# Universidad Católica de Córdoba

# Facultad de Ingeniería



# Trabajo Final:

# “Implementación de un registro de datos de un acelerómetro en una memoria microSD”

Autores: Lovera, Francisco Santiago

Venturi, Paulo Marco

Docente: Ing. Zaninetti, Walter

Fecha:

**Agradecimientos**

# Resumen

El trabajo final consiste en diseñar dos pulseras que recolectan datos del movimiento de ambas manos de un usuario escalando. La información del accionar del usuario luego se transfiere a una computadora donde dichos datos se procesan. Esta información es útil tanto para determinar la fuerza, estabilidad, control y velocidad del escalador como para establecer si la técnica empleada por el usuario fue correcta y en caso que no lo fuera, brindar instrucciones para mejorar el rendimiento. Desde el punto de vista del hardware, cada pulsera contiene un micro controlador, un acelerómetro de 3 ejes, una memoria microSD, una ficha miniUSB y una batería de litio polímero. (*Ver figura 1*.)

Se ha llevado a cabo el diseño y la construcción de un firmware que utiliza un bus de SPI para la transferencia de datos entre el acelerómetro y el micro controlador y a su vez se utiliza otro puerto para transferir estos datos a la memoria. Además, se debe respetar y utilizar el protocolo de USB para intercambiar datos con la computadora.

Posteriormente, se diseñó una carcasa de aluminio a medida con el fin de proteger estas pulseras y que sean ergonométricas al escalador ya que no deben interferir en su rendimiento.

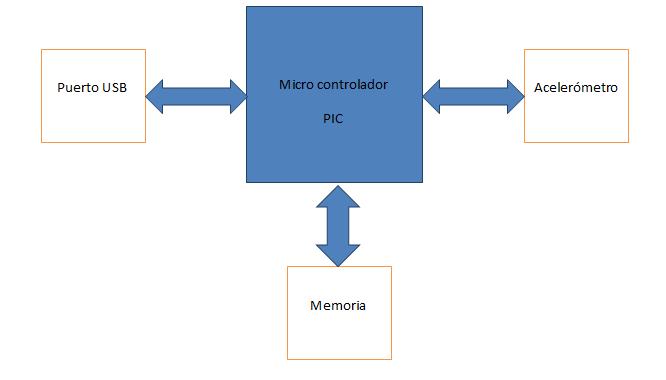


Figura 1

Índice

[Universidad Católica de Córdoba 1](#_Toc398711242)

[Facultad de Ingeniería 1](#_Toc398711243)

[Trabajo Final: 1](#_Toc398711244)

[“Implementación de un registro de datos de un acelerómetro en una memoria microSD” 1](#_Toc398711245)

[Resumen 2](#_Toc398711246)

[Introducción 5](#_Toc398711247)

[Objetivos del proyecto 5](#_Toc398711248)

[Marco teórico 6](#_Toc398711249)

[Acelerómetro ANALOG DEVICES ADXL345 6](#_Toc398711250)

[Micro controlador MICROCHIP PIC24FJ256GB106 6](#_Toc398711251)

[Real time clock calendar (RTCC) 8](#_Toc398711252)

[Analog to digital converter (ADC) 9](#_Toc398711253)

[Serial Peripheral Interface (SPI) 10](#_Toc398711254)

[IO ports 11](#_Toc398711255)

[Power saving features (SLEEP) 14](#_Toc398711256)

[USB 14](#_Toc398711257)

[Secure Digital (SD) Memory 14](#_Toc398711258)

[Implementación del Hardware 14](#_Toc398711259)

[Diseño de Esquemático 14](#_Toc398711260)

[Diseño Digital 14](#_Toc398711261)

[Diseño de Circuito de Carga de Batería 15](#_Toc398711262)

[Diseño del PCB 15](#_Toc398711263)

[Diseño de la Carcasa 15](#_Toc398711264)

[Implementación del Firmware 15](#_Toc398711265)

[MDD File System Library 16](#_Toc398711266)

[USB Library 16](#_Toc398711267)

[Periferical Library 16](#_Toc398711268)

[Driver Acelerómetro ADXL345 16](#_Toc398711269)

[Firmware de Aplicación 20](#_Toc398711270)

[Diagramas de Flujo 20](#_Toc398711271)

[Descripción de Código 20](#_Toc398711272)

[Archivo .CSV 21](#_Toc398711273)

[Problemas 21](#_Toc398711274)

[Correcciones de PCB 21](#_Toc398711275)

[Disponibilidad en el Mercado Local 21](#_Toc398711276)

[Impacto Ambiental 21](#_Toc398711277)

[Responsabilidad Social 21](#_Toc398711278)

[Conclusiones 21](#_Toc398711279)

[Índice de Figuras 21](#_Toc398711280)

[Referencias 21](#_Toc398711281)

[Anexo I Esquemático Completo 21](#_Toc398711282)

[Anexo II PCB Completo 21](#_Toc398711283)

[Anexo III Solid Edge 21](#_Toc398711284)

[Anexo IV Producto final 21](#_Toc398711285)

[Anexo V Organización Industrial 21](#_Toc398711286)

# Introducción

El siguiente trabajo final consiste en el realizado de un par de pulseras capaces de detectar, registrar y guardar las aceleraciones de ambas manos con el fin de interpolar dichos datos externamente y evaluar el accionar de un usuario al escalar.

Cada pulsera debe empezar a recolectar datos cuando se activa un switch al moverlo a la posición “*on”*. A partir de este momento, mediante un diodo led titilando, se le notifica al usuario que se empezó la recolección de datos. A partir de este momento, el usuario debe escalar normalmente y al finalizar la sesión, apaga el dispositivo moviendo el switch a la posición “*off*”. Se verá que el led deja de titilar, indicando que los datos están listos para ser visualizados.

La información coleccionada al escalar puede ser visualizada, tanto mediante USB o sino al extraer la memoria micro SD y conectarla a una computadora. Cada archivo tendrá el nombre de la hora en que se inicializo el archivo.

Previamente, se debe programar la hora actual al conectar un cable USB y ejecutar un script que sincronizara cada pulsera con la hora de la computadora. Al cargar la hora actual en cada pulsera, los dispositivos quedan sincronizados entre sí.

## Objetivos del proyecto

Para su correcto funcionamiento, los objetivos listados a seguir fueron pautados:

* Registrar las aceleraciones a un mínimo de 50 Hertz (Hz)
* Mantener el tiempo de forma precisa y que estén sincronizado entre ambas pulseras
* Realizar un producto resistente a golpes y que sea ergonométrico a la muñeca del usuario
* Minimizar los switches y leds
* Mantener un consumo mínimo y utilizar una batería recargable
* Exponer los datos recolectados en un formato simple y organizado
* Cada dato debe contener las aceleraciones de cada eje (y, z) y la hora en que cada dato fue recolectado con precisión de milisegundos (timestamp)
* Se debe monitorear el nivel de batería para evitar la pérdida de datos debido a una alimentación insuficiente.

# Marco teórico

## Acelerómetro ANALOG DEVICES ADXL345

Se seleccionó el acelerómetro ADXL345 de Analog Devices ya que este acelerómetro puede trabajar fácilmente a 50 Hz, tiene un consumo bajo y se puede programar en modo stand-by que permite reducir aún más el consumo. A su vez, al contar con una FIFO interna, se pueden leer hasta 32 valores de cada eje con un solo comando utilizando un multi-byte read. Por otro lado, la escritura y lectura de los registros del dispositivo está disponible tanto en SPI o I2C. Al contar con un puerto serie, se minimiza las cantidad de pines necesarios para utilizar el integrado y se facilita la recolección de datos. También, este acelerómetro tiene una precisión de 3.9 mg/LSB, es decir que un cambio del bit menos significativo equivale aproximadamente gravedades. Con esta precisión, se puede detectar inclinaciones de menos de un grado. Este número es suficientemente chico para la aplicación presentada en este trabajo final.

El acelerómetro elegido contiene 14 pines, de los cules cuatro son para la comunicación SPI o I2C, 4 son de alimentación y 2 son interrupciones programables. (ver figura 2).

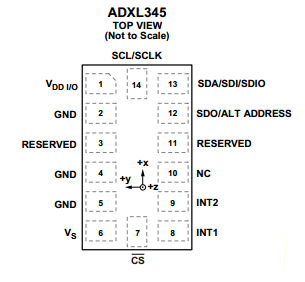


Figura 2

## Micro controlador MICROCHIP PIC24FJ256GB106

El micro controlador PIC24FJ256GB106 es de 16 bits, que puede funcionar a 16Mhz por instrucción. Este dispositivo cuenta con varios periféricos útiles para el proyecto a desarrollar como por ejemplo un conversor analógico a digital (ADC) de 10 bits, un puerto USB, varios timers, 3 módulos SPI y un “*Real Time Clock Calandar*” (RTCC ó RTC). A su vez, tiene una memoria SRAM suficientemente grande como para guardar varias mediciones antes de escribirlas a la memoria microSD. Este integrado trabaja con un voltaje de 3V, y tiene un consumo bajo.

Al visualizar las funciones de cada pin, se puede notar que cada pin puede trabajar como entrada o salida de una gran cantidad de módulos. (Ver figura 3) Esto se debe a que este micro trabaja con pines remappeables. Esto le da al usuario la posibilidad de diseñar PCB más chicos y menos complejos ya que se pueden elegir los pines de forma que se hagan mínimos los cruces de las rutas.

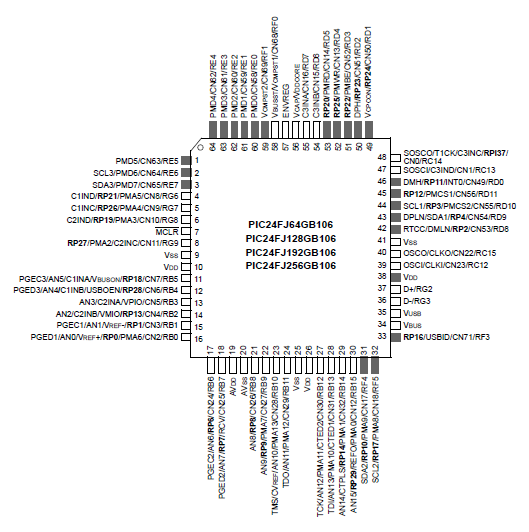


Figura 3

### Real time clock calendar (RTCC)

El Real Time Clock Calandar (RTCC) es un módulo dedicado a mantener la hora y fecha actual con mínimas intervenciones del microprocesador. Con solo agregar un cristal externo de 32.768 kHz y inicializar el modulo, se obtiene una señal cuadrada con pulsos cada medio segundo. Se lo puede calibrar para que sea más preciso utilizando el registro llamado RCFGCAL. Si la frecuencia de trabajo es muy rápida o muy lenta, se debe restar o sumar una frecuencia tal que el cristal oscile a 32.768 kHz. Esta operación se logra al escribir la diferencia entre las frecuencia ideal menos la frecuencia medida multiplicada por 60. (ver ecuación 1) Al escribir este valor en el registro de calibración del RTCC, se adicionaran esta cantidad de pulsos por minuto a la señal generada por el cristal secundario. El bit más significativo del registro indica si el número de pulsos a sumar es positivo o negativo. Por ejemplo, si la frecuencia de trabajo es más rápida que la ideal, el usuario debe escribir un valor negativo en RCFGCAL.

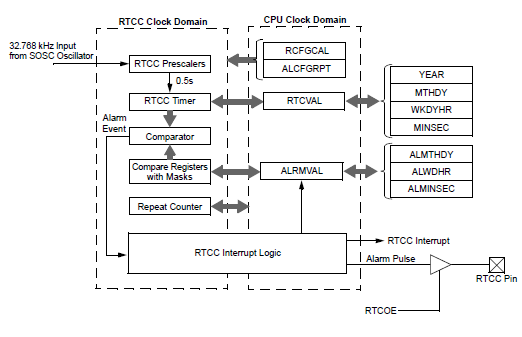


Figura 4

### Analog to digital converter (ADC)

El módulo ADC con el que cuenta este micro controlador trabaja por medio de aproximaciones sucesivas y funciona con una precision de 10 bits. Puede trabajar a una velocidad máxima de 500 ksps, cuenta con 16 entradas multiplexadas y 2 exclusivas para voltajes de referencia. Además tiene la opción de seleccionar la fuente de inicio de conversión. El diagrama de bloques presentado en la figura 5 ilustra el funcionamiento básico del módulo.

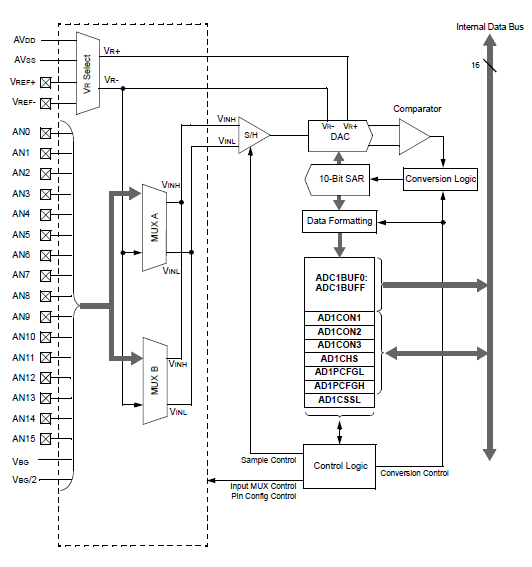


Figura 5

### Serial Peripheral Interface (SPI)

El puerto SPI es un puerto serie que utiliza 4 pines para establecer una comunicación bidireccional con uno o varios periférico. En la figura 5, se puede observar que hay un pin llamado SCK que genera un clock a partir de la frecuencia de ejecución de instrucciones del PIC. La frecuencia de trabajo es configurable ya que el micro controlador cuenta con un pre escalador primario y secundario que permite reducir la frecuencia del reloj de SPI. Hay un pin llamado SS (*slave select*) o CS (*Chip select*) que se utiliza para seleccionar con que periférico el master se quiere comunicar. Los otros dos pines son SDO y SDI que funcionan como la salida de datos y la entrada de datos respectivamente.

Por otro lado, se debe programar el micro controlador de forma que se actualicen los datos en el flanco correspondiente y que se lean los datos en el otro flanco. Por ejemplo, el acelerómetro funciona de forma tal que lee la entrada de datos en el flanco ascendiente y actualiza su salida en el flanco descendiente de SCK. Tanto el Slave, en este caso el acelerómetro, como el Master (el micro controlador) deben trabajar de igual manera sino la interpretación de datos será errónea.

En la figura 5 se ve que hay un solo buffer de entrada y salida. Debido a esto, para poder leer un byte, se debe escribir un byte de dato al bus.

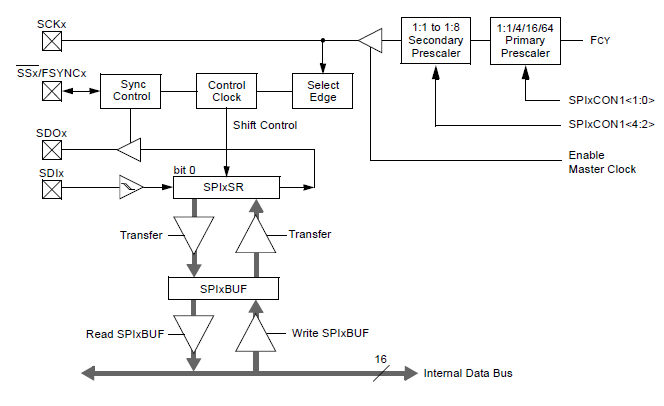


Figura 6

### IO ports

El micro controlador seleccionado tiene muchas entradas y salidas digitales que permiten al diseñador incorporar componentes como pulsadores, leds y relays por ejemplo. Se puede configurar el dispositivo para que los pines funcionen como entrada o salida al escribir un uno o cero lógico en el bit de TRIS correspondiente a dicho pin. A su vez, se puede leer el valor de una entrada verificando el bit de PORT, o se puede forzar una salida de un cierto valor escribiendo en el bit the LATCH correspondiente. La figura 7 muestra el funcionamiento básico de este módulo.

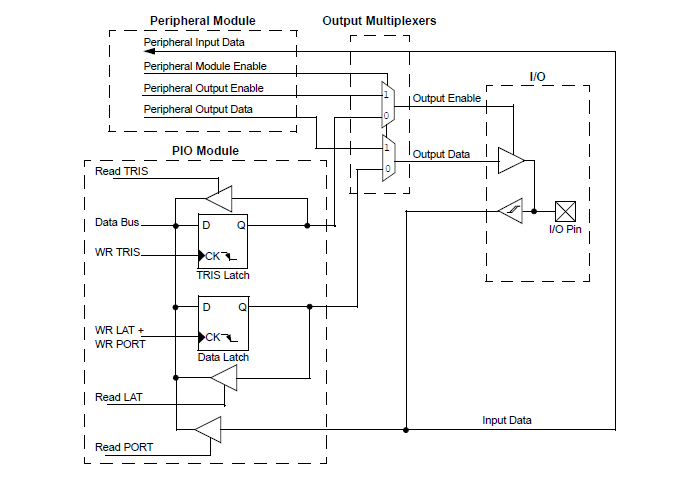


Figura 7

Si se observan los nombres de los pines, se ve que muchos de ellos ademas de servir como una entrada/salida digital, también pueden funcionar cono pines remappeables tanto de entrada como de salida. Dependiendo de como se los programa, el usuario puede utilizar dichos pines para comunicaciones seriales, clock de timers, comparadores y interrupts externos.

Para programar un pin como una entrada, se debe consultar la tabla 1, aquí vemos que al escribir el valor del número de pin remapeable seleccionado en el registro correspondiente a la función que se le quiere otorgar al pin, se mapea dicha función con el pin elegido. Por ejemplo si al pin RP21 se le quiere dar la función de SDI1, en el registro RPINR20, se escribe el valor 21.

Por otro lado, si se quiere programar una salida, al observar la tabla 2, se ve que se debe escribir el numero de la función a programar en el registro RPORxx correspondiente al pin seleccionado. Por ejemplo, se ve que la función 5 corresponde a la entrada de datos serie del puerto UART2. Si se escribe este numero en el byte mas significativo del registro RPOR3, se mapea el U2RX al pin RP7 (ver figura 8).

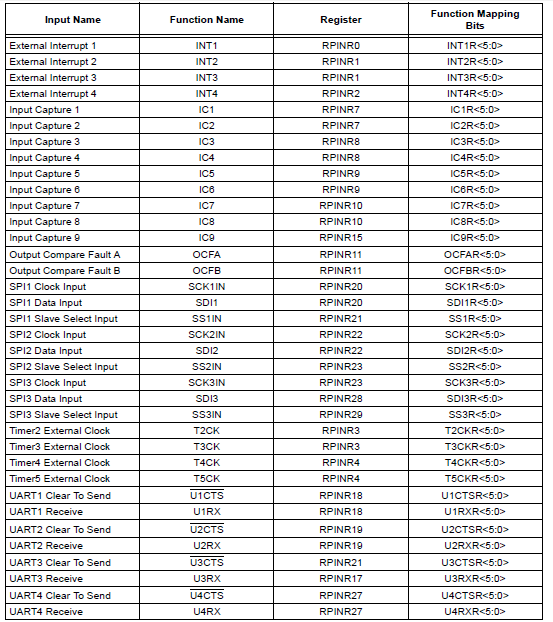


Tabla 1

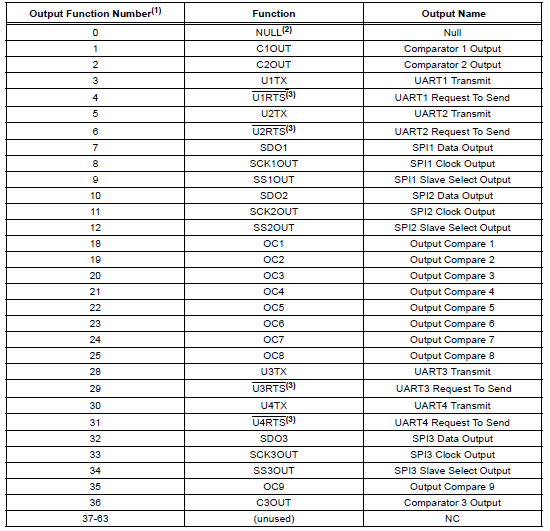


Tabla 2

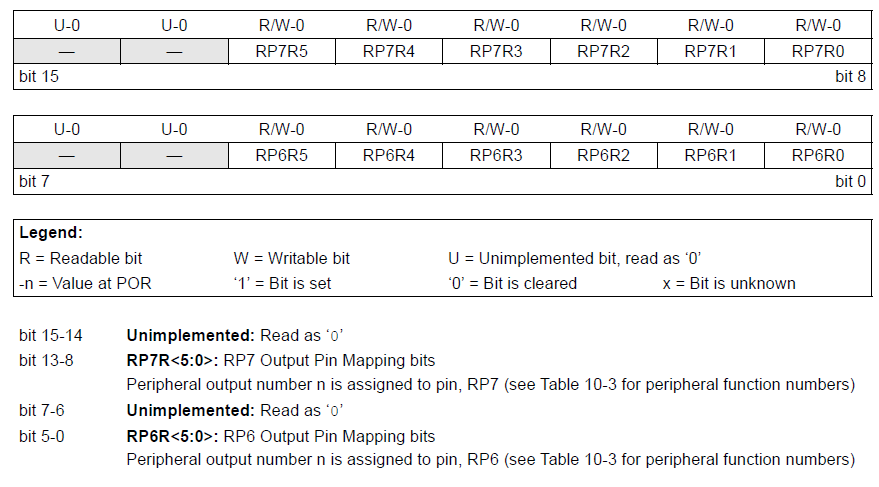


Figura 8

### Power saving features (SLEEP)

Cuando el dispositivo ingresa en modo sleep, el clock es desactivado y el consumo de corriente disminuye al mínimo, siempre y cuando ningún puerto este drenando carga. Además el watch dog timer es puesto a 0, en caso que estuviese activado. Algunos periféricos seguirán operando aun en modo sleep, tales como los puertos entrada\salida, con el fin de detectar cambios en la señal de entrada.

El procesador se despertara en cualquiera de los siguientes casos. Ante cualquier fuente de interrupción, ante un reset del sistema o cuando el watch dog timer de un time out.

Existe uun delay asociado al wake up del modo sleep para distintos modos de osiclacion, que se muestran a continuación en la tabla x.

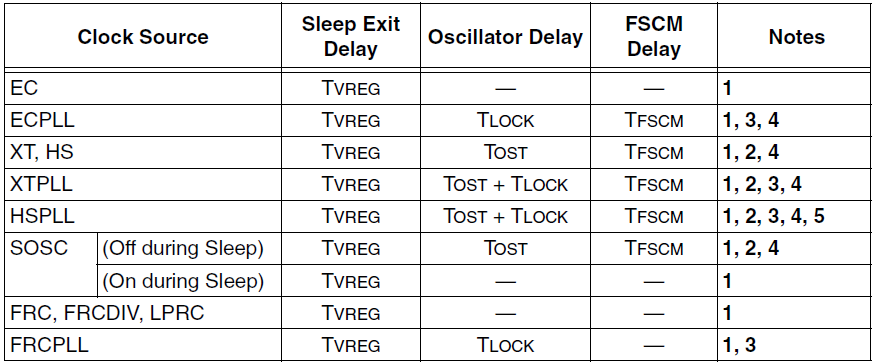


Tabla x

## USB

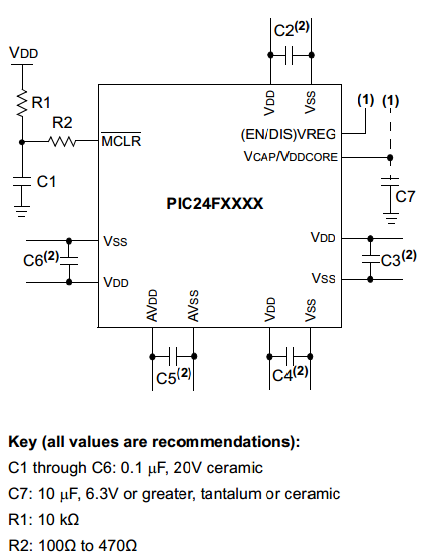
## Secure Digital (SD) Memory

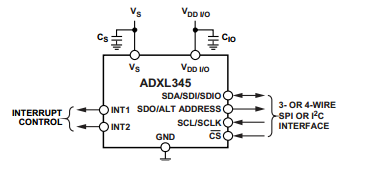
# Implementación del Hardware

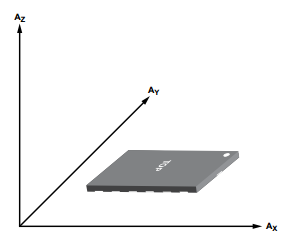
## Diseño de Esquemático

### Diseño Digital

Para el trabajo solicitado, se utilizó este módulo para censar continuamente el nivel de la batería, el cual se mantiene en el rango de 3 a 4,2 volts. Periódicamente el modulo interrumpe y verifica el valor de la batería. Se dejó un margen de 0,5 volts, es decir que cuando el valor de la batería está por debajo de 3,5 volts, el programa ingresa en modo sleep.







### Diseño de Circuito de Carga de Batería

## Diseño del PCB

## Diseño de la Carcasa

# Implementación del Firmware

Una de las principales razones por la cual se seleccionó un micro controlador de marca Microchip es por las librerías de aplicación disponible que permite que el programador se independice del bajo nivel del stack de USB o de Secure Data (SD), por ejemplo. El simple hecho de incluir las librerías, permite llamar funciones que facilitan el uso de

## MDD File System Library

## USB Library

## Periferical Library

## Driver Acelerómetro ADXL345

El acelerómetro seleccionado permite realizar 4 operaciones por el bus de SPI. Estas operaciones son leer un registro, escribir a un registro y leer/escribir múltiples registros. Al consultar el datasheet de este integrado, se ve que para leer un registro, primero se debe bajar la señal de chip select, aguardar un tiempo , y mandar el comando de read que es igual a 0x80. Consecuentemente, se puede mandar un byte igual a 0x40 que representa que se desea realizar un read de múltiples registros consecutivos. A seguir, se escribe el address del primer registro a recibir, o el registro a leer en caso de que sea un solo registro. Una vez enviado la dirección del registro correspondiente, el acelerómetro devuelve el dato almacenado en esta posición de memoria. Cuando se finaliza el intercambio de información, se debe aguardar un tiempo En la figura \_\_ se ve un diagrama de los tiempos para tanto leer como para escribir.

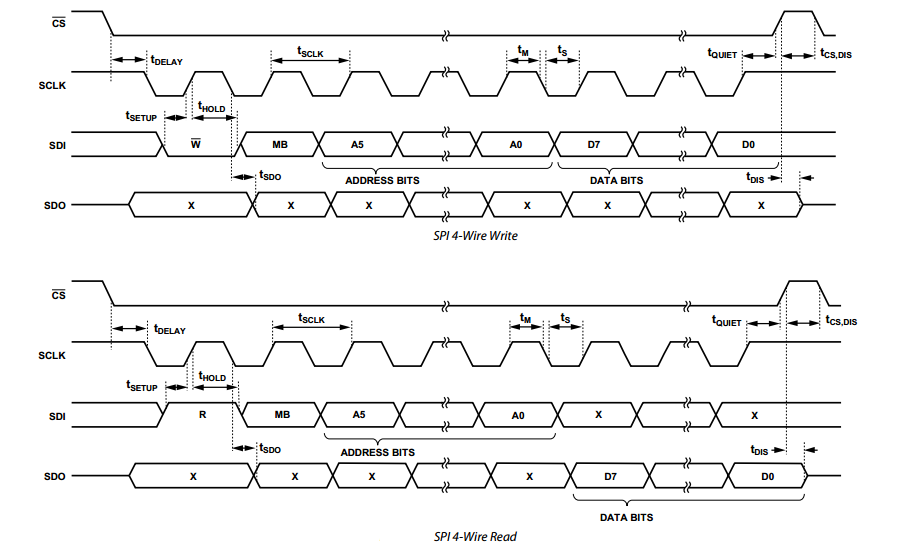
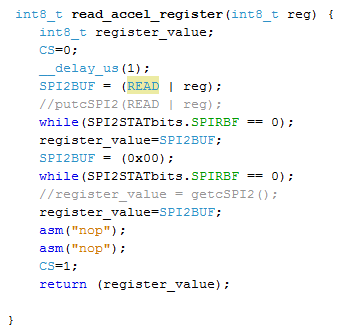
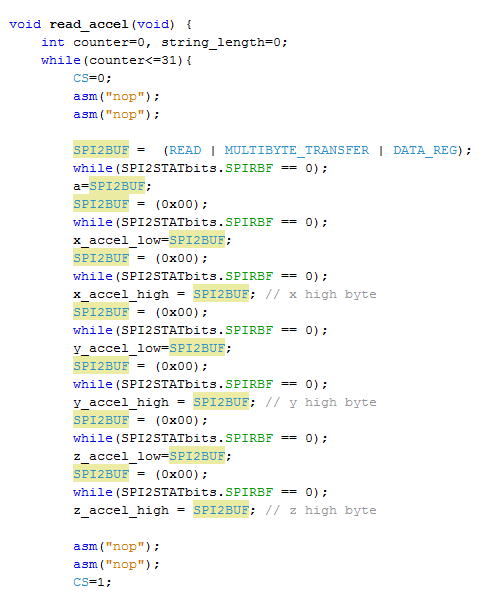


Figura 9

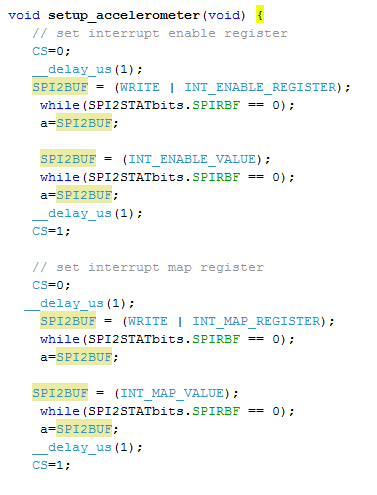
En el ejemplo de código 1, se puede visualizar como se escribió el driver de las funciones tanto de escribir a un registro, leer un registro y leer los registros de datos de todos los ejes del acelerómetro (desde la posición de memoria 0x32 hasta la 0x37). También se crearon funciones para colocar el acelerómetro en standby y para sacarlo de standby. Esto permite reducir el consumo de las pulseras cuando no están en uso.



Ejemplo de Codigo 1



Ejemplo de Codigo 2



Ejemplo de Codigo 3

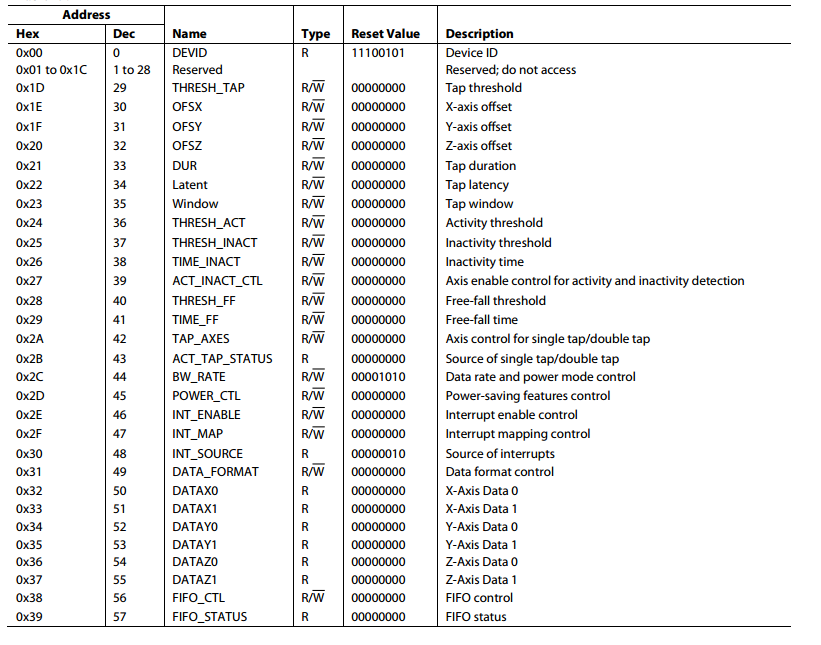


Tabla 3

## Firmware de Aplicación

### Diagramas de Flujo

### Descripción de Código

Primero se configura el acelerómetro, es decir, se habilita la interrupción del acelerómetro, la velocidad de conversión, se habilita el uso de fifo, al igual que el tamaño de la misma, entre otras configuraciones. Para leer las mediciones, primero se debe mandar un valor, dado a que es una comunicación serie por SPI. En este trabajo, se manda el valor 0x00. Cada valor recibido representa la parte baja y alta de la aceleración en cada eje, es decir, primero recibimos el byte bajo de la aceleración en el eje x, luego el byte alto de la aceleración en el eje x, luego en y y en z respectivamente. Estos 6 valores son guardados y agrupados de modo que solo haya 2 variables, 1 para cada eje. Para realizar la comunicación entre el micro controlador y el acelerómetro se debe tener en cuenta que hay que bajar el chip select, y esperar, en determinados momentos, a recibir un byte de confirmación que el dato está listo para ser leído. Además, al trabajar con una fifo de 32 niveles, se debe repetir el proceso 32 veces.

# Archivo .CSV

# Problemas

## Correcciones de PCB

## Disponibilidad en el Mercado Local

# Impacto Ambiental

# Responsabilidad Social

# Conclusiones

# Índice de Figuras

# Referencias

Datasheet adxl345

Datasheet pic24fj256gb106

# Anexo I Esquemático Completo

# Anexo II PCB Completo

# Anexo III Solid Edge

# Anexo IV Producto final

# Anexo V Organización Industrial