

Tarea 1

Profesor: Luciano Radrigan F.
Auxiliar: Camilo Araya.

Descripción

Para esta tarea a cada grupo se le pasara una ESP32 junto a un sensor inercial BOSCH BMI270, es responsabilidad de cada grupo cuidarlo y devolverlo a final de semestre en buen estado.

Utilizando el microcontrolador (ESP32) se hará una comunicación serial I2C con el sensor inercial, la cual busca tanto controlar la configuración como conseguir las mediciones de este, todo lo cual deberá ser manejado y recibido, respectivamente, desde una conexión serial UART a un computador, donde se mostrar estas mediciones.

El sensor inercial (IMU) BMI270 debe ser capaz de ponerse en los tres distintos modos de poder, los cuales son: bajo, medio y alto rendimiento, junto al modo suspensión. Para hacerlo use como base el programa subido en u-cursos para ello, él cual muestra como ponerlo en modo normal, y utilizando el datasheet del sensor para entender como poner los demás modos. Aparte de ello deberá poder cambiar la configuración de sensibilidad como la frecuencia de muestro (ODR) tanto para el giroscopio como para el acelerómetro de la unidad.

Del sensor deberá extraer las mediciones en los tres ejes (x,y,z) de aceleración, en $\frac{m}{seg^2}$ y en G, como de velocidad angular, en $\frac{rad}{seg}$. Junto a ello deberá mandar una transformada rapida de Fourier (FFT), una RMS (Root Mean Square), los últimos 5 peaks del RMS y los ultimos 5 peacks de la seria de tiempo de la aceleracion (por cada eje), cada una para la misma cierta ventana predefinida de los datos. Esta ventana es una temporal de una cantidad definida de últimas lecturas (pruebe comenzando con las últimas 500 lecturas o use la misma frecuencia de muestreo que tenga configurada), para cada eje independientemente.

- Recuerde que la RMS se calcula como, donde X_i son los valores de la muestra: $\sqrt{\frac{\sum_1^N X_i^2}{N}}$
- Para la transformada de Fourier puede usar la libreria de su preferencia
- Los últimos 5 peaks deben mantenerse guardadas a medida que aparezcan las medidas y ser reemplazadas cuando pase su ventana por las siguientes.
- Deberá visualizar y mostrar cada una de estas medidas en el computador que reciban. Siendo específicos en pantalla deberán al menos imprimirse (puede hacerlo a traves de graficos si quiere) los siguientes datos (recuerde que deberá calcularlos por ventana):
[$data_x, data_y, data_z, RMS_x, RMS_y, RMS_z, FFT_x, FFT_y, FFT_z, Peaks_x, Peaks_y, Peaks_z$]

Links utiles

- [Documentación ESP-IDF](#)
- [Página IMU BMI270](#)

Recomendaciones

- Dado que para la tarea se utilizaran componentes físicos, existirán dificultades para probar el software en grupo. Se recomienda juntarse para poder hacer las pruebas de su código, el cual deberían tenerlo en un repositorio (por ej en Github). Busquen la forma de coordinarse para poder trabajar en ello.
- Se va a trabajar conectando los pines a través de cables, ¡Conecte con cuidado antes de probar! Es posible quemar la ESP32 si es que el pin de voltaje (u otros de alto corriente) se entrecruza adonde no debe.
- Cualquier problema con el funcionamiento de sus componentes, recuerde revisar que el problema no sea uno físico (cables mal conectados o en mal estado) antes de revisar si es error de software. Consulte con el equipo docente si no es capaz de resolver o identificar el problema.
- Si tiene dudas acerca de la tarea o como manejar los componentes, no dude en publicar su duda en el foro o preguntar al cuerpo docente.
- Recuerde ser organizado para el código y el manejo de paquetes, dado a que estara trabajando para una comunicación entre la ESP32 (que estara escrito en C) y su computador (en Python) lo mejor es que sea claro que esta pasando en cada lado, más aún teniendo en cuenta que no podra monitorear a la ESP32.

Entrega

- La entrega del código estará abierta hasta el lunes 7 de mayo a las 23.59.
- En una fecha por definir se deberá realizar una demostración en vivo para revisar que cumpla el funcionamiento pedido.