

# **Trabajo Práctico 2 - Sensor Fusion**

**31.99 - Mecatrónica Aplicada**

**Fecha de entrega: 23 - 10 - 20**

**Docentes:**

- **Ing. Lucas Perfumo**
- **Ing. Nelson Ariel Fortunatti**

**23 - 9 - 2020**



### Tabla de contenido

Registros de cambios	3
Introducción	4
Implementación	5
Formato de Entrega	7

**Registros de cambios**

Revisión	Detalle de los cambios realizados	Fecha
1.0	Creación del documento	20/8/2020
1.1	Lineamientos del Trabajo Práctico	21/8/2020
1.2	Armado del trabajo práctico	20/9/2020
1.3	Versión final	23/9/2020

### Introducción

Dada las características mencionadas durante las clases teóricas los sensores MARG (Magnetic, Angular Rate and Gravity), el presente trabajo práctico consiste en utilizar el módulo MPU 9250 que posee un acelerómetro, giroscopio y magnetómetro obteniendo 9 grados de libertad. Para el procesamiento de cada uno de los sensores utilizaremos el algoritmo de Madgwick con el cual se pueden obtener muy buena precisión tanto para altas tasas de muestreo como para bajas. A su vez, al requerir una cantidad reducida de operaciones para cada actualización de la orientación puede ser utilizado en sistemas embebidos sin mayor dificultad.

El algoritmo de Madgwick es un desarrollo open source en los siguientes links se encuentra tanto la página web del desarrollador del algoritmo, paper de referencia y el código en C para implementar tanto el algoritmo para IMUs como para MARGs:

- [Página Web](#)
- [Paper](#)
- [Código en C](#)

Para utilizar este módulo, incorporar el algoritmo de sensor fusion y enviar los datos a la PC se utiliza la placa **STM32-Nucleo-F303RE**. Como IDE para la utilización de la placa se utilizará el **STM32CubeIDE** que se basa en un Eclipse customizado por ST el cual incorpora el **STM32CubeMX** que es una herramienta para inicializar los periféricos del microcontrolador de forma gráfica para luego crear código automático y facilitar el desarrollo. Ésta herramienta no posee limitaciones en su uso y puede ser descargada de forma gratuita [desde la página web de ST](#). Para los que no estén familiarizados con estas herramientas pueden seguir el [siguiente video](#).

### Implementación

El trabajo práctico consiste en realizar un sistema para la implementación de un algoritmo de **Sensor Fusion** y verificar su comportamiento graficando los datos mediante el uso de una PC.

En la **Figura 1**, se puede notar dos grandes bloques, uno relacionado a la placa de desarrollo **STM32-Nucleo** y otro correspondiente a la **PC**.

Para el bloque **STM32-Nucleo** se deben cumplir los siguientes requerimientos:

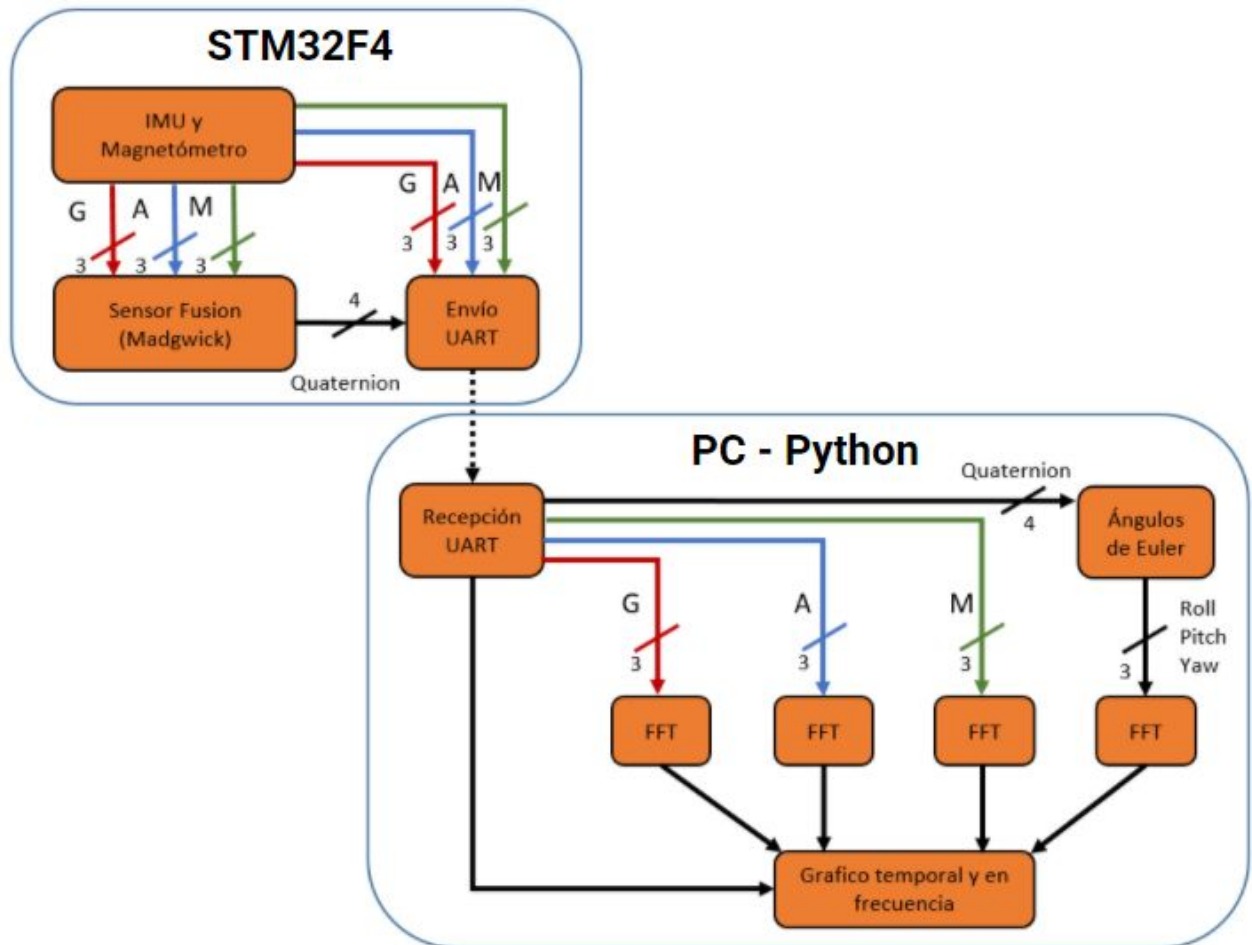
- Utilizar una biblioteca propia o de un tercero para realizar la comunicación entre la placa de desarrollo y el módulo MPU-9250.
- Utilizar el algoritmo de Madgwick para el procesamiento de las señales provenientes de los sensores según las consideraciones mencionadas en el paper del autor.
- El desarrollo del firmware encargado de realizar la intercomunicación de las tareas anteriores junto con la comunicación a la PC debe ser de autoría propia.
- Se debe enviar a la PC mediante el protocolo UART los valores obtenidos tanto del MPU-9250 sin procesar como los cuaternions obtenidos del filtro de orientación.
- Como los datos enviados por UART son valores en punto flotante, debe haber un mecanismo de sincronismo para poder obtener en la PC los valores en punto flotante enviados desde el microcontrolador.

Para el bloque **PC-Python** se deben cumplir los siguientes requerimientos:

- El script de python recibe 13 valores en punto flotante con datos útiles, se pueden utilizar variables adicionales que ayuden al sincronismo.
- Se deben presentar en pantalla para cada sensor los resultados de cada eje ortogonal junto a su transformada de Fourier para poder verificar el contenido espectral de cada señal analizada.
- Mediante la transformación de los cuaterniones a ángulos de Euler ([Ecuaciones 7, 8 y 9](#)) representar de forma gráfica los ángulos obtenidos para roll, pitch y yaw junto a su contenido espectral.

Luego de haber probado la utilización de la placa de desarrollo junto a la utilización de diversas funcionalidades. Es recomendable avanzar con un programa que sólo setee las características esenciales del módulo MPU 9250 para asegurarse una correcta comunicación y entendimiento del proceso de comunicación a través del protocolo I2C entre el microcontrolador y el módulo MARG. Para esto pueden usar la estructura del programa mostrados en [este link](#). Luego de haber obtenido resultados satisfactorios se puede utilizar la siguiente biblioteca para utilizar el MPU 9250 haciendo los cambios pertinentes para la plataforma de desarrollo utilizada:

- [Bolderflight](#)
- [Similar al anterior pero escrito en C](#)



**Figura 1: Diagrama en bloques del sistema a realizar.**

**Nota:** Es recomendable hacer un desarrollo modular del TP e implementar las funcionalidades una a una, verificando en cada paso el correcto funcionamiento, de esta manera se pueden resolver los problemas que se presenten con mayor facilidad.

### Formato de Entrega

- Confeccionar un documento donde se explique mediante un diagrama en bloques el funcionamiento de cada uno de los programas implementados tanto el referido a la placa STM32-Nucleo como al script de Python.
- Explicar cómo se llevó a cabo la sincronización entre la PC y la placa de desarrollo para la transmisión de valores en punto flotante.
- Analizar las gráficas obtenidas en base a las características mencionadas en clase para el caso del giroscopio, magnetómetro y acelerómetro. Hacer el mismo análisis de forma cualitativa para el caso de los ángulos de Euler. Presentar imágenes y evaluar los resultados obtenidos.
- Armar un breve video mostrando el funcionamiento del trabajo donde se pueda ver cómo varían las gráficas con respecto a cambios en la orientación por parte del MARG.
- Cualquier desarrollo adicional o información útil va a ser tomada en cuenta al momento de la evaluación del presente trabajo práctico.