Asier Marin

Aritz Ibañez

Ane Sanchiz

**Laboratorio 5**

**Sensores para Sistemas Embebidos**

**Instrucciones**

En este divertido laboratorio vais a aprender cómo comunicarse con sensores que usan el protocolo I2C. Además, aprenderás a buscar información en el *datasheet* del sensor. En estas actividades mostraremos los valores del sensor los mostraremos por pantalla, pero en el último tema del curso aprenderás cómo guardar los valores en una base de datos, o acceder a su lectura desde un navegador web desde tu smartphone.

* Entrega: en ALUD
* Este laboratorio se hará en grupos de 2-3 personas, aunque la entrega será individual.
* Revisad las diapositivas correspondientes a los sensores I2C.
* Antes de comenzar debéis instalar la librería smbus2 para I2C en python en la RPi

**$ sudo pip3 install smbus2**

**Parte 1 (4 puntos): Sensor digital de Luz**

**Entrega**: Este mismo documento habiendo respondido a las preguntas, y el fichero **luz\_sensor.py** completo.

En esta parte usaréis el sensor digital de Luz TSL2561. El *datasheet* está disponible en ALUD. Se proporciona el programa de ejemplo (luz\_sensor.py) Debéis conectar el sensor como se muestra en la figura de la siguiente página, y ejecutar el programa de ejemplo anterior. Comprobad que se muestran por pantalla los valores del sensor. A continuación, realizar las siguientes tareas/preguntas

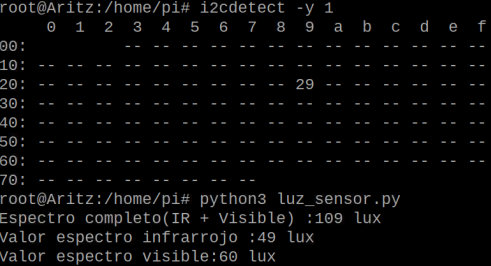
1. (1 punto) En el programa **luz\_sensor.py**, añadir una función que muestre por pantalla los valores de intensidad luminosa de una forma más *legible* o *intuitiva*, en vez de valores LUX. Por ejemplo, cuando el valor de LUX sea 125, se mostraría “Habitación interior bastante oscura”. Usad aquí vuestra creatividad para diseñar la función, y tomad como referencia los valores de la tabla de esta página web <https://docs.microsoft.com/en-us/windows/win32/sensorsapi/understanding-and-interpreting-lux-values>

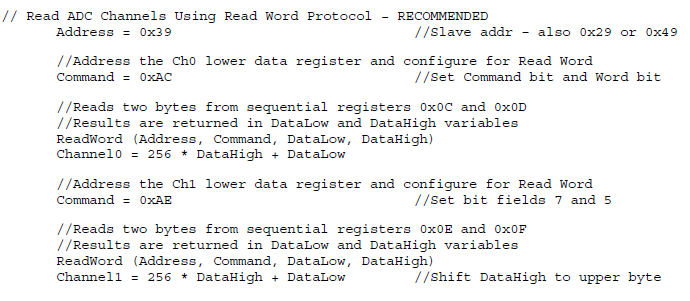
Como se puede observar en la imagen siguiente se puede ver como se muestra en pantalla la iluminación obtenida. Estas descripciones se basan en el rango de valores obtenidos a partir de la documentación oficial.

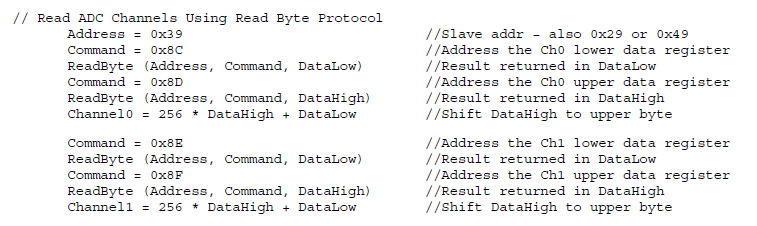
|  |
| --- |
|  |
|  |
|  |

1. (1 punto) Descubre la dirección I2C del sensor con el comando $i2cdetect -y 1  
   Escríbela debajo. ¿Coincide con la que se indica en el datasheet?

El comando describe desde qué dirección hexadecimal se obtienen los datos del sensor. Al igual que se indica en la documentación oficial, la dirección debe apuntar a 0x29 o 0x49.

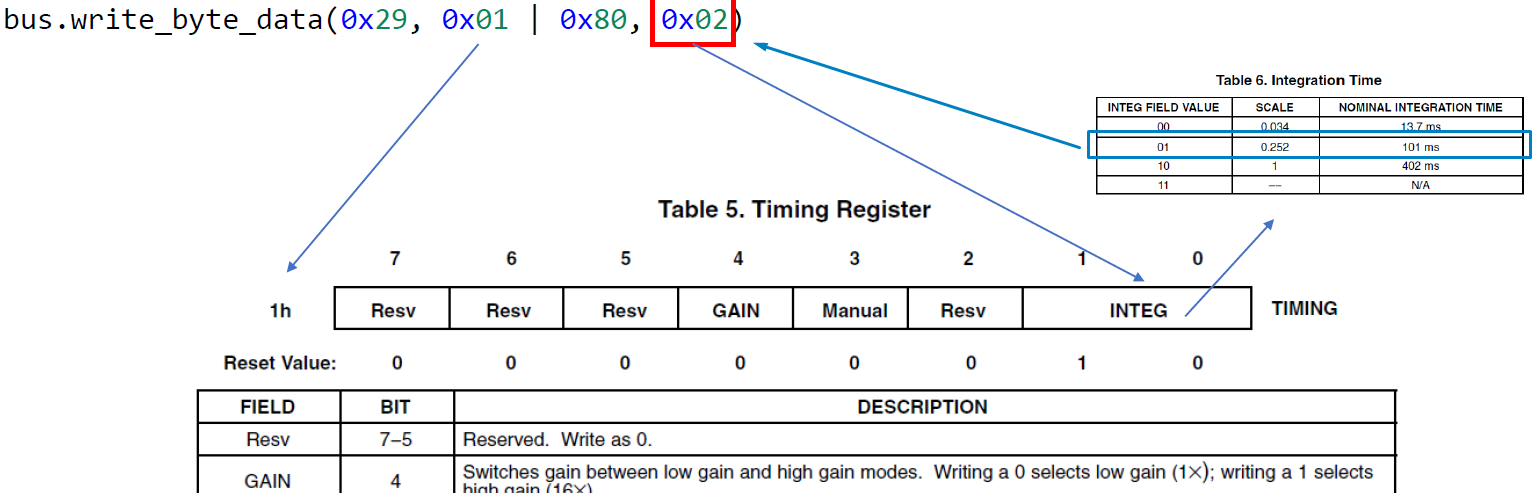


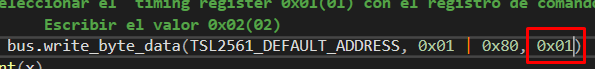


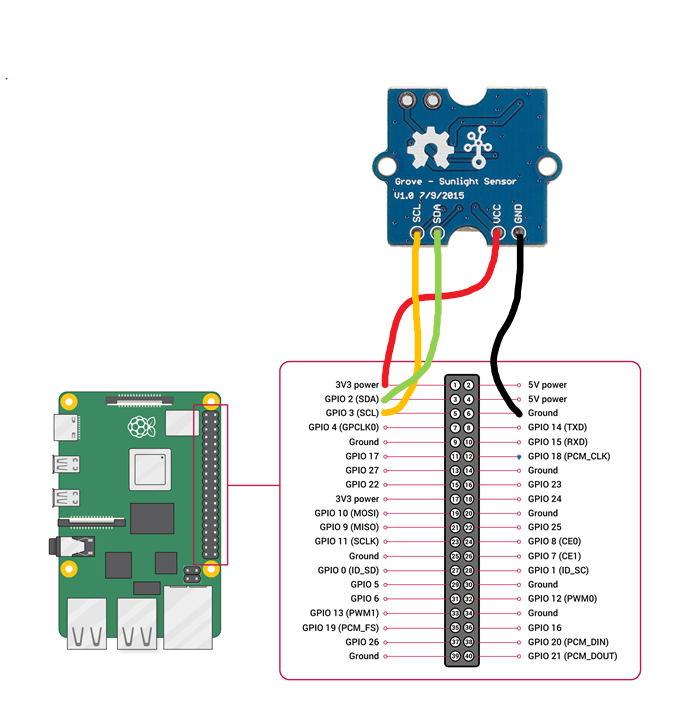


1. (2 puntos) ¿Cómo cambiarías el valor del tiempo de integración del conversor analógico-digital a un valor de 101 ms? (pista: mirad el registro de timing ó Timing Register). Consultad además las diapositivas de clase. Incluye la línea o líneas de código de Python que usarías.

Para obtener el valor con ese tiempo de respuesta es necesario que se establezca en el valor 0x01.





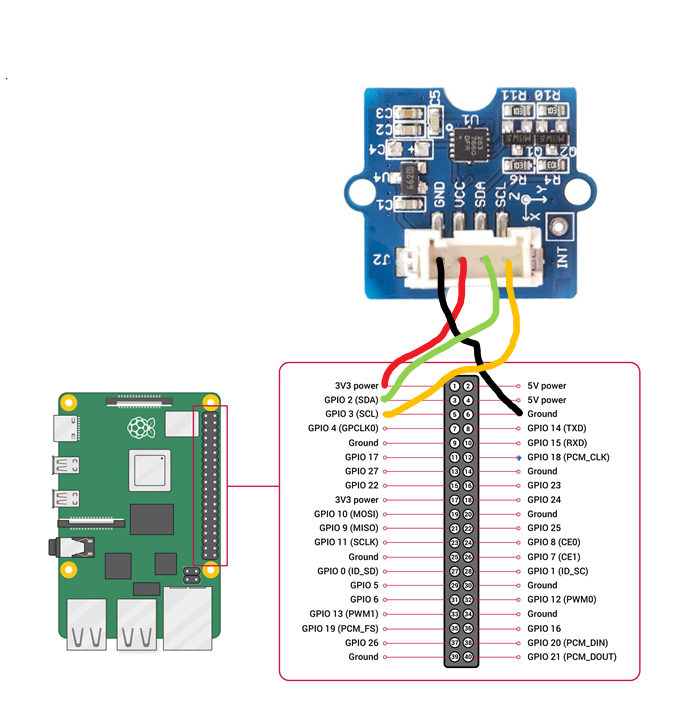


**Parte 2 (6 puntos): Acelerómetro**

El acelerómetro que tenéis es del tipo MEMs capacitivo, y una vez obtenido los valores analógicos de aceleración de los ejes x,y,z, los convierte a formato digital de 6 bits, y lo guarda en unos registros internos. Mediante el protocolo I2C y nuestra RPi, es posible leer los valores de dichos registros internos para acceder a los valores de la aceleración en x,y,z.

**Entrega**: Este mismo documento habiendo respondido a las preguntas, y el fichero **accel\_ejemplo.py** completo.

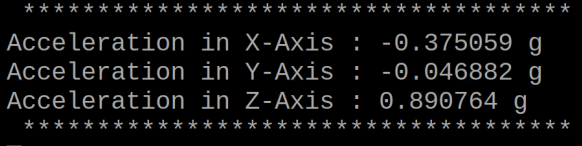
La siguiente figura muestra cómo debemos conectar el sensor a la RPI.



1. (0.5 puntos) Conecta el sensor a la RPi, y comprueba el funcionamiento con el programa que se proporciona **accel\_ejemplo.py**. Se debe mostrar por pantalla los valores de x,y,z. ¿A qué **unidades** corresponde la medida absoluta que devuelve el sensor?

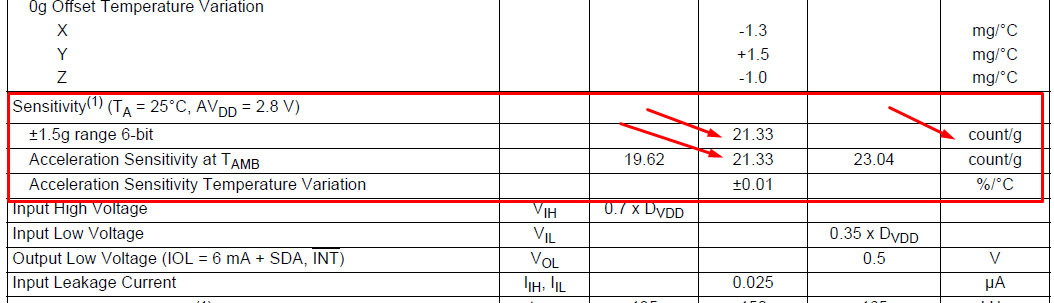
Corresponde a la intensidad gravitacional universal. De forma constante cada unidad corresponde a 9.8 m/s2.



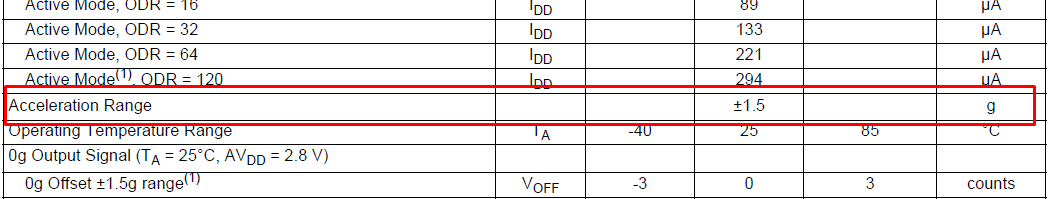


1. (0.5 puntos) ¿Por qué se dividen los valores que se leen del registro de datos entre **21.33**?

El valor predeterminado de entrada son 21.33, la idea es dividir por ese valor para obtener una referencia sobre 0. De esa manera en más fácil a la vista percibir los cambios en los diversos ejes.



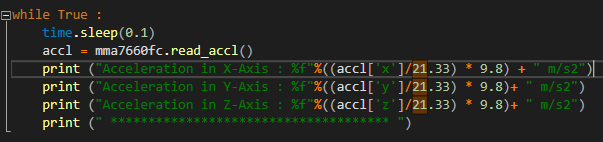
1. (0.5 puntos) ¿A qué se refiere el valor +/-1.5g, que describe el sensor en el datasheet?

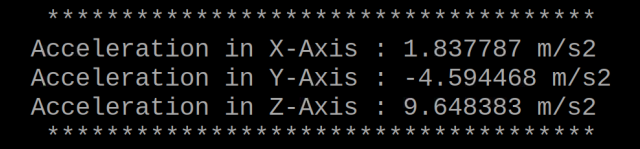


Es el rango de error del sensor.

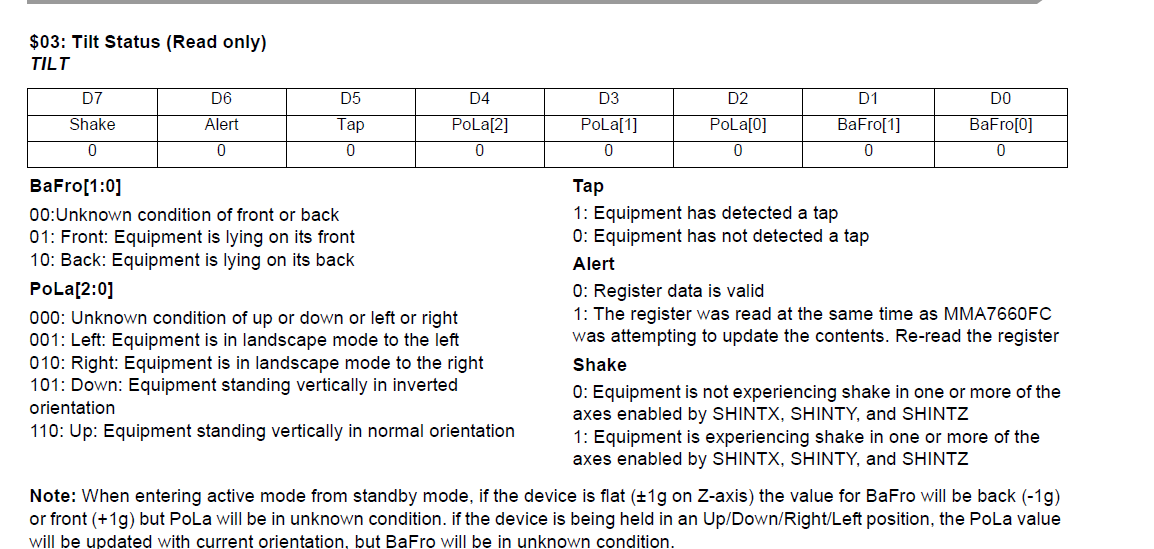
1. (0.5 puntos) Modifica ligeramente el programa anterior, de forma que se muestren por pantalla los valores en la unidad m/s^2.

Simplemente se ha multiplicado por su correcta conversión como constante.

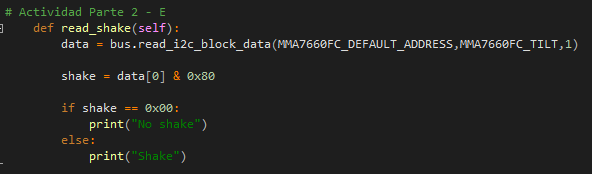


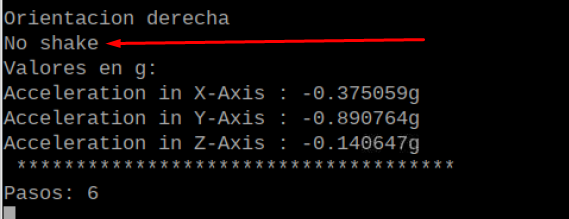


1. (1.5 puntos) Añade una función en el programa anterior (**accel\_ejemplo.py)** que detecte el evento de un “shake”. El sensor que tenéis ya implementa la detección del evento de shake, así que se puede leer el estado de shake o no-shake mediante la lectura del bit 7 del registro 03. Como ayuda, básate en la función que se incluye en el archivo de Python que lee los bits D4,D3,D2 para detectar la polarización (PoLa) del acelerómetro.



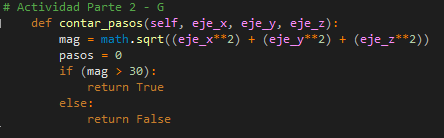
1. (1 punto) ¿Qué criterio usa el sensor para decidir si ha habido un evento de shake o no? (pista: consultad el datasheet)





1. (1.5 puntos) Realiza una función en el fichero Python anterior que implemente **contador de pasos simple basado en los valores leídos del acelerómetro**. No se pide que sea robusto, sino un programa sencillo. Cuando se llame a la función, el programa debe empezar a detectar pasos. Cada pasa detectado, imprime por pantalla el número actual del contador de pasos (desde el comiendo del contador de pasos). Para comprobar el funcionamiento, puedes mover suavemente el acelerómetro con la mano, emulando el movimiento de un paso (andando). Como ayuda, consulta las diapositivas de clase.

La referencia utilizada para contar pasos es 30. Se ha escogido a partir de ir probando poco a poco. La idea es utilizar la magnitud como filtro a la hora de contar pasos.



Este método por lo tanto se encuentra en el *loop* que hace la llamada para el filtrado constantemente.

