

# **Sistemas Basados en Microprocesador 2024**

## **Bloque 3**

**Integración y desarrollo de una aplicación:  
Controlador de un detector de aceleración**

## **Realización del diseño (IMPORTANTE)**

Para la realización de este bloque de la asignatura (Bloque 3) se dispone de 7 sesiones presenciales más una sesión de evaluación. La evaluación se realizará en la **última sesión presencial (jueves, 19 de diciembre)**, si bien puede realizarse con anterioridad, previa solicitud al profesor de su grupo de laboratorio y siempre que se haya entregado tanto los proyectos como la documentación requerida con anterioridad.

Para afrontar con garantías el diseño, éste debe dividirse en una serie de hitos que permitan ir comprobando el correcto funcionamiento de cada uno de los módulos que se vayan desarrollando.

Se deberán realizar entregas en Moodle con el progreso del diseño. Deben entregarse proyectos completos que demuestren el correcto funcionamiento de cada uno de los módulos que se van desarrollando y la documentación relativa a dicho módulos que formará parte de la memoria técnica final.

El calendario de entrega de módulos verificados es el siguiente:

	<b>Fecha</b>	<b>Módulos</b>
<b>E1</b>	Viernes, 29/11/2024, 14:00	JOYSTICK, HORA y <b>DOCUMENTACIÓN</b>
<b>E2</b>	Lunes, 9/12/2024, 14:00	ACELERÓMETRO, LCD, LEDs NUCLEO y <b>DOCUMENTACIÓN</b>
<b>E3</b>	Viernes, 13/12/2024, 14:00	COM-PC y <b>DOCUMENTACIÓN</b>
<b>E4</b>	Miércoles, 18/12/2024, 14:00	Proyecto FINAL y memoria técnica

Adicionalmente a la implementación del diseño es necesario tener en cuenta que se debe entregar una memoria técnica al finalizar el diseño, siguiendo el modelo que se encuentra en Moodle ("*B3\_Plantilla\_Memoria.docx*").

La **memoria**, junto al proyecto de **Keil µVision de la aplicación final**, deberán subirse a Moodle antes del **miércoles, 18 de diciembre de 2024, a las 14:00 horas.**

## **Introducción y objetivos del diseño**

El objetivo de este diseño es desarrollar una aplicación completa basada en la tarjeta NUCLEO STM32F429Zi a partir de unas especificaciones generales. Para poder realizar este diseño se necesita conocer cada uno de los elementos abordados (hardware y software) en los bloques anteriores de la asignatura. Además, los estudiantes deberán analizar el funcionamiento y utilizar de forma autónoma algunos elementos no estudiados hasta el momento e incluirlos en la aplicación propuesta.

Como se comprobará más adelante, la especificación del diseño es muy abierta para que se puedan tomar decisiones durante el desarrollo del mismo. Todas aquellas cuestiones que no estén claramente especificadas pueden implementarse del modo que se considere más oportuno, aplicando criterios basados en la simplicidad del diseño y la usabilidad del sistema.

Al igual que en prácticas anteriores, se propondrán algunos ejercicios opcionales que podrán realizarse por parte de los alumnos en función del tiempo disponible y que tendrán influencia sobre la calificación final de la asignatura.

## **Descripción del sistema**

El diseño que se va a desarrollar es un controlador de un detector de aceleración para un objeto en cualquiera de los ejes en los que se produzca: X, Y o Z. Para ello se utilizará un acelerómetro-giroscopo basado en el chip MPU6050, que puede encontrarse en distintas “breadboards” disponibles en tiendas online o físicas.

Este sistema se encargará de monitorizar de manera continua la aceleración respecto a una situación de reposo en cada uno de los ejes, tras someter dicha tarjeta a un movimiento que cambie su posición.

Los elementos básicos que se utilizarán para esta aplicación son los siguientes:

- Tarjeta NUCLEO STM32F429Zi.
- Tarjeta mbed App Board.
- Tarjeta con MPU6050 (GY-521, MikroE Accel 8 Click, etc) adquirido por cada pareja de estudiantes.

La figura 1 muestra un esquema simplificado del sistema.

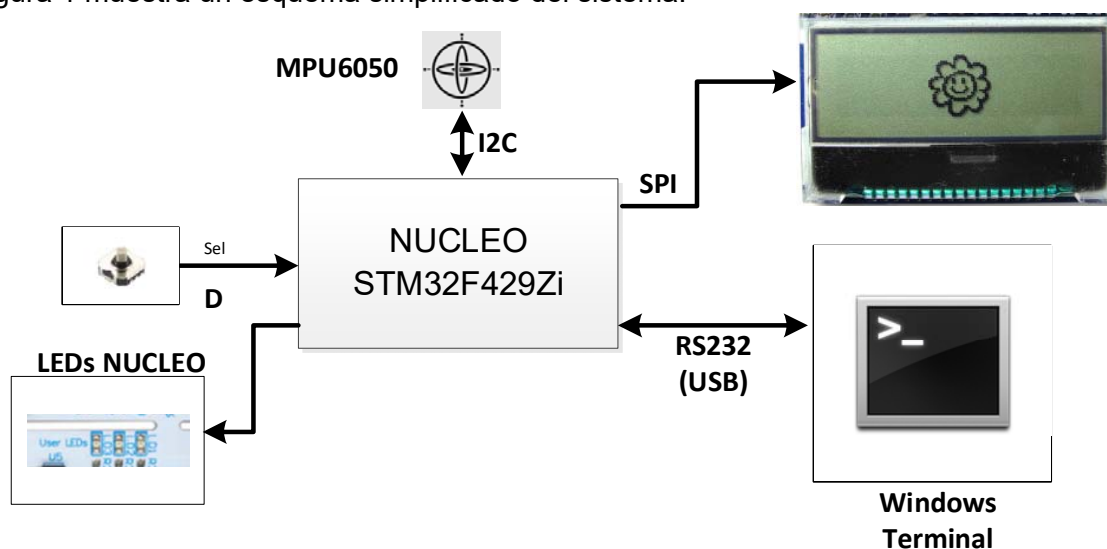


Figura 1

## **Especificaciones**

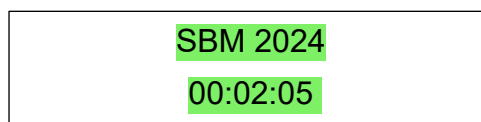
El funcionamiento del sistema debe responder a las siguientes especificaciones:

- a) Es obligatorio el uso del sistema operativo CMSIS RTOSv2 (RTX). La aplicación se construirá en torno a varios treads/timers que tendrán que dar soporte a una o varias funcionalidades del sistema.
- b) En todo momento el sistema debe mantener la hora con una resolución de un segundo.
- c) Entradas al sistema:
  - a. Acelerómetro/Giróscopo, proporcionará medidas de manera periódica (aceleración, temperatura).
  - b. Joystick, para poder interactuar con el sistema, el usuario utilizará las diferentes posiciones del joystick. La gestión del joystick se realizará por interrupciones, eliminando los posibles rebotes que éste origine, distinguiendo entre pulsaciones cortas y largas.
  - c. UART para recibir comandos de programación y depuración.
- d) Salidas del sistema:
  - a. LCD. El usuario podrá conocer el modo del sistema a través de la información proporcionada por el LCD
  - b. LEDs de usuario de la tarjeta NUCLEO (LD1, LD2 Y LD3) que mostrarán diferentes estados relativos a la posición del sensor.
  - c. UART para responder a los comandos de programación y depuración.
- e) El sistema tendrá tres modos de trabajo: REPOSO, ACTIVO y PROGRAMACIÓN/DEPURACIÓN.

Se podrá cambiar de modo en cualquier momento mediante una **pulsación larga** (se considera una pulsación larga aquella que dura más de 1 segundo) del botón CENTER del joystick. El sistema, tras un reset, arrancará en modo REPOSO.

- **Modo REPOSO.**

En este modo se presentará en el LCD el mensaje "SBM 2024" y la hora, tal y como se indica en la figura 2. Tras un *reset* el reloj marcará las 00:00:00. En este modo sólo se atenderá a una pulsación larga del botón CENTER del joystick, que hará que el sistema pase a modo ACTIVO.

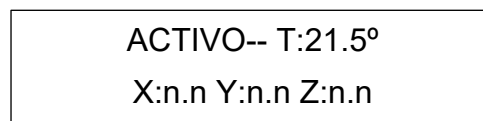


**Figura 2. LCD en modo REPOSO**

- **Modo ACTIVO.**

En este modo se estará leyendo la aceleración y la temperatura que entrega el sensor MPU6050 cada segundo. Los valores de la aceleración obtenidos en cada eje (Ax, Ay y Az) se compararán con los valores por defecto de referencia ( $Ax_r=Ay_r=Az_r=1.0$ ). Si los valores obtenidos son mayores que los de referencia se iluminará un led de la tarjeta NUCLEO: eje x->LD1, eje y->LD2, eje z->LD3. Si los valores obtenidos son menores se deberán apagar los leds correspondientes. El sensor debe inicializarse para utilizar el rango  $\pm 2g$ .

Cuando el sistema se encuentre en el modo ACTIVO, en el LCD debe representarse una información similar a la de la siguiente figura:



ACTIVO-- T:21.5°  
X:n.n Y:n.n Z:n.n

Figura 3. LCD en modo ACTIVO

Se debe mostrar la temperatura (T) obtenida del sensor y el valor de la aceleración para los tres ejes.

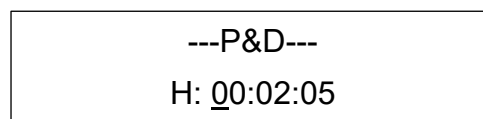
Adicionalmente el sistema debe mantener, cuando se encuentre en este modo, una tabla de 10 medidas almacenadas en un buffer circular. Cada entrada debe mantener en ASCII la siguiente información (hora, temperatura medida, y los valores de aceleración):

HH:MM:SS - -Tm:TT.T° -Ax:n.n-Ay:n.n-Az:n.n

Desde el modo ACTIVO podrá pasarse en cualquier momento al modo PROGRAMACIÓN/DEPURACIÓN pulsando la posición CENTER del joystick.

- **Modo PROGRAMACIÓN/DEPURACIÓN.**

En este modo se programará la hora del sistema y los valores de referencia de la aceleración para los tres ejes. La modificación de estos valores se realizará utilizando el joystick (pulsaciones cortas), presentando los cambios realizados en el LCD.



---P&D---  
H: 00:02:05

Figura 4. LCD en modo PROGRAMACIÓN/DEPURACIÓN para la hora

Utilizando los gestos UP/DOWN/LEFT/RIGHT se modificarán los valores y para hacerlos efectivos se realizará una pulsación corta en el gesto CENTER. Se modificarán sucesivamente los parámetros hora,  $Ax_r$ ,  $Ay_r$  y  $Az_r$  tras sucesivas pulsaciones del gesto CENTER.

Desde el modo PROGRAMACIÓN/DEPURACIÓN podrá pasarse en cualquier momento al modo REPOSO realizando una pulsación LARGA del gesto CENTER del joystick.

En este modo, el sistema además de responder a las peticiones del joystick, se controlará desde un PC a través de un canal serie de comunicaciones RS232. El formato de la comunicación RS232 será 115200 baudios, 8 bits de datos, 1 bit de stop y sin paridad. Es obligatorio que las comunicaciones serie se gestionen con CMSIS Driver (USART) utilizando la función de callback (interrupción/eventos).

Para el envío de estas tramas/comandos se recomienda el uso del programa **Teraterm**. Dentro de este programa se pueden utilizar macros para enviar tramas, ficheros de texto, en los que se definen cada uno de los bytes que componen dicha trama. En Moodle se encuentra un ejemplo de este tipo de ficheros.

El formato de las tramas de comunicación será el indicado en la Tabla 1.

**Tabla 1. Formato de las tramas de comunicación**

SOH CMD LEN Payload EOT		
Campo	Byte	Descripción
SOH	0x01	Todas las tramas comienzan con este carácter (Start of Head)
CMD	0xXX	Comando
LEN	0xXX	Longitud <b>TOTAL</b> de la trama, incluidos SOH y EOT.
Payload		Información del comando, puede ser de cualquier longitud e irá codificada en ASCII
EOT	0xFE	Fin de la trama

Las tramas que no estén bien formadas, no respetando el criterio indicado en la tabla anterior, deberán ser ignoradas por el sistema.

El PC podrá enviar diferentes comandos al equipo, especificados en la Tabla 2.

**Tabla 2. Comandos del PC al sistema<sup>1</sup>**

Comando / CMD	Ejemplo	Comentario
Puesta en hora [0x20]	0x01 <b>0x20</b> 0x0C "15:20:15" 0xFE "15:20:15" en hexadecimal 0x31 0x35 0x3A 0x32 0x30 0x3A 0x31 0x35	Establece la hora a las 15:20:15. Si el equipo acepta la hora, éste devolverá la misma trama que se ha recibido con el valor CMD en complemento a uno [0xDF]
Establece el valor de Ax_r [0x25]	0x01 <b>0x25</b> 0x08 "0.75" 0xFE	Establece el valor de la aceleración de referencia en el eje x en 0.75. El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xDA]
Establece el valor de Ay_r [0x26]	0x01 <b>0x26</b> 0x08 "0.50" 0xFE	Establece el valor de la aceleración de referencia en el eje y en 0.50. El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xD9]
Establece el valor de Az_r [0x27]	0x01 <b>0x27</b> 0x08 "0.25" 0xFE	Establece el valor de la aceleración de referencia en el eje z en 0.25. El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xD8]

<sup>1</sup> Los datos entre comillas ( " ") en esta tabla están representados en ASCII. Las comillas ( " ") no forman parte de la trama.

Todas las Medidas [0x55]	0x01 <b>0x55</b> 0x04 0xFE	Solicita todas las medidas almacenadas en el buffer circular. El equipo responde con tantas tramas del tipo “Medida” como medidas tenga almacenadas.
Borrar medidas [0x60]	0x01 <b>0x60</b> 0x04 0xFE	Borra las medidas almacenadas en el buffer circular. Cuando se ha borrado la memoria el equipo responde con la misma trama con el valor CMD en complemento a uno [0x9F]

El sistema debe responder con las tramas especificadas en la Tabla 3.

Tabla 3. Respuestas del sistema al PC

Respuestas / CMD	Ejemplo	Comentario
Respuesta al comando <i>Puesta en hora</i> [0x20]	0x01 <b>0xDF</b> 0x0C “15:20:15” 0xFE	
Respuesta al comando <i>Establece el valor de Ax_r</i>	0x01 <b>0xDA</b> 0x08 ”0.75” 0xFE	
Respuesta al comando <i>Establece el valor de Ay_r</i>	0x01 <b>0xD9</b> 0x08 ”0.50” 0xFE	
Respuesta al comando <i>Establece el valor de Az_r</i>	0x01 <b>0xD8</b> 0x08 ”0.25” 0xFE	
Medida [0xAF]	0x01 <b>0xAF</b> 0x27 “HH:MM:SS--Tm:TT.Tº-Ax:n.n-Ay:n.n-Az:n.n” 0xFE	Devuelve el valor de una medida almacenada.
Respuesta al comando <i>Borrar medidas</i> [0x60]	0x01 <b>0x9F</b> 0x04 0xFE	

## Conexiones del sistema

Como puede comprobarse del análisis de las especificaciones del sistema, el mismo estará compuesto por cuatro elementos diferentes: 1) la tarjeta NUCLEO STM32F429Zi, que incorpora el microcontrolador; 2) la tarjeta *mbed app board*, que proporciona el LCD y el joystick; 3) la tarjeta que incorpora el sensor MPU6050 y 4) el PC el cual transmite/recibe tramas serie del sistema.

Las conexiones entre estos elementos deben realizarse OBLIGATORIAMENTE siguiendo las indicaciones proporcionadas en la Tabla 4. La conexión de la tarjeta con el acelerómetro/giróscopo se realizará a la tarjeta NUCLEO según el criterio y selección del estudiante.

**Tabla 4. Conexiones entre la mbed app board y la tarjeta NUCLEO STM32F429Zi**

		Pin mbed app board	STM32F429	NUCLEO STM32F429Zi
<b>VCC</b>	+ 3.3V	DIP40	+ 3.3V	CN8.7 / +3.3V
<b>GND</b>	GND	DIP1	GND	CN8.11 / GND
<b>LCD (SPI)</b>	MOSI	DIP5	PA7	CN7.14 / D11
	SCK	DIP7	PA5	CN7.10 / D13
	CS	DIP11	PD14	CN7.16 / D10
	A0	DIP8	PF13	CN10.2 / D7
	RESET	DIP6	PA6	CN7.12 / D12
<b>Joystick</b>	Down	DIP12	PE12	CN10.26 / D39
	Left	DIP13	PE14	CN10.28 / D38
	Center	DIP14	PE15	CN10.30 / D37
	Up	DIP15	PB10	CN10.32 / D36
	Right	DIP16	PB11	CN10.34 / D35



## Módulos software del sistema

El software del sistema debe desarrollarse de manera modular, de forma que cada uno de los módulos que componen el mismo se encargue de la gestión de un aspecto determinado (tarea, periférico, sensor, funcionalidad, etc). Además de los módulos específicos, deberá desarrollarse un módulo principal que se encargará de la gestión y sincronización de todos ellos. Cada uno de los módulos específicos debe ser autocontenido y tendrá únicamente interacción con el módulo principal, sin posibilidad de ninguna comunicación directa entre módulos específicos que no esté controlada por el módulo principal.

El software correspondiente a cada uno de los módulos debe estar recogido en un fichero que contenga el código (.c) y un fichero de cabecera (.h).

**Se proporciona en Moodle un proyecto con un ejemplo de un módulo de salida (LEDs de la tarjeta NUCLEO) y el programa de test que verifica su funcionamiento.**

Para el desarrollo de la versión básica del sistema con las especificaciones obligatorias se proponen los siguientes módulos, cuya estructura se muestra en la Figura 5.

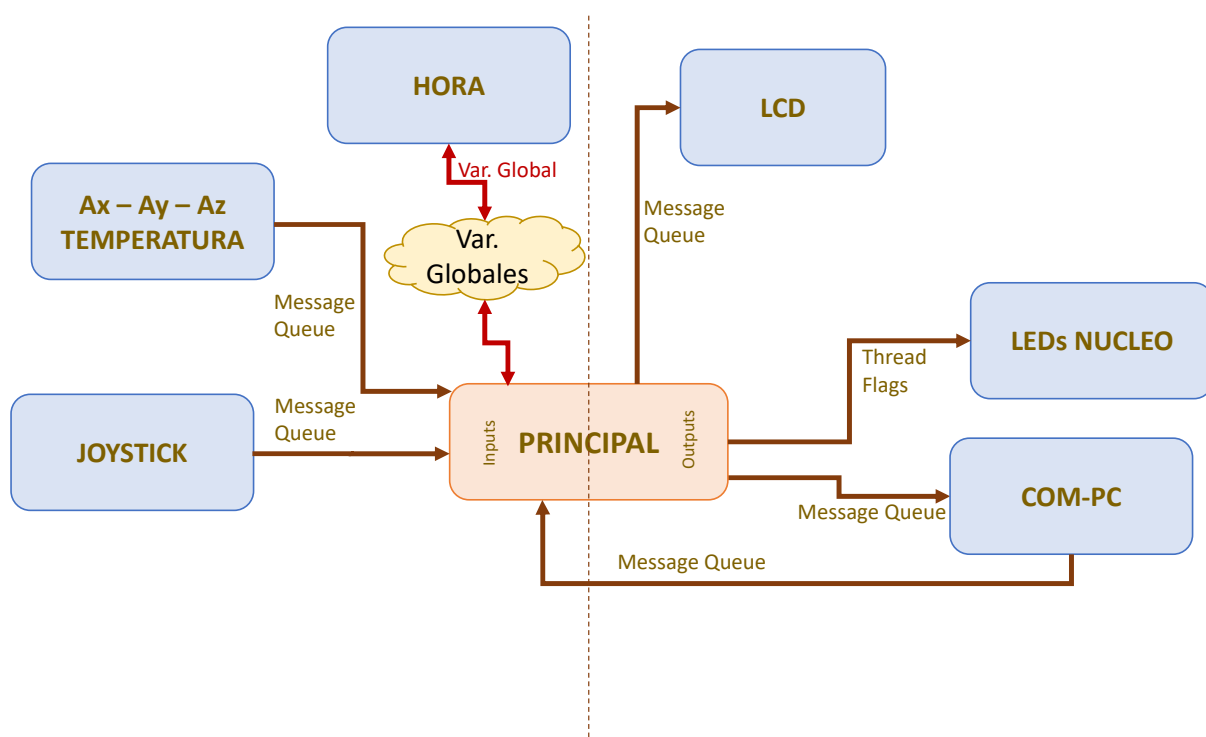


Figura 5. Estructura de los módulos que componen la aplicación

<b>Módulo HORA</b>	Módulo encargado de la gestión del reloj del sistema
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Variables Globales Actualiza las variables globales “horas”, “minutos” y “segundos”
<b>Ficheros</b>	clock.c y clock.h

<b>Módulo Ax – Ay – Az TEMPERATURA</b>	Módulo encargado de la lectura de la aceleración y la temperatura proporcionada por el sensor MPU6050 conectado al bus I2C
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Message Queue Envía un mensaje con la información cada segundo
<b>Ficheros</b>	Mpu6050.c y mpu6050.h

<b>Módulo JOYSTICK</b>	Módulo encargado de detectar e identificar una pulsación en el joystick
<b>Entrada</b>	Ninguna
<b>Salida</b>	Método de sincronización: Message Queue Cada vez que se detecta una pulsación se envía un mensaje a una cola de mensajes con información de la tecla pulsada y si la pulsación ha sido normal o larga
<b>Ficheros</b>	joystick.c y joystick.h

<b>Módulo LEDs NUCLEO</b>	Módulo encargado de encender/apagar los LEDs LD1, LD2 y LD3.
<b>Entrada</b>	Método de sincronización: Thread Flags Thread esperando la activación de algún flag para encender o apagar un led, en función del resultado de la comparación entre las aceleraciones proporcionadas por el sensor y las de referencia.
<b>Salida</b>	Ninguna
<b>Ficheros</b>	leds_N.c y leds_N.h

<b>Módulo LCD</b>	Módulo encargado de representación de información por el LCD conectado al bus SPI
<b>Entrada</b>	Método de sincronización: Message Queue Thread leyendo de cola de mensajes con la información que debe representarse en el LCD y la línea en que debe representarse
<b>Salida</b>	Ninguna
<b>Ficheros</b>	lcd.c y lcd.h

<b>Módulo COM-PC</b>	Módulo encargado de la comunicación con el PC a través de la línea serie integrada en el USB
<b>Entrada</b>	Método de sincronización: Message Queue Thread leyendo de una cola de mensajes con la información que se debe enviar al PC (Teraterm)
<b>Salida</b>	Método de sincronización: Message Queue Cada vez que se reciba una trama correcta desde el pc se enviará al módulo principal la información relevante para proporcionar la funcionalidad demandada
<b>Ficheros</b>	com.c y com.h

<b>Módulo PRINCIPAL</b>	Módulo principal del sistema que se encarga de coordinar todos los demás. Es el módulo que decide las acciones a tomar en función del modo del sistema y de la información que reciba del resto de los módulos
<b>Entradas</b>	Métodos de sincronización: Message Queue y Variables Globales Todas las salidas del resto de los módulos
<b>Salidas</b>	Métodos de sincronización: Message Queue y Thread Flags Todas las entradas al resto de los módulos
<b>Ficheros</b>	principal.c y principal.h

Estas especificaciones se deben implementar de manera obligatoria para superar el diseño. Todos aquellos detalles que no estén descritos previamente pueden ser implementados a criterio del estudiante aplicando fundamentalmente principios de usabilidad para la aplicación.

La calificación máxima que puede obtenerse implementando las opciones básicas de estas especificaciones es de **8 puntos**. Puede obtenerse una calificación superior implementando alguna de las mejoras que se indican a continuación. Como guía orientativa sobre la valoración de los principales ítems de evaluación, pueden considerarse las siguientes puntuaciones:

- El sistema no es robusto, se pierde el control del sistema: -5 puntos.
- No se ha implementado el envío/recepción de tramas: -5 puntos.
- No realizan lecturas de los parámetros entregados por el sensor: -5 puntos.
- No se ha implementado la modificación de parámetros con el joystick: -3 puntos.
- El joystick tiene rebotes: -3 puntos.
- No se ha implementado el buffer circular de medidas: -2 puntos.

### Mejoras del sistema (opcionales)

Con el fin de obtener un sistema con más prestaciones y de esta forma poder optar a una mejor calificación del trabajo, cada estudiante puede implementar, de manera opcional, una o varias de las mejoras que se describen a continuación. Para ello será necesario que se consulte con su profesor la forma de abordarlas.

La implementación de estas mejoras puede suponer modificaciones tanto en el código como en la estructura de los módulos ya desarrollados, por lo que se sugiere encarecidamente que se realice una copia de seguridad del sistema con las especificaciones básicas antes de acometer las posibles mejoras.

Cada una de las mejoras implementadas correctamente puede suponer un incremento máximo de la calificación del valor indicado.

También existe la posibilidad de que el estudiante proponga una mejora al margen de las que se detallan, siempre que se consulte antes con su profesor.

Mejoras propuestas:

- Lectura, visualización y almacenamiento de la velocidad angular proporcionada por el sensor: +1 punto.
- Implementación del modo de bajo consumo del sensor: +1 punto.
- Generación de una señal acústica cuando se supera algún valor de referencia en la aceleración: +1 punto.