La tabla 4.2 muestra los resultados de las 6 mediciones descritas en 4.2.1 realizadas para calibrar uno de los prototipos fabricados (ver capítulo 8).

Medición	Vector obtenido	Eje x	Eje y	Eje z
1	\mathbf{a}_{meas}^{+x}	0,988 g	-0,057 g	-0,051 g
2	\mathbf{a}_{meas}^{-x}	-0,996 g	-0,039 g	-0,056 g
3	\mathbf{a}_{meas}^{+y}	0,004 g	0,930 g	-0,069 g
4	\mathbf{a}_{meas}^{-y}	-0,002 g	-1,025 g	-0,048 g
5	\mathbf{a}_{meas}^{+z}	0,003 g	-0,045 g	0,895 g
6	\mathbf{a}_{meas}^{-z}	0,006 g	-0,040 g	-1,024 g

Tabla 4.2: Calibración del acelerómetro

A partir de las mediciones anteriores y siguiendo el método de calibración de 6 puntos descrito en 4.2.1, se obtiene la matriz \mathbb{G} y el vector \mathbf{a}_{offset} .

Capítulo 4. Acelerómetro

- Para el modelo con \mathbb{G} matriz diagonal (ver 4.2.1.1), se obtiene

$$\mathbb{G} = \begin{bmatrix} 0,992 & 0 & 0 \\ 0 & 0,977 & 0 \\ 0 & 0 & 0,961 \end{bmatrix} \text{ y } \mathbf{a}_{\text{offset}} = \begin{bmatrix} -0,004 \\ -0,047 \\ -0,066 \end{bmatrix} \text{ g.}$$

- Para el modelo con \mathbb{G} matriz genérica (no necesariamente diagonal, ver 4.2.1.2), se obtiene

$$\mathbb{G} = \begin{bmatrix} 0,992 & 0,004 & 0 \\ -0,008 & 0,977 & -0,004 \\ 0,004 & -0,012 & 0,961 \end{bmatrix} \text{ y } \mathbf{a}_{\text{offset}} = \begin{bmatrix} -0,004 \\ -0,047 \\ -0,066 \end{bmatrix} \text{ g.}$$

Se observa que para ambos casos, la matriz \mathbb{G} obtenida es similar a la matriz identidad y el vector $\mathbf{a}_{\text{offset}}$ obtenido es similar al vector nulo, lo cual era de esperar. Se observa también que ambos métodos difieren poco: ambas matrices \mathbb{G} son muy parecidas, mientras que se obtiene el mismo vector $\mathbf{a}_{\text{offset}}$.

De la hoja de datos del acelerómetro ADXL345 [3], se obtiene que

- la sensibilidad mínima para los ejes x, y, z es 230 LSB/g, mientras que la máxima es 282 LSB/g,
- el *offset* máximo para los ejes x e y es 150 mg (hacia ambos lados),
- el *offset* máximo para el eje z es 250 mg (hacia ambos lados).

Se observa entonces que los datos obtenidos están en el rango dado por la hoja de datos, ya que

$$\left\{ \begin{array}{l} G_x = 0,992 = 254 \text{ LSB/g} \\ G_y = 0,977 = 250 \text{ LSB/g} \\ G_z = 0,961 = 246 \text{ LSB/g} \\ a_{\text{offset}x} = -0,004 \text{ g} = -4 \text{ mg} \\ a_{\text{offset}y} = -0,047 \text{ g} = -47 \text{ mg} \\ a_{\text{offset}z} = -0,066 \text{ g} = -66 \text{ mg} \end{array} \right.$$

4.2.3. Resultados obtenidos luego de la calibración

Para validar ambos métodos de calibración, compararlos y cuantificar la precisión que se obtiene en la medida, se realizó la siguiente prueba.

Se colocó el dispositivo sobre una superficie plana. Luego, se inclinó dicha superficie un cierto ángulo θ como se muestra en la figura 4.5. Se registraron los valores de aceleración dados por el acelerómetro para ángulos θ entre 0° y 90° , con intervalos de $2,5^\circ$.

Teóricamente, aplicando trigonometría, la aceleración obtenida en la situación de la figura 4.5 debería ser

$$\mathbf{a}_{\text{meas}} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ g} - \mathbf{a}_{\mathbf{g}} = \begin{bmatrix} 0 \\ \sin(\theta) \\ \cos(\theta) \end{bmatrix} \text{ g}$$

4.2. Calibración del acelerómetro

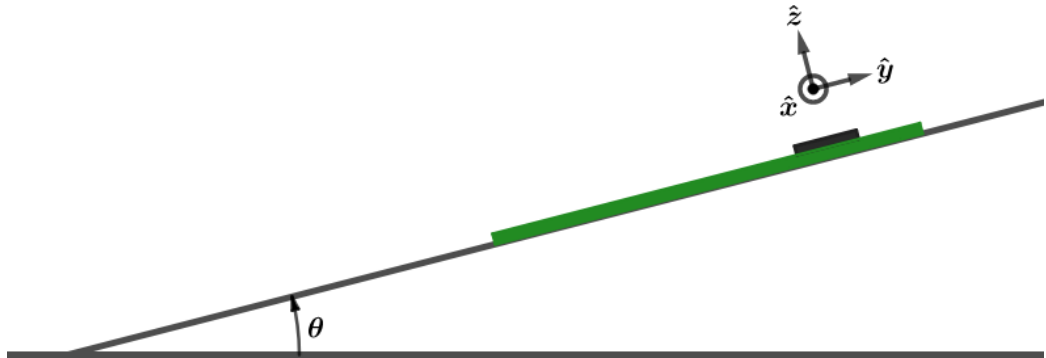


Figura 4.5: Prueba de la calibración realizada

A continuación, se muestran los resultados de la prueba realizada. Se grafican la aceleración teórica (curva), y los puntos experimentales: datos crudos de aceleración, datos calibrados según el modelo con matriz \mathbb{G} diagonal y datos calibrados según el modelo con matriz \mathbb{G} genérica.

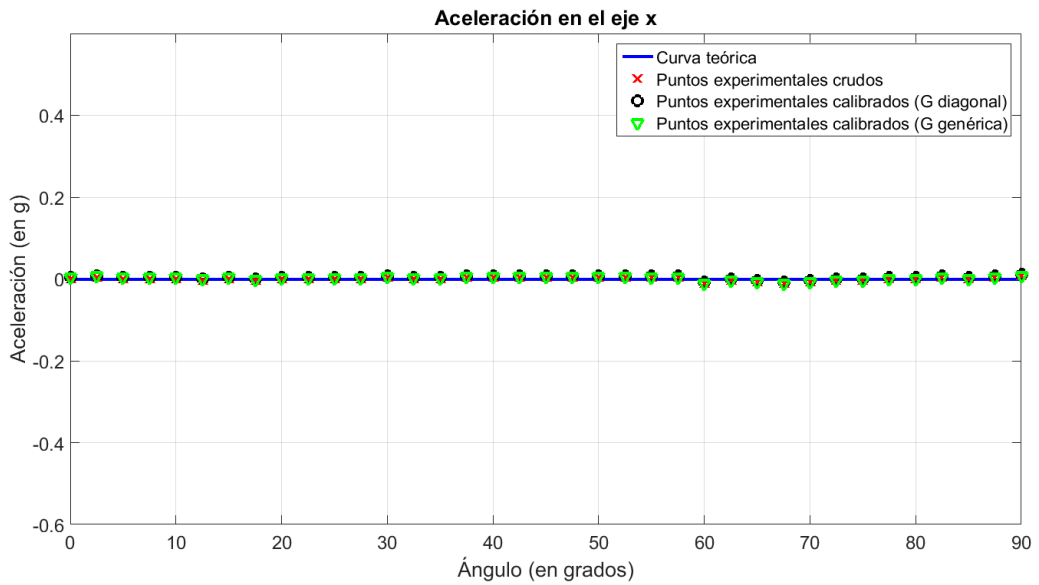


Figura 4.6: Aceleración a_x del eje x en función del ángulo θ

Se observa que en el caso particular del eje x (figura 4.6), el valor de $a_{offsetx}$ fue casi nulo por lo que todas las curvas coinciden, incluso la de los datos crudos. En un caso general, se esperaría que los datos crudos puedan variar un poco respecto a la curva teórica si $a_{offsetx}$ no fuera tan pequeño.

En los ejes y, z (figuras 4.7 y 4.8 respectivamente), se puede ver que los datos crudos de aceleración distan de la aceleración teórica, evidenciando la importancia de la calibración. Se puede ver también que los datos calibrados siguen correctamente a la curva de aceleración teórica, sin diferencias significativas entre ambos métodos de calibración.

Capítulo 4. Acelerómetro

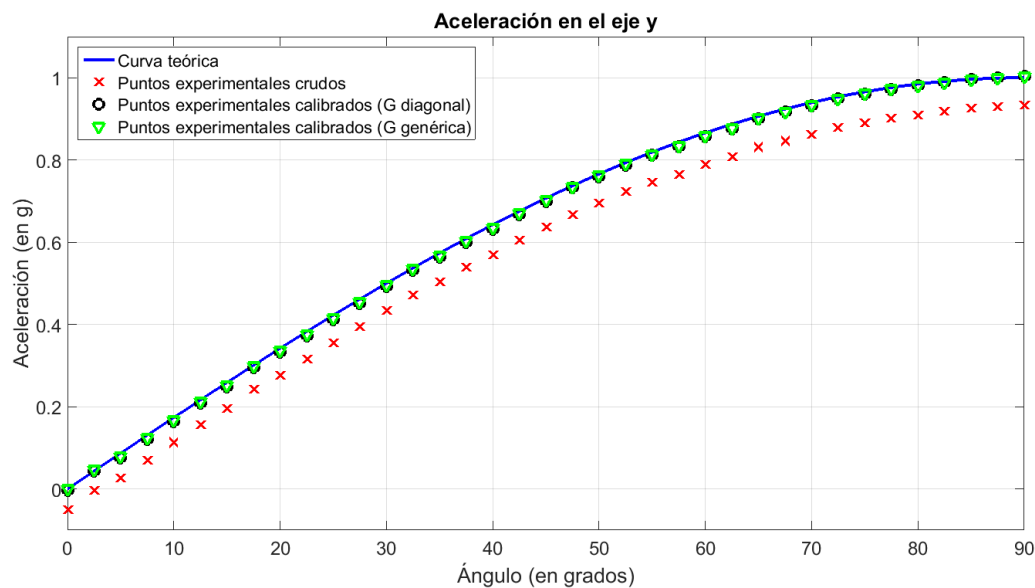


Figura 4.7: Aceleración a_y del eje y en función del ángulo θ

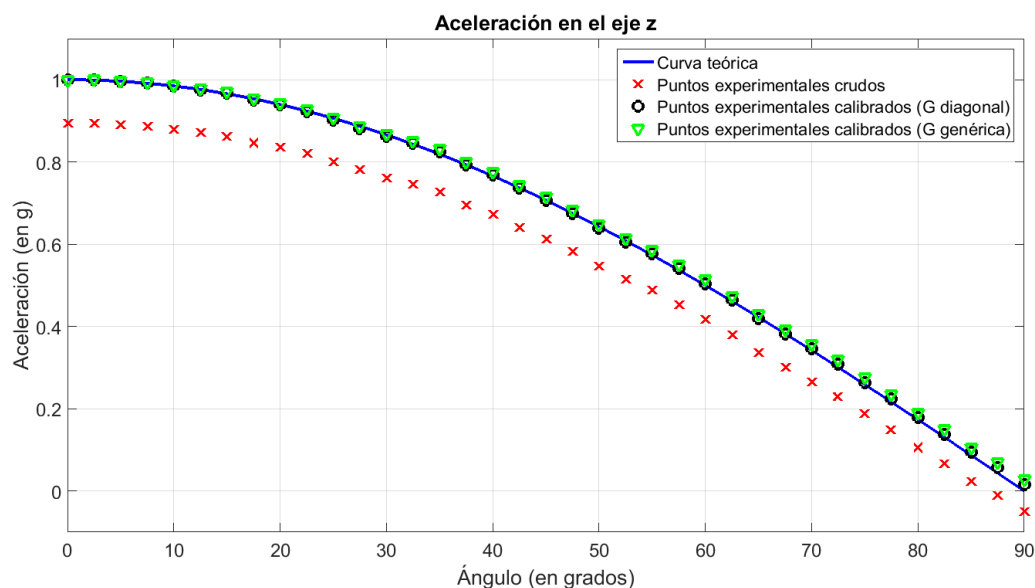


Figura 4.8: Aceleración a_z del eje z en función del ángulo θ

En lo que respecta al módulo de la aceleración, teóricamente debería ser $\|\mathbf{a}\|_2 = \sqrt{0^2 + \sin^2(\theta) + \cos^2(\theta)} g = 1 g$, lo cual es consistente con que se tomaron medidas con el acelerómetro quieto. Se puede ver que los datos calibrados se aproximan correctamente a la curva de aceleración teórica (figura 4.9). De hecho, todos los datos calibrados están comprendidos entre 0,99 g y 1,01 g. Sin embargo, los datos crudos caen todos fuera de dicho intervalo, lo que si no se calibrara el acelerómetro, llevaría a concluir erróneamente que la persona se está moviendo.

4.3. Posicionamiento del acelerómetro respecto a la persona

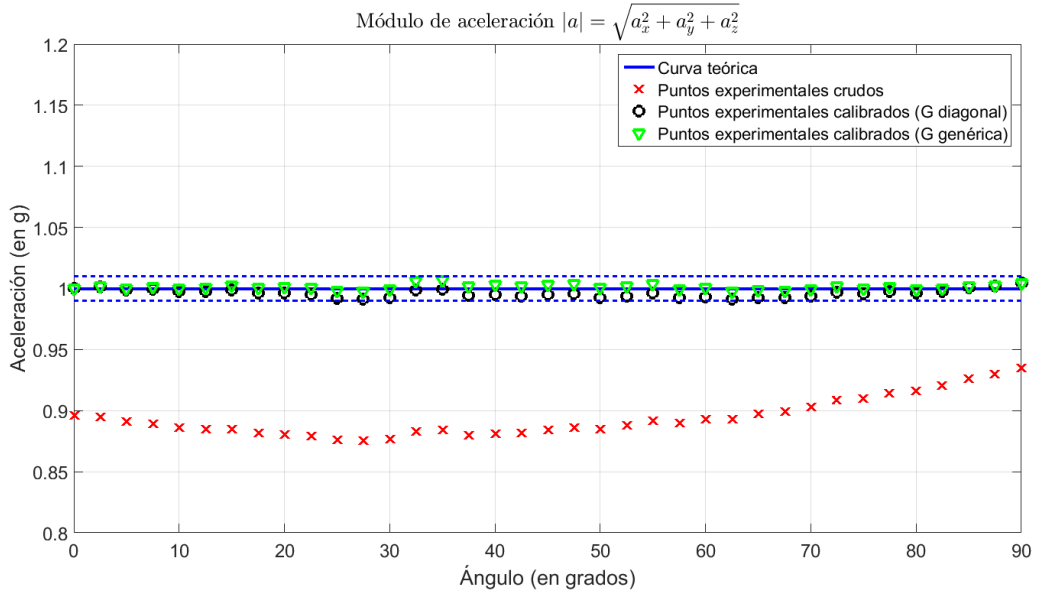


Figura 4.9: Módulo de la aceleración $\|\mathbf{a}\|_2 = \sqrt{a_x^2 + a_y^2 + a_z^2}$ en función del ángulo θ

Se concluye entonces que la calibración del acelerómetro es indispensable para obtener datos correctos y lograr determinar de forma precisa la posición y movimiento de la persona. También se concluye que no hay diferencias significativas entre ambos métodos estudiados, por lo que se utilizará el de la matriz \mathbb{G} diagonal por ser más sencillo.

4.3. Posicionamiento del acelerómetro respecto a la persona

Se llamará “posicionamiento” del acelerómetro respecto a la persona al procedimiento mediante el cual se obtiene un vector de aceleración posicionado respecto al torso de la misma según los ejes mostrados en la figura 4.10 a partir de un vector de aceleración ya calibrado según los ejes del acelerómetro.

4.3.1. Método de posicionamiento propuesto

El posicionamiento puede ser entendido como un cambio de base de \mathbb{R}^3 en \mathbb{R}^3 , de una base \mathcal{A} (relativa al acelerómetro) a una base \mathcal{P} (relativa a la persona, ver figura 4.10).

Se tiene entonces $\mathbf{a}_{\mathcal{P}} = ({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})\mathbf{a}_{\mathcal{A}}$, donde $\mathbf{a}_{\mathcal{P}}$ es el vector de aceleración según los ejes relativos a la persona, $\mathbf{a}_{\mathcal{A}}$ es el vector de aceleración según los ejes relativos al acelerómetro ya calibrado, y $({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})$ es la matriz de cambio de base de la base \mathcal{A} a la base \mathcal{P} .

Capítulo 4. Acelerómetro

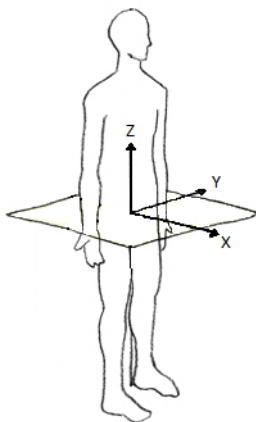


Figura 4.10: Ejes relativos a una persona

El objetivo es hallar la matriz $(\mathcal{P}\mathbb{T}_{\mathcal{A}})$, de forma de obtener $\mathbf{a}_{\mathcal{P}}$ en función de $\mathbf{a}_{\mathcal{A}}$. Para obtener dicha matriz, se realizan ciertas medidas con el acelerómetro ya posicionado firmemente en el torso de la persona.

Sea $(\mathcal{A}\mathbb{T}_{\mathcal{P}}) = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$ tal que $\mathbf{a}_{\mathcal{A}} = (\mathcal{A}\mathbb{T}_{\mathcal{P}})\mathbf{a}_{\mathcal{P}}$, es decir que es la matriz de cambio de base de la base \mathcal{P} a la base \mathcal{A} , por lo que $(\mathcal{A}\mathbb{T}_{\mathcal{P}}) = (\mathcal{P}\mathbb{T}_{\mathcal{A}})^{-1}$.

Medición 1: Persona parada

$$\mathbf{a}_{\mathcal{A}}^{+z} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \mathbf{g} = \begin{bmatrix} t_{13} \\ t_{23} \\ t_{33} \end{bmatrix} \mathbf{g}$$

Medición 2: Persona acostada sobre su hombro derecho

$$\mathbf{a}_{\mathcal{A}}^{+y} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \mathbf{g} = \begin{bmatrix} t_{12} \\ t_{22} \\ t_{32} \end{bmatrix} \mathbf{g}$$

Medición 3: Persona acostada boca arriba

$$\mathbf{a}_{\mathcal{A}}^{+x} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{g} = \begin{bmatrix} t_{11} \\ t_{21} \\ t_{31} \end{bmatrix} \mathbf{g}$$

4.3. Posicionamiento del acelerómetro respecto a la persona

Finalmente, se obtiene $({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})$:

$$({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}}) = ({}_{\mathcal{A}}\mathbb{T}_{\mathcal{P}})^{-1} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}^{-1} = [\mathbf{a}_{\mathcal{A}}^{+x} \quad \mathbf{a}_{\mathcal{A}}^{+y} \quad \mathbf{a}_{\mathcal{A}}^{+z}]^{-1} \mathbf{g}$$

Una vez calculada dicha matriz, se puede obtener $\mathbf{a}_{\mathcal{P}}$ a partir de $\mathbf{a}_{\mathcal{A}}$.

El procedimiento del posicionamiento del acelerómetro respecto al paciente sería entonces el siguiente.

- Colocar el dispositivo sujeto firmemente al torso de la persona.
- Colocar a la persona en la posición asociada a la medición 1.
- Presionar un botón en el dispositivo, indicando que la persona está en dicha posición.
- Colocar a la persona en la posición asociada a la medición 2.
- Presionar un botón en el dispositivo, indicando que la persona está en dicha posición.
- Colocar a la persona en la posición asociada a la medición 3.
- Presionar un botón en el dispositivo, indicando que la persona está en dicha posición.
- Una vez obtenidos dichos datos, se calcula la matriz $({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})$ y la almacena para obtener $\mathbf{a}_{\mathcal{P}}$ en función de $\mathbf{a}_{\mathcal{A}}$.

Está claro que ciertos errores podrían ocurrir, por ejemplo que se oprima el botón en movimiento, o que se oprima el botón dos veces consecutivas sin cambiar de posición. Este tipo de errores daría lugar a un posicionamiento incorrecto, por lo tanto a una errónea interpretación de los datos y a errores en la detección de posición y movimiento. Es por este motivo que deben realizarse ciertos chequeos para validar el posicionamiento antes de utilizarlo.

Se decidió tomar N muestras consecutivas de aceleración y no sólo una, promediándolas y obteniendo así las mediciones para determinar la matriz de posicionamiento:

$$\begin{cases} \mathbf{a}_{\mathcal{A}}^{+x} &= \frac{1}{N} \sum_{n=1}^N \mathbf{a}_{\mathcal{A}}^{+x} [n] \\ \mathbf{a}_{\mathcal{A}}^{+y} &= \frac{1}{N} \sum_{n=1}^N \mathbf{a}_{\mathcal{A}}^{+y} [n] \\ \mathbf{a}_{\mathcal{A}}^{+z} &= \frac{1}{N} \sum_{n=1}^N \mathbf{a}_{\mathcal{A}}^{+z} [n] \end{cases}$$

Capítulo 4. Acelerómetro

1. Se verifica que la diferencia entre muestras sucesivas de aceleración en cada eje sea menor a un cierto umbral ε_d para asegurar que la persona está quieta y la aceleración medida es estable. La condición que se verifica es entonces

$$\begin{cases} |a_x[n+1] - a_x[n]| < \varepsilon_d \\ |a_y[n+1] - a_y[n]| < \varepsilon_d \\ |a_z[n+1] - a_z[n]| < \varepsilon_d \end{cases} \quad \forall n \in \{1, \dots, N-1\}$$

2. Se verifica que los ejes sean normales al momento de tomar la medida, es decir que $\|\mathbf{a}_A\|_2 \approx 1 \text{ g}$. Por lo tanto la condición a verificar para cada eje es

$$1 \text{ g} - \varepsilon_a \leq \|\mathbf{a}_A\|_2 \leq 1 \text{ g} + \varepsilon_a$$

siendo $\varepsilon_a > 0$ un cierto umbral de aceleración a fijar mediante pruebas.

Como $\|\mathbf{a}_A\|_2 = \sqrt{a_{Ax}^2 + a_{Ay}^2 + a_{Az}^2}$, la condición que finalmente se utilizó para cada eje fue

$$(1 \text{ g} - \varepsilon_a)^2 \leq a_{Ax}^2 + a_{Ay}^2 + a_{Az}^2 \leq (1 \text{ g} + \varepsilon_a)^2$$

Se utilizó esta condición para no tener que computar una raíz cuadrada con el microcontrolador, sino un cuadrado.

3. Se verifica que los ejes de la persona sean “suficientemente ortogonales” es decir que el ángulo entre cualquier combinación de 2 ejes (x-y, x-z o y-z) es aproximadamente 90° . Por lo tanto las condiciones a verificar son

$$\begin{cases} 90^\circ - \varepsilon_\theta \leq \theta_{xy} \leq 90^\circ + \varepsilon_\theta \\ 90^\circ - \varepsilon_\theta \leq \theta_{xz} \leq 90^\circ + \varepsilon_\theta \\ 90^\circ - \varepsilon_\theta \leq \theta_{yz} \leq 90^\circ + \varepsilon_\theta \end{cases}$$

siendo $\varepsilon_\theta > 0$ un cierto umbral de ángulo a fijar mediante pruebas y θ_{ij} el ángulo formado entre los ejes i y j .

Se tiene entonces

$$\begin{cases} \cos(90^\circ + \varepsilon_\theta) \leq \cos(\theta_{xy}) \leq \cos(90^\circ - \varepsilon_\theta) \\ \cos(90^\circ + \varepsilon_\theta) \leq \cos(\theta_{xz}) \leq \cos(90^\circ - \varepsilon_\theta) \\ \cos(90^\circ + \varepsilon_\theta) \leq \cos(\theta_{yz}) \leq \cos(90^\circ - \varepsilon_\theta) \end{cases}$$

Como $\cos(90^\circ + \theta) = -\sin(\theta) \quad \forall \theta \in \mathbb{R}$, entonces se tiene

$$\begin{cases} -\sin(\varepsilon_\theta) \leq \cos(\theta_{xy}) \leq \sin(\varepsilon_\theta) \\ -\sin(\varepsilon_\theta) \leq \cos(\theta_{xz}) \leq \sin(\varepsilon_\theta) \\ -\sin(\varepsilon_\theta) \leq \cos(\theta_{yz}) \leq \sin(\varepsilon_\theta) \end{cases} \Rightarrow \begin{cases} |\cos(\theta_{xy})| \leq \sin(\varepsilon_\theta) \\ |\cos(\theta_{xz})| \leq \sin(\varepsilon_\theta) \\ |\cos(\theta_{yz})| \leq \sin(\varepsilon_\theta) \end{cases}$$

4.3. Posicionamiento del acelerómetro respecto a la persona

Como $\mathbf{u} \cdot \mathbf{v} = \|\mathbf{u}\|_2 \cdot \|\mathbf{v}\|_2 \cos(\theta_{uv}) \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^3$, las condiciones a verificar pueden traducirse

$$\left\{ \begin{array}{l} \left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+y} \right| \leq \left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 \cdot \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 \sin(\varepsilon_{\theta}) \\ \left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+z} \right| \leq \left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 \cdot \left\| \mathbf{a}_{\mathcal{A}}^{+z} \right\|_2 \sin(\varepsilon_{\theta}) \\ \left| \mathbf{a}_{\mathcal{A}}^{+y} \cdot \mathbf{a}_{\mathcal{A}}^{+z} \right| \leq \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 \cdot \left\| \mathbf{a}_{\mathcal{A}}^{+z} \right\|_2 \sin(\varepsilon_{\theta}) \end{array} \right.$$

Asumiendo que $\left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 = \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 = \left\| \mathbf{a}_{\mathcal{A}}^{+z} \right\|_2 \simeq 1 \text{ g}$ (observar que esta condición es justamente la que se verifica para asegurarse de que la persona esté quieta), se tiene finalmente

$$\left\{ \begin{array}{l} \left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+y} \right| \leq \sin(\varepsilon_{\theta}) \\ \left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+z} \right| \leq \sin(\varepsilon_{\theta}) \\ \left| \mathbf{a}_{\mathcal{A}}^{+y} \cdot \mathbf{a}_{\mathcal{A}}^{+z} \right| \leq \sin(\varepsilon_{\theta}) \end{array} \right.$$

¿Es razonable suponer que $\left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 = \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 = \left\| \mathbf{a}_{\mathcal{A}}^{+z} \right\|_2 \simeq 1 \text{ g}$? Con el chequeo de quieto mencionado anteriormente, se sabe para cada eje que

$$1 \text{ g} - \varepsilon_a \leq \|\mathbf{a}_{\mathcal{A}}\|_2 \leq 1 \text{ g} + \varepsilon_a$$

Por lo que si luego se verifica la condición aproximada

$$\left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+y} \right| \leq \sin(\varepsilon'_{\theta})$$

en lugar de la condición exacta

$$\left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+y} \right| \leq \left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 \cdot \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 \sin(\varepsilon_{\theta})$$

el umbral ε'_{θ} estaría errado ($\varepsilon'_{\theta} \neq \varepsilon_{\theta}$), y sería

$$\varepsilon'_{\theta} = \arcsin \left(\left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 \cdot \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 \sin(\varepsilon_{\theta}) \right)$$

En los casos límite, cuando $\left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 = \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 = 1 \text{ g} - \varepsilon_a$ y cuando $\left\| \mathbf{a}_{\mathcal{A}}^{+x} \right\|_2 = \left\| \mathbf{a}_{\mathcal{A}}^{+y} \right\|_2 = 1 \text{ g} + \varepsilon_a$, el umbral ε'_{θ} sería

$$\left\{ \begin{array}{l} \varepsilon'_{\theta \text{ mín}} = \arcsin \left((1 \text{ g} - \varepsilon_a)^2 \sin(\varepsilon_{\theta}) \right) \\ \varepsilon'_{\theta \text{ máx}} = \arcsin \left((1 \text{ g} + \varepsilon_a)^2 \sin(\varepsilon_{\theta}) \right) \end{array} \right.$$

Capítulo 4. Acelerómetro

Tomando valores razonables de $\varepsilon_a = 0,1 \text{ g}$ y $\varepsilon_\theta = 20^\circ$, se tiene

$$\begin{cases} \varepsilon'_{\theta min} &= \arcsin \left((1 \text{ g} - 0,1 \text{ g})^2 \sin(20^\circ) \right) \approx 16,08^\circ \\ \varepsilon'_{\theta max} &= \arcsin \left((1 \text{ g} + 0,1 \text{ g})^2 \sin(20^\circ) \right) \approx 24,45^\circ \end{cases}$$

No es indispensable que el umbral de ángulo ε_θ sea exacto. A los efectos del chequeo que se quiere realizar, no importa que $16,08^\circ < \varepsilon_\theta < 24,45^\circ$. La condición que finalmente se utilizó fue entonces la aproximada debido a su simplicidad:

$$\begin{cases} \left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+y} \right| \leq \sin(\varepsilon_\theta) \\ \left| \mathbf{a}_{\mathcal{A}}^{+x} \cdot \mathbf{a}_{\mathcal{A}}^{+z} \right| \leq \sin(\varepsilon_\theta) \\ \left| \mathbf{a}_{\mathcal{A}}^{+y} \cdot \mathbf{a}_{\mathcal{A}}^{+z} \right| \leq \sin(\varepsilon_\theta) \end{cases}$$

En caso de que no se cumpla alguna de todas estas condiciones, el posicionamiento no es aceptado por el dispositivo, entrando automáticamente a posicionar otra vez.

4.3.2. Resultados de posicionamientos

Con el objetivo de determinar los umbrales asociados a las verificaciones del posicionamiento mencionados anteriormente: umbral de diferencias sucesivas de aceleración ε_d para verificación de persona quieta, umbral de aceleración ε_a para verificación de ejes normales y umbral de ángulo ε_θ para verificación de ejes ortogonales, se realizaron posicionamientos “correctos”, que deberían pasar las verificaciones y “fallidos”, que no deberían pasar al menos una de las verificaciones.

En las figuras 4.11 y 4.12, se muestra respectivamente un posicionamiento “correcto” y uno “fallido” en lo que respecta al movimiento y a los ejes normales. Observar que la escala de las ordenadas de $\|\mathbf{a}\|_2$ es 0,01 g por división en la figura 4.11, y 0,05 g por división en la figura 4.12.

Está claro que la máxima diferencia entre aceleraciones sucesivas en el caso del posicionamiento “correcto” nos da una cota inferior del umbral ε_d , ya que si se eligiera un umbral ε_d menor, se tomaría erróneamente el posicionamiento como “fallido”. De la misma manera, el módulo de la aceleración en el caso del posicionamiento “correcto”, en particular $\max\{\|\mathbf{a}\|_2 - 1 \text{ g}\}$, nos da una cota inferior del umbral de aceleración ε_a .

Análogamente, en el caso del posicionamiento “fallido”, la máxima diferencia entre aceleraciones nos da una cota superior del umbral ε_d , mientras que $\max\{\|\mathbf{a}\|_2 - 1 \text{ g}\}$ nos da una cota superior de ε_a .

Luego de realizar varios posicionamientos “correctos” y “fallidos”, se llegó a las cotas

4.3. Posicionamiento del acelerómetro respecto a la persona

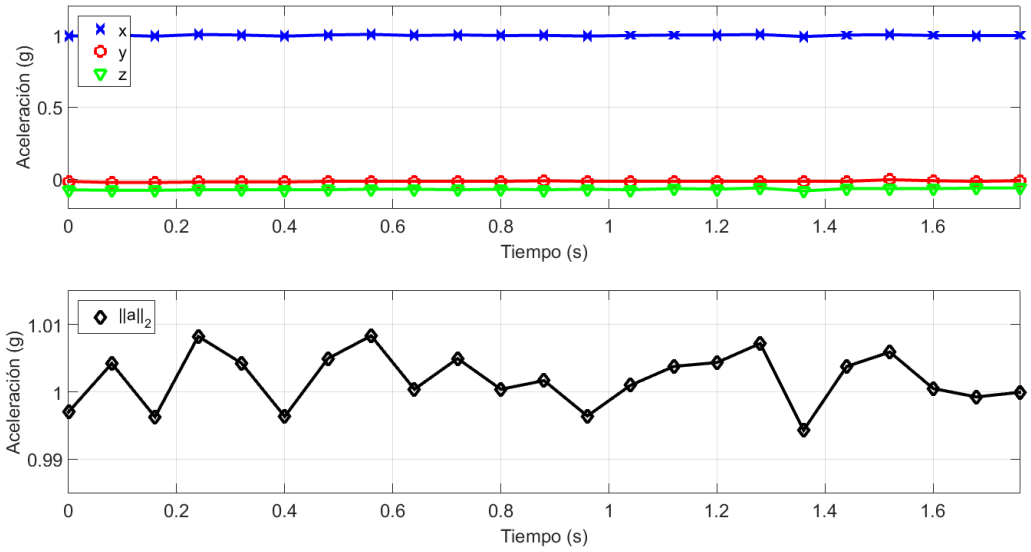


Figura 4.11: Posicionamiento “correcto”, a_x , a_y , a_z y $\|a\|_2$

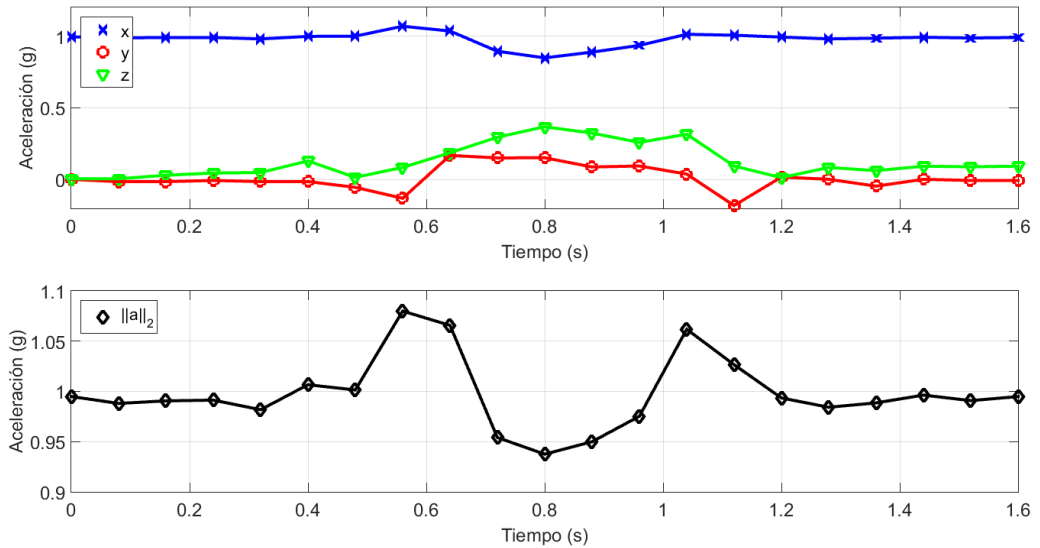


Figura 4.12: Posicionamiento “fallido” debido a movimiento de la persona

$$\begin{cases} \varepsilon_d < 0,15 \text{ g} \\ \varepsilon_d > 0,02 \text{ g} \\ \varepsilon_a < 0,12 \text{ g} \\ \varepsilon_a > 0,04 \text{ g} \end{cases}$$

Por lo tanto, se decidió elegir $\varepsilon_d = 0,06 \text{ g}$, que corresponde a $\frac{0,06 \text{ g}}{4 \text{ mg/LSB}} = 15 \text{ LSB}$, y $\varepsilon_a = 0,1 \text{ g}$, que corresponde a $\frac{0,1 \text{ g}}{4 \text{ mg/LSB}} = 25 \text{ LSB}$.

En las figuras 4.13 y 4.14, se muestra respectivamente un posicionamiento

Capítulo 4. Acelerómetro

“correcto” y uno “fallido” en lo que respecta a la ortogonalidad de los ejes.

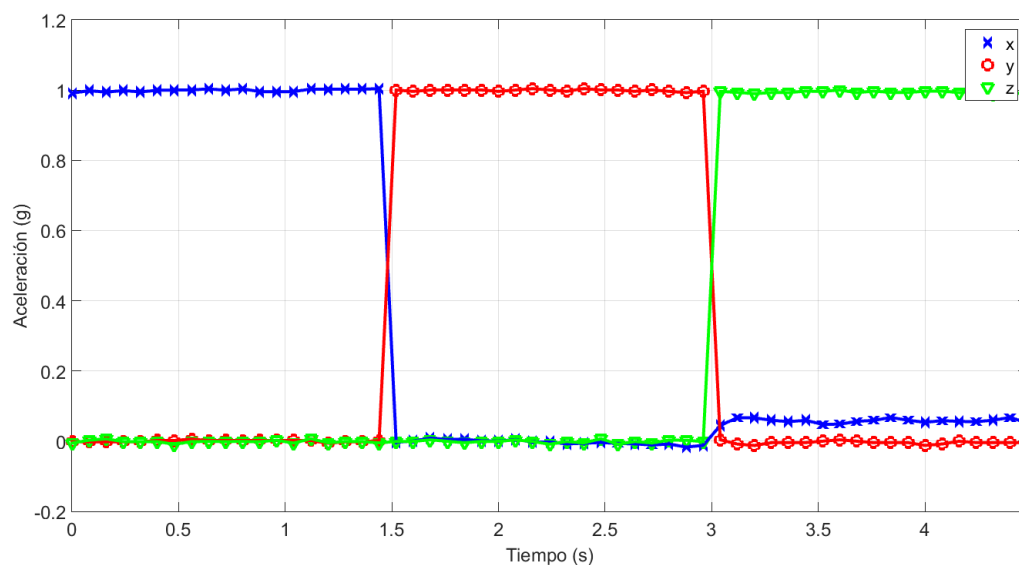


Figura 4.13: Posicionamiento “correcto”, a_x , a_y y a_z

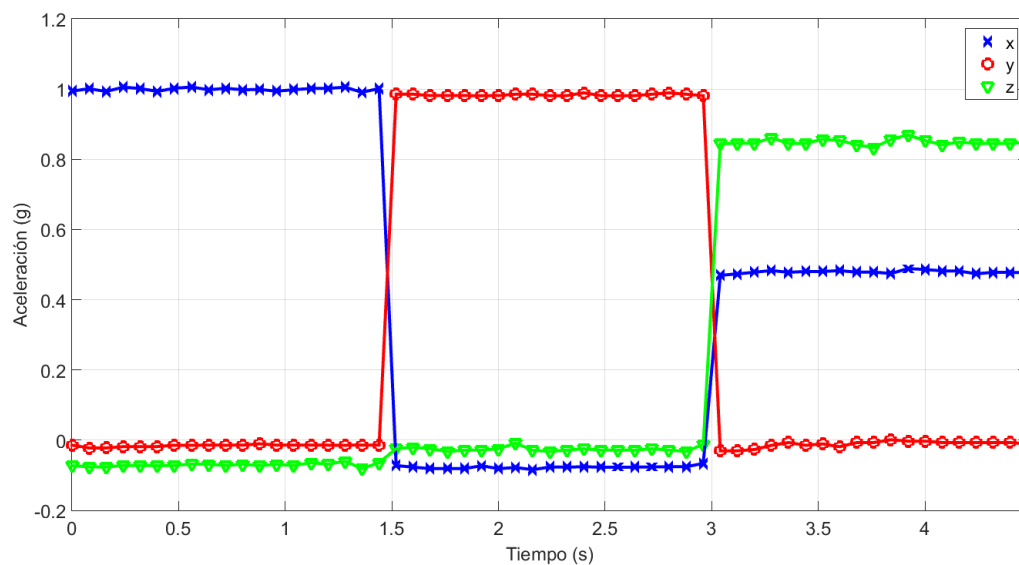


Figura 4.14: Posicionamiento “fallido” debido a la no-ortogonalidad de los ejes x-z

Observar que el posicionamiento “fallido” es solamente debido a la no-ortogonalidad de los ejes x-z, ya que el módulo de la aceleración en todos los casos pasa el chequeo, así como la ortogonalidad entre los ejes x-y e y-z.

Al igual que con el umbral de aceleración ε_a , se tiene que cada uno de los ángulos entre dos ejes distintos nos da una cota inferior de ε_θ en los casos en que deban pasar el chequeo, y una cota superior en los casos en que no.

4.3. Posicionamiento del acelerómetro respecto a la persona

Posicionamiento	θ_{xy}	θ_{xz}	θ_{yz}
“correcto” (figura 4.13)	0,2°	3,2°	0,3°
“fallido” (figura 4.14)	5,2°	25,4°	4,2°

Tabla 4.3: Posicionamiento del acelerómetro: ortogonalidad de los ejes

Luego de realizar varios posicionamientos “correctos” y “fallidos”, se llegó a las cotas

$$\begin{cases} \varepsilon_{\theta} < 25,4^{\circ} \\ \varepsilon_{\theta} > 5,2^{\circ} \end{cases}$$

Por lo tanto, se decidió elegir $\varepsilon_{\theta} = 20^{\circ}$.

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 5

Detección de posición y movimiento

Como se vio en el capítulo anterior, la aceleración medida por el acelerómetro se puede expresar de la siguiente manera

$$\mathbf{a}_{\text{meas}} = \mathbf{a}_{\text{case}} - \mathbf{a}_{\text{g}}$$

donde

- la aceleración \mathbf{a}_{g} es la aceleración gravitatoria: vector de magnitud 1 g con dirección y sentido hacia el centro de la Tierra, y es la que permite deducir la orientación de la persona, y
- la aceleración \mathbf{a}_{case} es la aceleración del encapsulado del acelerómetro, por lo tanto la asociada al movimiento de la persona y es la que permite deducir el nivel de movimiento de la persona.

Es necesario entonces separar de alguna manera $-\mathbf{a}_{\text{g}}$ de \mathbf{a}_{case} . Para determinar la mejor manera de hacerlo, se analizaron datos de aceleración de una persona caminando a distintas velocidades tomados con una frecuencia de muestreo de 100 Hz, obteniendo las figuras 5.1, 5.3 y 5.5. Dichas pruebas se realizaron con un dispositivo ya calibrado y posicionado respecto a la persona según los ejes de la figura 4.10.

Se observa que se obtienen formas de onda periódicas. El valor medio de las aceleraciones es aproximadamente 0 g para los ejes x e y, y 1 g para el eje z, lo cual corresponde con el hecho de que la persona está en posición vertical. Se observa también que a medida que la persona aumenta la velocidad de “lento” a “normal”, y finalmente a “rápido” las aceleraciones tienen mayor amplitud alrededor de su valor medio.

Para analizar los datos de aceleración anteriores en frecuencia, se les aplicó la FFT (*Fast Fourier Transform*), obteniendo los espectros mostrados en las figuras 5.2, 5.4 y 5.6, correspondientes a los datos de aceleración de las figuras 5.1, 5.3 y 5.5 respectivamente.

Se observa que a medida que la persona aumenta la velocidad de “lento” a “normal”, y finalmente a “rápido” la amplitud de los armónicos aumenta, pero no lo hace el valor de continua. Además, también aumenta la frecuencia de los mismos.

Capítulo 5. Detección de posición y movimiento

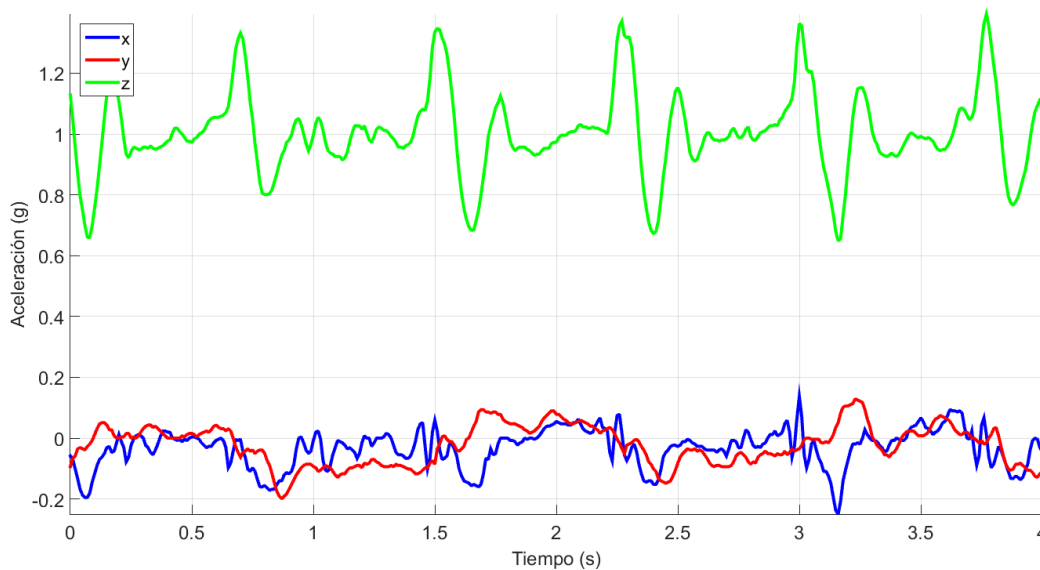


Figura 5.1: Datos de aceleración en los 3 ejes para persona caminando “lento”

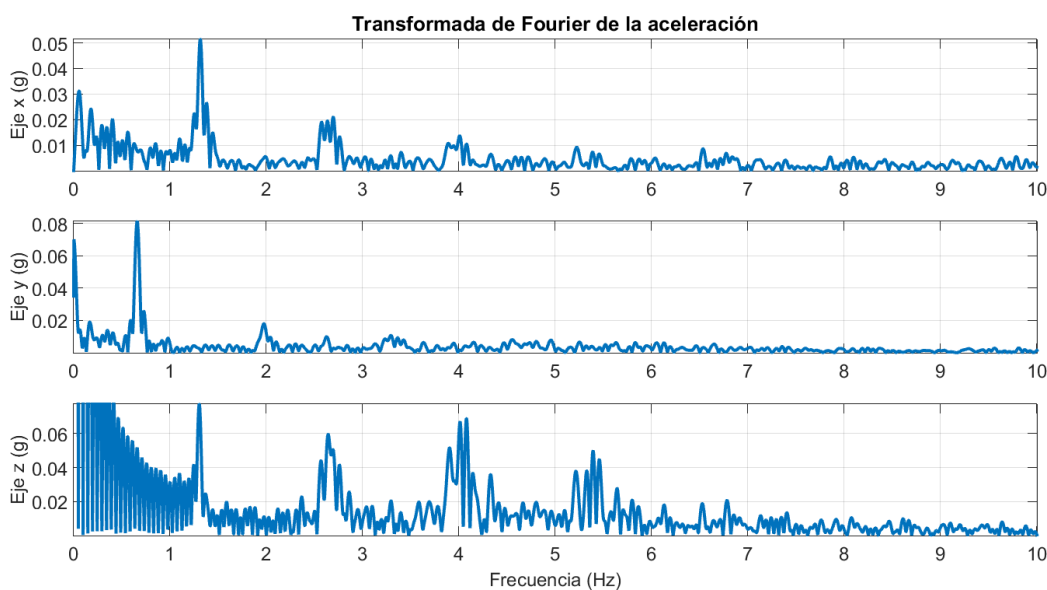


Figura 5.2: FFT de los datos para persona caminando “lento” (ver 5.1)

Un aspecto interesante a analizar es la frecuencia de la fundamental para cada eje: se puede observar que es la misma para los ejes x, z, pero es aproximadamente la mitad para el eje y. Dicho fenómeno se puede explicar considerando que el período de las señales de aceleración en los ejes x, z está dado por un paso de la persona, ya sea con el pie derecho o con el pie izquierdo, mientras que el período de la aceleración en el eje y está dado por dos pasos de la persona: uno izquierdo y uno derecho. Es por esto que $T_y = 2T_x = 2T_z$ (ver 5.1, 5.3, 5.5), y entonces $f_y = \frac{f_x}{2} = \frac{f_z}{2}$ (ver 5.2, 5.4, 5.6).

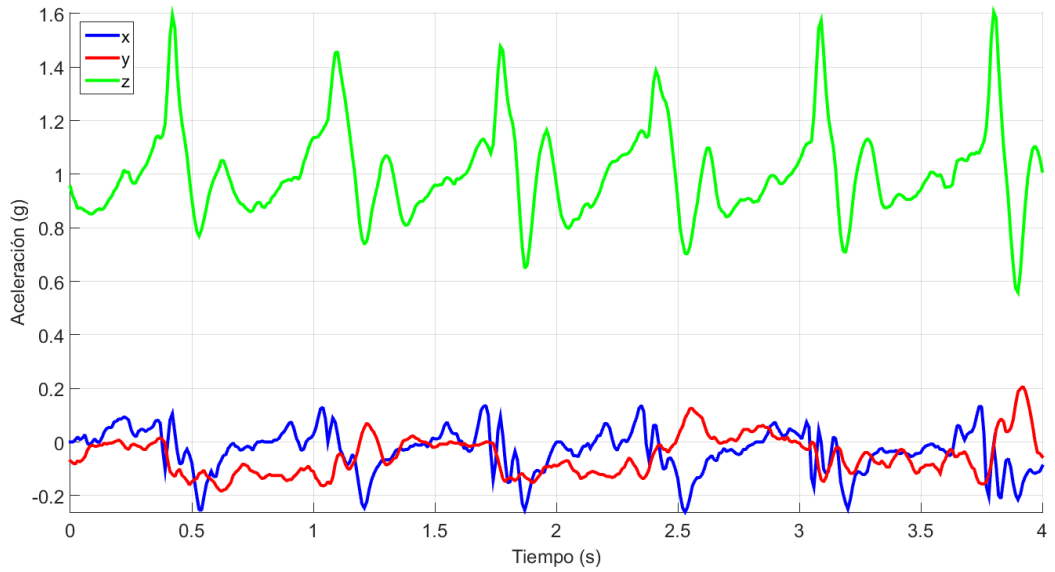


Figura 5.3: Datos de aceleración en los 3 ejes para persona caminando “normal”

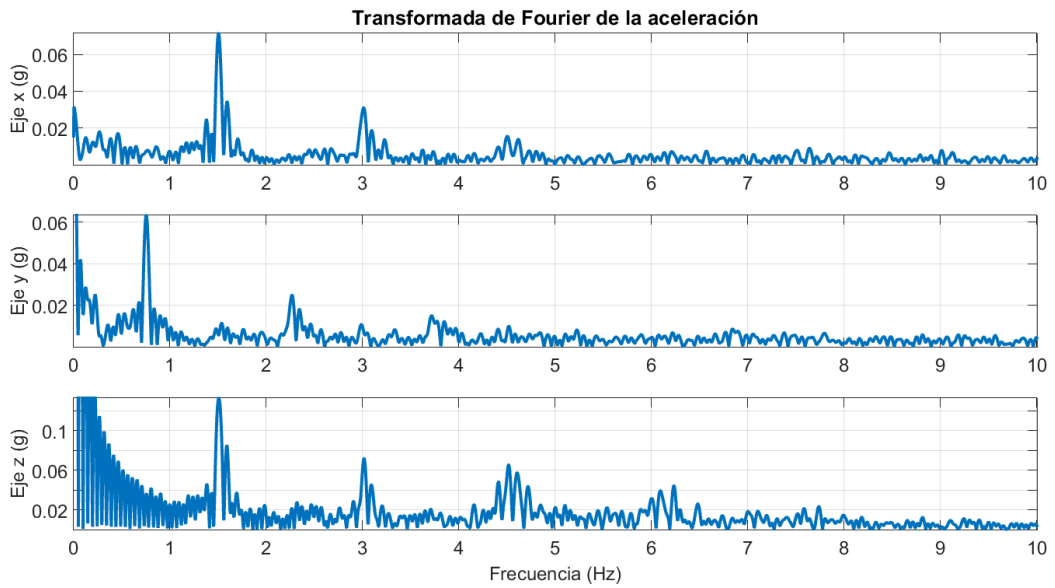


Figura 5.4: FFT de los datos para persona caminando “normal” (ver 5.3)

Se puede ver que $-\mathbf{a}_g$ es aproximadamente $\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ g, que corresponde a la componente de continua de \mathbf{a}_{meas} , y entonces que $\mathbf{a}_{case} = \mathbf{a}_{meas} + \mathbf{a}_g = \mathbf{a}_{meas} - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$ g, que corresponde a todas las frecuencias de \mathbf{a}_{meas} salvo la continua.

Una forma razonable de separar \mathbf{a}_{case} de $-\mathbf{a}_g$ es entonces mediante un filtro pasa-bajos si se quiere obtener $-\mathbf{a}_g$, o pasa-altos si se quiere obtener \mathbf{a}_{case} . A partir de los datos obtenidos, se puede ver que la frecuencia de corte de dicho filtro no deberá ser mayor a la frecuencia de la fundamental del eje y. Se eligió entonces la frecuencia de corte $f_c = 0,25$ Hz. Aplicando diferentes filtros pasa-bajos (variando

Capítulo 5. Detección de posición y movimiento

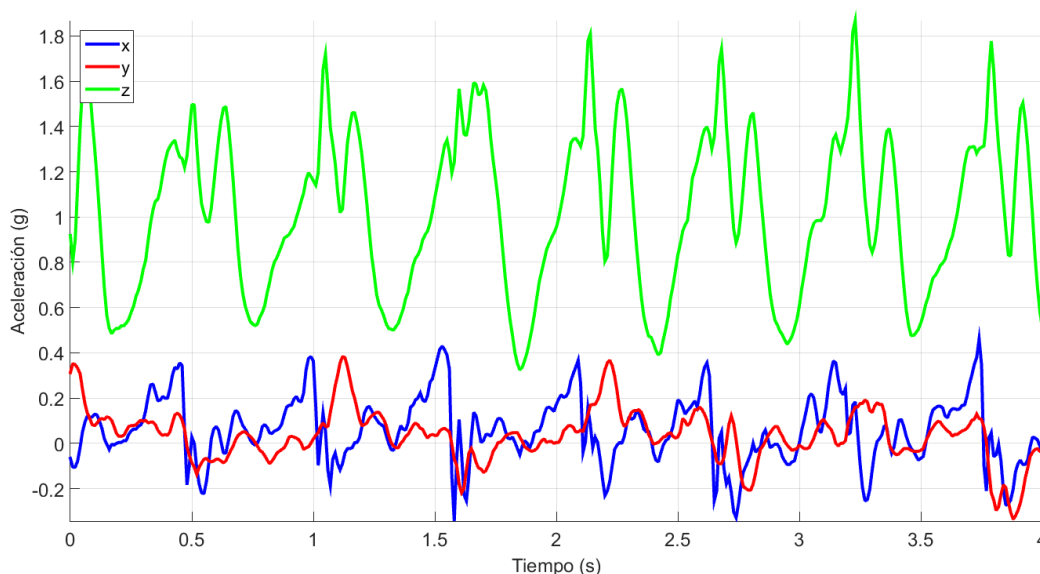


Figura 5.5: Datos de aceleración en los 3 ejes para persona caminando “rápido”

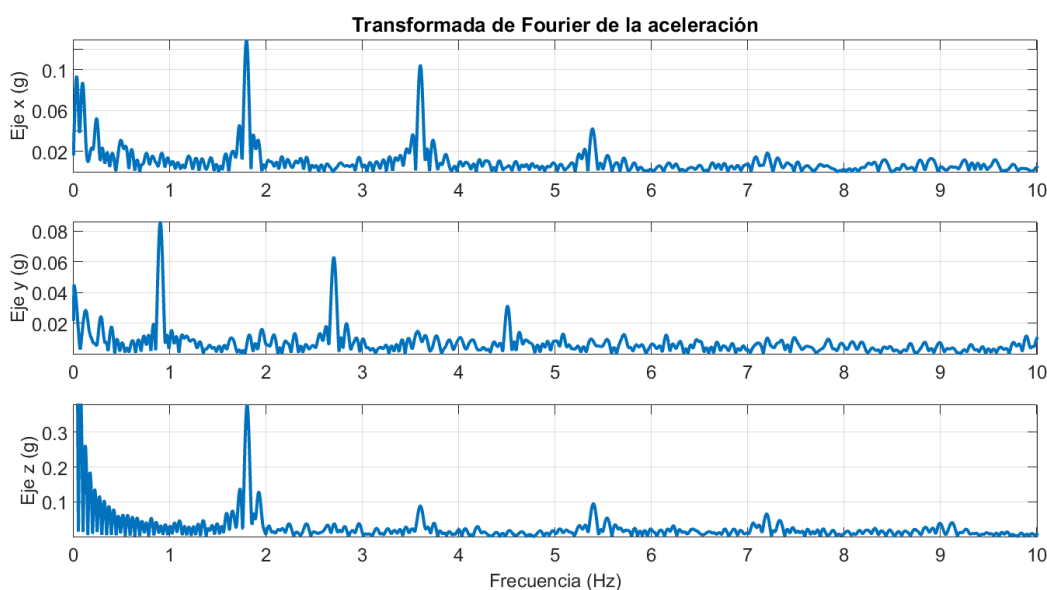


Figura 5.6: FFT de los datos para persona caminando “rápido” (ver 5.5)

el tipo de filtro: Butterworth, Chebyshev, elíptico, y el orden del mismo) a los datos mostrados anteriormente, se buscó el filtro que mejor se adaptara a dichos datos observando en particular el compromiso entre retardo del filtro y la pendiente de la ganancia del filtro en dB/dec. Se decidió finalmente utilizar un filtro Butterworth de orden 3. La figura 5.7 muestra el resultado de aplicar el filtro elegido a los datos de la figura 5.3.

Se observa que mediante dicho filtro, se separan adecuadamente \mathbf{a}_{case} de $-\mathbf{a}_g$. Algo importante a destacar es que esta descomposición es válida siempre y cuando

5.1. Frecuencia de muestreo de aceleraciones

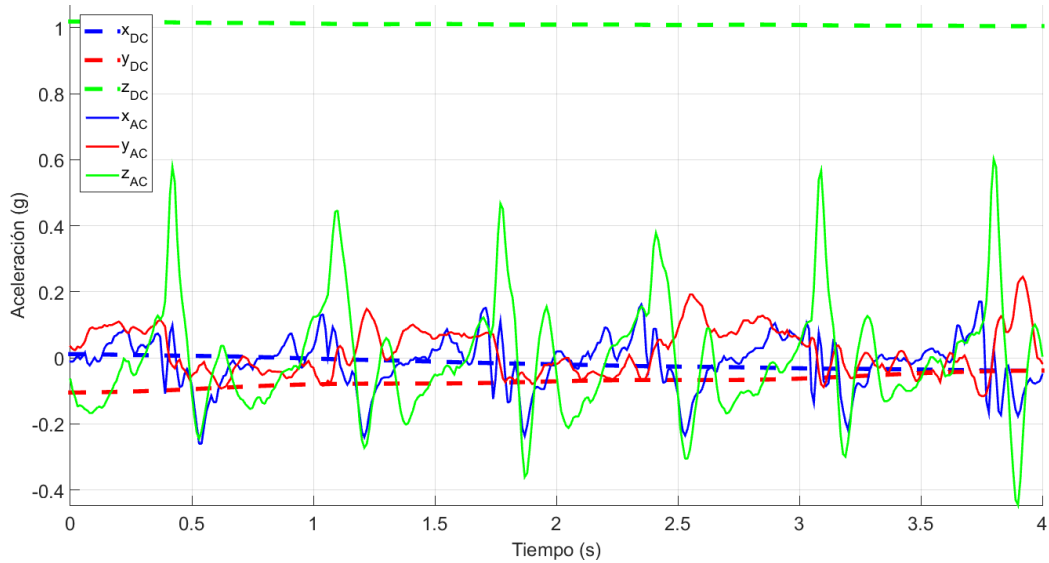


Figura 5.7: Datos filtrados (DC y AC) de aceleración en los 3 ejes para persona caminando “normal”

se cumpla que

- los movimientos de la persona sean alrededor de $-\mathbf{a}_g$, es decir que el valor de DC de \mathbf{a}_{case} sea cercano a 0, y que
- los movimientos de la persona sean tales que $-\mathbf{a}_g$ no tenga componentes en frecuencia mayores a la frecuencia de corte del filtro pasa-bajos, 0,25 Hz.

Se observa de las figuras anteriores que dichas condiciones se cumplen para una persona caminando.

A continuación, se elegirá la frecuencia de muestreo de datos de aceleración, y definirán los algoritmos para clasificar orientación (mediante \mathbf{a}_{case}) y niveles de esfuerzo (mediante $-\mathbf{a}_g$).

5.1. Frecuencia de muestreo de aceleraciones

Un parámetro muy importante del sistema es la frecuencia de muestreo de aceleraciones f_s . Si la misma es muy baja, por el teorema del muestreo de Nyquist-Shannon será imposible de reconstruir inequívocamente los fenómenos de aceleración que ocurran a frecuencia mayor a $\frac{f_s}{2}$ a partir de las muestras y se perdería información relevante de la señal. Por el contrario, si f_s es muy alta, se logran adquirir señales de aceleración de alta frecuencia, pero tanto el microcontrolador como el acelerómetro consumirán más corriente, reduciendo así la autonomía del dispositivo. Es entonces de vital importancia hallar la frecuencia f_s más adecuada para nuestro sistema.

Capítulo 5. Detección de posición y movimiento

Sin embargo, se presenta un problema con la frecuencia de muestreo de aceleraciones f_s . Si es adecuada para cuando la persona está quieta, no es suficientemente alta para cuando la persona está en movimiento. Si es adecuada cuando la persona está en movimiento, es innecesariamente alta cuando la persona está quieta, conllevando a un alto consumo de las baterías y por ende a una significativa reducción de la autonomía del dispositivo.

Se procedió entonces a implementar un algoritmo de frecuencia de muestreo de aceleraciones f_s variable entre dos valores f_{s0} y f_{s1} , con $f_{s0} < f_{s1}$.

- Cuando la persona está quieta (y lo estuvo durante los últimos 3 s), la frecuencia de muestreo es $f_s = f_{s0}$. Para considerar que la persona está quieta mientras $f_s = f_{s0}$, la condición a verificar es que $0,9 \text{ g} \leq \|\mathbf{a}_{\text{meas}}\|_2 \leq 1,1 \text{ g}$.
- Cuando se detecta que la persona está en movimiento, la frecuencia aumenta y pasa a valer $f_s = f_{s1}$.
- Cuando se detecta que la persona está quieta nuevamente (luego de estar en movimiento), se setea un *timer*, que al expirar vuelve a reducir la frecuencia a $f_s = f_{s0}$. Para considerar que la persona está quieta cuando $f_s = f_{s1}$, la condición a verificar es que hayan menos de la mitad de muestras de aceleración en 1 segundo que cumplan que $\|HPF\{\mathbf{a}_{\text{meas}}\}\|_2 > 0,05 \text{ g}$.
- Si se detecta movimiento (no se cumple la condición anterior) antes de que expire el *timer*, el mismo es reseteado.

La figura 5.8 muestra el diagrama de estados asociado a dicho algoritmo.

En el estado en blanco, la frecuencia de muestreo es f_{s0} , mientras que en los otros dos (en gris), la frecuencia de muestreo es f_{s1} .

Mediante pruebas, se verificó que una frecuencia $f_{s0} = 1,56 \text{ Hz}$ es suficiente para detectar movimiento cuando lo hay y aumentar la frecuencia a f_{s1} . A partir de los datos de aceleración tomados para persona caminando (ver figuras 5.2, 5.4 y 5.6), se puede ver que la mayoría de la energía de la señal se encuentra entre 0 Hz y 6 Hz. Se eligió entonces $f_{s1} = 12,5 \text{ Hz}$, pudiendo obtener información de frecuencias $f < \frac{12,5 \text{ Hz}}{2} = 6,25 \text{ Hz}$.

Se distinguen dos casos posibles:

1. Si se está en el estado de baja frecuencia de muestreo $f_s = f_{s0}$, se asume que la persona está quieta, entonces $\mathbf{a}_{\text{case}} \approx \vec{0}$ y $-\mathbf{a}_{\text{g}} \approx \mathbf{a}_{\text{meas}}$. Por lo tanto no se utiliza el filtro pasa-bajos para separar \mathbf{a}_{case} de $-\mathbf{a}_{\text{g}}$ ya que es innecesario.
2. Si se está en el estado de alta frecuencia de muestreo $f_s = f_{s1}$, es porque hay algún tipo de movimiento (o lo hubo recientemente), por lo que se filtra \mathbf{a}_{meas} con el filtro pasa-bajos de tipo Butterworth de orden 3 y frecuencia de corte $f_c = 0,25 \text{ Hz}$ de manera de separar $-\mathbf{a}_{\text{g}} = LPF\{\mathbf{a}_{\text{meas}}\}$, componente de continua y bajas frecuencias, de $\mathbf{a}_{\text{case}} = HPF\{\mathbf{a}_{\text{meas}}\}$, componente de altas frecuencias.

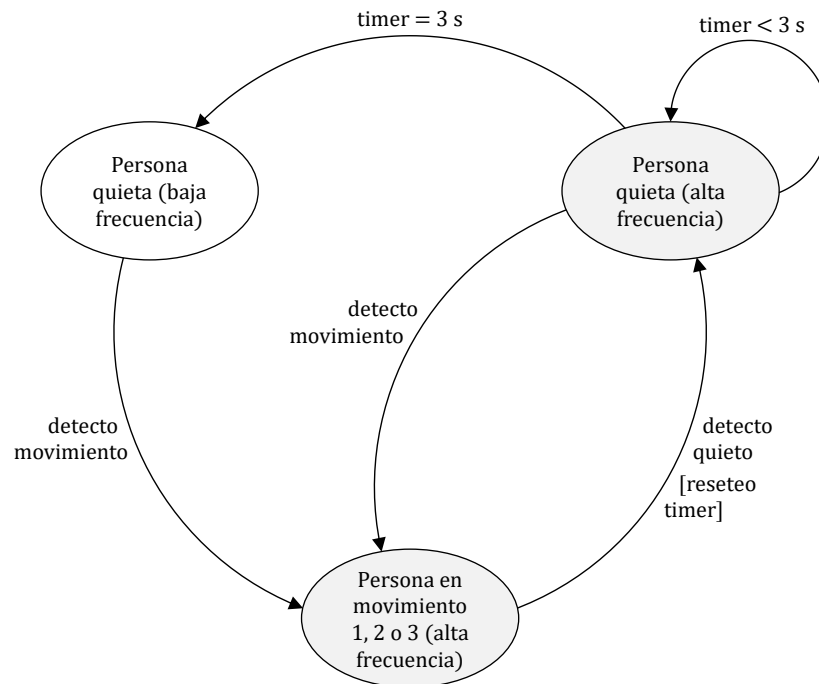


Figura 5.8: Diagrama de estados asociado a la variación de frecuencia de muestreo de aceleraciones

5.2. Detección de posición

Tal como se especifica en el documento de requerimientos detallados, el dispositivo deberá ser capaz de clasificar la posición (u orientación) de la persona entre los siguientes estados posibles:

- persona parada,
- persona acostada,
 - persona acostada boca arriba,
 - persona acostada boca abajo,
 - persona acostada sobre su hombro derecho,
 - persona acostada sobre su hombro izquierdo, y
- persona en estado intermedio.

Se incluyó un estado intermedio que corresponde a situaciones en las que la persona está inclinada (ni parada ni acostada). De esta manera, el tiempo que la persona está parada o acostada o en posición intermedia sumado debe ser igual al tiempo total de medición: $t_{parada} + t_{acostada} + t_{intermedia} = t_{total}$. El estado “persona acostada” engloba los 4 estados “boca arriba”, “boca abajo”, “sobre su hombro derecho” y “sobre su hombro izquierdo” por lo que el tiempo en cada uno de dichos

Capítulo 5. Detección de posición y movimiento

estados sumados debe ser igual al tiempo de persona acostada.

Como se mencionó anteriormente, para realizar la clasificación de la posición, se utiliza solamente el término asociado a la aceleración gravitatoria $-\mathbf{a}_g$ de la aceleración medida \mathbf{a}_{meas} .

La tabla 5.1 muestra el valor ideal de $-\mathbf{a}_g$ para algunas de las posiciones definidas.

Posición de la persona	$-\mathbf{a}_g$
parada	$\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} g$
acostada boca arriba	$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} g$
acostada boca abajo	$\begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} g$
acostada sobre su hombro derecho	$\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} g$
acostada sobre su hombro izquierdo	$\begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} g$

Tabla 5.1: Vector $-\mathbf{a}_g$ en función de la posición de la persona

Mediante pruebas, se determinaron entonces las condiciones para cada estado detalladas en la tabla 5.2 (en este caso, las aceleraciones a_x , a_y y a_z corresponden a la componente según el eje x, el eje y, y eje z respectivamente de la aceleración $-\mathbf{a}_g$).

Posición de la persona	Condición
parada	$a_z > 0,9 g$
acostada	$ a_z < 0,6 g$
acostada boca arriba	$(a_z < 0,6 g) \text{ and } (a_x \geq a_y) \text{ and } (a_x > -a_y)$
acostada boca abajo	$(a_z < 0,6 g) \text{ and } (a_x \leq a_y) \text{ and } (a_x < -a_y)$
acostada sobre su hombro derecho	$(a_z < 0,6 g) \text{ and } (a_x < a_y) \text{ and } (a_x \geq -a_y)$
acostada sobre su hombro izquierdo	$(a_z < 0,6 g) \text{ and } (a_x > a_y) \text{ and } (a_x \leq -a_y)$
estado intermedio	$(0,6 g \leq a_z \leq 0,9 g) \text{ or } (a_z \leq -0,6 g)$

Tabla 5.2: Posiciones de la persona

La figura 5.9 resume gráficamente las regiones de clasificación de posición dadas por las condiciones de la tabla anterior.

5.3. Detección de giros de 90° , levantadas y acostadas

El dispositivo deberá además detectar giros de 90° hacia la derecha y hacia la izquierda mientras la persona está acostada, “levantadas” y “acostadas”. Se consi-

5.4. Detección de movimiento

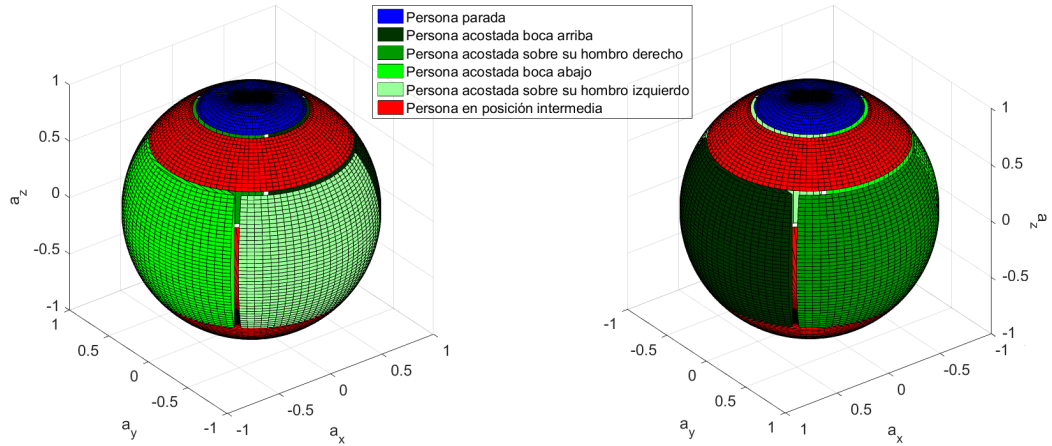


Figura 5.9: Clasificación de la posición de la persona según la aceleración $-\mathbf{a}_g$

dera que el giro es hacia la derecha si la persona gira en sentido horario mirándola desde arriba (sobre su cabeza), y que el giro es hacia la izquierda si la persona gira en sentido anti-horario. Se consideran solamente los giros que son transiciones de las posiciones “persona acostada boca arriba”, “persona acostada boca abajo”, “persona acostada sobre su hombro derecho” y “persona acostada sobre su hombro izquierdo”. Análogamente, se consideran las “levantadas” como una transición de posición de “persona acostada” a “persona parada” y las “acostadas” como una transición de posición de “persona parada” a “persona acostada”. En ambos casos podrá pasar por el estado intermedio.

De esta manera, para determinar dichas transiciones de clasificación de posición, se utiliza $-\mathbf{a}_g$ (correspondiente a \mathbf{a}_{meas} mientras la persona está quieta, y a $LPF\{\mathbf{a}_{meas}\}$ mientras la persona está en movimiento). Para esto, se debe recordar el último estado distinto de la posición intermedia, y comparándolo con el estado actual se puede determinar qué evento ocurre (por ejemplo, si el estado anterior era “persona acostada boca abajo” y el actual es “persona parada”, ocurrió una “levantada”).

5.4. Detección de movimiento

El objetivo de la clasificación del comportamiento de la persona en distintos rangos de movimiento es determinar indirectamente distintos niveles de actividad de la persona.

El primer indicador para saber si la persona está realizando alguna actividad y por lo tanto si se está moviendo, es el módulo de la aceleración. Si el mismo está en un cierto entorno de 1 g, se puede decir que la persona está quieta, mientras que si difiere lo suficiente de 1 g, se puede decir que la persona se está moviendo.

Sin embargo, es importante destacar que del posicionamiento, se tiene $\mathbf{a}_P = (\mathcal{P}\mathbb{T}_A)\mathbf{a}_A$ (ver capítulo 4.3), y por lo tanto $\|\mathbf{a}_P\|_2 = \|(\mathcal{P}\mathbb{T}_A)\mathbf{a}_A\|_2$, que no neces-

Capítulo 5. Detección de posición y movimiento

riamente es $\|\mathbf{a}_{\mathcal{A}}\|_2$ ya que la matriz $({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})$ no necesariamente será ortogonal en la práctica.

Por ejemplo, si se tiene $({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}}) = \begin{bmatrix} 1 & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$, entonces

- si $\mathbf{a}_{\mathcal{A}} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} g$, se tiene que $\|\mathbf{a}_{\mathcal{A}}\|_2 = 1 g$, mientras que

$$\|\mathbf{a}_{\mathcal{P}}\|_2 = \|({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})\mathbf{a}_{\mathcal{A}}\|_2 = \left\| \begin{bmatrix} 1 \\ \sin(\theta) \\ 0 \end{bmatrix} g \right\|_2 = \sqrt{1 + \sin^2(\theta)} g.$$

Si $\sqrt{1 + \sin^2(\theta)} g > 1 g + \varepsilon_a$, entonces se detectaría a la persona en movimiento a pesar de estar quieta.

- De manera similar, si $\mathbf{a}_{\mathcal{A}} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} g$, se tiene que $\|\mathbf{a}_{\mathcal{A}}\|_2 = 1 g$, mientras que

$$\|\mathbf{a}_{\mathcal{P}}\|_2 = \|({}_{\mathcal{P}}\mathbb{T}_{\mathcal{A}})\mathbf{a}_{\mathcal{A}}\|_2 = \left\| \begin{bmatrix} 0 \\ \cos(\theta) \\ 0 \end{bmatrix} g \right\|_2 = |\cos(\theta)| g.$$

Si $|\cos(\theta)| g < 1 g - \varepsilon_a$, entonces se detectaría a la persona en movimiento a pesar de estar quieta.

Es por este motivo que a la hora de detectar movimiento, el módulo de la aceleración que se toma es $\|\mathbf{a}_{\mathcal{A}}\|_2 = \|({}_{\mathcal{A}}\mathbb{T}_{\mathcal{P}})\mathbf{a}_{\mathcal{P}}\|_2$ y no $\|\mathbf{a}_{\mathcal{P}}\|_2$.

Para determinar de qué manera clasificar rangos de movimiento (para diferenciar niveles de actividad), se analizaron distintos procesamientos de la aceleración medida. Se desea, luego del procesamiento, obtener una “medida” del movimiento del usuario que sea monótona creciente con la velocidad, y que en lo posible tenga una relación lineal para tener la misma sensibilidad en todo el rango de velocidades.

Se tomaron en consideración distintas magnitudes de base para clasificar niveles de movimiento:

- Módulo de la aceleración restándole 1 g:
 $\|\mathbf{a}_{\text{meas}}\|_2 - 1 g$
- Módulo de la AC de la aceleración (que idealmente sería \mathbf{a}_{case}):
 $\|HPF\{\mathbf{a}_{\text{meas}}\}\|_2$
- Diferencias sucesivas (derivada discreta) de aceleración en el eje z:
 $|a_{\text{meas}_z}[n] - a_{\text{meas}_z}[n-1]|$
- Diferencias sucesivas (derivada discreta) del módulo de la aceleración:
 $|\|\mathbf{a}_{\text{meas}}[n]\|_2 - \|\mathbf{a}_{\text{meas}}[n-1]\|_2|$
- SMA (*normalized Signal Magnitude Area*, dicha magnitud es la que se usa para clasificar movimiento en [18]):
 $\frac{1}{t} \int_0^t \|\mathbf{a}_{\text{meas}}\|_1 dt = \frac{1}{t} \int_0^t (|a_{\text{meas}_x}| + |a_{\text{meas}_y}| + |a_{\text{meas}_z}|) dt$

Como la aceleración mientras la persona está en movimiento oscila alrededor de $-\mathbf{a}_g$ (ver 5.1, 5.3, 5.5), entonces las magnitudes anteriores también lo harán.

5.4. Detección de movimiento

Está claro que cualquiera de dichas magnitudes sin ningún procesado no son suficientes para clasificar adecuadamente rangos de movimiento ya que los puntos que caigan muy cerca de $-\mathbf{a}_g$ llevarían a concluir erróneamente que la persona está quieta. Es preciso agregarle algo de memoria al algoritmo de clasificación en rangos de movimiento (excepto a la magnitud SMA, ya que al ser una integral ya tiene memoria). Se decidió entonces utilizar una ventana móvil de 12s. Este promediado agrega un retardo a la “medida” de movimiento. Sin embargo, considerando la aplicación, no es relevante ya que no interesa el movimiento instantáneo del usuario sino que se desea identificar tiempos largos en diferentes rangos de movimiento.

Para evaluar el desempeño de los distintos algoritmos, se hicieron pruebas caminando a distintas velocidades, adquiriendo los datos de aceleración para visualizarlos y procesarlos luego. Para lograr la mayor objetividad posible, se utilizó una cinta caminadora, registrando el valor de velocidad en km/h desplegado por su *display*.

Se obtuvieron resultados similares para las distintas magnitudes. Las figuras 5.10, 5.11 y 5.12 muestran algunas de las magnitudes mencionadas en función de la velocidad de la caminadora. Dado que se muestra la media móvil de 12s de las magnitudes, se realizó lo mismo con la velocidad de la caminadora. Se muestran los puntos de todas las pruebas realizadas, con un color distinto cada prueba. Finalmente, también se muestra la recta que mejor aproxima dichos puntos utilizando el método de mínimos cuadrados, y los umbrales de velocidad elegidos $v_{01} = 1,1$ km/h, $v_{12} = 2,5$ km/h y $v_{23} = 4,5$ km/h.

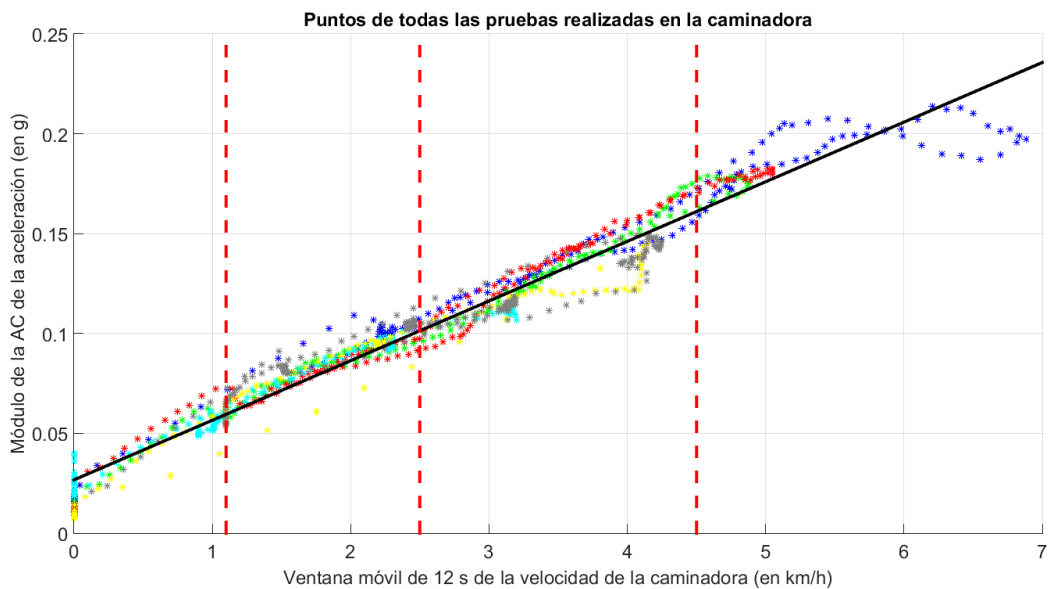


Figura 5.10: Módulo de la AC de la aceleración en función de la velocidad de la caminadora

Las 3 magnitudes mostradas (5.10, 5.11, 5.12) están correlacionadas con la velocidad de la caminadora, y por lo tanto con el movimiento de la persona. Se

Capítulo 5. Detección de posición y movimiento

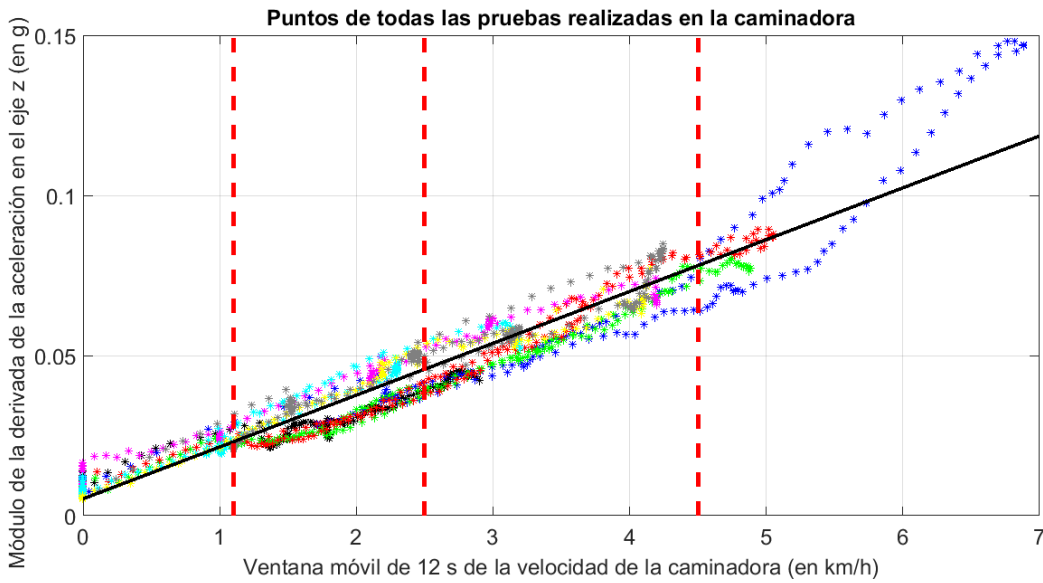


Figura 5.11: Diferencia sucesiva de la aceleración en el eje z en función de la velocidad de la caminadora

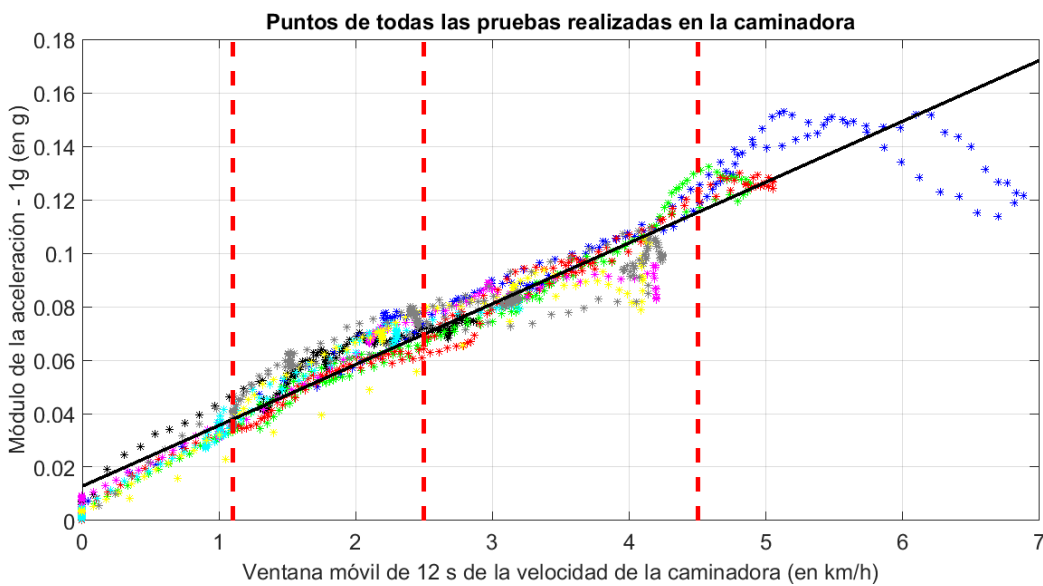


Figura 5.12: Módulo de la aceleración - 1g en función de la velocidad de la caminadora

observa que la asociada al módulo de la AC de la aceleración es la que tiene resultados más repetitivos para las distintas pruebas, es decir que tiene una “nube de puntos” más fina.

Se decidió entonces utilizar la ventana móvil de 12s del módulo de la componente AC de la aceleración $\|HPF\{\mathbf{a}_{meas}\}\|_2$. Se denotará M_A a dicha magnitud. Las figuras 5.13, 5.14 y 5.15 muestran los resultados de algunas de las pruebas

5.4. Detección de movimiento

realizadas con la magnitud M_A graficada junto a la velocidad de la caminadora y el promedio móvil de la misma.

La figura 5.13 muestra una prueba donde se aumentó la velocidad del *treadmill* en pasos de aproximadamente 0,5 km/h cada 10 s hasta llegar a 5 km/h (considerado caminata rápida) y luego se decrementó de igual manera.

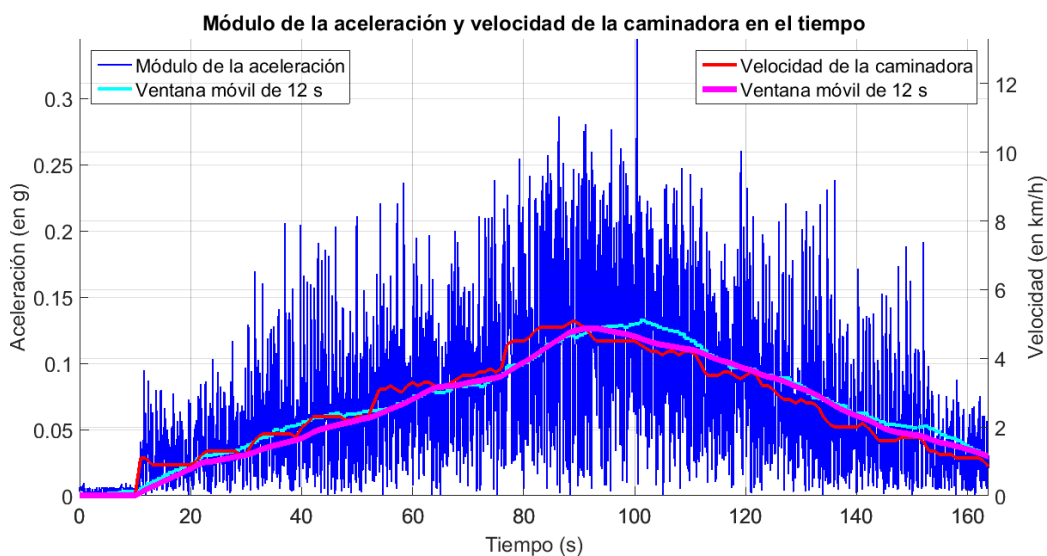


Figura 5.13: Prueba en la caminadora 1

La figura 5.14 presenta una prueba donde, empezando quieto, se aumentó la velocidad de la caminadora en pasos de 1 km/h cada 20 s. La figura 5.15 es similar con la diferencia que cuenta con incrementos y decrementos de velocidad intercalados.

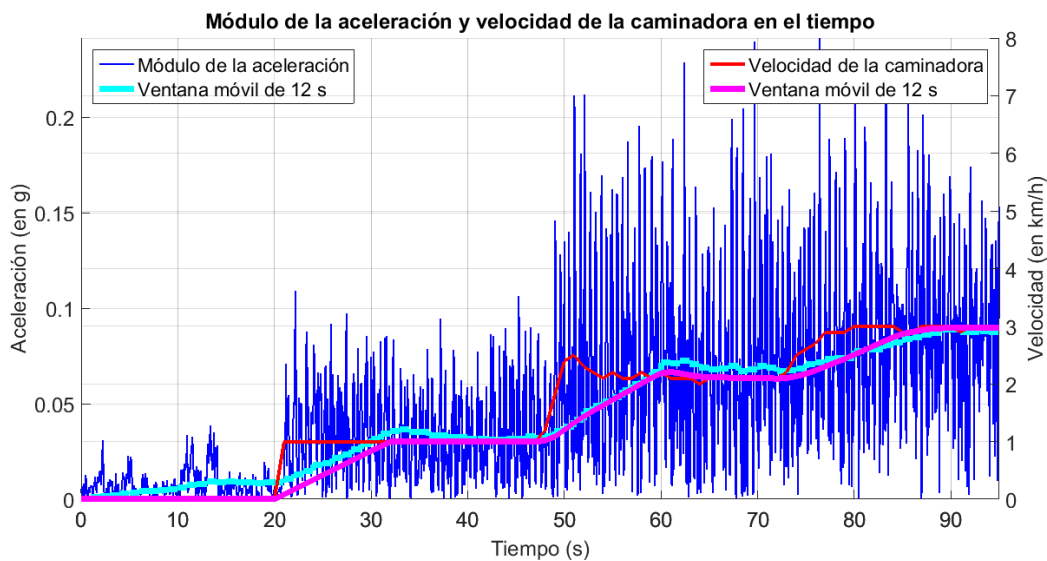


Figura 5.14: Prueba en la caminadora 2

Capítulo 5. Detección de posición y movimiento

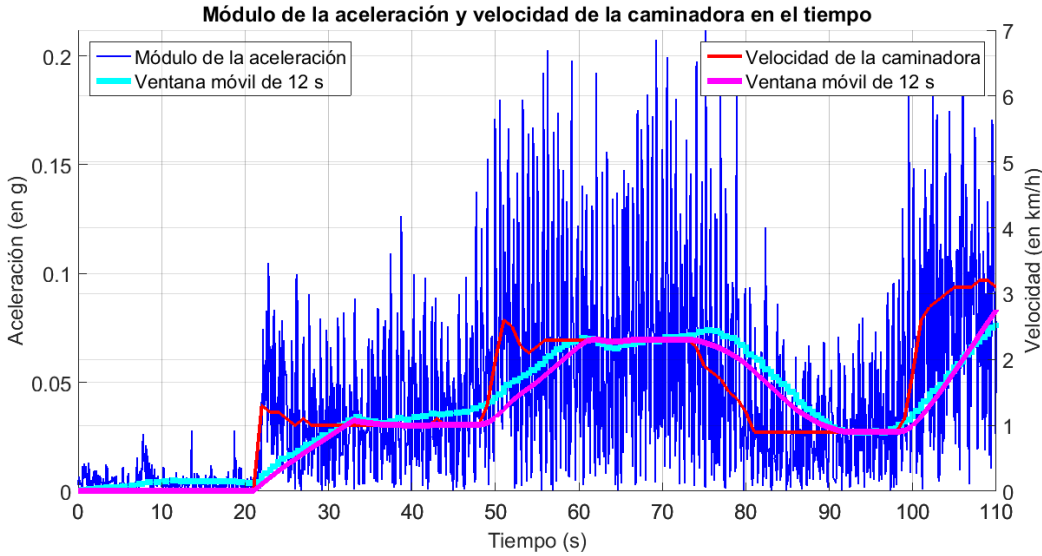


Figura 5.15: Prueba en la caminadora 3

En todas las pruebas realizadas, se obtuvo que el andamiento de la magnitud propuesta M_A acompaña al de la velocidad desplegada por el *display* de la caminadora (en particular se puede ver para las pruebas de las figuras 5.13, 5.14 y 5.15).

Finalmente, se busca encontrar umbrales de la magnitud M_A de manera de definir 3 niveles de actividad aproximados: un nivel 1 correspondiente a “caminar muy despacio”, un nivel 2 correspondiente al “caminar promedio de una persona de mediana edad” y el nivel 3 correspondiente a “caminar rápido”.

A partir de los umbrales de velocidad elegidos $v_{01} = 1,1$ km/h, $v_{12} = 2,5$ km/h, $v_{23} = 4,5$ km/h y la figura 5.10, se obtuvieron los umbrales de M_A : $\varepsilon_{01} = 0,065$ g, $\varepsilon_{12} = 0,10$ g y $\varepsilon_{23} = 0,16$ g.

Cada 1 s (*tick* de los contadores que llevan la cuenta del tiempo en cada posición y rango de movimiento), se compara M_A con los umbrales ε_{01} , ε_{12} y ε_{23} , clasificando finalmente

- rango de movimiento 0 si $M_A < \varepsilon_{01}$,
- rango de movimiento 1 si $\varepsilon_{01} \leq M_A < \varepsilon_{12}$,
- rango de movimiento 2 si $\varepsilon_{12} \leq M_A < \varepsilon_{23}$, y
- rango de movimiento 3 si $\varepsilon_{23} \leq M_A$.

Capítulo 6

Firmware del dispositivo

El *firmware* del dispositivo se programó en lenguaje de programación C con el entorno de desarrollo IAR versión 6.40.4.

6.1. Arquitectura

Se utilizó una arquitectura de *software* de Round-Robin con interrupciones [19]. Esta arquitectura atiende las acciones críticas mediante interrupciones y el resto del procesamiento con funciones (*handlers*) llamadas en el *loop* principal. Cada bandera (*flag*) corresponde a un evento, y es encendida en alguna de las rutinas de atención a las interrupciones (ISR del inglés *Interrupt Service Routine*). En caso que no haya eventos, permite al microcontrolador ir a un estado de bajo consumo. El algoritmo 1 ilustra el comportamiento fundamental de un *loop* Round-Robin.

Algoritmo 1 Pseudocódigo asociado a la arquitectura de Round-Robin

```
loop ▷ Loop Round-Robin
  if flag1 then
    flag1 = 0;
    handler_flag1();
  end if
  ...
  if flagN then
    flagN = 0;
    handler_flagN();
  end if
  if not (flag1 or ... or flagN) then
    low_power_mode();
  end if
end loop
```

Esta arquitectura presenta la ventaja que es flexible y escalable. Si se desea añadir otra tarea de procesamiento, sólo se debe crear un nuevo *handler* y agregarlo al *loop* principal. Esto es válido siempre que el requisito temporal de cada tarea sea menor a la suma de los tiempos de respuesta de cada *handler*.

6.1.1. Modos de funcionamiento

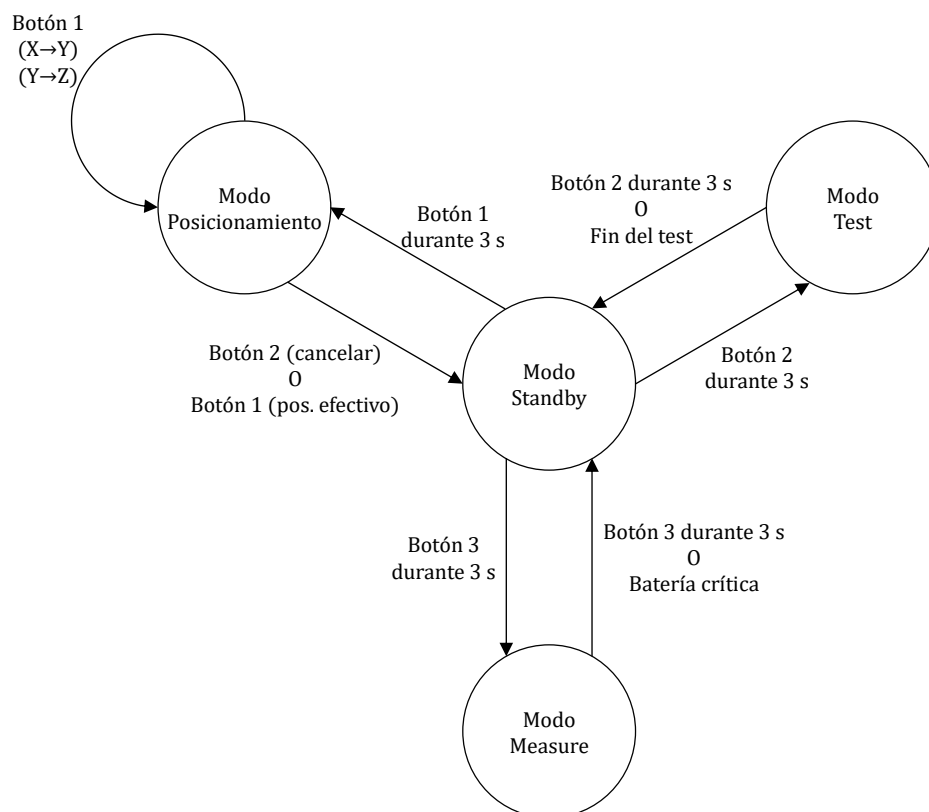


Figura 6.1: Diagrama mostrando los distintos modos de funcionamiento del dispositivo y las transiciones entre los mismos

El *firmware* cuenta con 4 modos distintos de funcionamiento.

- **Modo Standby:** Es el modo en el que inicia el dispositivo y el de menor consumo, ya que no se miden ni procesan aceleraciones. Sin embargo, es el único estado en el que se permite la comunicación mediante un intérprete de comandos (*shell*) y/o una GUI.
- **Modo Test:** Este modo facilita el desarrollo permitiendo adquirir aceleraciones para un posterior estudio de la señal. Se adquieren 2048 muestras de aceleración, las cuales son almacenadas en memoria Flash para luego poder ser descargadas al PC.
- **Modo Posicionamiento:** Este modo realiza el algoritmo descrito en la sección 4.3.1 para calcular la matriz $(\mathcal{P}\mathbb{T}_{\mathcal{A}})$. Una vez finalizada exitosamente la secuencia de posicionado, retorna automáticamente al modo Standby.
- **Modo Measure:** Es el modo principal del dispositivo. En este modo se procesan aceleraciones siguiendo el algoritmo planteado en el capítulo 5 y

6.2. Modularización

se clasifica el nivel de actividad y la posición del usuario, incrementando el contador correspondiente.

Para el manejo de los modos se cuenta con una variable global `estado` que almacena el modo activo. El usuario puede modificar esta variable mediante el accionado de los botones, generando una interrupción. Para minimizar el accionado accidental de los botones, los botones deben mantenerse presionados durante 3 s. En el dispositivo final, se utilizaron los LEDs verdes disponibles para señalar el modo activo, en la tabla 6.1 se ilustra la convención tomada.

Modo		LED1	LED2	LED3	Observación
Modo Posicionamiento	Eje X	✓	-	-	-
	Eje Y	-	✓	-	-
	Eje Z	-	-	✓	-
Modo Measure		-	✓	-	Se enciende cada 2 s durante 50 ms.
Modo Test		✓	✓	✓	Se encienden y se apagan secuencialmente.
Modo Standby		-	-	-	-

Tabla 6.1: Indicador del modo activo

La figura 6.1 muestra un diagrama con los distintos modos de funcionamiento del dispositivo y las transiciones entre los mismos. En general, el modo de operación es cambiado por el usuario mediante el accionado de los botones.

Cada modo de funcionamiento tiene tareas diferentes, por lo que se implementaron *loops* Round-Robin independientes para cada uno. Esta estructura presenta la ventaja de poder agregar nuevas tareas en cada modo o incluso nuevos modos de operación. La arquitectura general del programa principal sigue entonces el algoritmo 2.

6.2. Modularización

Se dividió el código asociado al *firmware* del dispositivo en módulos, con el objetivo de

- organizar mejor el código del proyecto,
- verificar el *firmware* por módulo antes de verificarlo funcionando completamente,
- tener la posibilidad de reutilizar módulos en otros proyectos, por ejemplo el módulo UART podría reutilizarse para otro dispositivo que también se comunique vía UART.

La figura 6.2 presenta el diagrama de módulos asociado al *firmware* del sistema. Está presentado de manera que cada capa corresponde a un nivel distinto

Algoritmo 2 Pseudocódigo asociado al programa principal

```

init_main();                                     ▷ main
loop
  switch estado do
    case STANDBY:
      init_standby();
      do
        ...                                     ▷ Loop Round-Robin Standby
        while ((estado = STANDBY) or flag1_standby or ... or flagN_standby);
        exit_standby();
    case TEST:
      init_test();
      do
        ...                                     ▷ Loop Round-Robin Test
        while ((estado = TEST) or ...);
        exit_test();
    case POSICIONAMIENTO:
      init_posicionamiento();
      do
        ...                                     ▷ Loop Round-Robin Posicionamiento
        while ((estado = POSICIONAMIENTO) or ...);
        exit_posicionamiento();
    case MEASURE:
      init_measure();
      do
        ...                                     ▷ Loop Round-Robin Measure
        while ((estado = MEASURE) or ...);
        exit_measure();
  end switch
end loop

```

de abstracción, comenzando desde abajo con módulos más básicos y más cercanos al *hardware*, y avanzando hacia arriba hasta llegar al programa principal del proyecto que utiliza todos los demás módulos. Cada módulo se construye a partir de módulos inferiores a él.

- **HAL_MCU**: Módulo de abstracción del *hardware* del microcontrolador. Contiene funciones que independizan al resto del *firmware* del microcontrolador elegido.
- **HAL_accel**: Módulo de abstracción del *hardware* del acelerómetro. Análogamente a HAL_MCU, permite que el resto del *firmware* se abstraiga del acelerómetro elegido. Incluye un archivo encabezado `accel_registers.h` que agrupa las direcciones de los registros del acelerómetro ADXL345.
- **SPI**: Módulo que gestiona la comunicación SPI del microcontrolador con el acelerómetro. Contiene funciones de inicialización y de envío y recepción de mensajes SPI.

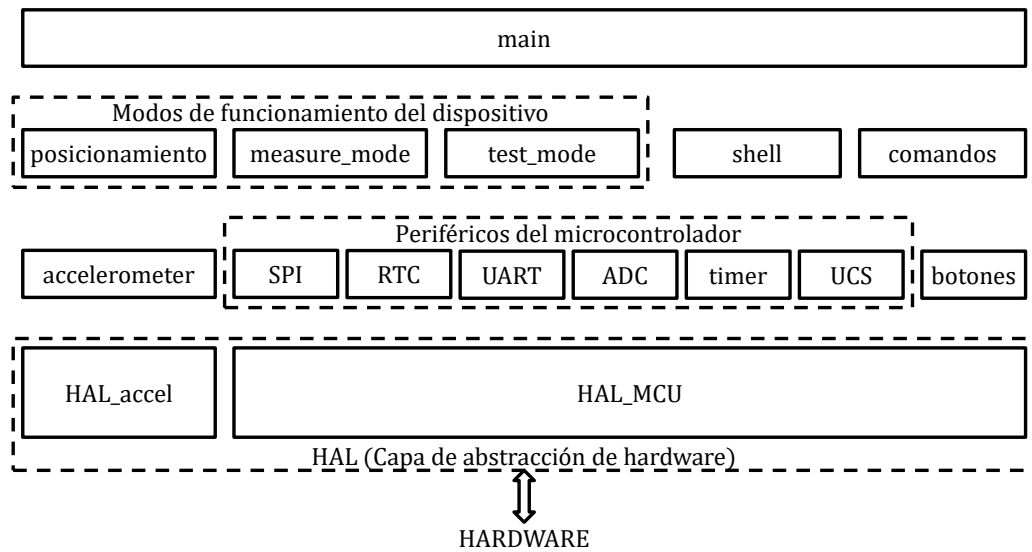


Figura 6.2: Diagrama de módulos asociado al *firmware* del sistema

- **RTC:** Módulo que maneja el periférico RTC del microcontrolador. Este periférico resuelve mediante *hardware* la implementación del reloj de tiempo real. El módulo contiene funciones para su inicialización y calibración, así como para la lectura y escritura de la hora actual.
- **UART:** Módulo que gestiona la comunicación UART del microcontrolador con el PC. Incluye la rutina de interrupción de envío/recepción de mensajes, así como funciones de inicialización del periférico USCI del microcontrolador y de manejo de *buffers* de recepción y transmisión.
- **ADC:** Módulo con funciones de inicialización y lectura del convertor ADC del microcontrolador. Mediante interrupciones, realiza un control periódico del nivel de batería del dispositivo. En caso de detectar nivel de batería bajo o crítico y estar funcionando en modo Measure, guarda los datos recabados y finaliza la medida, retorna a modo Standby y utiliza los LEDs para informar al usuario según el documento de requerimientos (apéndice A).
- **Timer:** Módulo para el manejo de los *timers* del sistema. Cuenta con las rutinas de interrupción de cada *timer*.
- **UCS:** Módulo que configura el periférico UCS del microcontrolador. Dicho periférico genera y controla las diferentes señales de reloj necesarias del sistema. Contiene una única función que configura el reloj principal MCLK en 4 MHz, el reloj ACLK (reloj fuente de periféricos RTC y *timer*) en 32,768 kHz (cristal externo) y el reloj SMCLK (reloj fuente de SPI y UART) en 1 MHz.
- **Botones:** Módulo que contiene la inicialización de los puertos utilizados para la detección de los botones y la rutina de atención a interrupción debido al accionado de un botón. Esta rutina cambia el estado del sistema dependiendo de cuál sea el botón y el tiempo de pulsado.

Capítulo 6. *Firmware* del dispositivo

- **Accelerometer:** Módulo con funciones para el manejo del acelerómetro. Contiene funciones de inicialización y de adquisición de aceleraciones así como la rutina de atención a interrupción del pin conectado a INT1 (el acelerómetro es configurado para que genere un flanco en dicho pin cada vez que tiene una nueva medida de aceleración). Además, este módulo cuenta con las constantes para la calibración de las medidas de aceleración según el método descrito en 4.2.1.1. La función de adquisición de aceleraciones utiliza dichas constantes de calibración para devolver las aceleraciones ya calibradas.
- **Shell:** Módulo que implementa el intérprete de comandos. Cuenta con un encabezado `shell_cmds.h` que contiene la tabla con todos los comandos implementados en el módulo `comandos`.
- **Comandos:** Módulo que reúne todas las funciones asociadas a cada comando.
- **Posicionamiento:** Módulo que implementa las funcionalidades del modo Posicionamiento. Contiene la matriz de posicionamiento y matrices auxiliares para su cálculo. Incluye funciones para realizar el algoritmo de posicionamiento diseñado.
- **Measure_mode:** Módulo que implementa el procesamiento de las aceleraciones para la detección de posición y nivel de actividad del usuario. En este módulo se encuentran los contadores estadísticos de la actividad del usuario y funciones para su almacenamiento en memoria Flash una vez finalizada la medida.
- **Test_mode:** Módulo con funciones para la adquisición de medidas y su posterior almacenamiento en memoria Flash.
- **Main:** Módulo que contiene el programa principal del *firmware* siguiendo la estructura definida en 6.1.

6.2.1. HAL (*Hardware Abstraction Layer*)

La capa de abstracción de *hardware* (HAL de sus siglas en inglés) es la única parte del *firmware* del proyecto que interactúa directamente con el *hardware* (ver figura 6.2). Todos los demás módulos se comunican con el *hardware* a través de dicha capa.

La HAL permite que el resto del *firmware* sea independiente del *hardware* utilizado. De esta manera, si se cambia el modelo del microcontrolador, el único *firmware* que debe modificarse es el asociado a dicha capa, que deberá adaptarse al nuevo microcontrolador. Está claro que esto es cierto en la medida que las prestaciones de *hardware* sean similares para encapsularlas en la HAL. Por ejemplo, una función `green_led_on()` sólo tendría sentido en plataformas de *hardware* que tengan un LED verde. La capa de abstracción de *hardware* permite entonces que el *firmware* del proyecto sea fácilmente portable a otras plataformas de *hardware*.

La HAL también reduce la complejidad del código y lo vuelve más legible, por ejemplo al utilizar la función `green_led_on()` en lugar de setear a 1 el contenido de cierto pin de salida mediante `P4OUT_bit.P4OUT4=1`.

6.2.2. RTC (*Real Time Clock*)

El microcontrolador que se utilizó (MSP430F5342 [1] [2] [20]) cuenta con un periférico RTC que resuelve la implementación de un reloj de tiempo real por *hardware*. Este periférico requiere una señal de reloj precisa, por lo que se utilizó como señal de reloj fuente un cristal externo de 32 768 Hz.

Además, el microcontrolador tiene registros de calibración del RTC (ver [2]), que corrigen desviaciones en la frecuencia del reloj. Es importante destacar que dicha calibración sólo corrige desviaciones iniciales en la frecuencia del reloj, y no desviaciones por variaciones de temperatura ni por envejecimiento de los componentes.

Para comprobar la precisión que tiene dicho RTC con el cristal externo de 32 768 Hz y calibrarlo a través de los registros del microcontrolador, se sincronizó la fecha y hora del RTC con la de una computadora, y luego de un cierto tiempo se midió el desfase temporal producido.

La sincronización de la fecha y hora del RTC con la computadora se realizó el 11/01/2018 a las 09:20:00. Luego, se verificó la fecha y hora del RTC el 18/01/2018 a las 09:20:00 de la computadora. La fecha y hora dada por el RTC fue 18/01/2018 09:20:20.

La tabla 6.2 resume lo obtenido.

	Computadora	RTC
Sincronización	11/01/2018 09:20:00	11/01/2018 09:20:00
Verificación	18/01/2018 09:20:00	18/01/2018 09:20:20
Tiempo transcurrido	7 días	7 días y 20 segundos

Tabla 6.2: Calibración del RTC

La precisión de la hora en ppm (partes por millón) a partir de dicha medida es entonces

$$\frac{(7 \text{ d} + 20 \text{ s}) - 7 \text{ d}}{7 \text{ d}} = \frac{20 \text{ s}}{7 \text{ d}} = \frac{20 \text{ s}}{168 \text{ h}} = \frac{20 \text{ s}}{10\,080 \text{ min}} = \frac{20 \text{ s}}{604\,800 \text{ s}} \approx 33 \text{ ppm}$$

Se observa que sin haber realizado la calibración, el reloj de tiempo real no cumple con los requerimientos (ver apéndice A), ya que la desviación obtenida fue mayor a dos segundos por día, que corresponde aproximadamente a 23 ppm.

Se ajustaron los registros de calibración del RTC correspondientes (RTCCALS y RTCCAL) según la hoja de datos de la familia del microcontrolador utilizado [2]:

- La frecuencia del reloj es más alta que la que debería ser, y por lo tanto se elige `RTCCALS=0`.

Capítulo 6. Firmware del dispositivo

- Para $RTCCALS=0$, cada LSB de $RTCCAL$ provoca un ajuste de aproximadamente -2 ppm, y por lo tanto se elije $RTCCAL=16$ (ya que $\frac{33 \text{ ppm}}{2 \text{ ppm}} = 16,5$).

Finalmente, para comprobar la precisión que tiene el RTC ya calibrado, se realizó la misma prueba que se mencionó anteriormente: se sincronizó la fecha y hora del RTC con la de una computadora, y luego de un cierto tiempo se midió el desfase temporal producido.

La tabla 6.3 resume lo obtenido.

	Computadora	RTC
Sincronización	04/02/2018 07:40:00	04/02/2018 07:40:00
Verificación	14/02/2018 07:40:00	14/02/2018 07:40:01
Tiempo transcurrido	7 días	7 días y 1 segundo

Tabla 6.3: RTC calibrado

La precisión de la hora en ppm (partes por millón) a partir de dicha medida es entonces

$$\frac{(7 \text{ d} + 1 \text{ s}) - 7 \text{ d}}{7 \text{ d}} = \frac{1 \text{ s}}{168 \text{ h}} = \frac{1 \text{ s}}{10\,080 \text{ min}} = \frac{1 \text{ s}}{604\,800 \text{ s}} \approx 1,7 \text{ ppm}$$

Se observa que luego de la calibración, la desviación de la hora del RTC disminuyó significativamente logrando cumplir con los requerimientos, ya que la misma fue menor a dos segundos por día.

6.3. Modo Measure

El modo Measure es el modo fundamental del dispositivo. En este modo, se realizan las siguientes tareas:

1. Se inicia una medida, reseteando todos los contadores de posición y nivel de movimiento.
2. Se adquieren aceleraciones a frecuencia variable siguiendo el algoritmo detallado en la sección 5.1.
3. Se procesan los datos de aceleración según los algoritmos de detección de posición y movimiento propuestos en el capítulo 5.
4. Según el resultado del procesamiento, se clasifica la posición y actividad del usuario en una de las posibles categorías y se incrementan los contadores correspondientes.
5. Una vez finalizada la medida, ya sea por cambio de día o por indicación del usuario, se guardan todos los datos estadísticos recabados en memoria Flash.

Teniendo en cuenta las mencionadas tareas, se implementó la estructura de Round-Robin con interrupciones mostrada por el algoritmo 3. A continuación, se describe cada una de las banderas (*flags*) utilizadas en el Round-Robin asociado al modo Measure.

Algoritmo 3 Pseudocódigo asociado al modo Measure

```

init_measure();
do
    if AcelRdy then
        AcelRdy = 0;
        start_data_read();
    end if
    if flag_SPI then
        flag_SPI = 0;
        get_accel();
        add_measure_data();
    end if
    if flag_1sec then
        flag_1sec = 0;
        update_measure_status();
        update_position();
    end if
    if flag_new_day then
        flag_new_day = 0;
        end_measure();
        init_measure();
    end if
    if not (AcelRdy or flag_SPI or flag_1sec or flag_new_day) then
        low_power_mode();
    end if
while (estado = MEASURE) or AcelRdy or flag_SPI or flag_1sec or flag_new_day
end_measure();

```

- **AcelRdy**: Indica que el acelerómetro tiene un nuevo dato pronto para leer. El acelerómetro genera un flanco en su pata INT1, la cual está conectada a un pin del microcontrolador. Dicho pin está configurado para generar una interrupción en la que únicamente se setea la bandera **AcelRdy**. Al detectar esta *flag* encendida, el programa principal envía la secuencia necesaria al acelerómetro para que este responda con el dato de aceleración a través de la función `start_data_read()`.
- **flag_SPI**: Indica que se recibió un nuevo dato de aceleración por SPI. Es seteada por la rutina de atención a interrupciones de la comunicación SPI cuando se detecta que se recibió un mensaje completo. La función `get_accel()` obtiene el nuevo dato de aceleración del *buffer* de recepción de SPI. Luego, la función `add_measure_data()` lo agrega al *buffer* de procesamiento del módulo Measure y si corresponde (depende si se está operando con frecuencia de muestreo alta o baja), le aplica el filtro pasa-bajos.

Capítulo 6. *Firmware* del dispositivo

- **flag_1sec**: Es seteada por la rutina de atención a interrupción del timer configurado para interrumpir cada 1 segundo. Cuando es seteada, la función `update_measure_status()` realiza el algoritmo de detección de movimiento e incrementa el contador correspondiente. Además, realiza el algoritmo de variación de la frecuencia de muestreo del acelerómetro dependiendo si se clasificó al usuario en movimiento o no. Luego, la función `update_position()` ejecuta el algoritmo de detección de posición e incrementa el contador correspondiente. A su vez, comparando la posición actual con la última registrada, detecta si se realizó un giro, una “acostada” o una “levantada” y actualiza los datos estadísticos en caso que así sea.
- **flag_new_day**: Indica que el RTC marca un nuevo día. La función `end_measure()` finaliza la medida en progreso y vuelca los datos estadísticos recabados en memoria Flash. Luego, la función `init_measure()` resetea los contadores e incrementa el puntero de día actual, iniciando una nueva medida.

En caso que ninguna de las *flags* esté seteada, el microcontrolador es llevado a un estado de bajo consumo (LPM0) a esperar alguna interrupción que lo despierte. Por otra parte, la ejecución sale del *loop* cuando el usuario acciona el botón para retornar a modo Standby. En ese caso, se llama a la función `end_measure()` que finaliza la medida y guarda los datos en memoria Flash.

6.3.1. Tiempos de respuesta

Cada tarea mencionada anteriormente tiene un requerimiento temporal (*deadline*) que, en caso de no respetarse, altera el correcto funcionamiento del programa. Los *deadlines* de cada evento están determinados por la frecuencia con la que suceden. La tabla 6.4 resume los tiempos de atención máximos para cada *flag*, consideradas independientes. Se analiza el caso con frecuencia de muestreo de aceleraciones alta ($f_s = 12,5$ Hz) ya que es el caso en el cual el procesamiento es mayor (persona en movimiento) y los tiempos de respuesta deben ser menores.

Tarea	Frecuencia	Deadline
AcelRdy	12,5 Hz	80 ms
flag_SPI	12,5 Hz	80 ms
flag_1seg	1 Hz	1 s
flag_new_day	Una vez por día	24 h

Tabla 6.4: Resumen de *deadlines* de tareas de modo Measure

El entorno de desarrollo IAR permite contar los ciclos que lleva ejecutar cierta parte del programa. A partir de esta medida y la frecuencia de operación del microcontrolador configurada ($MCLK = 4$ MHz), se registraron los tiempos de atención a cada tarea por separado en la tabla 6.5.

Tarea	Ciclos	Tiempo de ejecución
AcelRdy	2685	671 μ s
flag_SPI	13 622	3,41 ms
flag_1sec	2076	519 μ s
flag_new_day	255 428	63,86 ms
Total	273 811	68,45 ms

Tabla 6.5: Ciclos y tiempo de ejecución de cada tarea de modo Measure

Se observa que la tarea que más tiempo demanda es la asociada con la bandera `flag_new_day`. Cuando está seteada, se debe guardar todos los datos recabados en memoria no volátil para iniciar una nueva medida. Esto requiere varios ciclos de borrado y escritura de la memoria Flash. Debido a esto, los demás tiempos son prácticamente despreciables respecto al de dicha *flag*.

Para respetar todos los *deadlines*, se debe cumplir que la suma de los tiempos de ejecución de cada tarea (peor caso) sea menor al *deadline* más exigente. Esto se cumple ya que el tiempo de ejecución total resulta 68,45 ms mientras que la tarea con requisito temporal más exigente tiene un tiempo de atención máximo de 80 ms.

6.4. Modo Standby

El objetivo del modo Standby es mantener el dispositivo encendido pero consumiendo lo menos posible. En este modo, se desea que el acelerómetro y el microcontrolador permanezcan en modo de bajo consumo. El microcontrolador únicamente es despertado por interrupciones para, por ejemplo, actualizar la hora del RTC o para el monitoreo de la batería.

Con dicho fin, se decidió que el modo Standby sólo se encargue de la comunicación con el PC (en caso de haberla). Dicha comunicación está basada en un intérprete de comandos implementado en el módulo `shell`. Dicho módulo fue implementado en el marco del curso de Sistemas Embebidos para Tiempo Real, y su versión inicial fue provista por Conrado Rossi.

6.4.1. Intérprete de comandos

La comunicación con el PC está basada en mensajes predeterminados denominados comandos. Los mensajes son delimitados por el carácter ASCII `'\r'`. El encabezado `shell_cmds.h` contiene la tabla de comandos posibles y punteros a sus funciones asociadas.

La tabla 6.6 resume los comandos implementados. La respuesta a cada comando se implementó según los requerimientos en el apéndice A.

Capítulo 6. *Firmware* del dispositivo

Comando	Sintaxis
Consulta la hora actual.	RH
Setea la hora actual.	WH [dd] [mm] [aaaa] [hh] [mm] [ss]
Comienza envíos periódicos de aceleraciones.	GO
Detiene envíos periódicos de aceleraciones.	ST
Setea la frecuencia de envío de aceleraciones.	SO [odr] (ver tabla A.2 en el apéndice A)
Consulta el voltaje de la batería.	RB
Agrega un usuario.	AD [name]
Elimina un usuario.	DL [user]
Muestra los usuarios guardados.	SH
Selecciona el usuario actual.	SU [user]
Descarga los datos estadísticos.	DD [user] [day]
Descarga los datos de “levantadas”.	SD [user] [day]
Descarga los datos de “acostadas”.	LY [user] [day]
Descarga los datos del modo Test.	TM
Configura umbrales de actividad.	TH [x]
Consulta de posicionamiento realizado.	TP

Tabla 6.6: Comandos aceptados por el intérprete de comandos

Al recibir un mensaje completo (línea de caracteres), el *shell* lo analiza y separa en comando y argumentos (parámetros). Luego, recorre la tabla de comandos y en caso de encontrar el comando ingresado, ejecuta su función asociada con los parámetros ingresados.

6.5. Uso de memoria

La tabla 6.7 compara la memoria usada por el *firmware* con la estimada en el apéndice C y la disponible en el microcontrolador elegido.

	Usada	Estimada	Disponible
Memoria Flash	42,1 kB	50 kB	128 kB
Memoria RAM	4,1 kB	3,6 kB	10 kB

Tabla 6.7: Comparación de la memoria usada con la estimada y la disponible

La tabla 6.8 resume el mapa de memoria del *firmware* final del microcontrolador.

La estimación realizada inicialmente resultó aceptable, dentro de los márgenes de error esperables. Se utilizó un espacio de memoria significativamente menor al total disponible en el microcontrolador elegido. Este aspecto permitiría seguir expandiendo el *firmware* en caso que se quieran añadir funcionalidades al dispositivo. En caso contrario, en un re-diseño se podría optar por un microcontrolador de menor memoria (y por ende, de menor costo). Por ejemplo, el microcontrolador MSP430F5340 mantiene el mismo *footprint* y las mismas funcionalidades que el elegido (MSP430F342) y tiene 6 kB de memoria RAM y 64 kB de memoria Flash.

	Dirección	Uso
Periféricos	0x0000 - 0x0FFF	Reservado
BSL memory (Flash)	0x1000 - 0x17FF	Reservado
Info memory (Flash)	0x1800 - 0x1BFF	Libre
RAM	0x1C00 - 0x1C83	Datos a inicializar
	0x1C84 - 0x2BAB	Datos inicializados en cero
	0x2BAC - 0x435F	Libre
	0x4360 - 0x43FF	Stack
Main memory (Flash)	0x4400 - 0x4555	Código RTC
	0x4556 - 0x4FFF	Libre
	0x5000 - 0x7FFF	Buffer modo Test
	0x8000 - 0x81FF	Modo Measure - Users Info
	0x8200 - 0xABFF	Modo Measure - Users Data
	0xAC00 - 0xFFFF	Libre
	0xB000 - 0xB1CF	Constantes
	0xB1D0 - 0xB253	Valores iniciales de datos
	0xB254 - 0xFB11	Código
	0xFB12 - 0xFF7F	Libre
	0xFF80 - 0xFFFF	Vectores de interrupción
	0x10000 - 0x243FF	Libre

Tabla 6.8: Mapa de memoria del microcontrolador

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 7

Adquisidor de datos

7.1. Software Adquisidor de Datos

El adquisidor de datos se implementó con Python 3.6.2 y el módulo de desarrollo de GUI Tkinter. Con PyInstaller se crearon programas ejecutables para Windows y Linux en arquitecturas de 64 bits. Para poder utilizarlo es necesario tener conectado el dispositivo y haber instalados los *drivers* del mismo (en el apéndice E se detalla cómo instalarlos, tanto para Windows como para Linux).

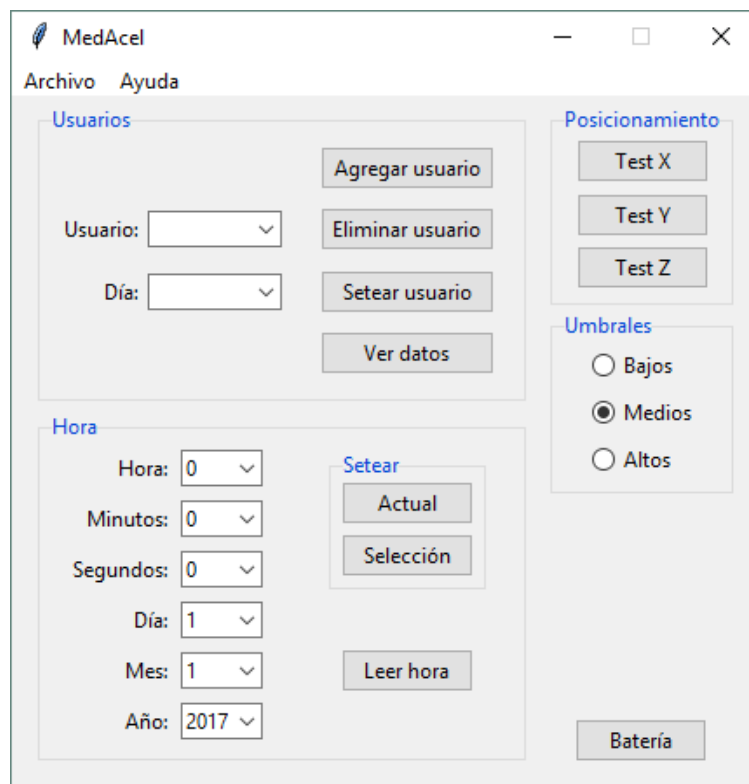


Figura 7.1: Ventana principal del adquisidor de datos

Capítulo 7. Adquisidor de datos

El *software* permite las siguientes acciones:

1. Seteo de hora del dispositivo, ya sea ingresada manualmente por el usuario o la actual del PC.
2. Consulta de hora del dispositivo.
3. Ingreso y borrado de usuario del dispositivo.
4. Descarga de los datos adquiridos por el dispositivo.
5. Testeo de posicionamiento.
6. Seteo de umbrales de actividad.
7. Consulta de nivel de batería del dispositivo.
8. Funciones avanzadas (descarga y gráfica de datos de modo Test o gráfica de datos de aceleración en tiempo real).

7.2. Manual de usuario

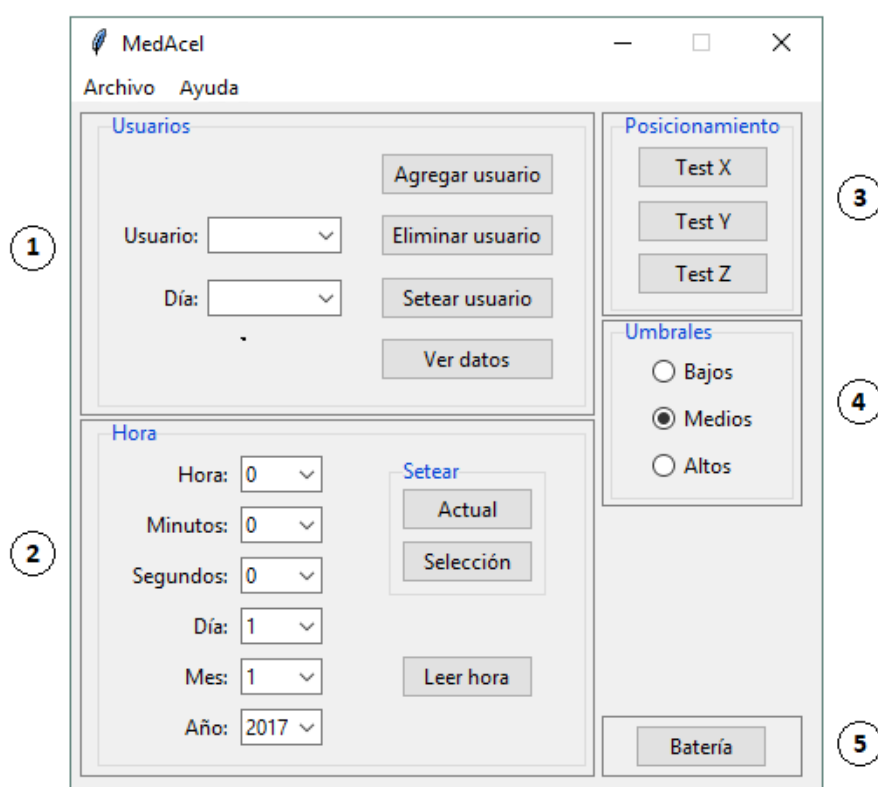


Figura 7.2: Secciones de la ventana principal

La ventana principal del adquisidor de datos está dividida en 5 secciones (figura 7.2):

1. configuración de usuarios y descarga de datos,
2. consulta y seteo de hora del dispositivo,
3. testeo de posicionamiento,
4. configuración de umbrales de actividad, y
5. consulta de batería.

Configuración de usuario y descarga de datos

En la sección de configuración de usuario y descarga se destacan los siguientes campos (figura 7.3):

1. Lista de selección de **usuario** (además permite el ingreso de un nuevo usuario).
2. Lista de selección de **día**.
3. **Agregar usuario**, agrega el usuario ingresado en el campo de usuario.
4. **Eliminar usuario**, elimina el usuario seleccionado.
5. **Setear usuario**, setea el usuario seleccionado como usuario actual. El usuario actual figura en la lista con su nombre en mayúsculas (figura 7.4).
6. **Ver datos**, muestra los datos del usuario y día seleccionado en una nueva ventana (figura 7.5).

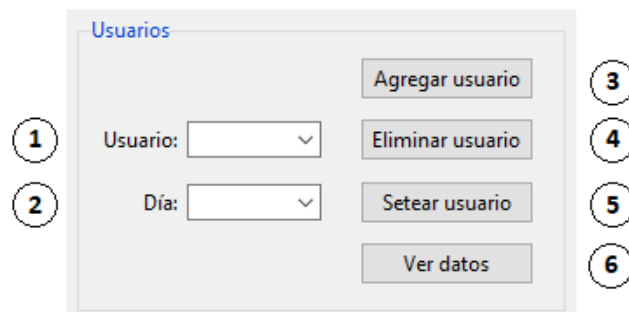


Figura 7.3: Sección de configuración de usuario y descarga de datos

Capítulo 7. Adquisidor de datos

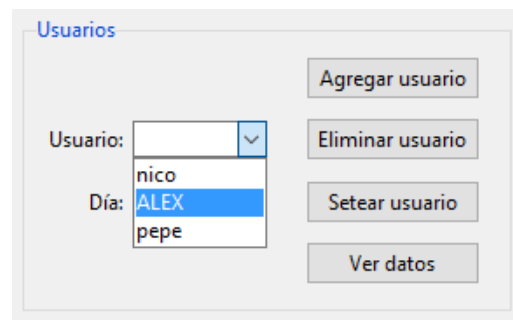


Figura 7.4: Listado de usuarios (usuario actual en mayúsculas)

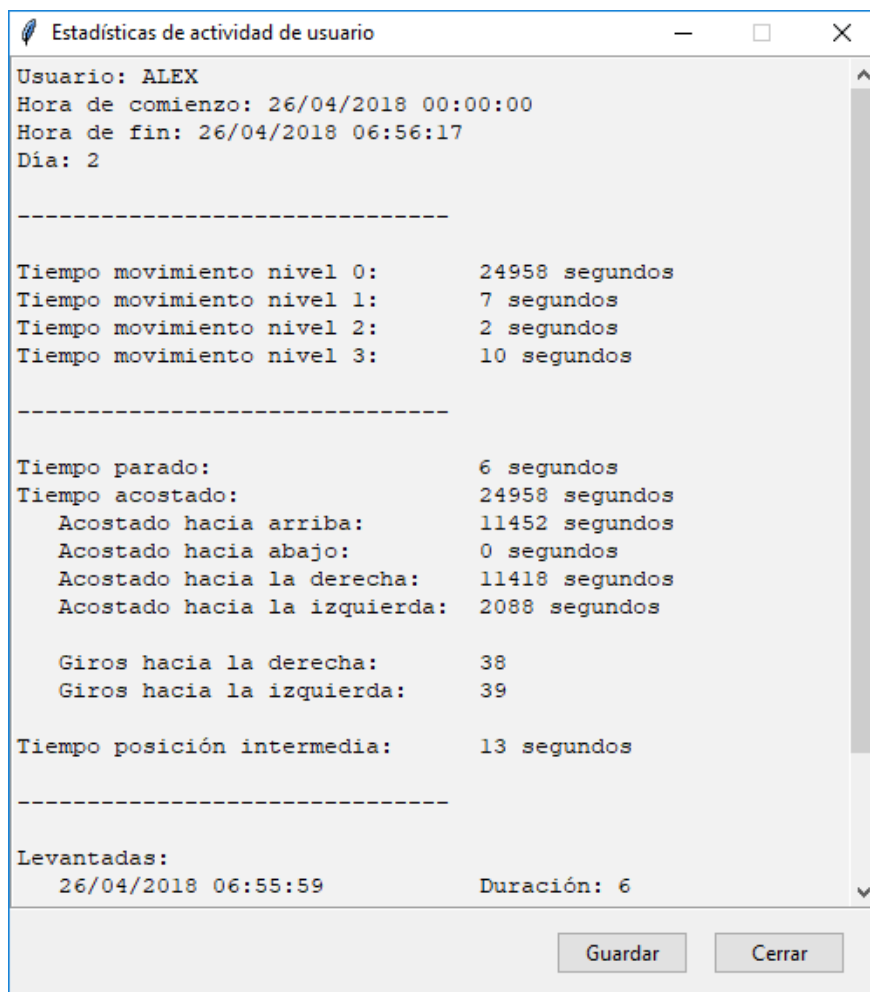


Figura 7.5: Ventana de visualización de datos adquiridos

Consulta y seteo de hora del dispositivo

La sección de consulta y seteo de hora (figura 7.6) incluye:

1. Campos de ingreso de hora a setear.

2. **Actual**, que setea la hora actual del PC como hora del dispositivo.
3. **Selección**, que setea la hora ingresada como hora del dispositivo.
4. **Leer hora**, que realiza una lectura de la hora del dispositivo.

The image shows a software interface titled 'Hora'. On the left, there is a form with six dropdown menus: 'Hora: 0', 'Minutos: 0', 'Segundos: 0', 'Día: 1', 'Mes: 1', and 'Año: 2018'. A circled '1' is placed to the left of these fields. On the right, there is a section titled 'Setear' containing three buttons: 'Actual' (with a circled '2'), 'Selección' (with a circled '3'), and 'Leer hora' (with a circled '4').

Figura 7.6: Sección de consulta y seteo de hora del dispositivo

Testeo de posicionamiento

Esta sección permite corroborar en el PC que el usuario haya posicionado correctamente el dispositivo. El usuario se puede colocar en cada una de las tres posiciones (parado, acostado boca arriba o acostado sobre el hombro derecho) y chequear que el eje correspondiente haya sido posicionado correctamente. Se tienen entonces 3 acciones.

- **Test X**, que verifica la posición del eje X (acostado boca arriba).
- **Test Y**, que verifica la posición del eje Y (acostado sobre hombro derecho).
- **Test Z**, que verifica la posición del eje Z (parado).

Cada una de estas acciones responde con un mensaje que indica si el dispositivo pudo detectar la posición del usuario correctamente o no. En caso que el dispositivo no haya sido posicionado aún, aparece un mensaje pidiendo al usuario que realice el posicionamiento del dispositivo.

Configuración de umbrales de actividad

Como se explicó en la sección 5.4, el dispositivo discierne el movimiento del usuario procesando las aceleraciones y comparando el resultado alcanzado con un conjunto de umbrales. El adquisidor de datos permite la configuración de dichos umbrales en 3 opciones: **Bajos**, **Medios** y **Altos**.

Capítulo 7. Adquisidor de datos

Los umbrales medios son los umbrales obtenidos según la sección 5.4 y son entonces la opción “por defecto”. Los umbrales bajos y altos tienen como finalidad considerar la edad y la condición física del usuario (por ejemplo, los umbrales bajos para una persona de avanzada edad y los umbrales altos para una persona deportista). De esta manera, la actividad detectada por el dispositivo tiene una mayor relación con el esfuerzo físico realizado por el usuario.

Consulta de batería

Al accionar el botón **Batería**, el adquisidor de datos responde con el nivel de batería según la figura 7.7.

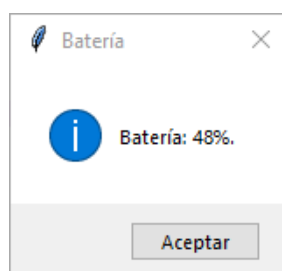


Figura 7.7: Ventana de consulta de nivel de batería

Para expresar el voltaje de las baterías como un porcentaje, se dejó el dispositivo funcionando varios días y se relevó el voltaje de las baterías al descargarse uniformemente en el tiempo. Con dichos datos (voltaje y tiempo de descarga), se implementó una *lookup table* (o “tabla de consulta”). Cuando se recibe el voltaje de las baterías (comando `RB\r`), se busca en la tabla y se obtiene el tiempo restante de las baterías interpolando linealmente con los voltajes de la tabla inmediatamente por arriba y por abajo. A partir de dicho tiempo, se calcula el nivel de batería como un porcentaje del tiempo total de descarga del dispositivo.

Funciones avanzadas

Las funciones avanzadas son principalmente de desarrollo. Por este motivo, se decidió que no fueran accesibles al usuario regular. Al seleccionar la opción Avanzado bajo el menú de Ayuda, se abrirá una ventana (figura 7.8) que solicita el ingreso de una contraseña (“ma2018”).

Al ingresar la contraseña solicitada, aparece una nueva sección en la ventana principal (figura 7.9). En la misma, existen 3 acciones disponibles:

- **Tiempo real**, que abre una ventana con una gráfica en tiempo real de la aceleración medida por el dispositivo.
- **Datos TM**, que permite la descarga de los datos de modo Test adquiridos en un archivo de texto *.txt*.

- **Plot TM**, que grafica los datos de modo Test previamente guardados en un archivo de texto *.txt*.

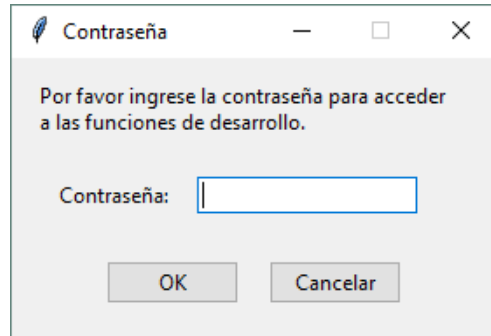


Figura 7.8: Ventana de ingreso de contraseña

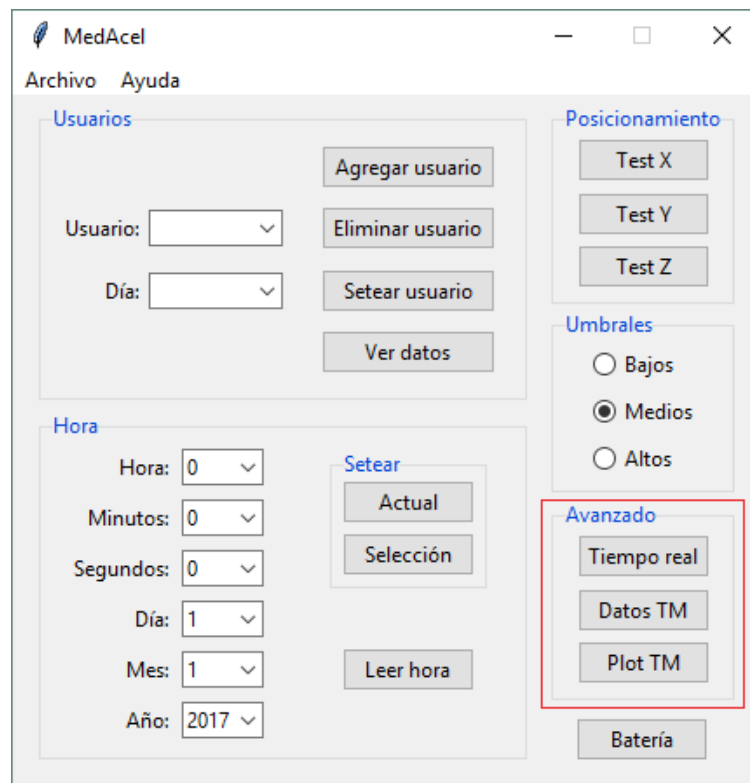


Figura 7.9: Ventana principal con funciones avanzadas

Esta página ha sido intencionalmente dejada en blanco.

Capítulo 8

Prototipo

Para poder comprobar el funcionamiento del sistema antes de hacer una versión final del dispositivo, se procedió a diseñar una versión prototipo del mismo.

El objetivo fue que dicho prototipo tuviera todas las funcionalidades que se esperan para el dispositivo final, pero sin restricciones de tamaño ni peso. Es decir que no se incluyó el diseño mecánico, en particular el método de sujeción del dispositivo al torso de la persona.

Se utilizó el *software* de diseño de PCB EAGLE (*Easily Applicable Graphical Layout Editor*) para diseñar el esquemático y el *layout*.

8.1. Esquemático

Para diseñar el esquemático del prototipo, se utilizaron las configuraciones y valores de los componentes descritos en el capítulo 3. A continuación, se muestra el esquemático asociado a cada uno de los componentes del sistema.

8.1.1. Step-up

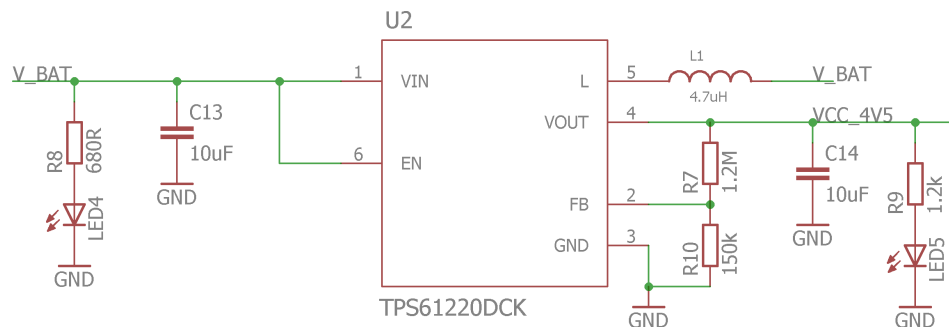


Figura 8.1: Esquemático asociado al *step-up*

Se incluyeron dos LEDs, uno al voltaje de entrada y uno al de salida, para tener un indicador visual de dichos voltajes. Las resistencias R_8 y R_9 asociadas a los respectivos LEDs se calculan a continuación.

Capítulo 8. Prototipo

De la hoja de datos de dichos LEDs (LS T67K, LY T67K y LG T67K de OSRAM) [21], se obtiene su corriente nominal $I_{LED} = 2\text{ mA}$ y el voltaje para el cual circula dicha corriente $V_{LED@2\text{mA}} = 1,8\text{ V}$.

$$R_8 = \frac{V_{R8}}{I_{R8}} = \frac{V_{BAT} - V_{LED4}}{I_{LED4}} \approx \frac{3,0\text{ V} - 1,8\text{ V}}{2\text{ mA}} = 600\ \Omega$$

$$R_9 = \frac{V_{R9}}{I_{R9}} = \frac{V_{CC4V5} - V_{LED5}}{I_{LED5}} \approx \frac{4,5\text{ V} - 1,8\text{ V}}{2\text{ mA}} = 1,35\text{ k}\Omega$$

Se eligió entonces $R_8 = 680\ \Omega$ y $R_9 = 1,2\text{ k}\Omega$.

8.1.2. Regulador lineal

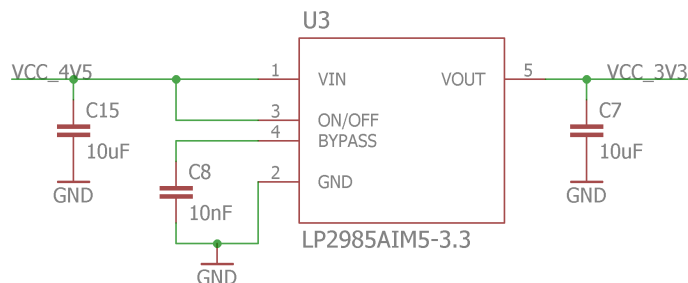


Figura 8.2: Esquemático asociado al regulador lineal

8.1.3. Acelerómetro

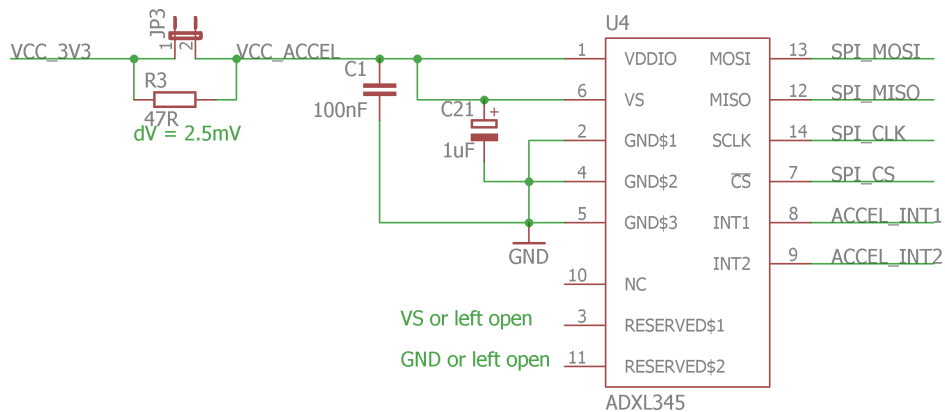


Figura 8.3: Esquemático asociado al acelerómetro

Se incluyó una resistencia R_3 en paralelo con un *jumper* JP_3 . Dichos componentes se agregaron para medir el consumo del acelerómetro, como se describirá más adelante.

8.1. Esquemático

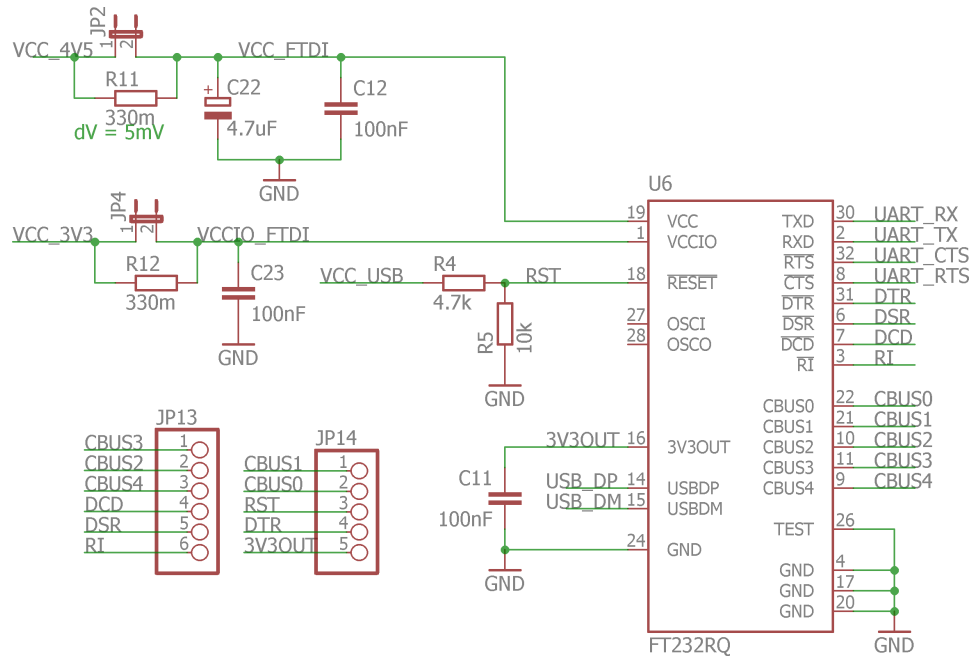


Figura 8.4: Esquemático asociado al FT232R

8.1.4. Conversor USB-UART

En este sub-circuito también se incluyeron resistencias R_{11} , R_{12} y *jumpers* JP_2 , JP_4 de medida de consumo, en este caso del integrado FT232R.

8.1.5. Conector mini-USB-A

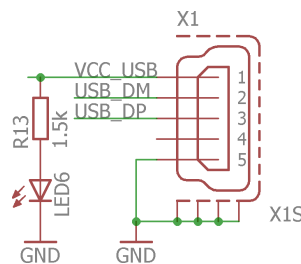


Figura 8.5: Esquemático asociado al conector USB

Se puede ver que se incluyó un LED, para tener un indicador visual de si hay voltaje en el conector mini-USB-A o no, es decir si está conectado o no. La resistencia R_{13} asociada a dicho LED se calcula a continuación.

$$R_{13} = \frac{V_{R13}}{I_{R13}} = \frac{V_{USB} - V_{LED6}}{I_{LED6}} \approx \frac{5,0 \text{ V} - 1,8 \text{ V}}{2 \text{ mA}} = 1,6 \text{ k}\Omega$$

Se eligió entonces $R_{13} = 1,5 \text{ k}\Omega$.

Capítulo 8. Prototipo

8.1.6. Conector JTAG (*Joint Test Action Group*)

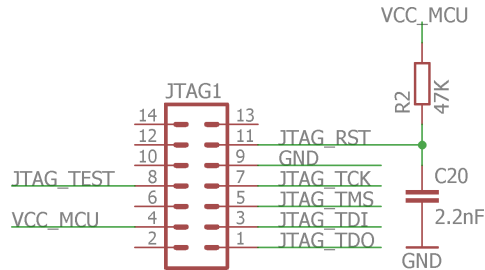


Figura 8.6: Esquemático asociado al conector JTAG

8.1.7. Protección contra descargas electrostáticas

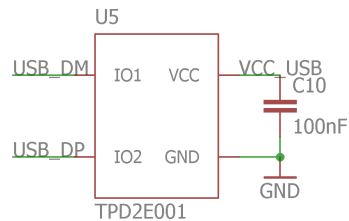


Figura 8.7: Esquemático asociado a la protección anti-estática

8.1.8. LEDs

Cada LED se conecta a un pin de salida del microcontrolador a través de una resistencia R .

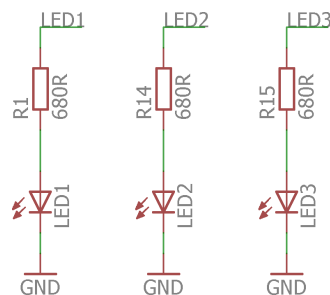


Figura 8.8: Esquemático asociado a los LEDs

Los voltajes LED1, LED2 y LED3 están conectados a pines de salida del microcontrolador.

El valor de las resistencias se calcula simplemente utilizando la ley de Ohm:

$$R = \frac{V_R}{I_R} = \frac{V_{OH} - V_{LED}}{I_{LED}} \approx \frac{3,3 \text{ V} - 1,8 \text{ V}}{2 \text{ mA}} = 750 \Omega$$

8.1. Esquemático

Se eligió entonces $R_1 = R_2 = R_3 = 680\ \Omega$ (serie E12).

Observación: Se consideró utilizar un transistor como *driver* de los LEDs. Sin embargo, se conectaron los LEDs directamente a los pines de salida del microcontrolador (sin transistor de por medio), ya que de la hoja de datos se sabe que dichos pines soportan la corriente nominal de los LEDs.

8.1.9. Botones

Los botones son activos por nivel bajo.

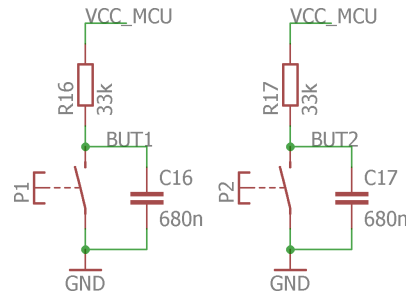


Figura 8.9: Esquemático del PCB asociado a los botones

Los voltajes BUT1 y BUT2 están conectados a pines de entrada del microcontrolador.

Las resistencias R_{16} y R_{17} limitan la corriente (evitan un cortocircuito entre fuente y tierra) al pulsar los botones P1 y P2 respectivamente. Para que la corriente por las resistencias sea $I_R = 100\ \mu\text{A}$ mientras los botones están pulsados, se eligen las resistencias de valor

$$R_{16} = R_{17} = \frac{V_{CC_MCU}}{I_R} = \frac{3,3\ \text{V}}{100\ \mu\text{A}} = 33\ \text{k}\Omega$$

Se agregaron los capacitores C_{16} y C_{17} , cuya función es eliminar rebotes.

Antes de pulsar un botón (por ejemplo P1), se tiene que no hay corriente por el capacitor C_{16} , y entonces $V_{BUT1} = V_{CC_MCU} = 3,3\ \text{V}$. Al pulsar el botón, el condensador C_{16} se descarga muy rápidamente y se tiene instantáneamente $V_{BUT1} = GND = 0\ \text{V}$.

Antes de soltar un botón (por ejemplo P1), se tiene $V_{BUT1} = GND = 0\ \text{V}$. Al soltar el botón, el condensador C_{16} se carga exponencialmente al valor $V_{CC_MCU} = 3,3\ \text{V}$ con constante de tiempo $\tau = R_{16}C_{16}$. De esta manera, si se elige la constante de tiempo τ convenientemente y se vuelve a pulsar el botón muy poco tiempo luego de soltarlo (de manera de no dejar que el condensador C_{16} se cargue), el microcontrolador no detectará que el botón fue soltado, sino que interpretará que quedó pulsado todo el tiempo. Es así que se eliminan los rebotes.

Se eligió la constante de tiempo $\tau = R_{16}C_{16} = 20\ \text{ms}$. Con $R_{16} = 33\ \text{k}\Omega$, se tiene

$$C_{16} = \frac{\tau}{R_{16}} = \frac{20\ \text{ms}}{33\ \text{k}\Omega} = 606\ \text{nF}$$

Se eligió entonces $C_{16} = 680\ \text{nF}$. Lo mismo se aplicó a C_{17} .

Capítulo 8. Prototipo

8.1.10. Conectores y *jumpers*

Se dejaron accesibles pines a los voltajes de las distintas alimentaciones. Se dejó accesible un conector para conectar la batería al circuito, y otros dos a los voltajes de 4,5 V y 3,3 V, así como algunos pines de tierra para verificar el circuito de forma práctica.

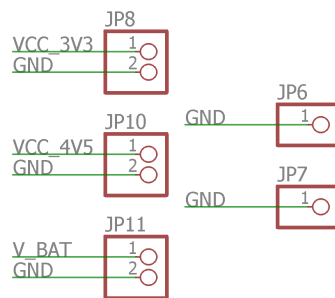


Figura 8.10: Esquemático del PCB asociado a los conectores y *jumpers*

8.1.11. Microcontrolador

En este sub-circuito también se incluyó una resistencia R_6 y un *jumper* JP_1 de medida de consumo del microcontrolador.

8.1. Esquemático

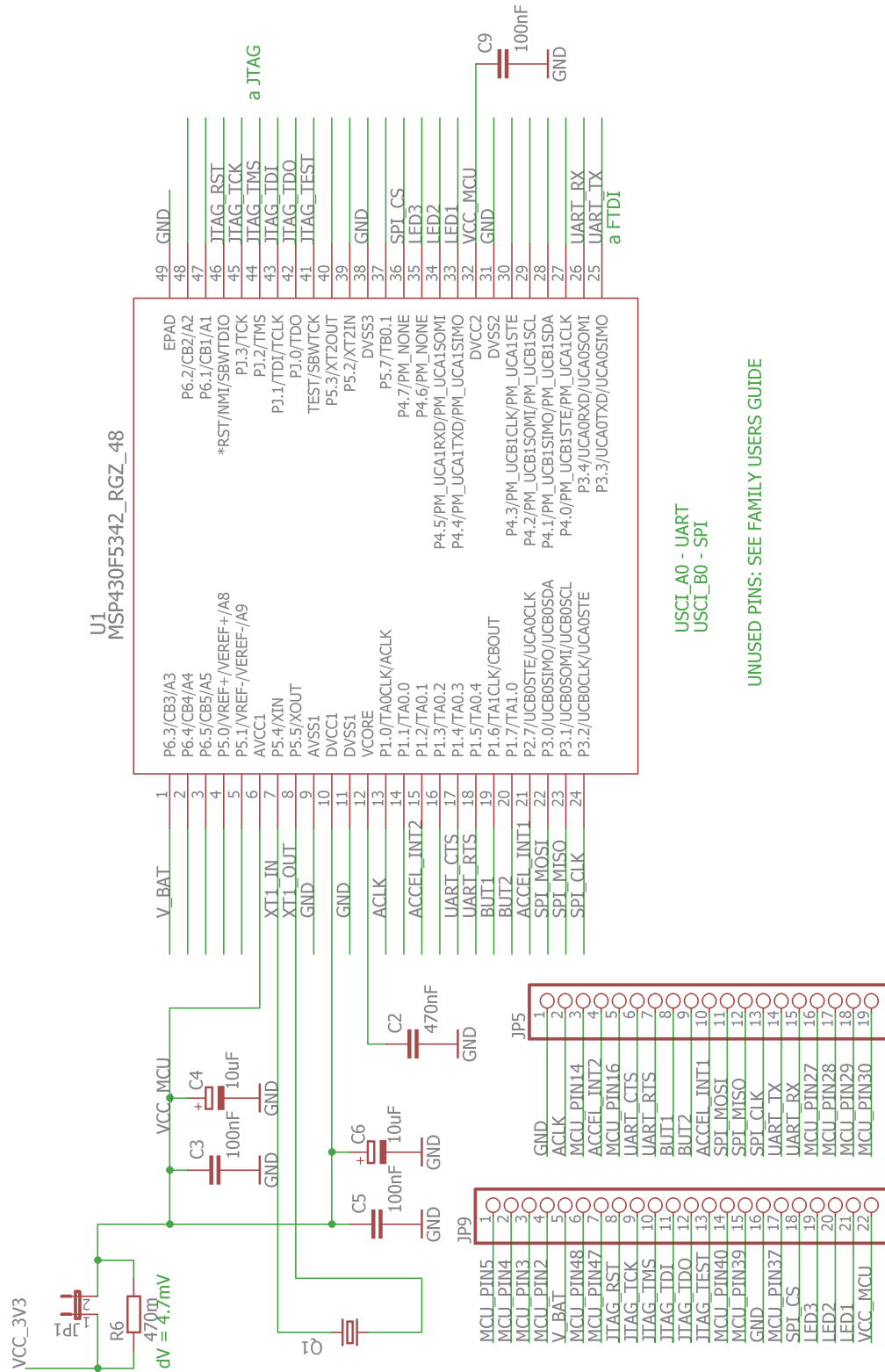


Figura 8.11: Esquemático del PCB asociado al microcontrolador

8.1.12. Resistencias de medida de consumo

Para poder medir el consumo de varios sub-circuitos del dispositivo, se agregaron resistencias en paralelo con un *jumper* en su respectiva alimentación. De esta manera, quitando el *jumper* se puede medir el voltaje en bornes de la resistencia, y por lo tanto (conociendo el valor de la resistencia) la corriente de entrada al sub-circuito.

Los sub-circuitos que tienen dicho *jumper* disponible para medir consumo son el microcontrolador (ver figura 8.11), el acelerómetro (ver figura 8.3) y el FTDI (ver figura 8.4).

Cada una de las resistencias se calcula utilizando el consumo I_{CC} esperado del sub-circuito (dado por las respectivas hojas de datos) de forma de que la caída de voltaje ΔV_{CC} en la resistencia no sea muy grande (del orden de mV o decenas de mV) utilizando la ley de Ohm:

$$R = \frac{\Delta V_{CC}}{I_{CC}}$$

En caso de que la corriente consumida tenga una forma de onda con picos, se puede poner un capacitor en paralelo con la resistencia de medida de consumo, de modo de suavizar los picos del consumo y poder medir un voltaje más constante. Además, dicho capacitor también sirve para evitar que los respectivos sub-circuitos deje de funcionar o tenga un funcionamiento anómalo debido a un pico de corriente muy grande pero de corta duración.

8.2. *Layout*

Para el ruteo del circuito se tuvo en cuenta ciertas consideraciones, teniendo especial cuidado con ciertas rutas críticas. Para el ruteo de alimentación y tierra se utilizaron pistas más anchas y en topología de estrella para evitar que el consumo de un sub-circuito haga caer el voltaje a la alimentación de los demás sub-circuitos. También se tuvo especial cuidado en el ruteo de *step-up*, intentando que las pistas que funcionan conmutando no pasen cerca de otras para no introducirles ruido de conmutación. En cuanto a los componentes, se intentó que los condensadores de desacople estén lo más cerca posible al pin del integrado a desacoplar. Muchas de estas recomendaciones fueron tomadas del libro “*The Circuit Designer’s Companion*” [22].

Luego de varias versiones e iteraciones, se llegó a un *layout* satisfactorio. El único fenómeno que resultó distinto a lo esperado estuvo relacionado con el voltaje de salida del *step-up*. Lo que ocurrió fue que el voltaje de salida deseado (4,5 V) caía al tomar más corriente. Por ejemplo, dicho voltaje caía al prender un LED, y caía aún más al prender dos LEDs.

Buscando en la literatura, se encontró una *Application Note* [23] en la que se explica que se puede agregar un capacitor de *feedforward* C_{FF} entre V_{OUT} y V_{FB} para aumentar el margen de fase del convertidor y aumentar así su estabilidad.

8.3. Consumo del dispositivo

Se soldó dicho capacitor $C_{FF} = 15\text{ pF}$ al prototipo y se observó que el voltaje de salida del *step-up* quedó en $4,5\text{ V}$ independientemente de la corriente extraída.

Pensando ya en el *layout* del diseño final, se decidió incluir el capacitor de *feed-forward* C_{FF} mencionado y tener más cuidado con el ruteo del *step-up* utilizando el sugerido por el fabricante en la hoja de datos [7].

La figura 8.12 muestra el *layout* del prototipo finalmente realizado. La figura 8.13 muestra uno de los tres prototipos fabricados.

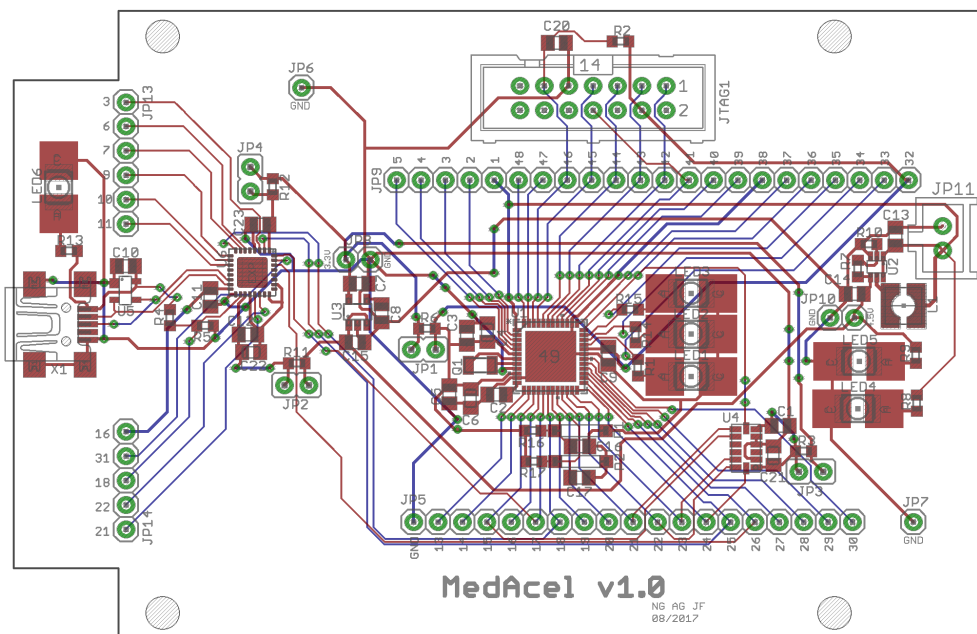


Figura 8.12: *Layout* del PCB asociado al prototipo realizado

8.3. Consumo del dispositivo

Para verificar que el dispositivo funciona como se espera en lo que respecta al consumo, se procedió a medir el consumo de varios sub-circuitos del dispositivo.

Como ya se explicó en 8.1.12, se agregaron resistencias en paralelo con un *jumper* en la alimentación de varios sub-circuitos. De esta manera, si se tiene el *jumper* cortocircuitado es como si la resistencia no estuviera, pero si se abre dicho *jumper*, se puede medir el voltaje en bornes del mismo y por lo tanto en bornes de la resistencia. Conociendo el valor de la resistencia, se halla la corriente que consume cada sub-circuito.

Los sub-circuitos que tienen dicho *jumper* disponible para medir consumo son el microcontrolador (ver figura 8.11), el acelerómetro (ver figura 8.3) y el FTDI (ver figura 8.4).

La tabla 8.1 compara el consumo del microcontrolador, acelerómetro y FTDI esperado (tomados de sus respectivas hojas de datos) con el consumo medido.

Capítulo 8. Prototipo

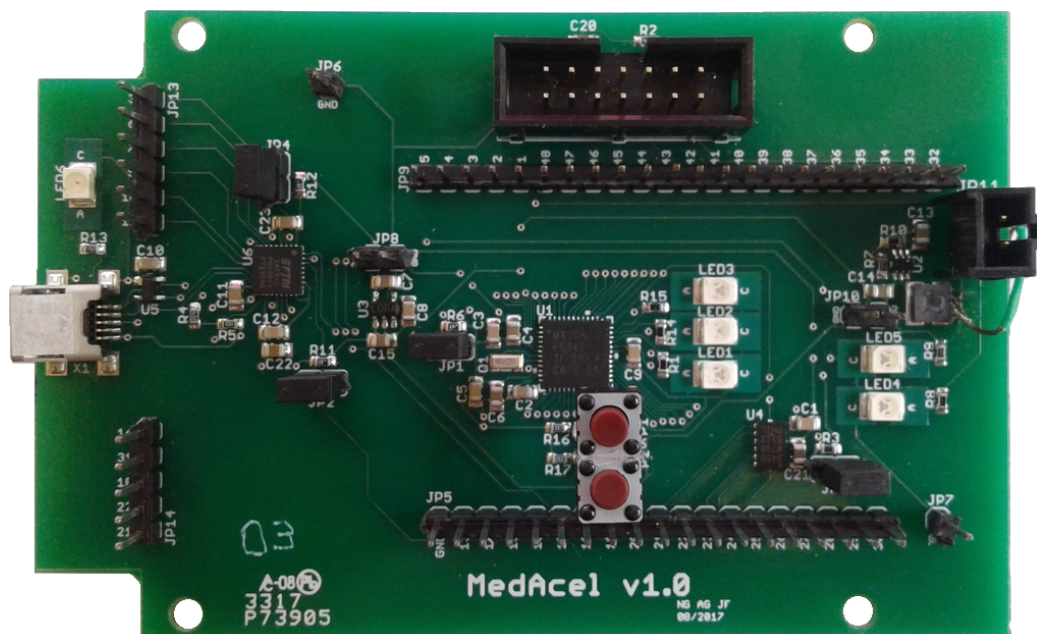


Figura 8.13: Foto del PCB asociado al prototipo realizado

Sub-circuito	Condiciones	Consumo esperado	Consumo medido
Microcontrolador	Activo	1,2 mA	1,87 mA
Microcontrolador	<i>Low Power Mode 0</i>	83 μ A	112 μ A
Acelerómetro	<i>ODR = 12,5 Hz</i>	50 μ A @2,5 V	66 μ A
Acelerómetro	<i>ODR = 1,56 Hz</i>	34 μ A @2,5 V	54 μ A
Acelerómetro	Standby	0,1 μ A @2,5 V	0,42 μ A
FTDI	Desconectado	70 μ A	98 μ A

Tabla 8.1: Consumo de cada sub-circuito a partir de las respectivas hojas de datos

Los valores medidos a través del osciloscopio en bornes de las respectivas resistencias de medida de consumo fueron en el orden de lo esperado, por lo que se prosiguió con el diseño del dispositivo final.

Capítulo 9

Diseño mecánico

A diferencia del diseño del prototipo (ver capítulo 8), al diseñar el dispositivo final sí se tuvieron en consideración los requerimientos detallados asociados a aspectos mecánicos del dispositivo como por ejemplo masa, dimensiones, grado de protección IP, y en particular los requerimientos asociados al método de sujeción: comodidad de la persona y firmeza del dispositivo respecto a la misma.

Se intentó reducir al máximo el tamaño del dispositivo tanto para no incomodar a la persona como para garantizar fácilmente la firmeza del dispositivo respecto a su torso. Se eligió entonces la caja más pequeña que contara con porta-pilas, encontrando la A9051117 *Soft-Case M* [24] de *OKW*. Dicha caja cuenta con un compartimiento para dos pilas AAA. Además, sus dimensiones son $105\text{ mm} \times 65\text{ mm} \times 19\text{ mm}$, cumpliendo el requerimiento de tamaño.

Una vez elegida la caja, se diseñó el *layout* del dispositivo con los tamaños de PCB recomendados por el fabricante de la misma en su hoja de datos [24].

Debido a la disminución del tamaño de la caja, se debió modificar el conector JTAG, ya que el mismo excedía la altura de la caja. Se buscó una solución sencilla y que altere lo menos posible lo realizado en el prototipo. Se decidió entonces mantener intacta la electrónica (ya descrita en 3.3.4) e implementar la conexión del programador con un conector *pogo pins*. Dicho conector se apoya directamente sobre el PCB para programar el microcontrolador durante varios segundos, y una vez programado se levanta. Cada pin tiene un resorte (ver figura 9.1), de manera de asegurar que todos los pines hacen contacto al mantener el conector sobre el PCB. Del lado del PCB, solamente quedan los agujeros metalizados asociados a los pines, de modo que entra en la caja.

Para ubicar los sub-circuitos en el *layout* se consideró su función y las recomendaciones de los fabricantes de los distintos integrados. En el caso de los circuitos de alimentación, fueron colocados lo más cerca posible del porta-pilas para facilitar el cableado. El conector mini-USB y el *switch* ON/OFF se colocaron tan cercanos al borde como permitía el fabricante de PCBs. Para la ubicación del acelerómetro, se siguió la recomendación de la hoja de datos [3] que sugiere ubicarlo cerca

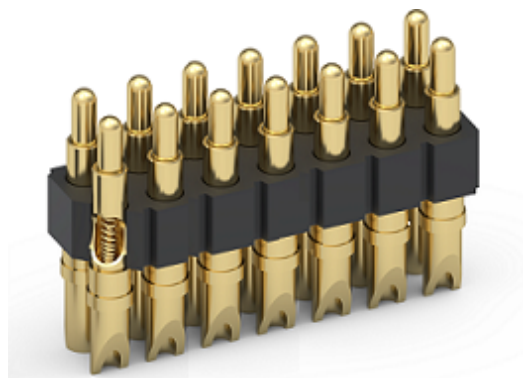


Figura 9.1: *Pogo pins*

de un agujero de montaje para que sea menos sensible a las vibraciones de la placa.

El paso siguiente fue idear cómo colocar los botones y LEDs. Se evaluó la opción de montar los botones y los LEDs (a través de porta-LEDs) directamente en agujeros a hacer en la caja y soldar los cables al PCB. Sin embargo, se descartó dicha opción optando por utilizar un teclado de membrana que ya cuenta con varios botones y LEDs y un cable *flex* que deberá conectarse al circuito. Se debió agregar el conector al PCB. La caja elegida cuenta con un pequeño relieve en el cual encaja adecuadamente el teclado de membrana, y evita que se pulsen los botones involuntariamente, por ejemplo al dormir.

También se diseñó el arte que se adhirió sobre el teclado de membrana, indicando el significado de los LEDs (LED1, LED2, LED3 y LED indicador de batería) y botones (botón 1 para entrar a modo Posicionamiento, botón 2 para entrar a modo Test y botón 3 para entrar a modo Measure). Además, se agregaron los logos asociados al Instituto de Ingeniería Eléctrica, la Facultad de Ingeniería y la Universidad de la República. La figura 9.2 muestra el arte del teclado de membrana.

Antes de fabricar el PCB asociado al *layout* diseñado, fue preciso asegurarse de que no hayan interferencias mecánicas entre ningún elemento del sistema, en particular el PCB y sus componentes soldados y la caja. En particular se verificó que la altura de todos los componentes sea menor a la de la caja. Para esto, se hizo un modelo 3D de la caja con el PCB diseñado posicionado dentro de la misma. El modelo 3D de la caja se obtuvo del fabricante de la misma, *OKW*. Las figuras 9.3 y 9.4 muestran distintas vistas de dicho modelo 3D.

Una vez verificado que no hayan interferencias mecánicas a través del modelo 3D, se fabricó el PCB asociado al *layout* diseñado (dicho *layout* se detallará más adelante en el capítulo 10).

Como se puede ver en las figuras 9.3 y 9.4, se debió hacer un calado en el costado de la caja a la altura del conector mini-USB y del *switch* para poder conectar el cable y usar el *switch* ON/OFF respectivamente.



Figura 9.2: Imagen del arte del teclado de membrana

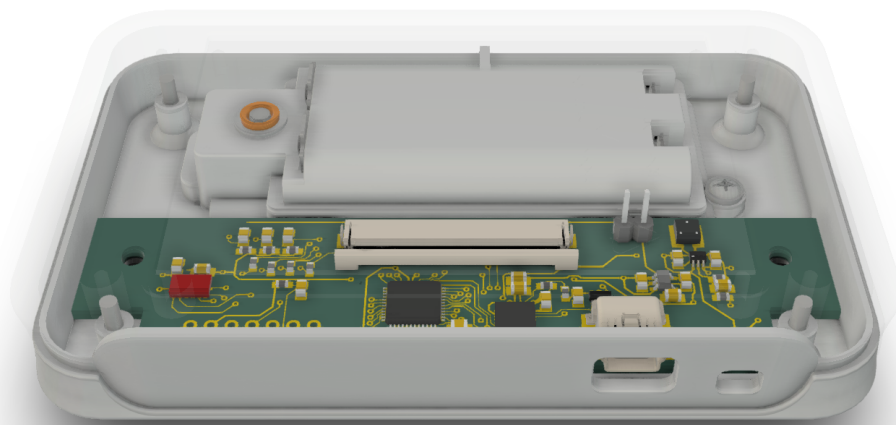


Figura 9.3: Imagen del modelo 3D realizado ya ensamblado (PCB y caja)

Para sujetar firmemente la caja al torso de la persona, se utilizó una funda de uso común en atletas para portar teléfonos móviles u otros dispositivos cosida a un elástico. Dicho elástico se coloca firmemente al torso de la persona con velcro. La figura 9.5 muestra el torso de una persona con el dispositivo colocado a través del elástico y la funda mencionadas.

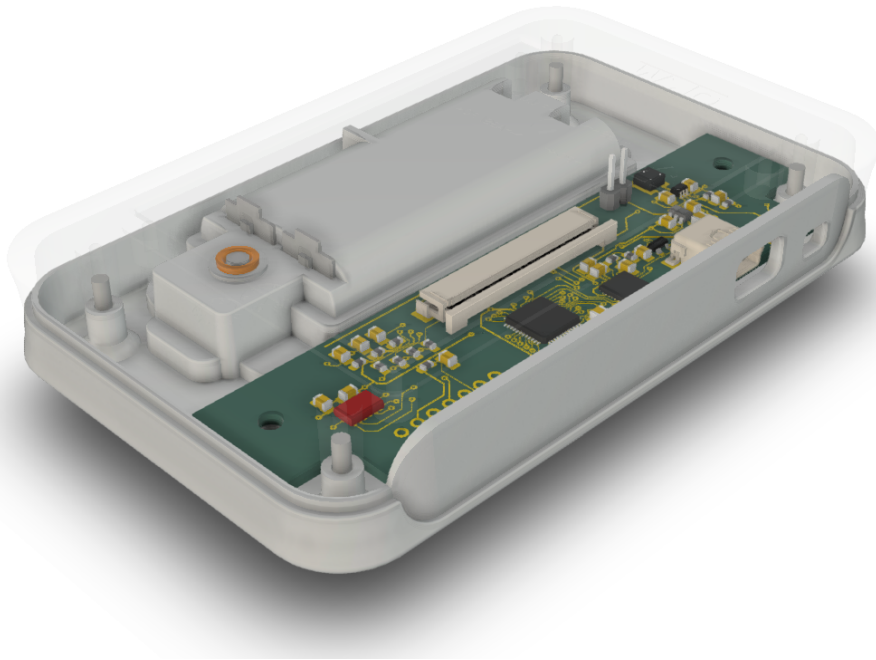


Figura 9.4: Imagen del modelo 3D realizado ya ensamblado (PCB y caja)



Figura 9.5: Dispositivo colocado en el torso de una persona

Capítulo 10

Dispositivo final

A diferencia del prototipo, el dispositivo final incluyó el diseño mecánico presentado en el capítulo 9. Este dispositivo final presenta entonces ciertas diferencias respecto al prototipo descrito en el capítulo 8, principalmente en lo que refiere al diseño mecánico: la caja, el tamaño del PCB realizado y el ordenamiento de los sub-circuitos en el mismo, botones y LEDs proveídos por el teclado de membrana. Además, se quitaron las resistencias de medida de consumo y se colocó un *switch* de encendido/apagado del dispositivo.

10.1. Esquemático

El esquemático del dispositivo final fue muy similar al del prototipo realizado, por lo que no se mostrarán todos los sub-circuitos del esquemático a nivel de detalle del prototipo (sección 8.1), sino que se mostrarán y explicarán solamente los cambios realizados. Sin embargo, los esquemáticos completos pueden verse en el apéndice G.

En primer lugar, se eliminaron los *jumpers* y resistencias de medida de consumo que se habían agregado a la entrada del microcontrolador, del acelerómetro y del FTDI (ver 8.1.12).

Se eliminaron también los LEDs que estaban presentes solamente como indicadores visuales de presencia de voltaje. Había uno para el voltaje de la batería V_{BAT} , uno para los 4,5 V de salida del *step-up*, y uno para el voltaje del conector mini-USB-A.

Como se comentó en el capítulo 9, se decidió utilizar un teclado de membrana, por lo que los botones y LEDs de interfaz de usuario con la persona pasan a estar en el teclado y no soldados sobre el PCB. Por otro lado, se tuvo que agregar un conector para dicho teclado de membrana. Las resistencias a través de las cuales se conectan los LEDs, y las resistencias y capacitores asociados a la eliminación de rebotes de los botones siguen estando presentes en el PCB.

Se agregó también un *switch* al voltaje de batería V_{BAT} para poder encender/apagar el dispositivo.

Capítulo 10. Dispositivo final

Como se explicó en 8.2, se decidió incluir un capacitor de *feedforward* $C_{FF} = 15 \text{ pF}$ entre V_{OUT} y V_{FB} para aumentar el margen de fase del convertidor y aumentar así su estabilidad y utilizar el ruteo del *step-up* sugerido por el fabricante en la hoja de datos [7].

Se agregó entonces dicho capacitor de *feedforward* $C_{FF} = 15 \text{ pF}$ al esquemático del dispositivo final.

10.2. Layout

Se tuvieron las mismas consideraciones que para el *layout* del prototipo: dimensionar el grosor de las pistas acorde a la corriente que circula por ellas (por ejemplo pistas de alimentación y tierra más gruesas), utilizar una conexión en estrella para la alimentación de cada sub-circuito para que el consumo de cada uno de ellos no influya en el voltaje de alimentación de otro, y ubicar los condensadores de desacople lo más cerca posible de sus respectivos integrados.

Al reducir el tamaño del PCB con respecto al del prototipo, se tuvo que reorganizar los distintos sub-circuitos para aprovechar al máximo el área disponible. También se tuvieron consideraciones mecánicas, como por ejemplo respecto a la posición del conector mini-USB-A y *switch* para que queden accesibles, y del conector del teclado de membrana para que pueda conectarse.

Además, como se tuvieron problemas con el voltaje de salida del *step-up*, se volvió a diseñar el *layout* asociado a dicho sub-circuito siguiendo las recomendaciones de la hoja de datos [7], con el objetivo de reducir el ruido de conmutación que puede propagarse a pistas sensibles (en particular a la de *feedback*).

La figura 10.1 muestra el *layout* del dispositivo finalmente realizado. La figura 10.2 muestra una de las tres placas finales fabricadas.

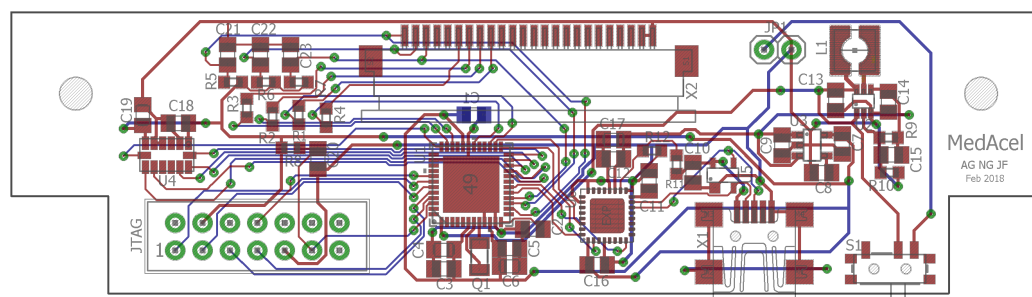


Figura 10.1: *Layout* del PCB asociado al dispositivo final realizado

10.3. Consumo y autonomía del dispositivo

Para verificar que el dispositivo funciona como se desea en lo que respecta al consumo, por ejemplo que el microcontrolador está entrando al modo de bajo consumo cuando no tiene interrupciones que atender, se procedió a analizar la corriente consumida por el dispositivo y observar las distintas interrupciones y sus

10.3. Consumo y autonomía del dispositivo

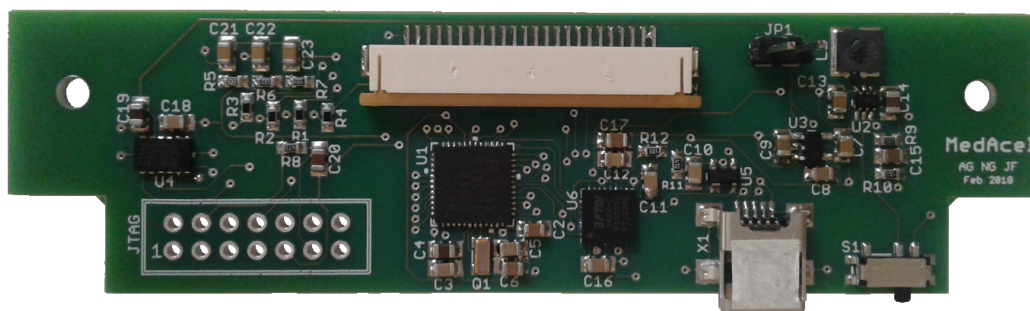


Figura 10.2: Foto del PCB asociado al dispositivo final realizado

consumos asociados. También se busca estimar la autonomía del dispositivo.

El modo Standby sirve para exportar los datos utilizándose típicamente durante algunos minutos, el modo Posicionamiento posiciona a la persona respecto al acelerómetro y está pensado que tarde algunos minutos como máximo, y el modo Test sirve para adquirir aceleraciones demorando menos de 3 minutos. Por lo tanto, el único modo en el cual se piensa usar el dispositivo continuamente es el modo Measure, es en este modo que es importante el consumo, y por lo tanto en el cual se midió.

Se calculará entonces la autonomía del dispositivo suponiendo que el mismo funciona siempre en modo Measure.

Para analizar la corriente consumida por el dispositivo y observar las distintas interrupciones y sus consumos asociados, se utilizó el producto ee-201 [25, 26] de *ee-quipment* consistente en un amplificador logarítmico de voltaje con una resistencia de 1Ω incorporada de tal forma que la corriente medida es simplemente $I = 10^{-V}$, siendo V el voltaje medido. El objetivo del amplificador logarítmico es permitir visualizar correctamente un rango dinámico de corriente de varios órdenes de magnitud (μA a mA).

La figura 10.3 muestra indirectamente la corriente total consumida por el dispositivo en el tiempo cuando el dispositivo está funcionando en HIGH ODR ($f_s = 12,5\text{Hz}$). Los picos de voltaje (y por lo tanto de corriente) que se observan corresponden a interrupciones asociadas a nuevos datos de aceleración. Se puede observar que su frecuencia es aproximadamente $12,5\text{Hz}$. Se observa que cuando el microcontrolador está en modo de bajo consumo, se tiene $V = 3,38\text{V}$ y por lo tanto $I = 10^{-3,38\text{V}} = 417\mu\text{A}$. Cuando llega la interrupción de aceleración, se tiene $V = 2,22\text{V}$ y por lo tanto $I = 10^{-2,22\text{V}} = 6,03\text{mA}$.

La figura 10.4 muestra en detalle la corriente consumida por el dispositivo cuando cae una de encendido del LED intermitente. Se observa que el LED está encendido aproximadamente durante 50ms (que corresponde a lo elegido).

Analizando cada una de las interrupciones se completó la tabla 10.1, que resume la corriente consumida por cada componente calculada a partir del voltaje

Capítulo 10. Dispositivo final

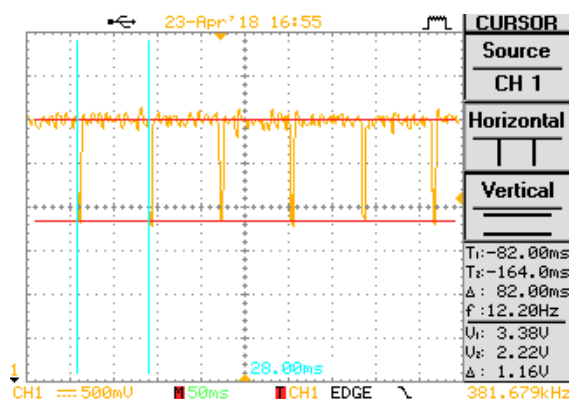


Figura 10.3: Captura del osciloscopio midiendo consumo, con el dispositivo funcionando con frecuencia de muestreo alta

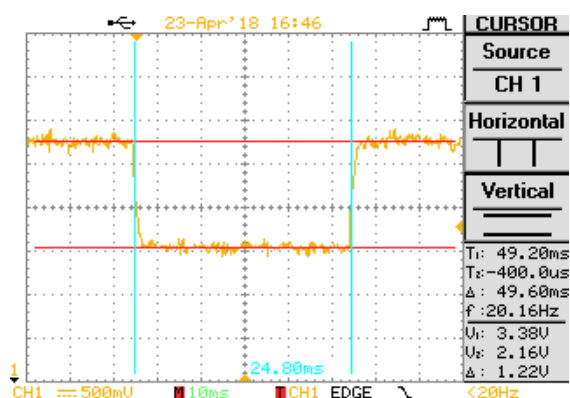


Figura 10.4: Interrupción del LED intermitente indicador de dispositivo encendido en modo Measure

medido, su duración y su frecuencia.

Fuente de consumo	Duración	Frecuencia	Voltaje	Consumo
Ninguna	-	-	3,38 V	417 μ A
Microcontrolador	2,0 ms	1,0 Hz	2,25 V	5,62 mA
Acelerómetro	4,28 ms	1,56 Hz	2,22 V	6,03 mA
LED2	50 ms	0,5 Hz	2,16 V	6,92 mA

Tabla 10.1: Consumo medido para las distintas interrupciones en modo Measure

A partir de dicha tabla, se procedió a estimar el consumo del dispositivo. Para esto, se utilizó la capacidad de una pila alcalina AAA, encontrando que su capacidad es de 1200 mA h [27] (de la hoja de datos de una pila AAA de *Energizer*, dato dado para descarga continua de 25 mA a 0,8 V y 21 °C). El dispositivo usará dos pilas AAA en serie, cuya capacidad equivalente es 1200 mA h. Luego, la autonomía

10.3. Consumo y autonomía del dispositivo

del dispositivo estará dada por

$$t = \frac{C}{\langle I \rangle}$$

A partir de los datos medidos, se calcula la corriente media que consume el dispositivo en modo Measure $\langle I \rangle$:

$$\langle I \rangle = \frac{1}{T} \int_0^T i(t) dt = \sum_{\text{modos}} I_{\text{modo}_i} \frac{t_{\text{modo}_i}}{T} = \sum_{\text{modos}} I_{\text{modo}_i} \delta_{\text{modo}_i}$$

Se distinguen dos casos: frecuencia de muestreo de aceleraciones baja $f_s = 1,56 \text{ Hz}$ y frecuencia de muestreo alta $f_s = 12,5 \text{ Hz}$ (mayor consumo).

1. En primer lugar, se estimó la autonomía del dispositivo suponiendo que $f_s = 1,56 \text{ Hz}$ en todo momento. La tabla 10.2 presenta el consumo medio total desglosado en las distintas interrupciones.

Fuente de consumo	Duty cycle	Consumo	Consumo medio
Ninguna	96,63 %	417 μA	403 μA (64,2 %)
Microcontrolador	0,2 %	5,62 mA	11,2 μA (1,8 %)
Acelerómetro	0,67 %	6,03 mA	40,2 μA (6,4 %)
LED2	2,5 %	6,92 mA	173 μA (27,6 %)
Total	-	-	627 μA (100 %)

Tabla 10.2: Cálculo del consumo medio total del dispositivo en modo Measure para LOW ODR

Finalmente, se tiene que la autonomía del dispositivo en LOW ODR es

$$t_{\text{LOW_ODR}} = \frac{C}{\langle I_{\text{LOW_ODR}} \rangle} = \frac{1200 \text{ mA h}}{627 \mu\text{A}} \approx 1913 \text{ h} \approx 80 \text{ d}$$

2. En segundo lugar, se estimó la autonomía del dispositivo suponiendo que $f_s = 12,5 \text{ Hz}$ en todo momento. La tabla 10.3 presenta el consumo medio total desglosado en las distintas interrupciones.

Fuente de consumo	Duty cycle	Consumo	Consumo medio
Ninguna	91,95 %	417 μA	383 μA (43,1 %)
Microcontrolador	0,2 %	5,62 mA	11,2 μA (1,3 %)
Acelerómetro	5,35 %	6,03 mA	322 μA (36,2 %)
LED2	2,5 %	6,92 mA	173 μA (19,4 %)
Total	-	-	890 μA (100 %)

Tabla 10.3: Cálculo del consumo medio total del dispositivo en modo Measure para HIGH ODR

Finalmente, se tiene que la autonomía del dispositivo en HIGH ODR es

$$t_{\text{HIGH_ODR}} = \frac{C}{\langle I_{\text{HIGH_ODR}} \rangle} = \frac{1200 \text{ mA h}}{889,90 \mu\text{A}} \approx 1348 \text{ h} \approx 56 \text{ d}$$

Capítulo 10. Dispositivo final

En realidad, la autonomía del dispositivo estará comprendida entre estos dos valores (56 d para $f_s = 12,5$ Hz y 80 d para $f_s = 1,56$ Hz). Dicho valor estará dado por la porción del tiempo δ que el dispositivo se encuentra en LOW ODR de la siguiente manera:

$$t = \frac{C}{\delta \langle I_{LOW_ODR} \rangle + (1 - \delta) \langle I_{HIGH_ODR} \rangle}$$

A partir del cálculo realizado, se obtuvo como mínimo una autonomía de 56 días, logrando cumplir con la meta de 14 días propuesta en los requerimientos.

Capítulo 11

Resultados

Este capítulo presenta resultados de verificación y de validación del dispositivo final. Se entiende la verificación como la prueba individual de cada requerimiento. En tanto, la validación intenta demostrar que el aspecto que se desea validar (puede ser el dispositivo completo o el diseño mecánico por ejemplo) cumple el propósito con el que fue diseñado.

11.1. Verificación de los requerimientos

Con el objetivo de determinar si el dispositivo diseñado cumple con cada uno de los requerimientos detallados descritos en el apéndice A, se realizaron tests de prueba. Cada test de prueba está asociado a la verificación de un requerimiento detallado particular.

Para algunos requerimientos seleccionados por el tutor, se redactaron protocolos de test. Los protocolos de test son instrucciones claras para la prueba del requerimiento, de manera de que se puedan repetir. Los requerimientos para los que no se redactaron protocolos de test fueron probados mediante pruebas informales o sólo funcionales.

El apéndice B resume todos los tests realizados.

11.1.1. Resultados

Se probaron la mayoría de los requerimientos exceptuando aquellos que requieren condiciones particulares (por ejemplo, temperatura de almacenamiento) y los que requieren tests muy extensos en el tiempo. Todos los requerimientos pasaron el testeado exceptuando el requerimiento de comodidad del usuario. Se observó que el dispositivo molesta al usuario cuando este se acuesta boca abajo.

11.2. Validación del diseño

11.2.1. Validación del algoritmo

11.2.1.1. Clasificación de movimiento

En el mismo proceso de diseño de la sección 5.4 se realizaron pruebas de validación. Se realizaron múltiples pruebas en una cinta caminadora y en todas se halló que hay un relación directa entre la magnitud deducida y la velocidad del usuario en el rango de velocidades de interés.

11.2.1.2. Clasificación de posición

Se dejó el dispositivo quieto, cambiando su orientación y registrando la nueva orientación y la hora a la que se produjo. En particular, se realizó lo detallado en la tabla 11.1.

Hora	Orientación de la persona	Duración
13:02	Parada	00:58
14:00	Acostada boca arriba	01:42
15:42	Acostada sobre su hombro derecho	02:10
17:52	Acostada boca arriba	00:34
18:26	Acostada sobre su hombro izquierdo	01:09
19:35	Acostada boca abajo	00:28
20:03	Parada	02:18
22:21	Acostada boca arriba	01:03
23:24	Parada	00:36

Tabla 11.1: Prueba de detección de posición realizada el 29/04/2018 con el dispositivo final

De la tabla 11.1, sumando el tiempo en cada posición, se tiene que los resultados deberían ser: persona parada 231 min = 13 860 s, persona acostada 426 min = 25 560 s, persona acostada boca arriba 199 min = 11 940 s, persona acostada boca abajo 28 min = 1680 s, persona acostada sobre su hombro derecho 130 min = 7800 s, persona acostada sobre su hombro izquierdo 69 min = 4140 s, 1 giro hacia la derecha (a las 15:42), 3 giros hacia la izquierda (a las 17:52, 18:26 y 19:35) y el tiempo total, es decir 657 min = 39 420 s en nivel de movimiento 0.

La figura 11.1 muestra una captura de pantalla del adquisidor de datos asociada a los datos de la prueba realizada.

La tabla 11.2 muestra los resultados esperados de la prueba dados por la tabla 11.1 y los resultados obtenidos dados por la figura 11.1. Se observa que todos los tiempos son consistentes con el párrafo anterior, habiendo clasificado tanto las posiciones como los giros, levantadas y acostadas de manera correcta.

11.2. Validación del diseño

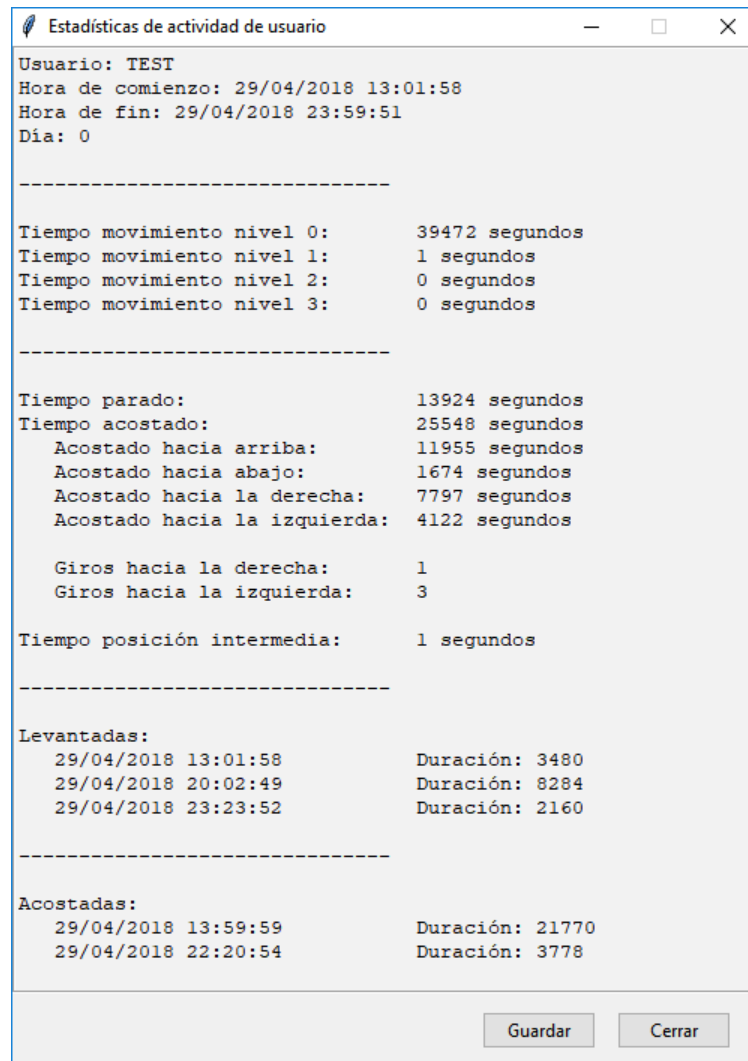


Figura 11.1: Datos obtenidos luego de la prueba de detección de posición

11.2.2. Validación del dispositivo

Para probar que el dispositivo cumple el propósito con el que se diseñó, se realizaron pruebas durante un día completo realizando las tareas cotidianas. Se hizo una medida en horas del día y una en horas de la noche para poder identificar más fácilmente los distintos eventos registrados.

La figura 11.2 muestra los datos obtenidos utilizando el dispositivo de día. Se utilizó el dispositivo durante más de 11 horas. Las estadísticas obtenidas coinciden con la actividad que se tuvo durante el día. Se observa que la mayoría del tiempo se registró como quieto (nivel 0) y parado. Se registró aproximadamente 2 horas en movimiento de las cuales la mayoría es en nivel 3, el cual se estima que representa el andar normal del usuario en cuestión.

Capítulo 11. Resultados

Posición	Resultados esperados	Resultados obtenidos
Persona parada	13 860 s	13 924 s
Persona acostada	25 560 s	25 548 s
Persona acostada boca arriba	11 940 s	11 955 s
Persona acostada boca abajo	1680	1674 s
Persona acostada sobre su hombro derecho	7800 s	7797 s
Persona acostada sobre su hombro izquierdo	4140 s	4122 s
Giros hacia la derecha	1	1
Giros hacia la izquierda	3	3
Tiempo en nivel de movimiento 0	39 420 s	39 472 s
Tiempo en nivel de movimiento 1	0 s	1 s
Tiempo en nivel de movimiento 2	0 s	0 s
Tiempo en nivel de movimiento 3	0 s	0 s

Tabla 11.2: Prueba de detección de posición realizada el 29/04/2018 con el dispositivo final: resultados esperados y obtenidos

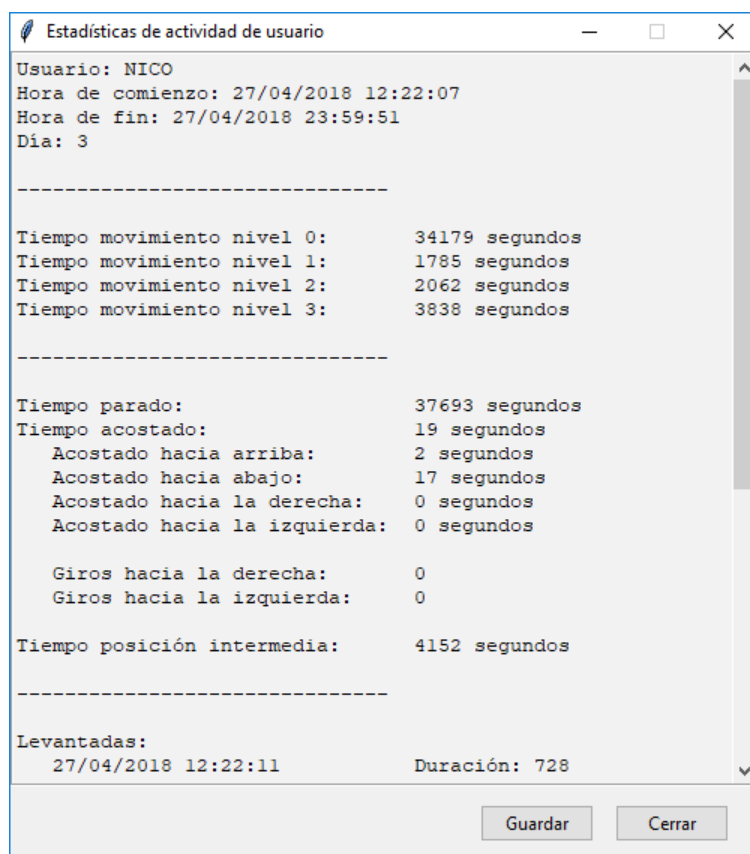


Figura 11.2: Datos obtenidos luego de utilizarlo de día

11.2. Validación del diseño

Si bien durante el día el usuario no se acostó, se registraron 19 segundos como acostado, de los cuales 17 son acostado boca abajo. Por lo corto de este tiempo, se considera que se debe a que el usuario se haya agachado por ejemplo. Debido a estos tiempos, se registraron falsas acostadas y levantadas que pueden ser identificadas por su corta duración.

En tanto, la figura 11.3 muestra los datos obtenidos en una medida en la noche. La medida comienza a las 00:00:00 debido a que a dicha hora el dispositivo comienza una nueva medida automáticamente. Como era de esperar, la mayoría del tiempo de duración de la medida fue registrado como persona quieta (nivel 0) y acostada. Se detectaron 11 giros hacia la derecha y 9 giros hacia la izquierda. Además la hora de levantado coincide con la hora a la que se levantó el usuario.

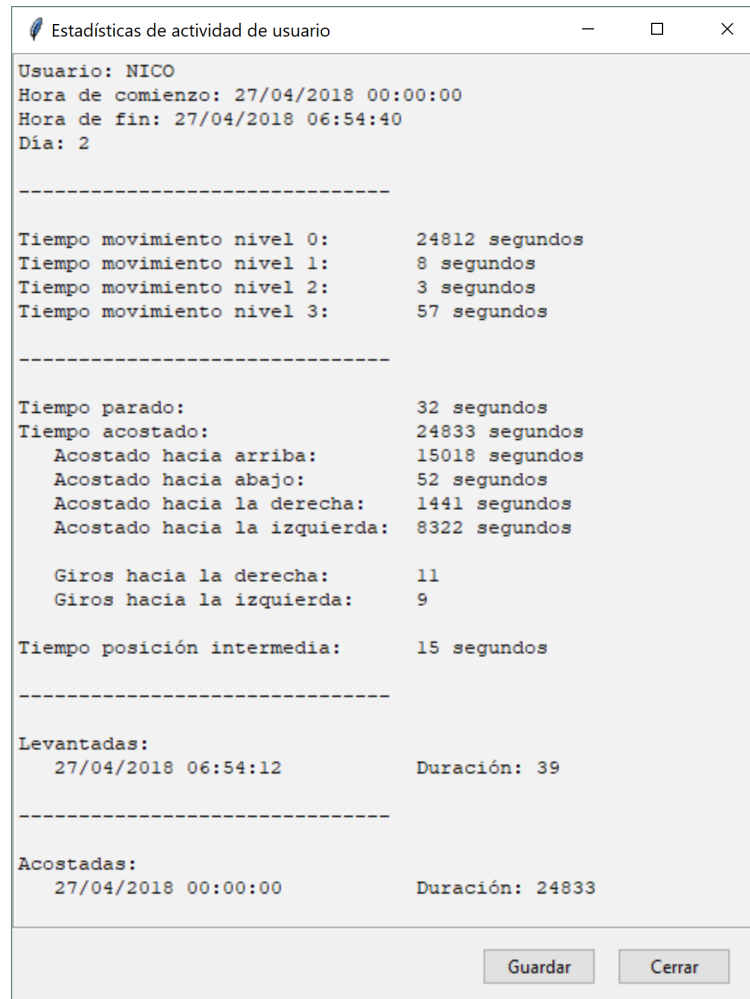


Figura 11.3: Datos obtenidos luego de utilizarlo de noche

Tanto de día como de noche, el dispositivo no alteró la vida cotidiana del usuario, por lo que estas pruebas también validan el diseño mecánico. Como ya se

Capítulo 11. Resultados

nombró, el dispositivo genera molestias al usuario si este se acuesta boca abajo. Sin embargo, tampoco impidió el sueño normal del usuario.

Capítulo 12

Conclusiones

Se obtuvo un dispositivo final que cumple con la mayoría de los requerimientos propuestos inicialmente. Principalmente, el dispositivo diseñado es capaz de clasificar la posición y el nivel de movimiento del usuario satisfactoriamente.

Estudiando la literatura, se encontraron métodos para la calibración del acelerómetro, la cual resultó indispensable para el correcto funcionamiento del sistema. Además, se ideó un método de posicionamiento robusto que permite calcular la aceleración del torso del usuario sin requerir una posición precisa del dispositivo.

Asimismo, mediante el procesamiento de las aceleraciones medidas en el andar cotidiano de una persona, se desarrolló un algoritmo para la detección de movimiento específico para la aplicación deseada. Dicho algoritmo fue validado mediante pruebas objetivas y luego implementado en *firmware* embebido obteniendo buenos resultados.

Se tuvo en especial consideración los aspectos mecánicos del diseño. Se logró un dispositivo relativamente pequeño y de bajo perfil que no molesta al usuario en sus actividades diurnas cotidianas. El sistema de sujeción propuesto es sencillo y con materiales de bajo costo, pero logra su propósito de mantener el dispositivo firme sobre el torso del usuario. El dispositivo fue utilizado durante varios días por el mismo usuario ocasionándole sólo leves molestias a la hora de dormir.

Por otra parte, se desea destacar el enfoque tenido en el diseño del dispositivo, donde se hizo hincapié en la redacción de un documento de Requerimientos detallados y un documento de Protocolos de test y Resultados. Paralelo a las primeras etapas de diseño, se realizó el documento de Requerimientos detallados que explicita de manera clara y concisa qué se espera del dispositivo final. Una vez finalizado el diseño, se redactó el documento de Protocolos de test y resultados, que resume las pruebas realizadas de cada uno de los requerimientos.

Por último, se resalta los conocimientos incorporados en el transcurso del proyecto. Se incursionó en el diseño de PCB y se mandó a fabricar dos PCB distintos los cuales resultaron ambos funcionales. Además, se aprendió Python desde cero y se logró desarrollar una interfaz gráfica para configurar el dispositivo y descargar los datos adquiridos. También se incorporó el manejo de nuevas herramientas informáticas importantes en la vida profesional como git para el control de versionado y GanttProject para organizar de manera eficiente los recursos humanos

Capítulo 12. Conclusiones

mediante diagramas de Gantt.

Apéndice A

Requerimientos detallados

A.1. Propósito

El propósito de este documento es la recopilación de los requerimientos detallados finales del proyecto de fin de carrera MedAcel.

A.2. Definiciones, abreviaturas y acrónimos

Siempre que se mencione alguna de las siguientes palabras o expresiones en el documento, se referirá a su definición correspondiente.

Sistema: Sistema completo de medida de posición y movimiento de una persona. Compuesto por una unidad central de recopilación de datos, un PC que corre el *software* adquisidor junto al cable de conexión y un elemento de fijación al torso del usuario.

MedAcel: Proyecto de fin de carrera “Dispositivo de medida de posición y movimiento de una persona”.

Dispositivo: Unidad central del sistema, se encarga de la recopilación de medidas de movimiento del usuario. Va sujeto al torso del individuo mediante el elemento de fijación.

SAD: *Software* Adquisidor de Datos. Unidad de *software* que corre en el ordenador conectado al dispositivo. Permite el despliegue de los datos recabados y la configuración de algunas funcionalidades del dispositivo.

Persona, individuo: Usuario portador del dispositivo.

GUI: *Graphical User Interface* (interfaz gráfica de usuario).

RTC: *Real-Time Clock* (reloj de tiempo real).

ODR: *Output Data Rate*, se utiliza en este documento como sinónimo de la frecuencia de muestreo del acelerómetro.

A.3. Convenciones

A.3.1. Requerimientos

Los requerimientos están identificados según el siguiente encabezado:

[Código] [Nombre] [[Fw],[Hw],[Sw]]

donde [Código] es el identificador numérico del requerimiento y [Nombre] es el título único del requerimiento. A su vez, cada requerimiento puede tener las siguientes etiquetas:

[Fw]: Indica si el requerimiento tiene que ser implementado a nivel de *Firmware*.

[Hw]: Indica si el requerimiento tiene que ser implementado a nivel de *Hardware*.

[Sw]: Indica si el requerimiento tiene que ser implementado a nivel de *Software*.

A.3.2. Comentarios

Los comentarios a los requerimientos (por ejemplo, justificaciones u observaciones) son identificados mediante letra *itálica*.

A.3.3. Posiciones

Se definen las siguientes posiciones del usuario según sus nombres clínicos:

- “Acostado boca arriba”: posición supina o decúbito dorsal.
- “Acostado boca abajo”: posición prona o decúbito ventral.
- “Acostado sobre hombro derecho”: posición decúbito lateral derecho.
- “Acostado sobre hombro izquierdo”: posición decúbito lateral izquierdo.

De igual manera, se definen los ejes de la figura A.1:

- Eje X: eje antero-posterior.
- Eje Y: eje izquierda-derecha.
- Eje Z: eje cráneo-caudal.

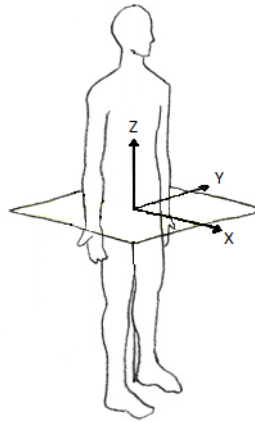


Figura A.1: Ejes relativos a una persona

A.3.4. Aceleración

Se definen los siguientes posibles datos de aceleración:

- \mathbf{a}_c aceleración calibrada pero no posicionada,
- \mathbf{a} aceleración calibrada y posicionada.

Cada uno de los datos serán expresados como múltiplos de la aceleración gravitatoria g . En caso de tratarse de la aceleración posicionada, se definirá como el vector de aceleración $\mathbf{a} = a_x \hat{\mathbf{x}} + a_y \hat{\mathbf{y}} + a_z \hat{\mathbf{z}}$, según los ejes de la figura A.1.

A.4. Dispositivo

A.4.1. Requerimientos mecánicos y físicos

A.4.1.1. Masa [Hw]

El dispositivo deberá tener una masa menor a 200 g.

A.4.1.2. Caja [Hw]

El dispositivo deberá contar con una caja de protección de forma prismática de base rectangular.

A.4.1.3. Dimensiones [Hw]

El dispositivo deberá entrar en un prisma de base rectangular de largo 12 cm, ancho 8 cm y altura 2 cm.

A.4.1.4. Rating de protección [Hw]

La caja del dispositivo deberá clasificarse como IPx1.

Comentario: la clasificación como IPx1 implica:

- *La protección contra el ingreso de sólidos no se especifica.*
- *El dispositivo deberá estar protegido contra el ingreso de gotas de agua verticales.*

A.4.1.5. Temperatura de funcionamiento [Hw]

El dispositivo deberá ser capaz de operar en un rango de temperatura entre -10°C y 40°C cumpliendo todos los requerimientos.

A.4.1.6. Temperatura de almacenamiento [Hw]

El dispositivo sin pilas no deberá romperse si se almacena a una temperatura entre -30°C y 70°C .

A.4.2. Interfaz de usuario

A.4.2.1. Llave de encendido/apagado [Hw]

El dispositivo deberá tener una llave de encendido/apagado.

A.4.2.2. Indicador de batería baja [Hw]

El dispositivo deberá tener un LED de color amarillo para indicar su nivel de batería.

A.4. Dispositivo

A.4.2.3. Indicadores de uso general [Hw]

El dispositivo deberá tener tres LEDs (LED1, LED2, LED3) de color verde para indicar en qué estado/modo se encuentra operando.

A.4.2.4. Botones de uso general [Hw]

El dispositivo deberá tener tres pulsadores (BUT1, BUT2, BUT3) para cambiar el modo de operación.

A.4.3. Requerimientos generales

A.4.3.1. Alimentación del dispositivo [Hw]

El dispositivo deberá alimentarse a partir de dos baterías alcalinas AA o AAA convencionales accesibles al usuario para su reemplazo.

A.4.3.2. Autonomía del dispositivo [Hw][Fw]

El dispositivo con baterías nuevas deberá tener una autonomía superior a una semana, con una meta de dos semanas.

A.4.3.3. Medida de alimentación [Hw][Fw]

El dispositivo deberá medir el voltaje de alimentación (ver A.4.3.1) cada $1\text{ s} \pm 0,01\text{ s}$ con un fondo de escala de $3,3\text{ V}$ y una precisión de $\pm 5\%$.

Nota: Dichas medidas serán almacenadas y procesadas mediante un entero positivo representando 0 V como 0 , y $3,3\text{ V}$ como 256 ($LSB \approx 13\text{ mV}$).

A.4.3.4. Indicado de nivel de batería [Hw][Fw]

El dispositivo deberá estimar el nivel de batería a partir de la medida de alimentación (ver A.4.3.3) e indicarlo mediante el indicador de batería baja (ver A.4.2.2) de acuerdo a la siguiente tabla:

Nivel de batería	Alimentación	Indicador de batería baja
Normal	Mayor a $2,0\text{ V}$	LED OFF
Bajo	Menor a $2,0\text{ V}$ Mayor a $1,6\text{ V}$	LED intermitente (1 s ON , 1 s OFF)
Crítico	Menor a $1,6\text{ V}$	LED ON

A.4.3.5. Reloj de tiempo real (RTC) [Fw]

El dispositivo deberá llevar cuenta de un reloj de tiempo real con registros de hora, minutos, segundos, día, mes y año. El desvío del reloj deberá ser menor a dos segundos por día.

Apéndice A. Requerimientos detallados

A.4.3.6. Medida de aceleración [Hw][Fw]

El dispositivo deberá ser capaz de medir la aceleración a la que se encuentra sujeto en tres ejes ortogonales, con un fondo de escala de ± 4 g en cada eje y una precisión de 4 mg.

Nota: Dichas medidas serán almacenadas y procesadas mediante un entero con signo representando ± 1 g como ± 256 ($LSB \simeq 4$ mg).

A.4.3.7. Calibración [Fw]

El dispositivo deberá contar con un método para su calibración, el cual deberá ser inaccesible para el usuario. Este método deberá asegurar que, con el dispositivo calibrado ubicado en un plano horizontal sobre una de sus caras, se mida una aceleración $a_c = 1 \text{ g} \pm 0,01 \text{ g}$ en dirección a dicha cara.

A.4.3.8. Múltiples usuarios [Fw]

El dispositivo deberá ser capaz de almacenar datos de tres usuarios. Cada uno de los usuarios deberá ser identificado por un número de 0 al 2 y por un nombre de usuario de cuatro caracteres máximo.

A.4.3.9. Modos de operación [Fw]

El dispositivo deberá contar con 4 modos de funcionamiento: modo Standby, modo Measure, modo Posicionamiento y modo Test.

A.4.4. Modo Standby

A.4.4.1. Inicio del dispositivo [Fw]

El dispositivo deberá iniciar en modo Standby al encenderse.

A.4.4.2. Indicado de dispositivo en modo Standby [Fw]

El dispositivo deberá indicar que se encuentra en modo Standby con todos los LEDs de uso general apagados.

A.4.4.3. Consumo [Hw][Fw]

El dispositivo en modo Standby deberá consumir una corriente de alimentación media menor a $500 \mu\text{A}$.

A.4.4.4. Nivel de batería crítico [Fw]

En modo Standby y con nivel de batería crítico (ver A.4.3.4), el dispositivo deberá permanecer en modo Standby.

A.4.5. Modo Test

A.4.5.1. Acceso al modo Test [Fw]

El dispositivo en el modo Standby deberá pasar al modo Test si se acciona de manera continua el pulsador BUT2 durante $3\text{ s} \pm 0,1\text{ s}$.

A.4.5.2. Indicado de dispositivo en modo Test [Fw]

El dispositivo deberá indicar que se encuentra en modo Test con todos los LEDs de uso general encendidos.

A.4.5.3. Adquisición de medidas [Fw]

Al ingresar en modo Test, el dispositivo deberá adquirir y almacenar en memoria no volátil 2048 medidas de aceleración a frecuencia $12,5\text{ Hz} \pm 5\%$. Una vez finalizada la adquisición de medidas, el dispositivo deberá retornar a modo Standby automáticamente.

A.4.5.4. Cancelación de adquisición [Fw]

El dispositivo en el modo Test deberá retornar al modo Standby si se acciona de manera continua el pulsador BUT3 durante $3\text{ s} \pm 0,1\text{ s}$.

A.4.6. Modo Posicionamiento

A.4.6.1. Algoritmo de posicionamiento [Fw]

En modo Posicionamiento, el dispositivo deberá ejecutar un algoritmo para referenciar las medidas de aceleración respecto a los ejes del usuario (ver figura A.1). Dicho algoritmo deberá constar de 3 etapas sucesivas:

1. Posicionado de eje X con el usuario acostado boca arriba,
2. Posicionado de eje Y con el usuario acostado sobre su hombro derecho,
3. Posicionado de eje Z con el usuario parado.

A.4.6.2. Medidas de aceleración en posicionamiento [Fw]

En cada una de las etapas de posicionado, el dispositivo deberá realizar 25 medidas de aceleración en cada eje a una frecuencia de $25\text{ Hz} \pm 5\%$.

Apéndice A. Requerimientos detallados

A.4.6.3. Posicionado de eje X [Fw]

Se considerará como eje X posicionado al promedio de las medidas de aceleración:

$$\begin{bmatrix} a_x^X \\ a_y^X \\ a_z^X \end{bmatrix} = \begin{bmatrix} \frac{1}{25} \sum_{i=1}^{25} a_{x_i} \\ \frac{1}{25} \sum_{i=1}^{25} a_{y_i} \\ \frac{1}{25} \sum_{i=1}^{25} a_{z_i} \end{bmatrix}$$

El posicionado del eje X deberá considerarse válido si se cumplen todas las siguientes condiciones:

- La diferencia absoluta entre medidas sucesivas de aceleración en cada eje es menor a 60 mg.
- Todas las medidas de aceleración satisfacen $\left| \sqrt{a_x^2 + a_y^2 + a_z^2} - 1 \right| < 0,1$ mg.

En caso de no cumplir, se deberá repetir la etapa de posicionado.

A.4.6.4. Posicionado de eje Y [Fw]

Se considerará como eje Y posicionado al promedio de las medidas de aceleración:

$$\begin{bmatrix} a_x^Y \\ a_y^Y \\ a_z^Y \end{bmatrix} = \begin{bmatrix} \frac{1}{25} \sum_{i=1}^{25} a_{x_i} \\ \frac{1}{25} \sum_{i=1}^{25} a_{y_i} \\ \frac{1}{25} \sum_{i=1}^{25} a_{z_i} \end{bmatrix}$$

El posicionado del eje Y deberá considerarse válido si se cumplen todas las siguientes condiciones:

- La diferencia absoluta entre medidas sucesivas de aceleración en cada eje es menor a 60 mg.
- Todas las medidas de aceleración satisfacen $\left| \sqrt{a_x^2 + a_y^2 + a_z^2} - 1 \right| < 0,1$ mg.
- El ángulo entre el eje X posicionado y el eje Y posicionado deberá ser mayor a 65° .

En caso de no cumplir, se deberá repetir la etapa de posicionado.

A.4.6.5. Posicionado de eje Z [Fw]

Se considerará como eje Z posicionado al promedio de las medidas de aceleración:

$$\begin{bmatrix} a_x^Z \\ a_y^Z \\ a_z^Z \end{bmatrix} = \begin{bmatrix} \frac{1}{25} \sum_{i=1}^{25} a_{x_i} \\ \frac{1}{25} \sum_{i=1}^{25} a_{y_i} \\ \frac{1}{25} \sum_{i=1}^{25} a_{z_i} \end{bmatrix}$$

El posicionado del eje Z deberá considerarse válido si se cumplen todas las siguientes condiciones:

- La diferencia absoluta entre medidas sucesivas de aceleración en cada eje es menor a 60 mg.
- Todas las medidas de aceleración satisfacen $\left| \sqrt{a_x^2 + a_y^2 + a_z^2} - 1 \right| < 0,1 \text{ mg}$.
- El ángulo entre el eje X posicionado y el eje Z posicionado deberá ser mayor a 65° .
- El ángulo entre el eje Y posicionado y el eje Z posicionado deberá ser mayor a 65° .

En caso de no cumplir, se deberá repetir la etapa de posicionado.

A.4.6.6. Acceso al modo Posicionamiento [Fw]

El dispositivo en el modo Standby deberá pasar al modo Posicionamiento si se acciona de manera continua el pulsador BUT1 durante $3 \text{ s} \pm 0,1 \text{ s}$.

A.4.6.7. Indicado de etapa de posicionamiento [Fw]

El dispositivo en modo Posicionamiento deberá señalar la etapa del algoritmo de posicionamiento en el que se encuentra a través de los indicadores de uso general de la siguiente manera:

- Posicionado de eje X: LED1 encendido, los restantes apagados.
- Posicionado de eje Y: LED2 encendido, los restantes apagados.
- Posicionado de eje Z: LED3 encendido, los restantes apagados.

A.4.6.8. Disparo de posicionado [Fw]

El dispositivo en modo Posicionamiento deberá disparar el posicionado de la etapa en la que se encuentra al accionar el pulsador BUT1.

Apéndice A. Requerimientos detallados

A.4.6.9. Referencia de aceleraciones respecto a usuario [Fw]

Una vez realizado el algoritmo de posicionamiento, todas las medidas de aceleración posteriores deberán ser referenciadas respecto a los ejes calculados de la siguiente manera:

$$\mathbf{a} = \begin{bmatrix} a_x^X & a_x^Y & a_x^Z \\ a_y^X & a_y^Y & a_y^Z \\ a_z^X & a_z^Y & a_z^Z \end{bmatrix}^{-1} \mathbf{a}_C$$

donde \mathbf{a} es el vector de aceleraciones referenciadas respecto al usuario y \mathbf{a}_C es el vector de aceleraciones medidas.

A.4.6.10. Posicionamiento exitoso [Fw]

Una vez realizado el algoritmo de posicionamiento, el dispositivo deberá encender todos los indicadores de uso general durante 1 s y retornar al modo Standby.

A.4.6.11. Cancelación de posicionamiento [Fw]

Estando en cualquier etapa del posicionamiento, el dispositivo deberá anular lo realizado al presionar el pulsador BUT2. En este caso, el dispositivo deberá encender 2 veces todos los indicadores de uso general (0,5 s on, 0,5 s off) y retornar al modo Standby.

A.4.7. Modo Measure

A.4.7.1. Generales

A.4.7.1.1. Acceso al modo Measure [Fw]

El dispositivo en el modo Standby deberá pasar al modo Measure e iniciar una medida si se acciona de manera continua el pulsador BUT3 durante $3\text{ s} \pm 0,1\text{ s}$.

A.4.7.1.2. Indicado de dispositivo en modo Measure [Fw]

El dispositivo deberá indicar que se encuentra en modo Measure encendiendo de manera cíclica el LED2 durante 50 ms cada 2 s.

A.4.7.1.3. Medidas [Fw]

El dispositivo deberá ser capaz de guardar hasta un máximo de 7 medidas en memoria no volátil. Cada medida deberá tener su conjunto de datos estadísticos independiente y deberá identificarse mediante un número de 0 a 6.

En caso de iniciar una nueva medida sin espacio de almacenamiento, el dispositivo deberá sobrescribir las estadísticas de la medida de mayor antigüedad.

A.4.7.1.4. Inicio de medida [Fw]

Al iniciar una medida, el dispositivo deberá resetear todos los contadores y registros estadísticos (ver A.4.7.6) y almacenar la fecha y hora actual del RTC como hora de comienzo.

A.4.7.1.5. Medidas diarias [Fw]

En caso de estar efectuando una medida y se tenga que el RTC marca un nuevo día (*hora 00:00:00*), el dispositivo deberá guardar los datos estadísticos (ver A.4.7.6) de la medida en proceso e iniciar una nueva medida.

A.4.7.1.6. Nivel de batería crítico durante medida [Fw]

Al detectar nivel de batería crítico en modo Measure (ver A.4.3.4), el dispositivo deberá almacenar los datos estadísticos de la medida en progreso y retornar a modo Standby.

A.4.7.1.7. Finalización de medida [Fw]

Si en modo Measure se acciona de manera continua el pulsador BUT3 durante $3\text{ s} \pm 0,1\text{ s}$, el dispositivo deberá almacenar la medida en progreso y la hora de finalización y retornar a modo Standby.

A.4.7.1.8. Protección contra bloqueo de microcontrolador [Fw]

El dispositivo deberá tener un timer de $1\text{ s} \pm 0,01\text{ s}$ que, en caso de no resetear, ocasione el reinicio automático.

Nota: Este timer se conoce también como watchdog timer. Previene que en caso de falla lógica, el dispositivo permanezca en modo Measure con un consumo alto.

A.4.7.2. Procesamiento de aceleraciones

A.4.7.2.1. Frecuencia de muestreo de aceleraciones [Fw]

En modo Measure, el dispositivo deberá ser capaz de operar en dos modalidades:

- Modo **LOW ODR**, en el cual el dispositivo deberá realizar medidas de aceleración en cada eje a una frecuencia de $1,56\text{ Hz} \pm 5\%$.
- Modo **HIGH ODR**, en el cual el dispositivo deberá realizar medidas de aceleración en cada eje a una frecuencia de $12,5\text{ Hz} \pm 5\%$ y almacenar las últimas 4 medidas realizadas.

Apéndice A. Requerimientos detallados

A.4.7.2.2. Cálculo de aceleración DC [Fw]

En modo HIGH ODR, el dispositivo deberá ser capaz de calcular con un error menor a 4 mg (*equivalente a 1 LSB*) y almacenar los últimos 4 resultados de la siguiente expresión:

$$\mathbf{a}_n^{\text{DC}} = 0,2196 \times 10^{-3} \mathbf{a}_n + 0,6588 \times 10^{-3} \mathbf{a}_{n-1} + 0,6588 \times 10^{-3} \mathbf{a}_{n-2} + 0,2196 \times 10^{-3} \mathbf{a}_{n-3} + 2,7488 \times \mathbf{a}_{n-1}^{\text{DC}} - 2,5282 \times \mathbf{a}_{n-2}^{\text{DC}} + 0,7776 \times \mathbf{a}_{n-3}^{\text{DC}}$$

Justificación: El cálculo anterior corresponde a un filtro digital pasa-bajos de tipo Butterworth, orden 3 y frecuencia de corte 0,25 Hz para una frecuencia de muestreo de 12,5 Hz. Al aplicarlo, se obtiene la componente continua de la aceleración medida.

A.4.7.2.3. Cálculo de aceleración AC [Fw]

En modo HIGH ODR, el dispositivo deberá ser capaz de realizar el siguiente cálculo con un error menor a 4 mg (*equivalente a 1 LSB*):

$$\mathbf{a}^{\text{AC}} = \mathbf{a} - \mathbf{a}^{\text{DC}}$$

Justificación: Se obtiene las componentes puramente de señal restándole a la aceleración medida su componente de continua.

A.4.7.2.4. Cálculo de módulo AC [Fw]

En modo HIGH ODR, el dispositivo deberá ser capaz de calcular con un error menor al 1 % y almacenar los últimos 12 resultados de la siguiente expresión:

$$\|\mathbf{a}^{\text{AC}}\|_2^2 = a_x^{\text{AC}2} + a_y^{\text{AC}2} + a_z^{\text{AC}2}$$

A.4.7.2.5. Procesamiento de medidas de aceleración [Fw]

En modo HIGH ODR, el dispositivo deberá realizar el siguiente procesamiento cada vez que realice una medida de aceleración:

1. Cálculo de aceleración DC (ver A.4.7.2.2).
2. Cálculo de aceleración AC (ver A.4.7.2.3).
3. Cálculo de módulo AC (ver A.4.7.2.4).

A.4.7.2.6. Promedio AC en un segundo [Fw]

En modo HIGH ODR, el dispositivo deberá ser capaz de promediar los últimos 12 módulos AC calculados (ver A.4.7.2.4) con un error menor al 1 % y almacenar los últimos 12 resultados alcanzados.

A.4.7.2.7. Promedio AC en 12 segundos [Fw]

En modo HIGH ODR, el dispositivo deberá ser capaz de promediar los últimos 12 promedios AC en un segundo calculados (ver A.4.7.2.6) con un error menor al 1 %.

A.4.7.2.8. Período de clasificación [Fw]

El dispositivo deberá clasificar el nivel de actividad del usuario y su posición cada $1\text{ s} \pm 0,001\text{ s}$.

Nota: Este tiempo también será denominado tick.

A.4.7.2.9. Medida de actividad [Fw]

En modo HIGH ODR, el dispositivo deberá realizar la siguiente serie de cálculos cada período de clasificación:

1. Promedio AC en un segundo (ver A.4.7.2.6).
2. Promedio AC en 12 segundos (ver A.4.7.2.7).

El resultado final alcanzado se denominará medida de actividad y se representará con el símbolo M_A . Será utilizado para la clasificación en niveles de actividad.

A.4.7.2.10. Umbrales de actividad [Fw]

El dispositivo deberá contar con 3 umbrales (ε_{01} , ε_{12} y ε_{23}) para la clasificación en niveles de actividad del usuario según la medida de actividad obtenida (ver A.4.7.2.9). El dispositivo deberá ser capaz de operar con cualquiera de los siguientes conjuntos de umbrales:

- Umbrales 0 (opción por defecto):
 - $\varepsilon_{01} = 440$
 - $\varepsilon_{12} = 1000$
 - $\varepsilon_{23} = 1600$
- Umbrales 1:
 - $\varepsilon_{01} = 330$
 - $\varepsilon_{12} = 750$
 - $\varepsilon_{23} = 1200$
- Umbrales 2:
 - $\varepsilon_{01} = 550$
 - $\varepsilon_{12} = 1200$

Apéndice A. Requerimientos detallados

- $\varepsilon_{23} = 2000$

Nota: La unidad de dichos umbrales es la misma que la de la magnitud M_A , es decir que $256^2 = 65536$ corresponde a $1 g^2$. Por ejemplo: un valor $\varepsilon_{01} = 440$ implica que el umbral entre nivel de movimiento 0 y 1 se obtiene cuando el promedio de 12 s de $\|\mathbf{a}^{AC}\|_2$ es igual a $\frac{\sqrt{440}}{256} g = 82 \text{ mg}$.

A.4.7.2.11. Pasaje de modo LOW ODR a modo HIGH ODR [Fw]

En modo LOW ODR, el dispositivo deberá pasar a modo HIGH ODR cuando mida una muestra de aceleración que cumpla una de las siguientes condiciones:

- $\|\mathbf{a}\|_2 > 1,1 g$
- $\|\mathbf{a}\|_2 < 0,9 g$

A.4.7.2.12. Pasaje de modo HIGH ODR a modo LOW ODR [Fw]

En modo HIGH ODR, el dispositivo deberá pasar a modo LOW ODR cuando durante $3 s \pm 0,3 s$ consecutivos se cumplen las siguientes condiciones:

- Más de 6 módulos AC calculados en menos de 1 segundo menor a 164.
- $M_A < \varepsilon_{01}$ (ver A.4.7.2.9).

A.4.7.3. Detección de posición

A.4.7.3.1. Posiciones posibles [Fw]

Cada período de clasificación (ver A.4.7.2.8), el dispositivo deberá clasificar la posición del usuario en uno de los siguientes estados:

- Persona parada,
- Persona acostada boca arriba,
- Persona acostada boca abajo,
- Persona acostada sobre hombro derecho,
- Persona acostada sobre hombro izquierdo,
- Persona en posición intermedia.

A.4.7.3.2. Posición persona parada [Fw]

El dispositivo deberá clasificar la posición del usuario como persona parada siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR: $a_z > 0,9 g$.
- En modo HIGH ODR: $a_z^{DC} > 0,9 g$.

A.4.7.3.3. Posición persona acostada boca arriba [Fw]

El dispositivo deberá clasificar la posición del usuario como persona acostada boca arriba siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR:

$$\left\{ \begin{array}{l} -0,6 \text{ g} < a_z < 0,6 \text{ g} \\ a_y \leq a_x \\ -a_y < a_x \end{array} \right.$$

- En modo HIGH ODR:

$$\left\{ \begin{array}{l} -0,6 \text{ g} < a_z^{DC} < 0,6 \text{ g} \\ a_y^{DC} \leq a_x^{DC} \\ -a_y^{DC} < a_x^{DC} \end{array} \right.$$

A.4.7.3.4. Posición persona acostada boca abajo [Fw]

El dispositivo deberá clasificar la posición del usuario como persona acostada boca abajo siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR:

$$\left\{ \begin{array}{l} -0,6 \text{ g} < a_z < 0,6 \text{ g} \\ a_x \leq a_y \\ a_x < -a_y \end{array} \right.$$

- En modo HIGH ODR:

$$\left\{ \begin{array}{l} -0,6 \text{ g} < a_z^{DC} < 0,6 \text{ g} \\ a_x^{DC} \leq a_y^{DC} \\ a_x^{DC} < -a_y^{DC} \end{array} \right.$$

A.4.7.3.5. Posición persona acostada sobre hombro derecho [Fw]

El dispositivo deberá clasificar la posición del usuario como persona acostada sobre hombro derecho siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR:

$$\left\{ \begin{array}{l} -0,6 \text{ g} < a_z < 0,6 \text{ g} \\ a_y < a_x \\ a_x \leq -a_y \end{array} \right.$$

- En modo HIGH ODR:

$$\left\{ \begin{array}{l} -0,6 \text{ g} < a_z^{DC} < 0,6 \text{ g} \\ a_y^{DC} < a_x^{DC} \\ a_x^{DC} \leq -a_y^{DC} \end{array} \right.$$

Apéndice A. Requerimientos detallados

A.4.7.3.6. Posición persona acostada sobre hombro izquierdo [Fw]

El dispositivo deberá clasificar la posición del usuario como persona acostada sobre hombro izquierdo siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR:

$$\begin{cases} -0,6 \text{ g} < a_z < 0,6 \text{ g} \\ a_x < a_y \\ -a_y \leq a_x \end{cases}$$

- En modo HIGH ODR:

$$\begin{cases} -0,6 \text{ g} < a_z^{DC} < 0,6 \text{ g} \\ a_x^{DC} < a_y^{DC} \\ -a_y^{DC} \leq a_x^{DC} \end{cases}$$

A.4.7.3.7. Posición intermedia [Fw]

El dispositivo deberá clasificar la posición del usuario como posición intermedia siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR: $0,6 \text{ g} \leq a_z \leq 0,9 \text{ g}$
- En modo HIGH ODR: $0,6 \text{ g} \leq a_z^{DC} \leq 0,9 \text{ g}$

A.4.7.4. Detección de giros, levantadas y acostadas

A.4.7.4.1. Giro de 90° hacia la derecha [Fw]

El dispositivo deberá detectar un giro de 90° hacia la derecha cada período de clasificación que se produzca una de las siguientes transiciones directas:

- persona acostada boca arriba → persona acostada sobre hombro derecho,
- persona acostada sobre hombro derecho → persona acostada boca abajo,
- persona acostada boca abajo → persona acostada sobre hombro izquierdo,
- persona acostada sobre hombro izquierdo → persona acostada boca arriba.

A.4.7.4.2. Giro de 90° hacia la izquierda [Fw]

El dispositivo deberá detectar un giro de 90° hacia la izquierda cada período de clasificación que se produzca una de las siguientes transiciones directas de estado:

- persona acostada boca arriba → persona acostada sobre hombro izquierdo,
- persona acostada sobre hombro izquierdo → persona acostada boca abajo,
- persona acostada boca abajo → persona acostada sobre hombro derecho,
- persona acostada sobre hombro derecho → persona acostada boca arriba.

A.4.7.4.3. Acostada [Fw]

El dispositivo deberá detectar una “acostada” cada período de clasificación que se produzca una de las siguientes transiciones de estado (admitiendo el pasaje por “posición intermedia”):

- persona parada → persona acostada boca arriba,
- persona parada → persona acostada boca abajo,
- persona parada → persona acostada sobre hombro derecho,
- persona parada → persona acostada sobre hombro izquierdo.

A.4.7.4.4. Levantada [Fw]

El dispositivo deberá detectar una “levantada” cada período de clasificación que se produzca una de las siguientes transiciones de estado (admitiendo el pasaje por “posición intermedia”):

- persona acostada boca arriba → persona parada,
- persona acostada boca abajo → persona parada,
- persona acostada sobre hombro derecho → persona parada,
- persona acostada sobre hombro izquierdo → persona parada.

A.4.7.5. Clasificación en niveles de actividad

A.4.7.5.1. Niveles de actividad [Fw]

Cada período de clasificación (ver A.4.7.2.8), el dispositivo deberá clasificar la medida de actividad obtenida en uno de los siguientes niveles de actividad:

- Nivel de actividad 0,
- Nivel de actividad 1,
- Nivel de actividad 2,
- Nivel de actividad 3.

A.4.7.5.2. Nivel de actividad 0 [Fw]

El dispositivo deberá clasificar la actividad del usuario como “nivel 0” siempre que se cumpla una de las siguientes condiciones:

- En modo LOW ODR.
- En modo HIGH ODR y $M_A < \varepsilon_{01}$.

Apéndice A. Requerimientos detallados

A.4.7.5.3. Nivel de actividad 1 [Fw]

El dispositivo deberá clasificar la actividad del usuario como “nivel 1” siempre que, en modo HIGH ODR, se tenga $\varepsilon_{01} \leq M_A < \varepsilon_{12}$.

A.4.7.5.4. Nivel de actividad 2 [Fw]

El dispositivo deberá clasificar la actividad del usuario como “nivel 2” siempre que, en modo HIGH ODR, se tenga $\varepsilon_{12} \leq M_A < \varepsilon_{23}$.

A.4.7.5.5. Nivel de actividad 3 [Fw]

El dispositivo deberá clasificar la actividad del usuario como “nivel 3” siempre que, en modo HIGH ODR, se tenga $M_A \geq \varepsilon_{23}$.

A.4.7.6. Recopilación de estadísticas

A.4.7.6.1. Contador diario de persona parada [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como persona parada (ver A.4.7.3.2). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.2. Contador diario de persona acostada [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como cualquiera de las siguientes posiciones:

- Persona acostada boca arriba (ver A.4.7.3.3),
- Persona acostada boca abajo (ver A.4.7.3.4),
- Persona acostada sobre hombro derecho (ver A.4.7.3.5),
- Persona acostada sobre hombro izquierdo (ver A.4.7.3.6).

El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.3. Contador diario de persona acostada boca arriba [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como persona acostada boca arriba (ver A.4.7.3.3). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.4. Contador diario de persona acostada boca abajo [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como persona acostada boca abajo (ver A.4.7.3.4). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.5. Contador diario de persona acostada sobre hombro derecho [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como persona acostada sobre hombro derecho (ver A.4.7.3.5). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.6. Contador diario de persona acostada sobre hombro izquierdo [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como persona acostada sobre hombro izquierdo (ver A.4.7.3.6). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.7. Contador diario de persona en posición intermedia [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la posición del usuario sea clasificada como posición intermedia (ver A.4.7.3.7). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.8. Contador diario de giros de 90° hacia la derecha [Fw]

El dispositivo deberá llevar cuenta de los giros de 90° hacia la derecha detectados (ver A.4.7.4.1). El contador deberá tener un máximo de al menos 255 giros.

A.4.7.6.9. Contador diario de giros de 90° hacia la izquierda [Fw]

El dispositivo deberá llevar cuenta de los giros de 90° hacia la izquierda detectados (ver A.4.7.4.2). El contador deberá tener un máximo de al menos 255 giros.

A.4.7.6.10. Contador diario de nivel de actividad 0 [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la actividad del usuario sea clasificada como nivel de actividad 0 (ver A.4.7.5.2). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.11. Contador diario de nivel de actividad 1 [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la actividad del usuario sea clasificada como nivel de actividad 1 (ver A.4.7.5.3). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

Apéndice A. Requerimientos detallados

A.4.7.6.12. Contador diario de nivel de actividad 2 [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la actividad del usuario sea clasificada como nivel de actividad 2 (ver A.4.7.5.4). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.13. Contador diario de nivel de actividad 3 [Fw]

El dispositivo deberá llevar cuenta de los períodos de clasificación en los que la actividad del usuario sea clasificada como nivel de actividad 3 (ver A.4.7.5.5). El contador deberá tener un máximo de al menos 86400 (*equivalente a 1 día*).

A.4.7.6.14. Registro de acostadas [Fw]

El dispositivo deberá almacenar hasta 20 registros con la siguiente información de cada “acostada” registrada (ver A.4.7.4.3):

Hora: Número de *ticks* desde que se inició la medida hasta que se detectó la “acostada”. El máximo valor de este contador deberá ser por lo menos 86400 (*24 horas*).

Duración: Número de *ticks* desde que se detectó la “acostada” hasta que se detecte una “levantada” posterior. El máximo valor de este contador deberá ser por lo menos 86400 (*24 horas*).

En caso de que detectar más de 20 “acostadas”, el dispositivo deberá almacenar registros de las de mayor duración.

A.4.7.6.15. Registro de levantadas [Fw]

El dispositivo deberá almacenar hasta 20 registros con la siguiente información de cada “levantada” registrada (ver A.4.7.4.4):

Hora: Número de *ticks* desde que se inició la medida hasta que se detectó la “levantada”. El máximo valor de este contador deberá ser por lo menos 86400 (*24 horas*).

Duración: Número de *ticks* desde que se detectó la “levantada” hasta que se detecte una “acostada” posterior. El máximo valor de este contador deberá ser por lo menos 86400 (*24 horas*).

En caso de que detectar más de 20 “levantadas”, el dispositivo deberá almacenar registros de las de mayor duración.

A.4.8. Comunicación con SAD

A.4.8.1. Estado del dispositivo [Fw]

La comunicación con el SAD sólo deberá ser posible con el dispositivo encendido y en modo Standby.

A.4.8.2. Interfaz de comunicación [Hw][Fw]

El dispositivo deberá tener un puerto mini-USB para la conexión con el SAD. El envío/recepción de mensajes (ver A.4.8.3) deberá implementarse mediante UART vía USB.

A.4.8.3. Protocolo de comunicación [Fw]

La comunicación entre el dispositivo y el SAD deberá implementarse utilizando el byte como unidad mínima de envío/recepción. La comunicación deberá realizarse a 57600 baudios y utilizar 1 bit de parada entre bytes. Los mensajes intercambiados deberán codificarse utilizando ASCII. Un mensaje podrá contar con un máximo de 32 caracteres y deberá finalizar con el caracter “\r”.

A.4.8.4. Envío periódico de aceleraciones [Fw]

El dispositivo deberá ser capaz de enviar periódicamente la aceleración medida en cada uno de los tres ejes. Por defecto, la frecuencia entre medidas deberá ser $6,25\text{ Hz} \pm 5\%$. El formato de los envíos deberá ser $\pm\text{xxxx}$, $\pm\text{yyyy}$, $\pm\text{zzzz};\text{\r}$, donde:

- $\pm\text{xxxx}$ es la aceleración en el eje X,
- $\pm\text{yyyy}$ es la aceleración en el eje Y,
- $\pm\text{zzzz}$ es la aceleración en el eje Z.

A.4.8.5. Comandos [Fw]

La comunicación desde el SAD al dispositivo deberá basarse en comandos (mensajes predeterminados) asociados a una acción o respuesta por parte del dispositivo. Cada una de las acciones de la tabla A.1 deberá tener el comando asociado correspondiente.

A.4.8.6. Comando WH [Fw]

El dispositivo deberá setear el RTC a la hora ingresada al recibir el comando WH hh mm ss DD MM YYYY\r, donde:

- hh es la hora a setear (entre 00 y 23),
- mm son los minutos a setear (entre 00 y 59),
- ss son los segundos a setear (entre 00 y 59),
- DD es el día a setear (entre 01 y 31),
- MM es el mes a setear (entre 01 y 12),
- YYYY es el año a setear (mayor a 0000).

Apéndice A. Requerimientos detallados

Acciones	Comando asociado
Seteo de RTC	WH
Lectura de RTC	RH
Lectura de voltaje de alimentación	RB
Disparo de lecturas de aceleraciones	GO
Detención de lecturas de aceleraciones	ST
Seteo de frecuencia de lecturas de aceleraciones	SO
Lecturas de aceleraciones en modo Test	TM
Ingreso de nuevo usuario	AD
Borrado de usuario	DL
Seteo de usuario actual	SU
Listado de usuario ingresados	SH
Descarga de datos estadísticos	DD
Descarga de “levantadas”	SD
Descarga de “acostadas”	LY
Seteo de umbrales de actividad	TH
Consulta de posicionamiento realizado	TP

Tabla A.1: Comandos del SAD

El dispositivo deberá chequear que cada uno de los parámetros sea válido. En caso afirmativo, deberá setear el RTC y responder `OK\r`. En caso contrario, deberá responder `ERROR\r`.

A.4.8.7. Comando RH [Fw]

El dispositivo deberá responder con la hora actual del RTC al recibir el comando `RH\r`. El formato de la respuesta deberá ser `DD/MM/YYYY hh:mm:ss\r`, donde:

- `hh` es la hora actual,
- `mm` son los minutos actuales,
- `ss` son los segundos actuales,
- `DD` es el día actual,
- `MM` es el mes actual,
- `YYYY` es el año actual.

A.4.8.8. Comando RB [Fw]

Al recibir el comando `RB\r`, el dispositivo deberá enviar la última medida de alimentación realizada (ver A.4.3.3).

A.4.8.9. Comando GO [Fw]

Al recibir el comando GO\r, el dispositivo deberá disparar el envío periódico de aceleraciones (ver A.4.8.4).

A.4.8.10. Comando ST [Fw]

Al recibir el comando ST\r, el dispositivo deberá detener el envío periódico de aceleraciones (ver A.4.8.4).

A.4.8.11. Comando SO [Fw]

Al recibir el comando SO [ODR]\r, el dispositivo deberá modificar la frecuencia de envío periódico de aceleraciones (ver A.4.8.4). La asignación deberá seguir la lógica que se muestra en la tabla A.2.

[ODR]	Frecuencia
0	0,10 Hz
1	0,20 Hz
2	0,39 Hz
3	0,78 Hz
4	1,56 Hz
5	3,13 Hz
6	6,25 Hz
7	12,5 Hz
8	25 Hz
9	50 Hz
10	100 Hz
11	200 Hz
12	400 Hz
13	800 Hz
14	1600 Hz
15	3200 Hz

Tabla A.2: Correspondencia entre comando y frecuencia

A.4.8.12. Comando TM [Fw]

Al recibir el comando TM\r, el dispositivo deberá enviar las medidas realizadas en modo test en orden cronológico. El formato de los envíos deberá ser $\pm\text{xxxx}$, $\pm\text{yyyy}$, $\pm\text{zzzz}$;\r, donde:

- $\pm\text{xxxx}$ es la aceleración en el eje X,
- $\pm\text{yyyy}$ es la aceleración en el eje Y,
- $\pm\text{zzzz}$ es la aceleración en el eje Z.

Apéndice A. Requerimientos detallados

A.4.8.13. Comando AD [Fw]

Al recibir el comando AD [name]\r, el dispositivo deberá agregar un usuario con nombre de usuario [name]. El dispositivo deberá responder 1\r si la operación fue exitosa o 0\r en caso contrario (*por ejemplo, por falta de capacidad de almacenamiento*).

A.4.8.14. Comando DL [Fw]

Al recibir el comando DL [id]\r, el dispositivo deberá eliminar el usuario con número identificador [id] y sus datos estadísticos asociados. El dispositivo deberá responder 1\r si la operación fue exitosa o 0\r en caso contrario (*por ejemplo, no existe usuario ingresado con el [id] ingresado*).

A.4.8.15. Comando SU [Fw]

Al recibir el comando SU [id]\r, el dispositivo deberá setear el usuario con número identificador [id] como usuario actual. El dispositivo deberá responder 1\r si la operación fue exitosa o 0\r en caso contrario (*por ejemplo, no existe usuario ingresado con el [id] ingresado*).

A.4.8.16. Comando SH [Fw]

Al recibir el comando SH\r, el dispositivo deberá responder por cada usuario ingresado con un mensaje [id] [active] [name]\r, donde [id] es número identificador del usuario, [active] es 1 si es el usuario actual o 0 en caso contrario, y [name] es el nombre de usuario.

A.4.8.17. Comando DD [Fw]

Al recibir el comando DD [user_id] [meas_id]\r, el dispositivo deberá responder con Y\r, en caso que el usuario con número identificador [id] tenga almacenado una medida con número identificador [meas_id]; o N\r, en caso contrario. En el primer caso, posteriormente se deberá enviar los datos estadísticos asociados en el siguiente orden:

- Hora de comienzo de la medida en el siguiente formato: DD/MM/YYYY hh:mm:ss\r,
- Contador diario de persona parada,
- Contador diario de persona acostada,
- Contador diario de persona acostada boca arriba,
- Contador diario de persona acostada boca abajo,
- Contador diario de persona acostada sobre hombro derecho,
- Contador diario de persona acostada sobre hombro izquierdo,

- Contador diario de persona en posición intermedia,
- Contador diario de nivel de actividad 0,
- Contador diario de nivel de actividad 1,
- Contador diario de nivel de actividad 2,
- Contador diario de nivel de actividad 3,
- Contador diario de giros de 90° hacia la derecha,
- Contador diario de giros de 90° hacia la izquierda.

A.4.8.18. Comando SD [Fw]

Al recibir el comando SD [user_id] [meas_id]\r, el dispositivo deberá responder con 1\r, en caso que el usuario con número identificador [id] tenga almacenado una medida con número identificador [meas_id]; o 0\r, en caso contrario. En el primer caso, deberá enviar posteriormente todas las “levantadas” registradas en formato [Hora] [Duración]\r (ver A.4.7.6.15). Por último, como mensaje de terminación deberá enviar E\r.

A.4.8.19. Comando LY [Fw]

Al recibir el comando SD [user_id] [meas_id]\r, el dispositivo deberá responder con 1\r, en caso que el usuario con número identificador [id] tenga almacenado una medida con número identificador [meas_id]; o 0\r, en caso contrario. En el primer caso, deberá enviar posteriormente todas las “acostadas” registradas en formato [Hora] [Duración]\r (ver A.4.7.6.14). Por último, como mensaje de terminación deberá enviar E\r.

A.4.8.20. Comando TH [Fw]

Al recibir el comando TH [number]\r, el dispositivo deberá setear como umbrales a utilizar el conjunto de umbrales [number] (ver A.4.7.2.10). El dispositivo deberá responder 1\r si la operación fue exitosa o 0\r en caso contrario.

A.4.8.21. Comando TP [Fw]

Al recibir el comando TP\r, el dispositivo deberá responder 1\r en caso de ya haber realizado el posicionamiento o 0\r en caso contrario.

A.5. Sujeción

A.5.0.1. Comodidad de la persona [Hw]

El método de sujeción del dispositivo al torso de la persona deberá ser tal que no incomode a la misma en actividad normal, en particular mientras camina, corre,

Apéndice A. Requerimientos detallados

salta, o está acostada en cualquier posición.

A.5.0.2. Firmeza respecto a la persona [Hw]

El método de sujeción del dispositivo deberá ser tal que sujeto a un cilindro de madera de 40 cm de diámetro y 40 cm de altura el dispositivo no se mueva en ninguna dirección cuando se lo tira tres veces en la dirección de cada eje desde una distancia de 10 cm sobre una superficie rígida.

A.6. Adquisidor de datos

A.6.0.1. Entorno de funcionamiento del adquisidor [Sw]

El SAD deberá funcionar en un PC con puerto USB y sistema operativo Windows 7 o posterior o Linux.

A.6.0.2. GUI (*Graphical User Interface*) [Sw]

El SAD deberá tener una GUI a través de la cual interactuar con el dispositivo. Dicha GUI deberá permitir al usuario realizar las acciones listadas a continuación.

A.6.1. Posibles acciones

A.6.1.1. Seteo de hora [Sw]

El SAD deberá permitir setear el reloj del dispositivo con la hora actual del PC o con una hora válida ingresada.

A.6.1.2. Consulta de hora [Sw]

El SAD deberá permitir la consulta de la hora actual del reloj del dispositivo.

A.6.1.3. Creación de un usuario [Sw]

El SAD deberá permitir la creación de un usuario con un nombre de 4 caracteres ASCII de largo.

A.6.1.4. Cambio de usuario [Sw]

El SAD deberá permitir la selección de un usuario como usuario actual del dispositivo.

A.6.1.5. Eliminación de un usuario [Sw]

El SAD deberá permitir la eliminación de un usuario.

A.6. Adquisidor de datos

A.6.1.6. Descarga de datos de modo Test [Sw]

El SAD deberá permitir la descarga de los datos de aceleración adquiridos en el modo test.

A.6.1.7. Descarga de datos de modo Measure [Sw]

El SAD deberá permitir la descarga de los datos estadísticos adquiridos en el modo Measure y el guardado de los mismos en el PC como un archivo de texto plano. Se deberá poder elegir el usuario y la medida que se quiere descargar.

A.6.1.8. Seteo de umbrales de actividad [Sw]

El SAD deberá permitir setear los umbrales de actividad (ver A.4.7.2.10) del dispositivo en: “bajos” (umbrales 1), “medios”(umbrales 0) o “altos”(umbrales 2).

A.6.1.9. Testeo de posicionamiento [Sw]

El SAD deberá permitir el testeo del posicionamiento de cada eje del dispositivo.

A.6.1.10. Consulta de nivel de batería [Sw]

El SAD deberá permitir la consulta del nivel de batería del dispositivo.

Esta página ha sido intencionalmente dejada en blanco.

Apéndice B

Protocolos de test y resultados

Apéndice B. Protocolos de test y resultados

B.1. Propósito

El propósito de este documento es la recopilación de los protocolos de test realizados y los resultados de los mismos. La tabla B.1 indica para cada requerimiento si se redactó un protocolo de test formal y si efectivamente se testeó (formal o informalmente).

	Protocolo de test	Test		Protocolo de test	Test		Protocolo de test	Test
A.4.1.1	✓	✓	A.4.7.1.5	✓	✓	A.4.7.6.11	-	✓
A.4.1.2	✓	✓	A.4.7.1.6	✓	✓	A.4.7.6.12	-	✓
A.4.1.3	✓	✓	A.4.7.1.7	-	✓	A.4.7.6.13	-	✓
A.4.1.4	-	-	A.4.7.1.8	-	✓	A.4.7.6.14	-	✓
A.4.1.5	✓	-	A.4.7.2.1	✓	✓	A.4.7.6.15	-	✓
A.4.1.6	✓	-	A.4.7.2.2	✓	✓	A.4.8.1	-	✓
A.4.2.1	✓	✓	A.4.7.2.3	✓	✓	A.4.8.2	-	-
A.4.2.2	✓	✓	A.4.7.2.4	✓	✓	A.4.8.3	-	✓
A.4.2.3	✓	✓	A.4.7.2.5	✓	✓	A.4.8.4	-	✓
A.4.2.4	✓	✓	A.4.7.2.6	✓	✓	A.4.8.5	-	✓
A.4.3.1	✓	✓	A.4.7.2.7	✓	✓	A.4.8.6	-	✓
A.4.3.2	-	-	A.4.7.2.8	✓	✓	A.4.8.7	-	✓
A.4.3.3	✓	✓	A.4.7.2.9	✓	✓	A.4.8.8	-	✓
A.4.3.4	✓	✓	A.4.7.2.10	-	-	A.4.8.9	-	✓
A.4.3.5	✓	✓	A.4.7.2.11	✓	✓	A.4.8.10	-	✓
A.4.3.6	-	-	A.4.7.2.12	-	✓	A.4.8.11	-	✓
A.4.3.7	-	✓	A.4.7.3.1	-	✓	A.4.8.12	-	✓
A.4.3.8	-	✓	A.4.7.3.2	✓	✓	A.4.8.13	-	✓
A.4.3.9	✓	✓	A.4.7.3.3	-	✓	A.4.8.14	-	✓
A.4.4.1	✓	✓	A.4.7.3.4	-	✓	A.4.8.15	-	✓
A.4.4.2	✓	✓	A.4.7.3.5	-	✓	A.4.8.16	-	✓
A.4.4.3	✓	✓	A.4.7.3.6	-	✓	A.4.8.17	-	✓
A.4.4.4	-	✓	A.4.7.3.7	-	✓	A.4.8.18	-	✓
A.4.5.1	-	✓	A.4.7.4.1	✓	✓	A.4.8.19	-	✓
A.4.5.2	✓	✓	A.4.7.4.2	-	✓	A.4.8.20	-	✓
A.4.5.3	-	✓	A.4.7.4.3	-	✓	A.4.8.21	-	✓
A.4.5.4	-	✓	A.4.7.4.4	-	✓	A.5.0.1	-	✓
A.4.6.1	-	✓	A.4.7.5.1	-	✓	A.5.0.2	-	✓
A.4.6.2	-	✓	A.4.7.5.2	-	✓	A.6.0.1	-	-
A.4.6.3	✓	✓	A.4.7.5.3	✓	✓	A.6.0.2	-	-
A.4.6.4	-	✓	A.4.7.5.4	-	✓	A.6.1.1	-	-
A.4.6.5	-	✓	A.4.7.5.5	-	✓	A.6.1.2	-	-
A.4.6.6	-	✓	A.4.7.6.1	✓	✓	A.6.1.3	-	-
A.4.6.7	-	✓	A.4.7.6.2	✓	✓	A.6.1.4	-	-
A.4.6.8	-	✓	A.4.7.6.3	✓	✓	A.6.1.5	-	-
A.4.6.9	-	✓	A.4.7.6.4	✓	✓	A.6.1.6	-	-
A.4.6.10	-	✓	A.4.7.6.5	✓	✓	A.6.1.7	-	-
A.4.6.11	-	✓	A.4.7.6.6	✓	✓	A.6.1.8	-	-
A.4.7.1.1	-	✓	A.4.7.6.7	✓	✓	A.6.1.9	-	-
A.4.7.1.2	-	✓	A.4.7.6.8	-	✓	A.6.1.10	-	-
A.4.7.1.3	✓	✓	A.4.7.6.9	-	✓			
A.4.7.1.4	✓	✓	A.4.7.6.10	-	✓			

Tabla B.1: Requerimientos con protocolo de test redactado y con test realizado

B.2. Definiciones, abreviaturas y acrónimos

Sistema: Sistema completo de medida de posición y movimiento de una persona. Compuesto por una unidad central de recopilación de datos, un PC que corre el *software* adquisidor junto al cable de conexión y un elemento de fijación al torso del usuario.

MedAcel: Proyecto de fin de carrera “Dispositivo de medida de posición y movimiento de una persona”.

Dispositivo: Unidad central del sistema, se encarga de la recopilación de medidas de movimiento del usuario. Va sujeto al torso del individuo mediante el elemento de fijación.

SAD: *Software* Adquisidor de Datos. Unidad de *software* que corre en el ordenador conectado al dispositivo. Permite el despliegue de los datos recabados y la configuración de algunas funcionalidades del dispositivo.

Persona, individuo: Usuario portador del dispositivo.

B.3. Convenciones

B.3.1. Formato de cada test

Con el fin de mantener una correlación con el documento de requerimientos, cada protocolo de test y su resultado conserva el nombre y numeración del requerimiento a testear.

Cada protocolo de test contiene un apartado **Procedimiento** y un apartado **Resultados**:

- El apartado **Procedimiento** lista los pasos a seguir del test.
- El apartado **Resultados** detalla los resultados esperados del test y los resultados obtenidos al realizarlo. Además, especifica con qué unidad se realizó el test y la conclusión final del mismo (**PASA** o **FALLA**).

B.3.2. Comentarios

Los comentarios de cada test (por ejemplo, justificaciones u observaciones) son identificados mediante letra *itálica*.

B.3.3. *Firmware* del dispositivo

Los test serán realizados sobre unidades con una de las siguientes entregas de *firmware*:

- **Firmware Final:** Versión final de *firmware* del dispositivo final.

Apéndice B. Protocolos de test y resultados

- **Firmware Prototipo:** Versión final de *firmware* del dispositivo prototipo. Es equivalente al Firmware Final salvo las siguientes modificaciones:
 - Adaptaciones al *hardware* del dispositivo prototipo.
 - Se conmuta el pin 14 del prototipo cada vez que se realiza una medida de alimentación.
 - Se conmuta el pin 28 del prototipo cada vez que se realiza una medida de aceleración.
 - En modo Measure, cada período de clasificación se envía mediante UART al PC el contador que fue incrementado y si se detectó un giro o una “levantada”/“acostada”.
 - En modo Measure, se conmuta el pin 16 cada período de clasificación para poder medirlo con un osciloscopio.
 - En modo Measure, se enciende el LED3 del prototipo cuando la frecuencia de muestreo de aceleraciones es alta (**HIGH ODR**) y se apaga el LED3 cuando la frecuencia de muestreo es baja (**LOW ODR**).
- **Firmware Bench:** Versión modificada de Firmware Prototipo. En modo Measure, guarda 512 muestras de aceleraciones en memoria Flash, les realiza el procesamiento y guarda los resultados. Luego, permite descargar estos datos para su posterior comparación con el mismo algoritmo corrido *offline* (en Matlab) sobre el mismo conjunto de datos de aceleración.

A menos que se indique lo contrario, los test serán realizados con el Firmware Final o el Firmware Prototipo (según la unidad testeada sea un dispositivo final o dispositivo prototipo).

B.4. Dispositivo

B.4.1. Requerimientos mecánicos y físicos

B.4.1.1. Masa [Hw]

Procedimiento:

1. **Pesar** el dispositivo con una balanza calibrada, **verificar** que la masa del dispositivo es menor a 200 g.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	La masa del dispositivo es menor a 200 g.	Masa: 120 g	Sí

Unidad testada: Final 01

Conclusión: **PASA**

B.4.1.2. Caja [Hw]

Procedimiento:

1. **Verificar** mediante inspección que el dispositivo tiene una caja de protección prismática de base rectangular.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	El dispositivo tiene una caja de protección prismática de base rectangular.	Caja prismática de base rectangular.	Sí

Unidad testada: Final 01

Conclusión: **PASA**

B.4.1.3. Dimensiones [Hw]

Procedimiento:

1. **Medir** el largo del dispositivo, **verificar** que es menor a 12 cm.
2. **Medir** el ancho del dispositivo, **verificar** que es menor a 8 cm.
3. **Medir** la altura del dispositivo, **verificar** que es menor a 2 cm.

Apéndice B. Protocolos de test y resultados

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	La longitud del largo del dispositivo es menor a 12 cm.	Largo: 10,6 cm	Sí
2	La longitud del ancho del dispositivo es menor a 8 cm.	Ancho: 6,6 cm	Sí
3	La longitud de la altura del dispositivo es menor a 2 cm.	Altura: 1,9 cm	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.1.4. Rating de protección [Hw]

No se redacta protocolo de test.

Unidad testeada: -

Conclusión: **SIN TESTEAR**

B.4.1.5. Temperatura de funcionamiento [Hw]

Procedimiento:

1. Con el dispositivo a 40 °C, **verificar** que las principales funcionalidades del mismo (modo Measure) se desempeñan según las especificaciones.
2. Con el dispositivo a -10 °C, **verificar** que las principales funcionalidades del mismo (modo Measure) se desempeñan según las especificaciones.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	Las funcionalidades principales operan según sus especificaciones a 40 °C.	-	-
2	Las funcionalidades principales operan según sus especificaciones a -10 °C.	-	-

Unidad testeada: -

Conclusión: **SIN TESTEAR**

B.4.1.6. Temperatura de almacenamiento [Hw]

Procedimiento:

1. Almacenar el dispositivo sin pilas durante más de 5 horas a 70 °C, **verificar** que sus principales funcionalidades (modo Measure) se desempeñan según las especificaciones.

B.4. Dispositivo

2. Almacenar el dispositivo sin pilas durante más de 5 horas a -30°C , **verificar** que sus principales funcionalidades (modo Measure) se desempeñan según las especificaciones.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	Las funcionalidades principales operan según sus especificaciones luego de almacenado a 70°C .	-	-
2	Las funcionalidades principales operan según sus especificaciones luego de almacenado a -30°C .	-	-

Unidad testada: -

Conclusión: **SIN TESTEAR**

B.4.2. Interfaz de usuario

B.4.2.1. Llave de encendido/apagado [Hw]

Procedimiento:

1. Inspeccionar el dispositivo y **verificar** que tiene una llave ON/OFF.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	El dispositivo tiene una llave ON/OFF.	Llave ON/OFF.	Sí

Unidad testada: Final 01

Conclusión: **PASA**

B.4.2.2. Indicador de batería baja [Hw]

Procedimiento:

1. Inspeccionar el dispositivo y **verificar** que tiene un LED de color amarillo.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	El dispositivo tiene un LED amarillo.	Un LED amarillo.	Sí

Unidad testada: Final 01

Conclusión: **PASA**

Apéndice B. Protocolos de test y resultados

B.4.2.3. Indicadores de uso general [Hw]

Procedimiento:

1. Inspeccionar el dispositivo y **verificar** que tiene 3 LEDs de color verde.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	El dispositivo tiene 3 LEDs verdes.	3 LEDs verdes.	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.2.4. Botones de uso general [Hw]

Procedimiento:

1. Inspeccionar el dispositivo y **verificar** que tiene 3 botones.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	El dispositivo tiene 3 botones.	3 botones.	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.3. Requerimientos generales

B.4.3.1. Alimentación del dispositivo [Hw]

Procedimiento:

1. **Verificar** mediante inspección que el dispositivo es compatible con pilas alcalinas convencionales AA o AAA.
2. **Verificar** que las pilas son de fácil acceso para su reemplazo por parte del usuario.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	El dispositivo es compatible con pilas alcalinas convencionales AA o AAA.	Compatible con pilas AAA.	Sí
2	Las pilas son de fácil acceso para su reemplazo.	Fácil acceso.	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.3.2. Autonomía del dispositivo [Hw][Fw]

No se redacta protocolo de test.

Unidad testeada: -

Conclusión: **SIN TESTEAR**

B.4.3.3. Medida de alimentación [Hw][Fw]

Procedimiento - parte 1:

- 1.a. Alimentar el dispositivo con una fuente de alimentación variable. Fijar el voltaje en $3\text{ V} \pm 1\%$.
- 1.b. Conectar el dispositivo al PC y abrir un terminal serial.
- 1.c. Enviar un comando RB (Read Battery) y **registrar** la respuesta. **Verificar** que es un número de 0 a 255 y que corresponde al rango esperado.
- 1.d. **Repetir** el paso 1.c con el voltaje de alimentación fijado en $1,8\text{ V} \pm 1\%$.

Resultados - parte 1:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1.c	La respuesta del dispositivo es 233 ± 11 (corresponde a $3\text{ V} \pm 5\%$).	Respuesta: 243	Sí
1.d	La respuesta del dispositivo es 140 ± 7 (corresponde a $1,8\text{ V} \pm 5\%$).	Respuesta: 135	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

Procedimiento - parte 2:

- 2.a. Conectar la punta de prueba de un osciloscopio al pin 14 del dispositivo prototipo.
- 2.b. **Verificar** que se observa una señal cuadrada periódica, **registrar** el tiempo entre flancos.

Resultados - parte 2:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
2.b	Señal cuadrada periódica con flancos cada $1\text{ s} \pm 0,01\text{ s}$.	Cuadrada periódica, flancos cada 1,003 s.	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

Apéndice B. Protocolos de test y resultados

B.4.3.4. Indicado de nivel de batería [Hw][Fw]

Procedimiento:

1. Alimentar el dispositivo con una fuente de alimentación variable. Fijar el voltaje en $3\text{ V} \pm 1\%$. **Verificar** que el indicador de batería baja está apagado.
2. Decrementar el voltaje en pasos de $0,1\text{ V}$ hasta que el indicador de batería baja parpadee, **registrar** dicho voltaje.
3. Decrementar el voltaje en pasos de $0,1\text{ V}$ hasta que el indicador de batería baja permanezca encendido continuamente, **registrar** dicho voltaje.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	LED OFF	LED OFF	Sí
2	LED intermitente: $2,0\text{ V} \pm 10\%$	$2,0\text{ V}$	Sí
3	LED ON: $1,6\text{ V} \pm 10\%$	$1,5\text{ V}$	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.3.5. Reloj de tiempo real (RTC) [Fw]

Procedimiento:

1. Conectar el dispositivo al PC y abrir el SAD.
2. Setear el dispositivo con la hora 00:00:00.
3. Luego de 168 horas (7 días), realizar una consulta de la hora del dispositivo y **registrar**.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Hora: 23:59:46 - 00:00:14	Hora: 00:00:01	Sí

Unidad testeada: Prototipo 03

Conclusión: **PASA**

B.4.3.6. Medida de aceleración [Hw][Fw]

No se redacta protocolo de test.

Unidad testeada: -

Conclusión: **SIN TESTEAR**

B.4.3.7. Calibración [Fw]

No se redacta protocolo de test.

Test informal: Se colocó el dispositivo ya calibrado sin posicionar en un plano horizontal sobre cada una de sus caras y se verificó que se mida una aceleración de $1\text{ g} \pm 0,01\text{ g}$ en el eje correspondiente.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.3.8. Múltiples usuarios [Fw]

No se redacta protocolo de test.

Test informal: Se agregaron 3 usuarios y se verificó que cada uno puede almacenar medidas independientes.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.3.9. Modos de operación [Fw]

Procedimiento:

1. Setear el dispositivo en modo Standby, **verificar** que todos los indicadores LEDs estén apagados.
2. Setear el dispositivo en modo Measure, **verificar** que LED1 esté encendido y el resto estén apagados.
3. Setear el dispositivo en modo Posicionamiento, **verificar** que LED2 esté encendido y el resto estén apagados.
4. Setear el dispositivo en modo Test, **verificar** que LED3 esté encendido y el resto estén apagados.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	LED1, LED2 y LED3 apagados.	LED1, LED2 y LED3 apagados.	Sí
2	Sólo LED1 encendido.	Sólo LED1 encendido	Sí
3	Sólo LED2 encendido.	Sólo LED2 encendido.	Sí
4	Sólo LED3 encendido.	Sólo LED3 encendido.	Sí

Unidad testada: Final 01

Conclusión: **PASA**

Apéndice B. Protocolos de test y resultados

B.4.4. Modo Standby

B.4.4.1. Inicio del dispositivo [Fw]

Procedimiento:

1. Encender el dispositivo, **verificar** que LED1, LED2 y LED3 estén apagados.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	LED1, LED2 y LED3 apagados.	LED1, LED2 y LED3 apagados.	Sí

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.4.2. Indicado de dispositivo en modo Standby [Fw]

La prueba de este requerimiento se cubre en el test B.4.4.1.

B.4.4.3. Consumo [Hw][Fw]

Nota: En este test se utiliza el amplificador logarítmico de corriente ee-201. Este aparato genera una señal de voltaje a partir de la corriente medida, cuya relación está dada por la expresión $i = 10^{-V}$.

Procedimiento:

1. Conectar los pines positivo y negativo de alimentación del dispositivo a los terminales DUT y GND del amplificador respectivamente.
2. Conectar los terminales positivo y negativo de una fuente de alimentación variable a los terminales SUPPLY y GND del amplificador respectivamente. Fijar el voltaje de la fuente en $3V \pm 1\%$.
3. Conectar la punta de prueba de un osciloscopio al terminal SCOPE del amplificador.
4. Medir y **registrar** el valor medio de la señal observada en el osciloscopio. **Registrar** la corriente correspondiente a ese voltaje y **verificar** que esté en el rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Corriente menor a $500\mu A$	Valor medio: 3,331 V Corriente: $467\mu A$	Sí

Unidad testeada: Prototipo 03

Conclusión: **PASA**

B.4.4.4. Nivel de batería crítico [Fw]

No se redacta protocolo de test.

Se alimentó el dispositivo con una fuente variable y se fijó el voltaje en $1,5\text{ V} \pm 1\%$. Se encendió el dispositivo y se verificó que no sea posible pasar a modo Test, Posicionamiento o Measure.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.5. Modo Test

B.4.5.1. Acceso al modo Test [Fw]

No se redacta protocolo de test.

Test informal: Con el dispositivo en modo Standby, se cronometró el tiempo que es necesario mantener apretado el botón 2 y se verificó que este fuera aproximadamente 3 s.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.5.2. Indicado de dispositivo en modo Test [Fw]

No se redacta protocolo de test.

Test informal: Se seteó el dispositivo en modo Test y se verificó que se enciendan los 3 LEDs de color verde.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.5.3. Adquisición de medidas [Fw]

Procedimiento:

1. Setear el dispositivo en modo Test y cronometrar el tiempo desde que ingresó al modo Test hasta que retorna a modo Standby. **Registrar** dicho tiempo y **verificar** que se encuentre en el rango esperado.
2. Conectar el dispositivo al PC y abrir el SAD.
3. Descargar los datos de modo Test, **verificar** que se hayan obtenido la cantidad de muestras de aceleración esperadas.
4. Apagar y encender el dispositivo y volver a descargar los datos de modo Test. **Verificar** que los datos descargados sean los mismos a los obtenidos en el paso 3.

Apéndice B. Protocolos de test y resultados

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1	Tiempo Test: $163\text{ s} \pm 5\text{ s}$	Valor medio: 162 s	Sí
3	Muestras: 2048	Muestras: 2048	Sí
4	Mismos datos	Mismos datos	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.5.4. Cancelación de adquisición [Fw]

No se redacta protocolo de test.

Test informal: Se seteo el dispositivo en modo Test, se cronometró el tiempo que hay que mantener apretado el botón 2 para retornar a modo Standby y se obtuvo que dicho tiempo es aproximadamente 3 s.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.6. Modo Posicionamiento

B.4.6.1. Algoritmo de posicionamiento [Fw]

La prueba de este requerimiento está cubierta en los tests B.4.6.3, B.4.6.4 y B.4.6.5.

B.4.6.2. Medidas de aceleración en posicionamiento [Fw]

No se redacta protocolo de test.

Test informal: Se conectó la punta de prueba de un osciloscopio al pin 28 del prototipo. Se seteo el dispositivo en modo Posicionamiento y se disparo el posicionamiento del eje X. En el osciloscopio, se obtuvo una señal cuadrada con 25 flancos y frecuencia 24,8 s.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.6.3. Posicionado de eje X [Fw]

Procedimiento - parte 1:

- 1.a. Con el dispositivo quieto sobre una superficie, disparar el posicionamiento del eje X y **verificar** que es tomado como válido.

B.4. Dispositivo

- 1.b Con el dispositivo en movimiento apreciable, disparar el posicionamiento del eje X y **verificar** que es tomado como inválido y que es necesario su repetición.

Resultados esperados y obtenidos - parte 1:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
1.a	Posicionamiento válido	Posicionamiento válido	Sí
1.b	Posicionamiento inválido	Posicionamiento inválido	Sí

Nota: En este test, se fija un breakpoint en la línea 173 del archivo `posicionamiento.c` y se utiliza el programador MSP-FET430UIF y el entorno de desarrollo IAR para editar en ejecución variables en memoria.

Procedimiento - parte 2:

- 2.a. Disparar el posicionamiento del eje X, verificar que la ejecución se detiene en el breakpoint. Editar la variable `z` de manera que la diferencia absoluta con la variable `z_ant` sea igual a 16 (*equivalente a 64 mg*) y continuar la ejecución del programa. **Verificar** que el posicionamiento es tomado como inválido.
- 2.b. Disparar el posicionamiento del eje X, verificar que la ejecución se detiene en el breakpoint. Editar la variable `z` de manera que la diferencia absoluta con la variable `z_ant` sea igual a 14 (*equivalente a 56 mg*) y continuar la ejecución del programa. **Verificar** que el posicionamiento es tomado como válido.
- 2.c. Disparar el posicionamiento del eje X, verificar que la ejecución se detiene en el breakpoint. Editar las variables `x`, `y`, `z` con los valores 0, 0 y 282 respectivamente (*módulo aproximado a 1,102 g*) y continuar la ejecución del programa. **Verificar** que el posicionamiento es tomado como inválido.
- 2.d. Disparar el posicionamiento del eje X, verificar que la ejecución se detiene en el breakpoint. Editar las variables `x`, `y`, `z` con los valores 0, 0 y 281 respectivamente (*módulo aproximado a 1,098 g*) y continuar la ejecución del programa. **Verificar** que el posicionamiento es tomado como válido.
- 2.e. Disparar el posicionamiento del eje X, verificar que la ejecución se detiene en el breakpoint. Editar las variables `x`, `y`, `z` con los valores 0, 0 y 230 respectivamente (*módulo aproximado a 0,898 g*) y continuar la ejecución del programa. **Verificar** que el posicionamiento es tomado como inválido.
- 2.f. Disparar el posicionamiento del eje X, verificar que la ejecución se detiene en el breakpoint. Editar las variables `x`, `y`, `z` con los valores 0, 0 y 231 respectivamente (*módulo aproximado a 0,902 g*) y continuar la ejecución del programa. **Verificar** que el posicionamiento es tomado como válido.

Resultados esperados y obtenidos - parte 2:

Apéndice B. Protocolos de test y resultados

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
2.a	Posicionamiento inválido	Posicionamiento inválido	Sí
2.b	Posicionamiento válido	Posicionamiento válido	Sí
2.c	Posicionamiento inválido	Posicionamiento inválido	Sí
2.d	Posicionamiento válido	Posicionamiento válido	Sí
2.e	Posicionamiento inválido	Posicionamiento inválido	Sí
2.f	Posicionamiento válido	Posicionamiento válido	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.6.4. Posicionado de eje Y [Fw]

No se redacta protocolo de test.

Test informal: Primero, se intentó realizar el posicionado del eje Y con el dispositivo en movimiento y se verificó que obtenga un posicionamiento inválido. Luego, se colocó el dispositivo en la misma posición del eje X, y se intentó posicionar el eje Y aumentando el ángulo con el eje X en pasos de 10° hasta que el posicionamiento sea tomado como válido. Se obtuvo que dicho ángulo fue 70° .

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.6.5. Posicionado de eje Z [Fw]

No se redacta protocolo de test.

Test informal: Primero, se intentó realizar el posicionado del eje Z con el dispositivo en movimiento y se verificó que se obtenga un posicionamiento inválido. Luego, se colocó el dispositivo en la misma posición del eje X, y se intentó posicionar el eje Z aumentando el ángulo con el eje X en pasos de 10° hasta que el posicionamiento sea tomado como válido. Se obtuvo que dicho ángulo fue 70° . Se repitió el mismo procedimiento con ángulo Y-Z y se obtuvo que el ángulo umbral también fue 70° .

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.6.6. Acceso a modo Posicionamiento [Hw]

No se redacta protocolo de test.

Test informal: Con el dispositivo en modo Standby, se cronometró el tiempo que hay que mantener apretado el botón 1 para que pase a modo Posicionamiento.

Se obtuvo que dicho tiempo es aproximadamente 3 s.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.6.7. Indicado de etapa de posicionamiento [Fw]

No se redacta protocolo de test.

Test informal: Se seteo el dispositivo en modo Posicionamiento, y se verificó que en cada etapa de posicionado se encienda sólo el LED correspondiente.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.6.8. Disparo de posicionado [Fw]

No se redacta protocolo de test.

Test informal: Con el dispositivo seteado en modo Posicionamiento, se verificó que al apretar BUT1 se dispare el posicionamiento del eje correspondiente.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.6.9. Referencia de aceleraciones respecto al usuario [Fw]

No se redacta protocolo de test.

Test informal: Se posicionó el dispositivo y se verificó que en cada una de las posiciones en las que se posicionó se mida $1\text{ g} \pm 0,01\text{ g}$.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.6.10. Posicionamiento exitoso [Fw]

No se redacta protocolo de test.

Test informal: Se seteo el dispositivo en modo Posicionamiento, se posicionaron los 3 ejes y se verificó que todos los LEDs verdes se enciendan durante aproximadamente 1 s.

Unidad testada: Final 01

Conclusión: **PASA**

Apéndice B. Protocolos de test y resultados

B.4.6.11. Cancelación de posicionamiento [Fw]

No se redacta protocolo de test.

Test informal: Se seteo el dispositivo en modo Posicionamiento y se verificó que al apretar BUT2 todos los LEDs verdes parpaddeen dos veces.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7. Modo Measure

B.4.7.1. Generales

B.4.7.1.1. Acceso al modo Measure [Fw]

No se redacta protocolo de test.

Test informal: Con el dispositivo en modo Standby, se cronometró el tiempo que hay que mantener apretado el botón 3 para que pase a modo Measure. Se obtuvo que dicho tiempo es aproximadamente 3s.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.1.2. Indicado de dispositivo en modo Measure [Fw]

No se redacta protocolo de test.

Test informal: Se conectó un osciloscopio en el pin del LED2 y se verificó que se obtuviera una onda cuadrada de período 2s con tiempo de encendido 50 ms. Se obtuvo que el período es de 2,01s y el tiempo de encendido es de 54ms.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.1.3. Medidas [Fw]

Procedimiento:

1. Iniciar y finalizar 7 medidas consecutivas.
2. Conectar el dispositivo al PC y abrir la GUI.
3. Descargar y **verificar** que se pueden descargar los datos de cada una de las 7 medidas.
4. Iniciar y finalizar una nueva medida.

B.4. Dispositivo

5. Descargar los datos y **verificar** que se sobrescribieron los datos de la primer medida (medida con hora de comienzo más antigua).

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Datos de 7 medidas disponibles	Datos de 7 medidas disponibles	Sí
5	Datos de medida de mayor antigüedad sobrescritos	Datos de medida de mayor antigüedad sobrescritos	Sí

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.1.4. Inicio de medida [Fw]

Procedimiento:

1. Conectar el dispositivo al PC y abrir el SAD. Sincronizar la hora del dispositivo con la del PC.
2. Iniciar y finalizar una medida. Registrar la hora de comienzo de la medida.
3. Descargar los datos de la medida. **Verificar** que todos los contadores y registros estadísticos hayan sido reseteados. **Verificar** que la hora de comienzo de la medida sea correcta.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Datos estadísticos reseteados	Datos estadísticos reseteados	Sí
3	Hora de comienzo OK	Hora de comienzo OK	Sí

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.1.5. Medidas diarias [Fw]

Procedimiento:

1. Conectar el dispositivo al PC y abrir el SAD. Sincronizar la hora del dispositivo con la del PC.
2. Iniciar una medida. Dejar operando durante más de 24 horas y finalizar la medida.
3. Descargar los datos estadísticos. **Verificar** que se registraron datos en dos medidas, **verificar** que la hora de comienzo de la segunda medida es 00:00:00.

Apéndice B. Protocolos de test y resultados

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Dos medidas registradas	Dos medidas registradas	Sí
3	Hora de comienzo de la segunda medida 00:00:00	Hora de comienzo de la segunda medida 00:00:00	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.1.6. Nivel de batería crítico durante medida [Fw]

Procedimiento:

1. Alimentar el dispositivo con una fuente de alimentación variable. Fijar el voltaje en $3\text{ V} \pm 1\%$.
2. Iniciar una nueva medida entrando en modo Measure.
3. Fijar el voltaje de alimentación en $1,5\text{ V} \pm 1\%$. **Verificar** que el dispositivo finalice la medida y retorne a modo Standby.
4. Conectar el dispositivo al PC y abrir el SAD.
5. Descargar los datos de la medida y **verificar** que sean correctos (hora de comienzo, hora de fin, tiempo total, etc).

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Retorna a modo Standby	Retorna a modo Standby	Sí
5	Datos OK	Datos OK	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.1.7. Finalización de medida [Fw]

No se redacta protocolo de test.

Test informal: Con el dispositivo en modo Measure, se cronometró el tiempo que hay que mantener apretado BUT3 para retornar a modo Standby. Luego, se descargaron los datos y se verificó que los datos y la hora de finalización se hayan guardado correctamente.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.1.8. Protección contra bloqueo de microcontrolador [Fw]

No se redacta protocolo de test.

Test informal: En modo Measure, se agregó al código un *loop* que espere el reseteo de un timer de 1,1s y se verificó que el dispositivo es reseteado antes de salir del *loop*.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.2. Procesamiento de aceleraciones

B.4.7.2.1. Frecuencia de muestreo de aceleraciones

Procedimiento:

1. Conectar la punta de prueba de un osciloscopio al pin 28 del prototipo.
2. Setear el dispositivo en modo Measure.
3. Dejar el dispositivo quieto de manera que LED3 esté apagado (LOW ODR). **Verificar** que la señal observada en el osciloscopio es cuadrada, **registrar** el tiempo entre flancos y **verificar** que esté dentro de lo esperado.
4. Mover el dispositivo de manera que LED3 esté encendido (HIGH ODR). **Verificar** que la señal observada en el osciloscopio es cuadrada, **registrar** el tiempo entre flancos y **verificar** que esté dentro de lo esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Se observa una onda cuadrada con tiempo entre flancos $0,64\text{s} \pm 5\%$.	Flancos cada 0,65 s	Sí
4	Se observa una onda cuadrada con tiempo entre flancos $0,08\text{s} \pm 5\%$.	Flancos cada 0,082 s	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.2. Cálculo de aceleración DC [Fw]

Procedimiento:

1. Programar el dispositivo con el Firmware Bench.
2. Encender el dispositivo y setearlo en modo Measure. Mover el dispositivo hasta que retorne a modo Standby.
3. Conectar el dispositivo al PC y abrir Matlab. Correr el script `bench_test.m`.

Apéndice B. Protocolos de test y resultados

4. **Registrar** el error máximo en la aceleración DC y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Error máximo: 1 LSB	Error máximo: 1 LSB	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.3. Cálculo de aceleración AC [Fw]

Procedimiento:

1. Programar el dispositivo con el Firmware Bench.
2. Encender el dispositivo y setearlo en modo Measure. Mover el dispositivo hasta que retorne a modo Standby.
3. Conectar el dispositivo al PC y abrir Matlab. Correr el script `bench_test.m`.
4. **Registrar** el error máximo en la aceleración AC y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Error máximo: 1 LSB	Error máximo: 1 LSB	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.4. Cálculo de módulo AC [Fw]

Procedimiento:

1. Programar el dispositivo con el Firmware Bench.
2. Encender el dispositivo y setearlo en modo Measure. Mover el dispositivo hasta que retorne a modo Standby.
3. Conectar el dispositivo al PC y abrir Matlab. Correr el script `bench_test.m`.
4. **Registrar** el error máximo en el módulo AC y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Error máximo: 1 %	Error máximo: 0 %	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.5. Procesamiento de medidas de aceleración [Fw]

La prueba de este requerimiento se cubre en los tests B.4.7.2.2, B.4.7.2.3 y B.4.7.2.4.

B.4.7.2.6. Promedio AC en un segundo [Fw]

Procedimiento:

1. Programar el dispositivo con el Firmware Bench.
2. Encender el dispositivo y setearlo en modo Measure. Mover el dispositivo hasta que el dispositivo retorne a modo Standby.
3. Conectar el dispositivo al PC y abrir Matlab. Correr el script `bench_test.m`.
4. **Registrar** el error máximo en el promedio de un segundo y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Error máximo: 1 %	Error máximo: 0.89 %	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.7. Promedio AC en 12 segundos [Fw]

Procedimiento:

1. Programar el dispositivo con el Firmware Bench.
2. Encender el dispositivo y setearlo en modo Measure. Mover el dispositivo hasta que el dispositivo retorne a modo Standby.
3. Conectar el dispositivo al PC y abrir Matlab. Correr el script `bench_test.m`.
4. **Registrar** el error máximo en la medida de actividad y **verificar** que esté dentro del rango esperado.

Apéndice B. Protocolos de test y resultados

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Error máximo: 1 %	Error máximo: 0.59 %	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.8. Periodo de clasificación [Fw]

Procedimiento:

1. Iniciar una medida.
2. Conectar la punta de prueba de un osciloscopio al pin 14 del dispositivo prototipo.
3. **Verificar** que se observa una señal cuadrada periódica, **registrar** el tiempo entre flancos.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Señal cuadrada periódica con flancos cada $1\text{ s} \pm 0,001\text{ s}$.	Cuadrada periódica, flancos cada 1,001 s.	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.9. Medida de actividad [Fw]

La prueba de este requerimiento se cubre en los tests B.4.7.2.6 y B.4.7.2.7.

B.4.7.2.10. Umbrales de actividad [Fw]

No se redacta protocolo de test.

Unidad testeada: -

Conclusión: **SIN TESTEAR**

B.4.7.2.11. Pasaje de modo LOW ODR a modo HIGH ODR [Fw]

Nota: Para este test, se fija un breakpoint en la línea 344 del archivo `measure_mode.c` y se utiliza el programador `MSP-FET430UIF` y el entorno de desarrollo para editar en ejecución variables en memoria.

Procedimiento:

B.4. Dispositivo

1. Setear el dispositivo en modo Measure, verificar que la ejecución se detiene en el breakpoint.
2. Modificar las variables `accel.data[accel.index].x`, `accel.data[accel.index].y`, `accel.data[accel.index].z` con los valores 0, 0, y 230 (*módulo aproximado a 0,898 g*) y continuar con la ejecución del programa. **Verificar** que se enciende el LED3 del prototipo.
3. Esperar a que el dispositivo retorne a modo LOW ODR (se apaga LED3). Verificar que la ejecución se detiene en el breakpoint.
4. Modificar las variables `accel.data[accel.index].x`, `accel.data[accel.index].y`, `accel.data[accel.index].z` con los valores 0, 0, y 282 (*módulo aproximado a 1,102 g*) y continuar con la ejecución del programa. **Verificar** que se enciende el LED3 del prototipo.
5. Esperar a que el dispositivo retorne a modo LOW ODR (se apaga LED3). Verificar que la ejecución se detiene en el breakpoint.
6. Modificar las variables `accel.data[accel.index].x`, `accel.data[accel.index].y`, `accel.data[accel.index].z` con los valores 0, 0, y 256 (*módulo igual a 1 g*) y continuar con la ejecución del programa. **Verificar** que el LED3 permanece apagado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
2	Se enciende LED3	Se enciende LED3	Sí
4	Se enciende LED3	Se enciende LED3	Sí
6	LED3 apagado	LED3 apagado	Sí

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.2.12. Pasaje de modo HIGH ODR a modo LOW ODR [Fw]

No se redacta protocolo de test.

Test informal: Con el dispositivo calibrado y posicionado, mover el dispositivo durante 6s a partir de que se encienda el LED3 indicando que se entró en modo HIGH ODR. Dejar el dispositivo en reposo y esperar a que el LED3 se apague. Verificar que luego de dejar quieto el dispositivo el LED3 se apaga en menos de 18 s.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.3. Detección de posición

B.4.7.3.1. Posiciones posibles [Fw]

La prueba de este requerimiento se cubre en los tests B.4.7.3.2, B.4.7.3.3, B.4.7.3.4, B.4.7.3.5, B.4.7.3.6 y B.4.7.3.7.

Apéndice B. Protocolos de test y resultados

B.4.7.3.2. Posición persona parada [Fw]

Nota: Para este test, se agrega un breakpoint en la línea 254 del archivo `measure_mode.c` y se utiliza el programador `MSP-FET430UIF` y el entorno de desarrollo `IAR` para editar en ejecución variables en memoria.

Procedimiento:

1. Conectar el dispositivo al PC y abrir un terminal serial.
2. Setear el dispositivo en modo Measure, verificar que la ejecución se detiene en el breakpoint.
3. Modificar la variable `z` con el valor 229 (*aproximadamente 0,895 g*) y continuar la ejecución del programa.
4. **Verificar** que en el terminal serial se recibe `STANDING++\r`.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Se recibe <code>STANDING++\r</code>	Se recibe <code>STANDING++\r</code>	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.3.3. Posición persona acostada boca arriba [Fw]

No se redacta protocolo de test.

Test informal: Se realizó un test funcional de este requerimiento con el dispositivo prototipo. Se colocó y posicionó el dispositivo a un usuario, se conectó el dispositivo al PC y se abrió un terminal serial. Luego, se verificó que cuando el usuario está acostado boca arriba se recibe `LAYING_DOWN_UP++\r`.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.3.4. Posición persona acostada boca abajo [Fw]

No se redacta protocolo de test.

Test informal: Análogo al test B.4.7.3.3, se verifica que cuando el usuario está acostado boca abajo se recibe `LAYING_DOWN_DOWN++\r`.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.3.5. Posición persona acostada sobre hombro derecho [Fw]

No se redacta protocolo de test.

Test informal: Análogo al test B.4.7.3.3, se verifica que cuando el usuario está acostado sobre su hombro derecho se recibe LAYING_DOWN_RIGHT++\r.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.3.6. Posición persona acostada sobre hombro izquierdo [Fw]

No se redacta protocolo de test.

Test informal: Análogo al test B.4.7.3.3, se verifica que cuando el usuario está acostado sobre su hombro izquierdo se recibe LAYING_DOWN_LEFT++\r.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.3.7. Posición intermedia [Fw]

No se redacta protocolo de test.

Test informal: Análogo al test B.4.7.3.3, se verifica que cuando el usuario está en posición intermedia (entre parado y acostado) se recibe INTERMEDIATE++\r.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.4. Detección de giros, levantadas y acostadas

B.4.7.4.1. Giro de 90° hacia la derecha [Fw]

Procedimiento:

1. Posicionar el dispositivo.
2. Conectar el dispositivo al PC y abrir un terminal serial.
3. Colocar el dispositivo de manera que se reciba LAYING_DOWN_UP++\r, girarlo en torno al eje Z en sentido horario hasta que se reciba LAYING_DOWN_RIGHT++\r. **Verificar** que se recibe TURN_RIGHT++\r.
4. Colocar el dispositivo de manera que se reciba LAYING_DOWN_RIGHT++\r, girarlo en torno al eje Z en sentido horario hasta que se reciba LAYING_DOWN_DOWN++\r. **Verificar** que se recibe TURN_RIGHT++\r.
5. Colocar el dispositivo de manera que se reciba LAYING_DOWN_DOWN++\r, girarlo en torno al eje Z en sentido horario hasta que se reciba LAYING_DOWN_LEFT++\r. **Verificar** que se recibe TURN_RIGHT++\r.

Apéndice B. Protocolos de test y resultados

- Colocar el dispositivo de manera que se reciba `LAYING_DOWN_LEFT++\r`, girarlo en torno al eje Z en sentido horario hasta que se reciba `LAYING_DOWN_UP++\r`. **Verificar** que se recibe `TURN_RIGHT++\r`.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
3	Se recibe <code>TURN_RIGHT++\r</code>	Se recibe <code>TURN_RIGHT++\r</code>	Sí
4	Se recibe <code>TURN_RIGHT++\r</code>	Se recibe <code>TURN_RIGHT++\r</code>	Sí
5	Se recibe <code>TURN_RIGHT++\r</code>	Se recibe <code>TURN_RIGHT++\r</code>	Sí
6	Se recibe <code>TURN_RIGHT++\r</code>	Se recibe <code>TURN_RIGHT++\r</code>	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.4.2. Giro de 90° hacia la izquierda [Fw]

No se redacta protocolo de test.

Test informal: Se realizó un test funcional de este requerimiento con el dispositivo prototipo. Se colocó y posicionó el dispositivo a un usuario, se conectó el dispositivo al PC y se abrió un terminal serial. Luego, se verificó que cada vez que el usuario realiza acostado un giro hacia la izquierda se recibe `TURN_LEFT++\r`.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.4.3. Acostada [Fw]

No se redacta protocolo de test.

Test informal: Se realizó un test funcional de este requerimiento con el dispositivo prototipo. Se colocó y posicionó el dispositivo a un usuario, se conectó el dispositivo al PC y se abrió un terminal serial. Luego, se verificó que cada vez que el usuario pase de estar acostado a estar parado se reciba en el terminal la hora de la acostada y su duración.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.4.4. Levantada [Fw]

No se redacta protocolo de test.

Test informal: Se realizó un test funcional de este requerimiento con el dispositivo prototipo. Se colocó y posicionó el dispositivo a un usuario, se conectó el

dispositivo al PC y se abrió un terminal serial. Luego, se verificó que cada vez que el usuario pase de estar parado a estar acostado se reciba en el terminal la hora de la acostada y su duración.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.5. Clasificación en niveles de actividad

B.4.7.5.1. Niveles de actividad [Fw]

La prueba de este requerimiento se cubre en los tests B.4.7.5.2, B.4.7.5.3, B.4.7.5.4 y B.4.7.5.5.

B.4.7.5.2. Nivel de actividad 0 [Fw]

No se redacta protocolo de test.

Test informal: Se realizó un test funcional de este requerimiento con el dispositivo prototipo. Se conectó el dispositivo al PC y se abrió un terminal serial. Luego, se seteo el dispositivo en modo Measure y se verificó que cuando LED3 está apagado (LOW ODR) se reciba LEVEL0++\r. Después, se verificó lo mismo cuando LED3 está encendido y M_A está en el rango correspondiente.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.5.3. Nivel de actividad 1 [Fw]

Nota: Para este test, se agrega un breakpoint en la línea 463 del archivo `measure_mode.c` y se utiliza el programador `MSP-FET430UIF` y el entorno de desarrollo `IAR` para editar en ejecución variables en memoria.

Procedimiento:

1. Conectar el dispositivo al PC y abrir un terminal serial.
2. Setear el dispositivo en modo Measure. Mover el dispositivo de manera notoria, verificar que la ejecución se detiene en el breakpoint.
3. Modificar la variable `mean_movavg` con la expresión `threshold_01+1` y continuar la ejecución del programa.
4. **Verificar** que en el terminal serial se recibe LEVEL1++\r.
5. Esperar que la ejecución se detenga de nuevo, modificar la variable `mean_movavg` con la expresión `threshold_12-1` y continuar la ejecución del programa.
6. **Verificar** que en el terminal serial se recibe LEVEL1++\r.

Apéndice B. Protocolos de test y resultados

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Se recibe LEVEL1++\r	Se recibe LEVEL1++\r	Sí
6	Se recibe LEVEL1++\r	Se recibe LEVEL1++\r	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.5.4. Nivel de actividad 2 [Fw]

No se redacta protocolo de test.

Test informal: De igual manera que en el test B.4.7.5.3, se modificó la variable `mean_movavg` con valores entre `threshold_12` y `threshold_23` y se verificó que en el terminal serial se recibe `LEVEL2++\r`.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.5.5. Nivel de actividad 3 [Fw]

No se redacta protocolo de test.

Test informal: De igual manera que en el test B.4.7.5.3, se modificó la variable `mean_movavg` con valores mayores a `threshold_23` y se verificó que en el terminal serial se recibe `LEVEL3++\r`.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.6. Recopilación de estadísticas

B.4.7.6.1. Contador diario de persona parada [Fw]

Procedimiento:

1. Posicionar el dispositivo sobre una superficie de manera que registre “persona parada”.
2. Conectar el dispositivo al PC y abrir el SAD. Configurar la hora del dispositivo en 00:00:00.
3. Setear el dispositivo en modo Measure, dejar el dispositivo quieto sobre la superficie durante 24 horas y finalizar la medida.
4. Conectar el dispositivo al PC y abrir el SAD. Descargar los datos de la medida, **registrar** el tiempo detectado como “persona parada” y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Tiempo persona parada: 86400 ± 100	Tiempo persona parada: 86396	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.2. Contador diario de persona acostada [Fw]

La prueba de este requerimiento se cubre en los tests B.4.7.6.3, B.4.7.6.4, B.4.7.6.5 y B.4.7.6.6.

B.4.7.6.3. Contador diario de persona acostada boca arriba [Fw]

Procedimiento:

1. Posicionar el dispositivo sobre una superficie de manera que registre “persona acostada boca arriba”.
2. Conectar el dispositivo al PC y abrir el SAD. Configurar la hora del dispositivo en 00:00:00.
3. Setear el dispositivo en modo Measure, dejar el dispositivo quieto sobre la superficie durante 24 horas y finalizar la medida.
4. Conectar el dispositivo al PC y abrir el SAD. Descargar los datos de la medida, **registrar** el tiempo detectado como “persona acostada boca arriba” y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Tiempo persona acostada hacia arriba: 86400 ± 100	Tiempo persona acostada hacia arriba: 86397	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.4. Contador diario de persona acostada boca abajo [Fw]

Procedimiento:

1. Posicionar el dispositivo sobre una superficie de manera que registre “persona acostada boca abajo”.
2. Conectar el dispositivo al PC y abrir el SAD. Configurar la hora del dispositivo en 00:00:00.

Apéndice B. Protocolos de test y resultados

3. Setear el dispositivo en modo Measure, dejar el dispositivo quieto sobre la superficie durante 24 horas y finalizar la medida.
4. Conectar el dispositivo al PC y abrir el SAD. Descargar los datos de la medida, **registrar** el tiempo detectado como “persona acostada boca abajo” y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Tiempo persona acostada hacia abajo: 86400 ± 100	Tiempo persona acostada hacia abajo: 86398	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.5. Contador diario de persona acostada sobre hombro derecho [Fw]

Procedimiento:

1. Posicionar el dispositivo sobre una superficie de manera que registre “persona acostada sobre hombro derecho”.
2. Conectar el dispositivo al PC y abrir el SAD. Configurar la hora del dispositivo en 00:00:00.
3. Setear el dispositivo en modo Measure, dejar el dispositivo quieto sobre la superficie durante 24 horas y finalizar la medida.
4. Conectar el dispositivo al PC y abrir el SAD. Descargar los datos de la medida, **registrar** el tiempo detectado como “persona acostada sobre hombro derecho” y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Tiempo persona acostada hacia derecha: 86400 ± 100	Tiempo persona acostada hacia derecha: 86399	Sí

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.6. Contador diario de persona acostada sobre hombro izquierdo [Fw]

Procedimiento:

1. Posicionar el dispositivo sobre una superficie de manera que registre “persona acostada sobre hombro izquierdo”.

B.4. Dispositivo

2. Conectar el dispositivo al PC y abrir el SAD. Configurar la hora del dispositivo en 00:00:00.
3. Setear el dispositivo en modo Measure, dejar el dispositivo quieto sobre la superficie durante 24 horas y finalizar la medida.
4. Conectar el dispositivo al PC y abrir el SAD. Descargar los datos de la medida, **registrar** el tiempo detectado como “persona acostada sobre hombro izquierdo” y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Tiempo persona acostada hacia izquierda: 86400 ± 100	Tiempo persona acostada hacia izquierda: 86399	Sí

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.7. Contador diario de persona en posición intermedia [Fw]

Procedimiento:

1. Posicionar el dispositivo sobre una superficie de manera que registre “persona en posición intermedia”.
2. Conectar el dispositivo al PC y abrir el SAD. Configurar la hora del dispositivo en 00:00:00.
3. Setear el dispositivo en modo Measure, dejar el dispositivo quieto sobre la superficie durante 24 horas y finalizar la medida.
4. Conectar el dispositivo al PC y abrir el SAD. Descargar los datos de la medida, **registrar** el tiempo detectado como “persona en posición intermedia” y **verificar** que esté dentro del rango esperado.

Resultados esperados y obtenidos:

Paso	Resultado esperado	Resultado obtenido	¿Aceptable? Sí/No
4	Tiempo posición intermedia: 86400 ± 100	Tiempo posición intermedia: 86398	Sí

Unidad testada: Prototipo 01

Conclusión: **PASA**

Apéndice B. Protocolos de test y resultados

B.4.7.6.8. Contador diario de giros de 90° hacia la derecha [Fw]

No se redacta protocolo de test.

Test informal: Se le colocó el dispositivo a una persona y se le realizó el posicionamiento. Se ingresó al modo Measure y se le pidió al usuario que haga la siguiente secuencia de posiciones acostado con la boca hacia arriba → acostado sobre el hombro derecho → acostado con la boca hacia abajo → acostado sobre el hombro izquierdo → acostado con la boca hacia arriba.

Se conectó el dispositivo a la PC y se corroboró que el dispositivo registró 4 giros hacia la derecha.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.6.9. Contador diario de giros de 90° hacia la izquierda [Fw]

No se redacta protocolo de test.

Test informal: Se le colocó el dispositivo a una persona y se le realizó el posicionamiento. Se ingresó al modo Measure y se le pidió al usuario que haga la siguiente secuencia de posiciones acostado con la boca hacia arriba → acostado sobre el hombro izquierdo → acostado con la boca hacia abajo → acostado sobre el hombro derecho → acostado con la boca hacia arriba.

Se conectó el dispositivo a la PC y se corroboró que el dispositivo registró 4 giros hacia la izquierda.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.6.10. Contador diario de nivel de actividad 0 [Fw]

No se redacta protocolo de test.

Test informal: Se le colocó el dispositivo a una persona y se le realizó el posicionamiento. Se ingresó al modo Measure con el dispositivo conectado al PC. Se le pidió al usuario que se mantuviera quieto durante 10s mientras se verificaba que la magnitud que se compara con los umbrales no supere **threshold_01**.

Luego se le consultó al dispositivo el contador de nivel de actividad 0 y se corroboró que el dispositivo responda de acuerdo a lo esperado.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.11. Contador diario de nivel de actividad 1 [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.7.6.10 se le pidió al usuario que caminara durante 10s mientras otra persona controlaba que la magnitud a comparar con los umbrales se mantuviera entre `threshold_01` y `threshold_12`. Luego se le consultó al dispositivo el contador de nivel de actividad 1 y se corroboró que el dispositivo responda de acuerdo a lo esperado.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.12. Contador diario de nivel de actividad 2 [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.7.6.10 se le pidió al usuario que caminara rápidamente durante 10s mientras otra persona controlaba que la magnitud a comparar con los umbrales se mantuviera entre `threshold_12` y `threshold_23`.

Luego se le consultó al dispositivo el contador de nivel de actividad 2 y se corroboró que el dispositivo responda de acuerdo a lo esperado.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.13. Contador diario de nivel de actividad 3 [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.7.6.10 se le pidió al usuario que corriera durante 10s mientras otra persona controlaba que la magnitud a comparar con los umbrales supere `threshold_23`.

Luego se le consultó al dispositivo el contador de nivel de actividad 3 y se corroboró que el dispositivo responda de acuerdo a lo esperado.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.7.6.14. Registro de acostadas [Fw]

No se redacta protocolo de test.

Test informal: Se conectó el dispositivo al PC y se configuró la hora actual. Luego, se posicionó el dispositivo y se realizaron más de 20 “acostadas”, registrando la hora de la misma y su duración. Finalmente, se descargaron los datos y se

Apéndice B. Protocolos de test y resultados

verificó que se hayan almacenado correctamente (hora y duración) las 20 “acostadas” de mayor duración.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.7.6.15. Registro de levantadas [Fw]

No se redacta protocolo de test.

Test informal: Se conectó el dispositivo al PC y se configuró la hora actual. Luego, se posicionó el dispositivo y se realizaron más de 20 “levantadas”, registrando la hora de la misma y su duración. Finalmente, se descargaron los datos y se verificó que se hayan almacenado correctamente (hora y duración) las 20 “levantadas” de mayor duración.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8. Comunicación con SAD

B.4.8.1. Estado del dispositivo [Fw]

No se redacta protocolo de test.

Test informal: Se colocó el dispositivo en modo Standby y se le envió el comando `G0\r`, se verificó que el dispositivo comenzó a enviar aceleraciones. Luego se colocó el dispositivo en cada uno de los modos de operación restantes, se reenvió el mismo comando y se corroboró que no se recibe nada.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.2. Interfaz de comunicación [Fw]

No se redacta protocolo de test.

Unidad testeada: -

Conclusión: **SIN TESTEAR**

B.4.8.3. Protocolo de comunicación [Fw]

No se redacta protocolo de test.

Test informal: Se conectó el dispositivo y se configuró un terminal serial con los siguientes parámetros: 57600 baudios y 1 bit de parada. Se enviaron cada uno

de los comandos y se corroboró que el dispositivo responde correctamente.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.8.4. Envío periódico de aceleraciones [Fw]

No se redacta protocolo de test.

Test informal: Estando el dispositivo en modo Standby se le envió el comando `GO\r` y se verificó que el dispositivo comience el envío periódico de aceleraciones con el formato `±xxxx, ±yyyy, ±zzzz\r` a una frecuencia de 6,25 Hz.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.8.5. Comandos [Fw]

No se redacta protocolo de test.

Test informal: Se envió cada uno de los comandos de la tabla A.1 y se verifico que el dispositivo **no** responda `Función no encontrada\r`

Unidad testada: Final 01

Conclusión: **PASA**

B.4.8.6. Comando WH [Fw]

No se redacta protocolo de test.

Test informal: Se colocó el dispositivo en modo Standby y se le envió el comando `WH hh mm ss DD MM\r`, se corroboró que se recibe `ERROR\r`. Luego enviar el comando `WH hh mm ss DD MM YYYY\r` donde los argumentos son tomados de otro reloj, se verifica que se recibe `OK\r`.

Se envió el comando `RH\r` en tiempos aleatorios, se corroboró que el dispositivo responde `DD/MM/YYYY hh:mm:ss\r` y que aún esté sincronizado con la fuente de reloj de referencia.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.8.7. Comando RH [Fw]

La prueba de este requerimiento se cubre en el test B.4.8.6.

Apéndice B. Protocolos de test y resultados

B.4.8.8. Comando RB [Fw]

No se redacta protocolo de test.

Test informal: Estando el dispositivo en modo Standby, se envió el comando RB\r mientras se medía la tensión en bornes de las pilas. Se verificó que el dispositivo responde x,xxV\r y que la respuesta es similar a la tensión medida en bornes de las pilas.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.9. Comando GO [Fw]

La prueba de este requerimiento se cubre en el test B.4.8.4.

B.4.8.10. Comando ST [Fw]

No se redacta protocolo de test.

Test informal: Estando el dispositivo enviando aceleraciones periódicamente se envió el comando ST\r y se notó que el dispositivo detiene el envío de aceleraciones.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.11. Comando SO [Fw]

No se redacta protocolo de test.

Test informal: Estando el dispositivo en modo Standby se envió el comando GO\r y el dispositivo comenzó el envío de aceleraciones. Se seteó la frecuencia de envío de aceleraciones en 0,10 Hz con el comando SO 0\r, se verificó que se recibe 1 aceleración cada 10 s. Se reenvió el comando incrementando el argumento y se notó que la frecuencia de los envíos se duplica en cada incremento.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.12. Comando TM [Fw]

No se redacta protocolo de test.

Test informal: Luego de haber ingresado al modo test y estando en modo Standby se envió el comando TM\r. Se verificó que comienza el envío de las acele-

raciones almacenadas con el formato $\pm xxxx$, $\pm yyyy$, $\pm zzzz$ \r.

Unidad testada: Prototipo 01

Conclusión: **PASA**

B.4.8.13. Comando AD [Fw]

No se redacta protocolo de test.

Test informal: Estando el dispositivo en modo Standby se enviaron los siguientes comandos AD $aaaa$ \r, AD $bbbb$ \r, AD $cccc$ \r y se verificó que el dispositivo respondió 1\r luego de cada envío.

Se consultó la tabla de usuarios con el comando SH\r, el dispositivo respondió : 0 0 $aaaa$ \r, 1 0 $bbbb$ \r, 2 0 $cccc$ \r.

Finalmente se envió el comando AD $dddd$ \r y se corroboró que el dispositivo respondió 0\r.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.8.14. Comando DL [Fw]

No se redacta protocolo de test.

Test informal: Luego de haber realizado el test B.4.8.13 se enviaron los siguientes comandos DL 0\r, DL 1\r, DL 2\r y se verificó que el dispositivo respondió 1\r luego de cada envío.

Se consultó la tabla de usuarios con el comando SH\r, se corroboró que el dispositivo no responde nada.

Finalmente se envió el comando DL 0\r y el dispositivo respondió 0\r.

Unidad testada: Final 01

Conclusión: **PASA**

B.4.8.15. Comando SU [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.8.13 se envió el comando SU 1\r y el dispositivo respondió 1\r.

Se consultó la tabla de usuarios y el dispositivo envió las siguientes líneas: 0 0 $aaaa$ \r, 1 1 $bbbb$ \r, 2 0 $cccc$ \r.

Finalmente se envió el comando DL 0\r para borrar el usuario 0 y se intentó setear dicho usuario con el comando SU 0\r, el dispositivo respondió 0\r.

Apéndice B. Protocolos de test y resultados

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.16. Comando SH [Fw]

La prueba de este requerimiento se cubre en el test B.4.8.13.

B.4.8.17. Comando DD [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.8.13 con el usuario 0 activo, se realizó una medida de prueba cronometrando los tiempos en cada posición y contando la cantidad de giros realizados. Se envió el comando DD 0 0 y se verificó que el dispositivo envíe Y\r.

A continuación se corroboró que el dispositivo envíe la hora de comienzo de la medida seguido de cada uno de los contadores separados por \r. Se corroboró que los contadores eran similares a los tiempos medidos.

Finalmente se envió el comando DD 0 1 y se verificó que se recibe N\r.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.18. Comando SD [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.8.17, se envió el comando SD 0 0\r y se recibió 1\r.

A continuación se recibió cada una de las levantadas en el formato [Hora] [Duración]\r y se verificó que coincidieran con el tiempo medido. Se verificó que el dispositivo envíe E\r como fin de mensaje.

Se corroboró que ante cualquier otra elección válida de argumentos el dispositivo responde 0\r.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.19. Comando LY [Fw]

No se redacta protocolo de test.

Test informal: Estando en las condiciones del test B.4.8.17, se envió el comando LY 0 0\r y se recibió 1\r.

A continuación se recibió cada una de las acostadas en el formato [Hora] [Duración]\r y se verificó que coincidieran con el tiempo medido. Se verificó que el dispositivo

envíe `E\r` como fin de mensaje.

Se corroboró que ante cualquier otra elección válida de argumentos el dispositivo responde `0\r`.

Unidad testeada: Final 01

Conclusión: **PASA**

B.4.8.20. Comando TH [Fw]

No se redacta protocolo de test.

Nota: Para este test, se agrega un breakpoint en la línea 316 del archivo `comandos.c` y se utiliza el programador `MSP-FET430UIF` y el entorno de desarrollo `IAR` para observar en ejecución variables en memoria.

Test informal: Estando en modo Standby, se envió el comando `TH 0\r`. Se verificó que el dispositivo envíe `1\r` y quede detenido en el breakpoint. Se observaron las variables `threshold_01`, `threshold_12` y `threshold_23` y se corroboró que tomaron el valor esperado.

Se repitió la prueba con distintos argumentos válidos y se observó que las variables cambiaron de forma coherente. Por último se envió el comando con un argumento inválido y el dispositivo respondió `0\r`.

Unidad testeada: Prototipo 01

Conclusión: **PASA**

B.4.8.21. Comando TP [Fw]

No se redacta protocolo de test.

Test informal: Se accionó el switch ON/OFF para reiniciar el dispositivo y se envió el comando `TP\r`, el dispositivo respondió `0\r`.

Se le colocó el dispositivo a una persona y se le aplicó el algoritmo de posicionamiento.

Se volvió a enviar el comando y se corroboró que se recibe `1\r`.

Unidad testeada: Final 01

Conclusión: **PASA**

B.5. Sujeción

B.5.0.1. Comodidad de la persona [Hw]

No se redacta protocolo de test.

Apéndice B. Protocolos de test y resultados

Test informal: Se le colocó el dispositivo a una persona y se le pidió que se ponga en cada una de las siguientes posiciones:

- Torso en posición vertical.
- Torso en posición horizontal.
- Acostado boca hacia arriba.
- Acostado sobre el hombro derecho.
- Acostado sobre el hombro izquierdo.
- Acostado boca hacia abajo.

Si bien el usuario no manifestó molestia en las 5 primeras posiciones, acostado con la boca hacia abajo sintió presión en la zona donde se encontraba el dispositivo.

Unidad testeada: Final 01

Conclusión: **FALLA**

B.5.0.2. Firmeza respecto a la persona [Hw]

No se redacta protocolo de test.

Test informal: Estando el dispositivo sujeto y ya posicionado, se tomaron fotografías de la ubicación inicial del dispositivo. Posteriormente, el usuario utilizó la caminadora durante 10 min incrementando progresivamente la velocidad. Se comparó la posición actual con la que se registró en las fotografías y se corroboró que la traslación del dispositivo es despreciable.

Unidad testeada: Final 01

Conclusión: **PASA**

B.6. Adquisidor de datos

Los requerimientos asociados al Software Adquisidor de Datos no fueron testeados ni se redactaron protocolos de test para un futuro testeo.

Apéndice C

Estimación de memoria de microcontrolador

C.1. Memoria RAM

C.1.1. Datos estadísticos

Se almacenará en memoria RAM una estructura con los datos de la medida en curso. Una vez finalizada dicha medida, los datos recabados serán volcados en memoria FLASH. En la tabla C.1, se detalla el contenido de la estructura y el espacio de memoria mínimo necesario.

Estadística		Bytes
Hora de comienzo	Segundos: 1 byte	7
	Minutos: 1 byte	
	Hora: 1 byte	
	Día: 1 byte	
	Mes: 1 byte	
	Año: 2 bytes	
Hora de finalización (como <i>offset</i> a partir de hora de arranque)		4
Contador Parado		2
Contador Acostado		2
Contador Acostado - derecha		2
Contador Acostado - izquierda		2
Contador Acostado - arriba		2
Contador Acostado - abajo		2
Contador Intermedio		2
Contador Nivel 0		2
Contador Nivel 1		2
Contador Nivel 2		2
Contador Nivel 3		2
Registro Acostada ($\times 20$)	Hora: 4 bytes	120
	Duración: 2 bytes	
Registro Levantada ($\times 20$)	Hora: 4 bytes	120
	Duración: 2 bytes	
Total		273

Tabla C.1: Estimación de memoria RAM para estadísticas diarias.

C.1.2. Matrices de posicionamiento

Será necesario espacio de memoria para almacenar la matriz de posicionamiento y su inversa, así como una matriz auxiliar para facilitar el cálculo del módulo de los datos de aceleración posicionados.

Cada una de dichas matrices tiene 9 entradas (son 3×3) de *float* (4 bytes). Por lo tanto, se requerirá un espacio de al menos **108 bytes**.

C.1.3. *Buffers*

Serán necesarios *buffers* de recepción y transmisión tanto para la comunicación UART como la comunicación SPI. A su vez, en modo Measure se utilizarán diferentes *buffers* para el procesamiento de las aceleraciones y la posterior detección de posición y actividad del usuario. En la tabla C.2, se listan los *buffers* a utilizar y la cantidad mínima de memoria requerida.

Buffer		Bytes
UART	<i>Buffer</i> de transmisión	1024
	<i>Buffer</i> de recepción	32
SPI	<i>Buffer</i> de transmisión	1024
	<i>Buffer</i> de recepción	16
Procesamiento	<i>Buffer</i> de aceleraciones (últimas 4 aceleraciones en cada eje)	24
	<i>Buffer</i> de aceleraciones DC (últimas 4 aceleraciones DC en cada eje)	24
	<i>Buffer</i> de promedio de módulos (últimos 12 módulos)	48
	<i>Buffer</i> de media móvil	48
Total		2240

Tabla C.2: Estimación de memoria RAM para *buffers*

C.1.4. Variables auxiliares

Se estima que se utilizará un mínimo de **1024 bytes** en variables auxiliares. Se considera en este apartado las *flags*, estados, contadores auxiliares, índices de *buffers*, *buffers* auxiliares, variables de debuggeo y demás.

C.1.5. Estimación final

En la tabla C.3 se resume los resultados alcanzados previamente. Se concluye que será necesario un mínimo de 3,6 kB de memoria RAM.

Datos estadísticos	273 B
Matrices de posicionamiento	108 B
<i>Buffers</i>	2240 B
Variables auxiliares	1024 B
Estimación final	3645 B

Tabla C.3: Resumen de estimación de uso de memoria RAM.

C.2. Memoria FLASH

C.2.1. Datos estadísticos

Una vez finalizada una medida, se volcarán los datos estadísticos del cuadro C.1 en memoria FLASH. El microcontrolador deberá tener suficiente espacio en memoria FLASH para almacenar los datos estadísticos de cada medida de cada usuario. Teniendo en cuenta que se permitirá un máximo de 7 medidas y 3 usuarios, se ocupará un mínimo de **6,6 kB** de memoria FLASH.

C.2.2. Modo Test

En modo Test, se almacenarán un máximo de 2048 muestras de aceleración. Cada muestra de aceleración ocupa 6 bytes (2 bytes por eje), por lo que se necesita un espacio de **12 kB** de memoria FLASH.

C.2.3. Memoria de programa

La memoria requerida por el programa es difícil de estimar. De todos modos, para poder hacer el análisis, se realiza la sobre-estimación de **30 kB** de memoria de programa. Se espera que, una vez finalizado, se obtenga un programa de menor tamaño.

C.2.4. Inicialización de datos y constantes

Para los datos que requieren ser inicializados en el arranque y las constantes del programa, se reserva un mínimo de **1 kB** de memoria FLASH.

C.2.5. Estimación final

En la tabla C.4 se resume estimaciones realizadas. Se concluye que será necesario un mínimo de 50 kB de memoria FLASH.

Datos estadísticos	6,6 kB
Modo Test	12 kB
Memoria de programa	30 kB
Inicialización de datos y constantes	1 kB
Estimación final	50 kB

Tabla C.4: Comparación de estimación de uso de memoria FLASH.

Apéndice D

Resumen de acelerómetros considerados

ACELERÓMETROS ANALÓGICOS

Fabricante	Modelo	Rango	Sensibilidad	Noise ($\mu\text{g}/\sqrt{\text{Hz}}$)	Bandwidth	Alimentación	Consumo
Analog Devices	ADXL327	2g	420mV/g	250	X, Y - 1.6kHz Z - 550Hz	1,8 - 3,6V	350uA @3V
NXP	FXLN8361Q	2g	229mV/g	130 (XY) 200 (Z)	X, Y - 1,1kHz Z - 600Hz	1,71 - 3,6V	180uA
NXP	FXLN8371Q	2g	229mV/g	200 (XY) 280 (Z)	X, Y - 2,7kHz Z - 600Hz	1,71 - 3,6V	180uA
STMicro	LIS344ALH	2g, 6g	Vdd/5	50 @3,3V,2g	1,5kHz	2,4 - 3,6V	680uA
Kionix	KXTC9-2050	2g	660mV/g	125	50Hz	1,8 - 3,6V	Normal: 240uA Standby: 5uA

Figura D.1: Comparación de acelerómetros analógicos

Apéndice D. Resumen de acelerómetros considerados

ACELERÓMETROS DIGITALES

Fabricante	Modelo	Rango	Resolución	Sensibilidad	Noise (ug/vHz)	Output Data Rate (max)	Interfaz	Alimentación	Consumo	Características
Analog Devices	ADXL312	1.5g, 3g, 6g, 12g	1.5g - 10 bits 4g - 11 bits 3g - 11 bits	1.5g - 3mg/LSB 3g - 6mg/LSB	340u	3.2kHz	SPI I2C	2V - 3,6V	170uA ODR=100Hz 55uA ODR=10Hz 0.1uA standby	FIFO buffer Standby mode
	ADXL313	0.5g, 1g, 2g, 4g	0.5g - 10 bits 1g - 11 bits 2g - 12 bits	1mg/LSB	150u	3.2kHz	SPI I2C	2V - 3,6V	170uA ODR=100Hz 55uA ODR=10Hz 0.1uA standby	FIFO buffer Standby mode
	ADXL345	2g, 4g, 8g, 16g	2g - 10 bits 4g - 11 bits	4mg/LSB	420u	3.2kHz	SPI I2C	2V - 3,6V	140uA ODR=100Hz 30uA ODR=10Hz 0.1uA standby	FIFO buffer Standby mode
	ADXL346	2g, 4g, 8g, 16g	2g - 10 bits 4g - 11 bits	4mg/LSB	420u	3.2kHz	SPI I2C	1,7 - 2,75V	140uA ODR=100Hz 30uA ODR=10Hz 0.1uA standby	FIFO buffer Standby mode
	ADXL362	2g, 4g, 8g	12 bits	2g - 1mg/LSB 4g - 2mg/LSB	175u	400Hz	SPI	1,6 - 3,5V	Normal: 1.8uA Low noise: 3.3uA Ultralow noise: 13uA Standby: 0.01uA	FIFO buffer Standby mode
NXP	FXLS8471Q	2g, 4g, 8g	14 bits	2g - 0.25mg/LSB 4g - 0.5mg/LSB	126u	800Hz	SPI I2C	1,95 - 3,6V	35uA ODR=50Hz 130uA ODR=200Hz 240uA ODR=800Hz	FIFO buffer Standby mode
Bosch	BMA280	2g, 4g, 8g, 16g	14 bits	2g - 4096 LSB/g 4g - 2048 LSB/g	120u	2kHz	SPI I2C	1,62 - 3,6V	Normal: 130uA Standby: 62uA	FIFO buffer Standby mode
	BMA222E	2g, 4g, 8g, 16g	8 bits	2g - 64 LSB/g 4g - 32 LSB/g	600u	2kHz	SPI I2C	1,62 - 3,6V	Normal: 130uA Standby: 62uA	FIFO buffer Standby mode
	BMA250E	2g, 4g, 8g, 16g	10 bits	2g - 256 LSB/g 4g - 128 LSB/g	400u	2kHz	SPI I2C	1,62 - 3,6V	Normal: 130uA Standby: 62uA	FIFO buffer Standby mode
	BMA255	2g, 4g, 8g, 16g	12 bits	2g - 1024 LSB/g 4g - 512 LSB/g	150u	2kHz	SPI I2C	1,62 - 3,6V	Normal: 130uA Standby: 62uA	FIFO buffer Standby mode
STMicro	LIS2HH12	2g, 4g, 8g	16 bits	2g - 61ug/LSB 4g - 122ug/LSB		800Hz	SPI I2C	1,71 - 3,6V	180uA ODR=100Hz 5uA power-down	FIFO buffer
	LIS2DS12	2g, 4g, 8g, 16g	16 bits	2g - 61ug/LSB 4g - 122ug/LSB	2g - 120u 4g - 150u	6.4kHz	SPI I2C	1,62 - 1,98V	Normal: 150uA Power-down: 0.7uA	FIFO buffer
	LIS2DH12	2g, 4g, 8g, 16g	12 bits	2g - 4mg/LSB 4g - 8mg/LSB	220u	5.3kHz	SPI I2C	1,71 - 3,6V	11uA ODR=50Hz 2uA ODR=1Hz 0.5uA standby	FIFO buffer Standby mode
	LIS2DE12	2g, 4g, 8g, 16g	8 bits	2g - 15.6mg/LSB 4g - 31.2mg/LSB		5.3kHz	SPI I2C	1,71 - 3,6V	6uA ODR=50Hz 0.5uA power-down	
Kionix	KX124-1051	2g, 4g, 8g	16 bits	2g - 61ug/LSB 4g - 122ug/LSB		25.6kHz	SPI I2C	1,71 - 3,6V	Normal: 145uA Standby: 0.9uA	FIFO buffer Standby mode
	KX122-1037	2g, 4g, 8g	16 bits	2g - 61ug/LSB 4g - 122ug/LSB		25.6kHz	SPI I2C	1,71 - 3,6V	Normal: 145uA Standby: 0.9uA	FIFO buffer Standby mode
	KX022-1020	2g, 4g, 8g	16 bits	2g - 61ug/LSB 4g - 122ug/LSB		1.6kHz	SPI I2C	1,71 - 3,6V	Normal: 145uA Standby: 0.9uA	FIFO buffer Standby mode

Figura D.2: Comparación de acelerómetros digitales

Apéndice E

Instalación de los drivers del dispositivo

En Windows 7 o posterior:

En caso de contar con una conexión a Internet, Windows instala los *drivers* necesarios automáticamente. En caso contrario, se pueden instalar manualmente siguiendo los siguientes pasos.

1. Descargar los drivers del sitio web de FTDI: <http://www.ftdichip.com/Drivers/VCP.htm>. Descomprimir el archivo *.zip* descargado en un directorio.
2. Abrir el Administrador de dispositivos.
3. Conectar el dispositivo a la PC.
4. Identificar el ítem que apareció en la lista del Administrador de dispositivos al conectar el dispositivo. Si aún no se instalaron los *drivers* para la comunicación con el dispositivo, probablemente se encuentre en “Otros dispositivos”, con el nombre “USB-SERIAL ADAPTER”.
5. Hacer click derecho sobre dicho ítem y seleccionar “Actualizar software de controlador”.
6. En la nueva ventana abierta, seleccionar “Buscar software de controlador en el equipo”.
7. Ingresar la ruta al directorio del paso 1 y verificar que la opción “Incluir subcarpetas” esté seleccionada. Luego seleccionar “Siguiente”.
8. Aparecerá un mensaje de “Instalando los controladores” y luego “Windows actualizó correctamente los controladores”. Seleccionar “Cerrar”.
9. En el Administrador de dispositivos, buscar un dispositivo con el nombre “USB Serial Port” bajo la categoría “Otros dispositivos”. Repetir pasos del 5 al 8 con dicho dispositivo.

Apéndice E. Instalación de los drivers del dispositivo

10. En el Administrador de dispositivos, verificar que en la categoría “Controladoras de bus serie universal” figura “USB Serial Adapter” y en la categoría “Puertos (COM y LPT)” figura “USB Adapter Serial Port” con un puerto COM asignado.

Luego de instalados los drivers, se debe modificar el puerto COM asignado al dispositivo siguiendo los pasos a continuación.

1. Conectar el dispositivo y abrir el Administrador de dispositivos.
2. En la categoría “Puertos (COM y LPT)”, buscar el ítem “USB Adapter Serial Port” y realizar click derecho sobre él.
3. En la pestaña “Configuración de puerto” de la nueva ventana, seleccionar “Opciones avanzadas”.
4. Modificar el campo “Número de puerto COM” con la opción “COM9” y seleccionar “Aceptar”.

En Linux:

1. Conectar el dispositivo a la PC.
2. Abrir terminal.
3. Ingresar el comando

```
$ lsusb
```

(debería aparecer en el listado el FTDI con su PID y VID).

4. Ingresar el comando

```
$ dmesg | grep ftdi
```

(no debería aparecer nada).

5. Ingresar los comandos

```
$ cd /etc/udev/rules.d
$ sudo touch 85-ftdi.rules
$ sudo nano 85-ftdi.rules
```

(o abrir con editor de preferencia, por ejemplo `sublime` o `gedit`).

6. Insertar en el archivo lo siguiente:

```
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0403", ATTRS{idProduct}=="6001", GROUP="dialout"
```

```
ACTION=="add", ATTRS{idVendor}=="0403", ATTRS{
  idProduct}=="6001", RUN+="/sbin/modprobe
  ftdi_sio", RUN+="/bin/sh -c 'echo 0403 6001 > /
  sys/bus/usb-serial/drivers/ftdi_sio/new_id'"
```

7. Desconectar el dispositivo de la PC.
8. Reiniciar la PC (o cerrar la sesión).
9. Conectar el dispositivo a la PC.
10. Ingresar el comando

```
$ dmesg | grep ftdi
```

(ahora debería aparecer el FTDI detectado).

11. Ingresar los comandos

```
$ cd /dev/
```

```
$ ls
```

(chequear que aparece un ttyUSB nuevo).

Esta página ha sido intencionalmente dejada en blanco.

Apéndice F

Esquemático del prototipo

Apéndice F. Esquemático del prototipo

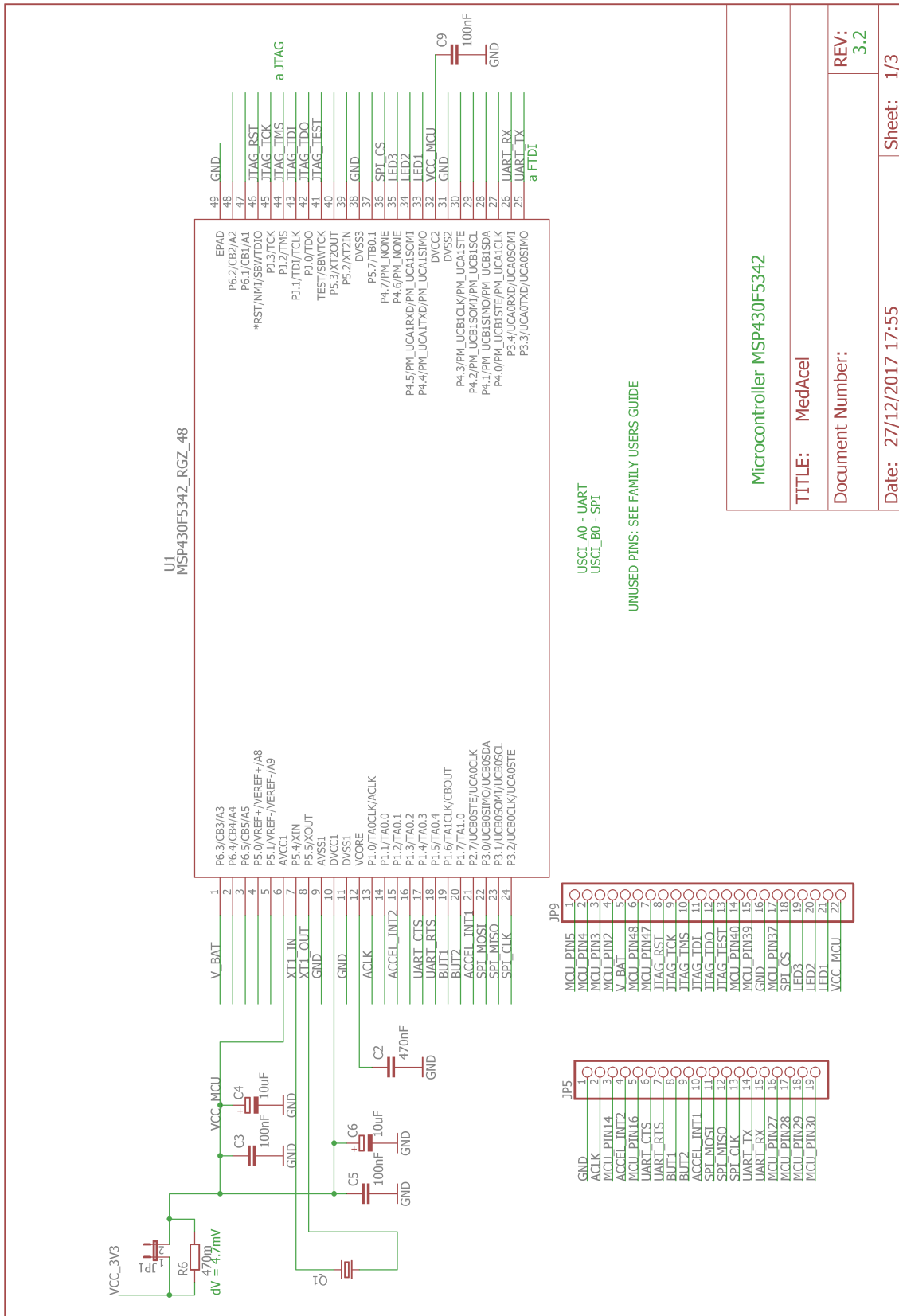


Figura F.1: Esquemático del prototipo - hoja 1

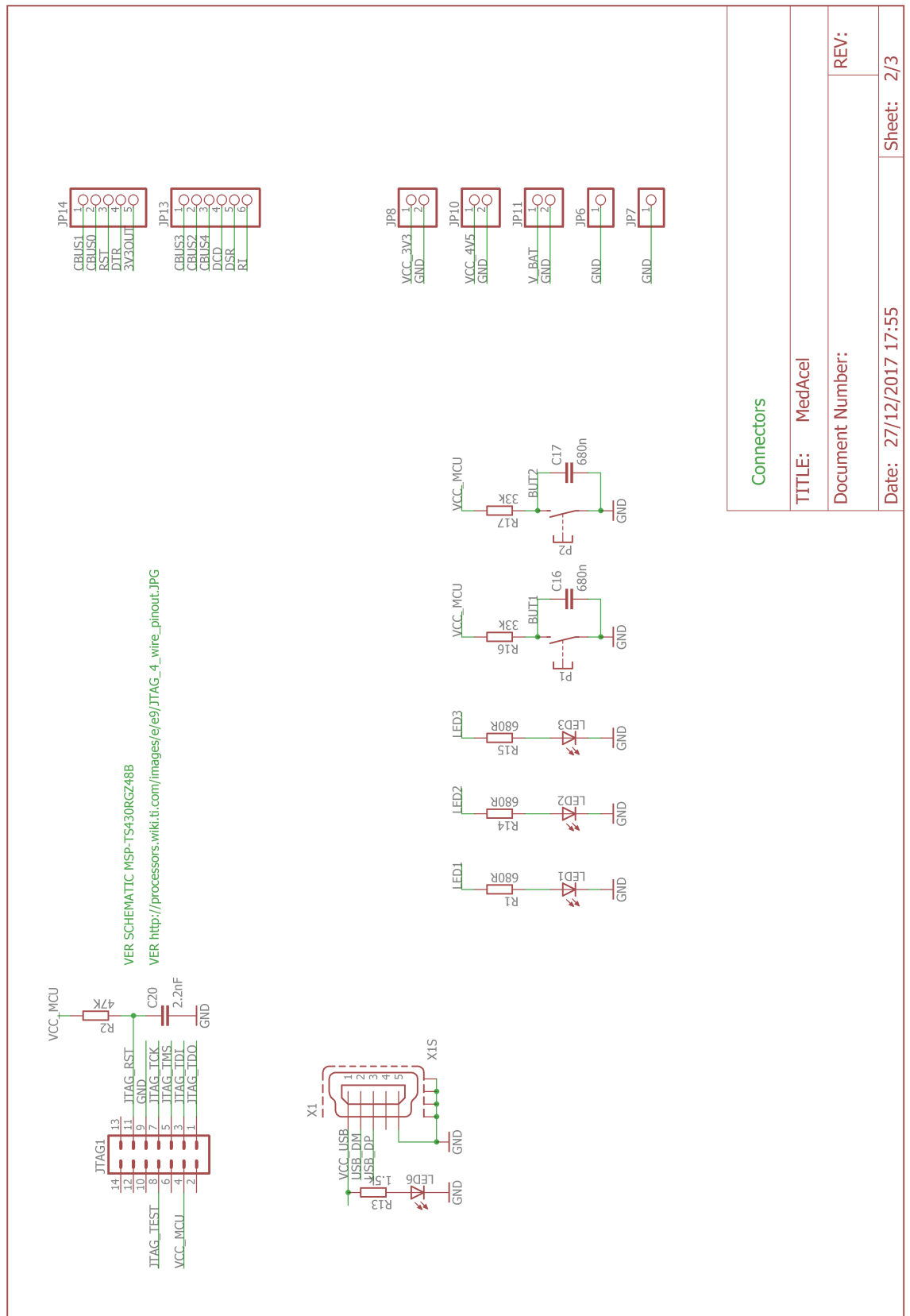
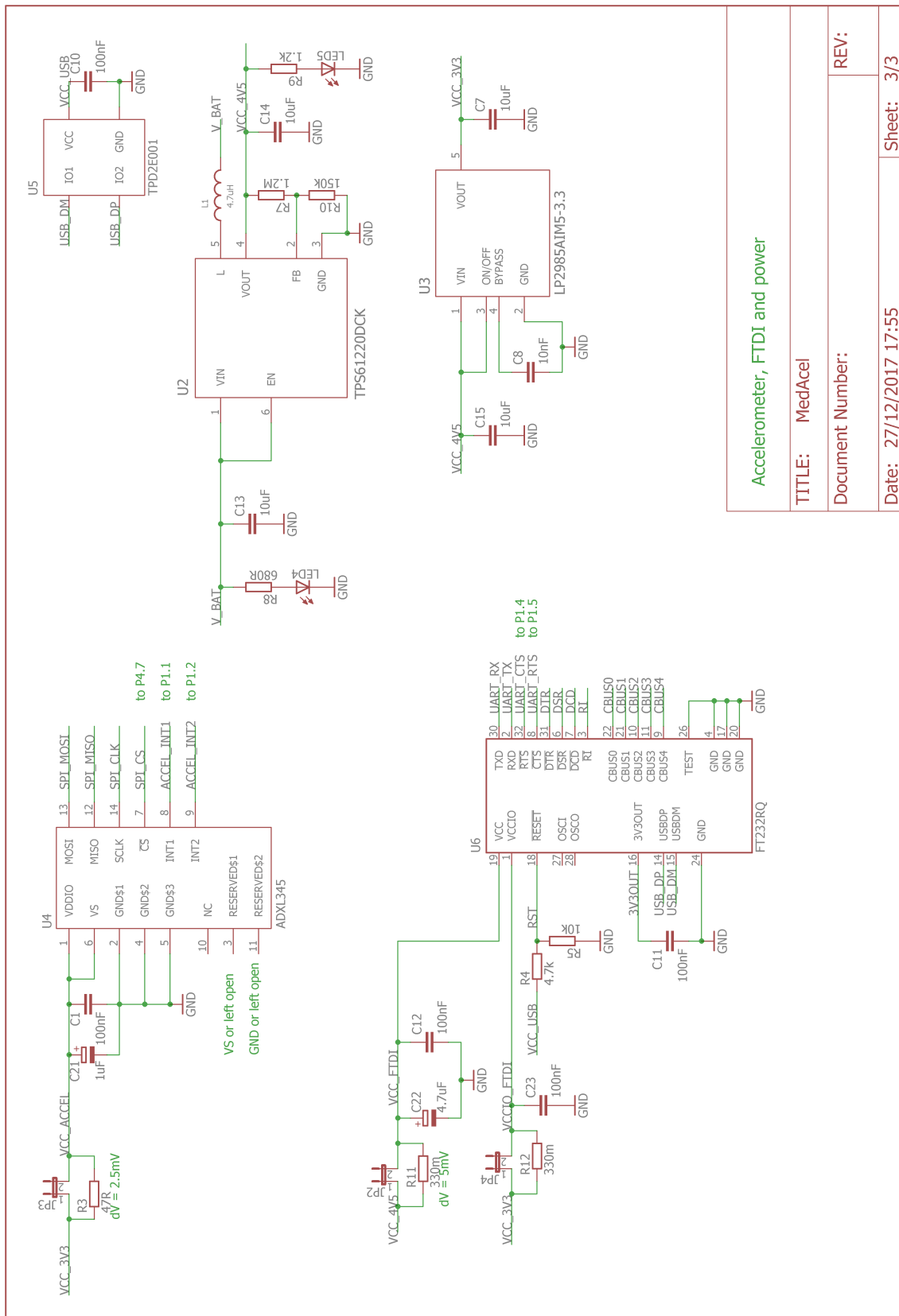


Figura F.2: Esquemático del prototipo - hoja 2

Apéndice F. Esquemático del prototipo



Accelerometer, FTDI and power	
TITLE: MedAccel	REV:
Document Number:	
Date: 27/12/2017 17:55	Sheet: 3/3

Figura F.3: Esquemático del prototipo - hoja 3

Apéndice G

Esquemático del diseño final

Apéndice G. Esquemático del diseño final

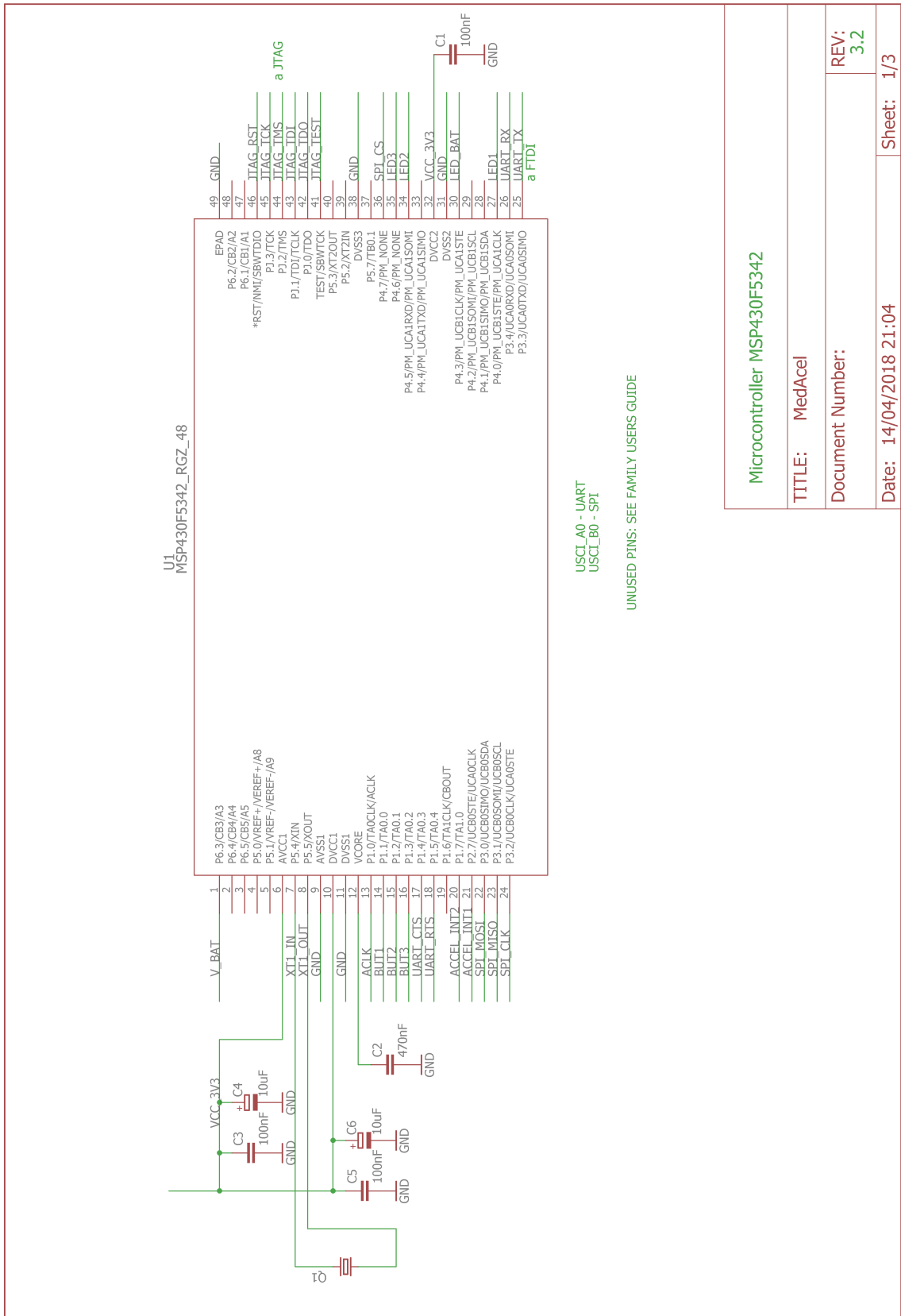


Figura G.1: Esquemático del diseño final - hoja 1

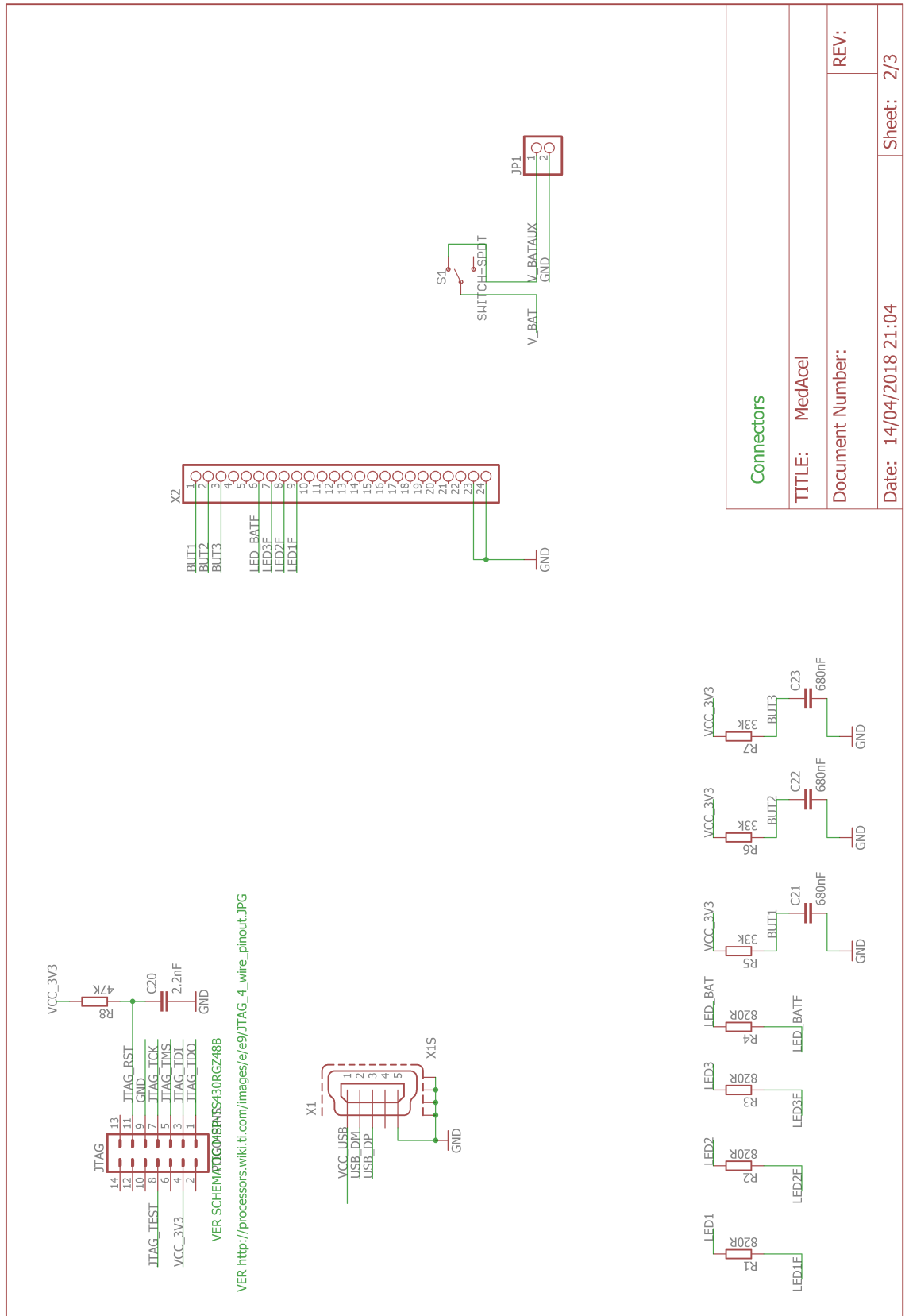
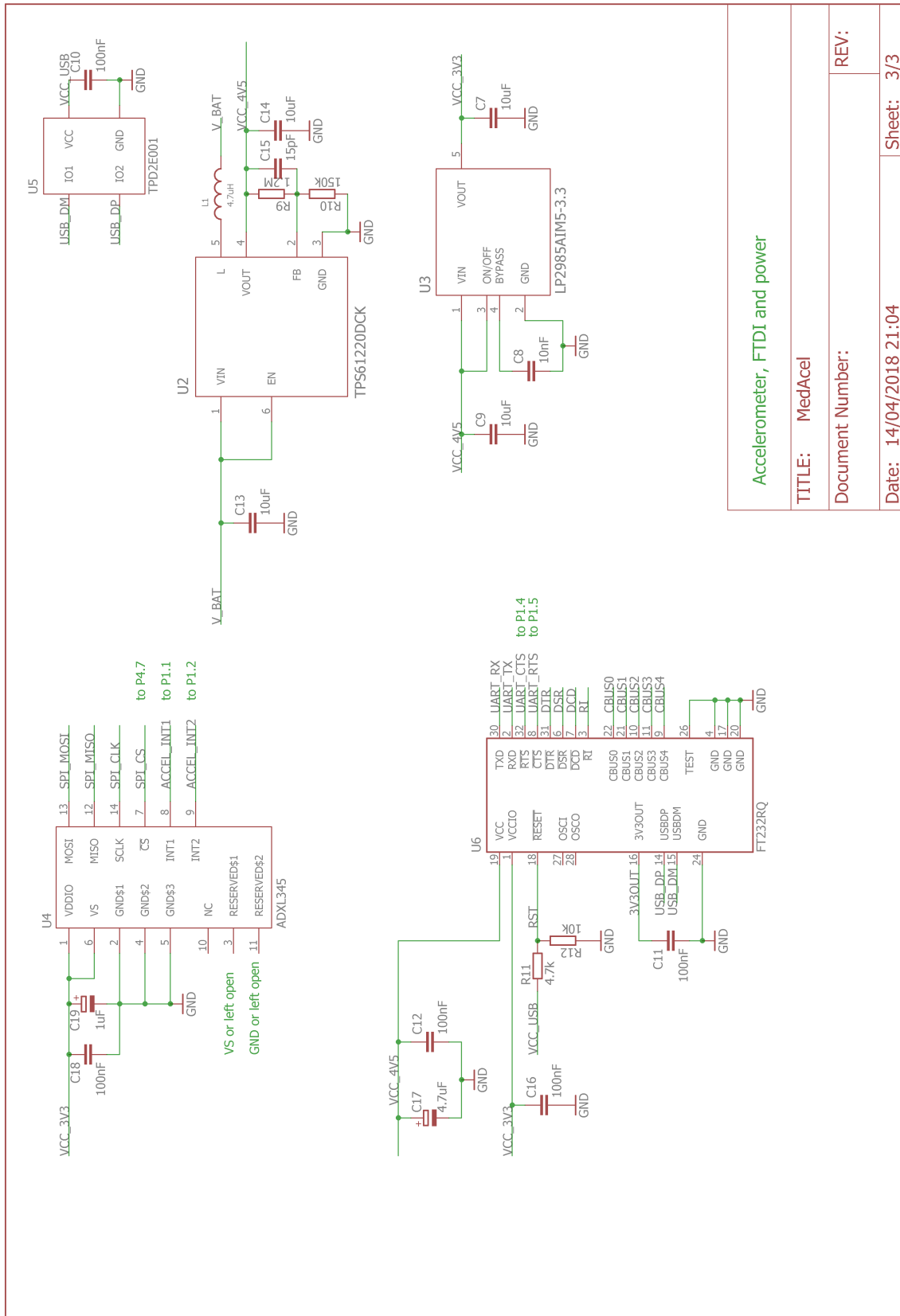


Figura G.2: Esquemático del diseño final - hoja 2

Apéndice G. Esquemático del diseño final



Accelerometer, FTDI and power	
TITLE: MedAccel	REV:
Document Number:	
Date: 14/04/2018 21:04	Sheet: 3/3

Figura G.3: Esquemático del diseño final - hoja 3

Referencias

- [1] Texas Instruments, *MSP430F534x Datasheet*, August 2013. Rev. E.
- [2] Texas Instruments, *MSP430x5xx and MSP430x6xx Family User's Guide*, October 2016. Rev. P.
- [3] Analog Devices, *ADXL345 Datasheet*. Rev. E.
- [4] Texas Instruments, *TPD2E001 Datasheet*, March 2016. Rev. I.
- [5] Future Technology Devices International Limited, *FT232R USB UART IC Datasheet*, 2015. Version 2.13.
- [6] Texas Instruments, *LP2985-N Datasheet*, December 2016. Rev. Y.
- [7] Texas Instruments, *TPS6122x Datasheet*, November 2014. Rev. B.
- [8] "IEEE Standard Test Access Port and Boundary Scan Architecture," *IEEE Std 1149.1-2001*, pp. 1–212, July 2001.
- [9] Texas Instruments, "JTAG (MSP430) - Wiki Page, Texas Instruments." [http://processors.wiki.ti.com/index.php/JTAG_\(MSP430\)](http://processors.wiki.ti.com/index.php/JTAG_(MSP430)). Online; accessed: 6-April-2018.
- [10] "Why does USB 2.0 need Circuit Protection?," Application Note EC616, Littelfuse.
- [11] Analog Devices, "Instrumentation: Test and Measurement Methods and Solutions - Design Conference 2013." <https://www.slideshare.net/AnalogDevicesInc/instrumentation-test-and-measurement-methods-and-solutions-design-conference-2013>. Online; accessed: 7-April-2018.
- [12] Analog Devices, "ADXL345 - FAQ / Analog Devices." <http://www.analog.com/en/education/education-library/product-faqs/ADXL345.html>. Online; accessed: 24-January-2018.
- [13] "Parameters and calibration of a low-g 3-axis accelerometer," Application Note AN4508, STMicroelectronics, June 2014. Rev. 1.
- [14] "6-point tumble sensor calibration," Design Tip DT0053, STMicroelectronics, December 2015. Rev. 1.

Referencias

- [15] M. Pedley, “High-Precision Calibration of a Three-Axis Accelerometer,” Application Note AN4399, Freescale Semiconductor, October 2015. Rev. 2.0.
- [16] K. Tuck, “Offset calibration of the MMA8450Q,” Application Note AN3916, Freescale Semiconductor, February 2010. Rev. 0.
- [17] C. J. Fisher, “Using an Accelerometer for Inclination Sensing,” Application Note AN-1057, Analog Devices, 2010. Rev. 0.
- [18] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 156–167, Jan 2006.
- [19] Curso Sistemas Embebidos para Tiempo Real, IIE - Fing - UdelaR, “Diagnostics de teórico.” <https://eva.fing.edu.uy/course/view.php?name=sisem#section-1>. Online; accessed: 10-April-2018.
- [20] Texas Instruments, *MSP430F5342 Device Erratasheet*, June 2017. Rev. V.
- [21] OSRAM, *TOPLED Datasheet LG T67K*, December 2015. Version 1.1.
- [22] P. Wilson, *The Circuit Designer’s Companion*. Newnes, third ed., 2012.
- [23] D. Fernandez, “Design considerations for a resistive feedback divider in a DC/DC converter,” Application Note SLYT469, Texas Instruments, 2012.
- [24] OKW, “A9051117 SOFT-CASE M.” <https://www.okw.com/en/Soft-Case/A9051117.htm>. Online; accessed: 2-April-2018.
- [25] ee-equipment, *Real-Time Current Monitor - Product Overview*, 2013. Rev 2.0.
- [26] ee-equipment, *Real-Time Current Monitor - Quick-Start Guide*, 2013. Rev 1.0.
- [27] Energizer, *Energizer AAA Alkaline E92 - Product Datasheet*. Version 1103-M.

Índice de tablas

4.1. Salida del acelerómetro \mathbf{a}_{meas} para \mathbf{a}_{case} dado	20
4.2. Calibración del acelerómetro	29
4.3. Posicionamiento del acelerómetro: ortogonalidad de los ejes	41
5.1. Vector $-\mathbf{a}_g$ en función de la posición de la persona	50
5.2. Posiciones de la persona	50
6.1. Indicador del modo activo	59
6.2. Calibración del RTC	63
6.3. RTC calibrado	64
6.4. Resumen de <i>deadlines</i> de tareas de modo Measure	66
6.5. Ciclos y tiempo de ejecución de cada tarea de modo Measure	67
6.6. Comandos aceptados por el intérprete de comandos	68
6.7. Comparación de la memoria usada con la estimada y la disponible	68
6.8. Mapa de memoria del microcontrolador	69
8.1. Consumo de cada sub-circuito a partir de las respectivas hojas de datos	88
10.1. Consumo medido para las distintas interrupciones en modo Measure	96
10.2. Cálculo del consumo medio total del dispositivo en modo Measure para LOW ODR	97
10.3. Cálculo del consumo medio total del dispositivo en modo Measure para HIGH ODR	97
11.1. Prueba de detección de posición realizada el 29/04/2018 con el dispositivo final	100
11.2. Prueba de detección de posición realizada el 29/04/2018 con el dispositivo final: resultados esperados y obtenidos	102
A.1. Comandos del SAD	128
A.2. Correspondencia entre comando y frecuencia	129
B.1. Requerimientos con protocolo de test redactado y con test realizado	136
C.1. Estimación de memoria RAM para estadísticas diarias.	178
C.2. Estimación de memoria RAM para <i>buffers</i>	179

Índice de tablas

C.3. Resumen de estimación de uso de memoria RAM.	179
C.4. Comparación de estimación de uso de memoria FLASH.	180

Índice de figuras

1.1. Diagrama de bloques del sistema	2
3.1. Diagrama de bloques del sistema	8
3.2. Aplicación típica del <i>step-up</i> TPS6122x	11
3.3. Aplicación del regulador lineal LP2985	11
3.4. Aplicación del acelerómetro ADXL345	12
3.5. Aplicación del integrado FT232R en configuración <i>Self Powered</i> . .	13
3.6. Esquemático asociado al conector JTAG	14
3.7. Aplicación típica del chip TPD2E001 de protección contra descargas electrostáticas	14
4.1. Vista superior del circuito integrado de los acelerómetros MEMS de la familia ADXL de <i>Analog Devices</i>	18
4.2. Giros de 90° del acelerómetro.	21
4.3. Movimiento del acelerómetro en cada eje.	22
4.4. Caída del acelerómetro.	23
4.5. Prueba de la calibración realizada	31
4.6. Aceleración a_x del eje x en función del ángulo θ	31
4.7. Aceleración a_y del eje y en función del ángulo θ	32
4.8. Aceleración a_z del eje z en función del ángulo θ	32
4.9. Módulo de la aceleración $\ \mathbf{a}\ _2 = \sqrt{a_x^2 + a_y^2 + a_z^2}$ en función del ángulo θ	33
4.10. Ejes relativos a una persona	34
4.11. Posicionamiento “correcto”, a_x , a_y , a_z y $\ \mathbf{a}\ _2$	39
4.12. Posicionamiento “fallido” debido a movimiento de la persona . . .	39
4.13. Posicionamiento “correcto”, a_x , a_y y a_z	40
4.14. Posicionamiento “fallido” debido a la no-ortogonalidad de los ejes x-z	40
5.1. Datos de aceleración en los 3 ejes para persona caminando “lento”	44
5.2. FFT de los datos para persona caminando “lento” (ver 5.1)	44
5.3. Datos de aceleración en los 3 ejes para persona caminando “normal”	45
5.4. FFT de los datos para persona caminando “normal” (ver 5.3) . . .	45
5.5. Datos de aceleración en los 3 ejes para persona caminando “rápido”	46
5.6. FFT de los datos para persona caminando “rápido” (ver 5.5) . . .	46

Índice de figuras

5.7. Datos filtrados (DC y AC) de aceleración en los 3 ejes para persona caminando “normal”	47
5.8. Diagrama de estados asociado a la variación de frecuencia de muestreo de aceleraciones	49
5.9. Clasificación de la posición de la persona según la aceleración $-a_g$	51
5.10. Módulo de la AC de la aceleración en función de la velocidad de la caminadora	53
5.11. Diferencia sucesiva de la aceleración en el eje z en función de la velocidad de la caminadora	54
5.12. Módulo de la aceleración - 1 g en función de la velocidad de la caminadora	54
5.13. Prueba en la caminadora 1	55
5.14. Prueba en la caminadora 2	55
5.15. Prueba en la caminadora 3	56
6.1. Diagrama mostrando los distintos modos de funcionamiento del dispositivo y las transiciones entre los mismos	58
6.2. Diagrama de módulos asociado al <i>firmware</i> del sistema	61
7.1. Ventana principal del adquisidor de datos	71
7.2. Secciones de la ventana principal	72
7.3. Sección de configuración de usuario y descarga de datos	73
7.4. Listado de usuarios (usuario actual en mayúsculas)	74
7.5. Ventana de visualización de datos adquiridos	74
7.6. Sección de consulta y seteo de hora del dispositivo	75
7.7. Ventana de consulta de nivel de batería	76
7.8. Ventana de ingreso de contraseña	77
7.9. Ventana principal con funciones avanzadas	77
8.1. Esquemático asociado al <i>step-up</i>	79
8.2. Esquemático asociado al regulador lineal	80
8.3. Esquemático asociado al acelerómetro	80
8.4. Esquemático asociado al FT232R	81
8.5. Esquemático asociado al conector USB	81
8.6. Esquemático asociado al conector JTAG	82
8.7. Esquemático asociado a la protección anti-estática	82
8.8. Esquemático asociado a los LEDs	82
8.9. Esquemático del PCB asociado a los botones	83
8.10. Esquemático del PCB asociado a los conectores y <i>jumpers</i>	84
8.11. Esquemático del PCB asociado al microcontrolador	85
8.12. <i>Layout</i> del PCB asociado al prototipo realizado	87
8.13. Foto del PCB asociado al prototipo realizado	88
9.1. <i>Pogo pins</i>	90
9.2. Imagen del arte del teclado de membrana	91
9.3. Imagen del modelo 3D realizado ya ensamblado (PCB y caja)	91

Índice de figuras

9.4. Imagen del modelo 3D realizado ya ensamblado (PCB y caja) . . .	92
9.5. Dispositivo colocado en el torso de una persona	92
10.1. <i>Layout</i> del PCB asociado al dispositivo final realizado	94
10.2. Foto del PCB asociado al dispositivo final realizado	95
10.3. Captura del osciloscopio midiendo consumo, con el dispositivo fun- cionando con frecuencia de muestreo alta	96
10.4. Interrupción del LED intermitente indicador de dispositivo encen- dido en modo Measure	96
11.1. Datos obtenidos luego de la prueba de detección de posición	101
11.2. Datos obtenidos luego de utilizarlo de día	102
11.3. Datos obtenidos luego de utilizarlo de noche	103
A.1. Ejes relativos a una persona	109
D.1. Comparación de acelerómetros analógicos	181
D.2. Comparación de acelerómetros digitales	182
F.1. Esquemático del prototipo - hoja 1	188
F.2. Esquemático del prototipo - hoja 2	189
F.3. Esquemático del prototipo - hoja 3	190
G.1. Esquemático del diseño final - hoja 1	192
G.2. Esquemático del diseño final - hoja 2	193
G.3. Esquemático del diseño final - hoja 3	194

Esta es la última página.
Compilado el domingo 16 septiembre, 2018.
<http://iie.fing.edu.uy/>