

Sistemas Basados en Microprocesador

Bloque 3

**Desarrollo de un sistema de control de
iluminación ambiental**

Convocatoria extraordinaria

Curso 2023/2024

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 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 de este. 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.

Descripción del sistema

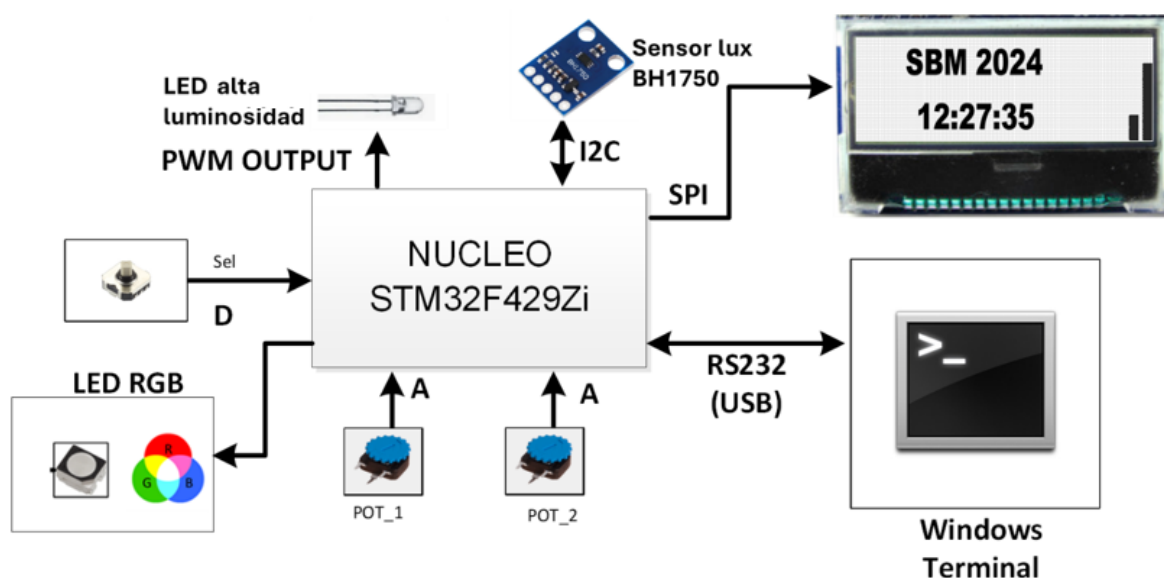
El diseño que se va a desarrollar es un **sistema de control de iluminación ambiental automático**.

Este sistema se encargará de **monitorizar de manera continua la luminosidad de un recinto, de compararla con un valor de referencia** y de **actuar sobre un sistema de iluminación para ajustar la luminosidad ambiental** al valor de referencia deseado.

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

- Tarjeta NUCLEO STM32F429Zi.
- Tarjeta *mbed App Board*.
- **Sensor de luz ambiental BH1750 de Rohm semiconductor**
- **1 diodo LED de alta luminosidad¹**

La figura 1 muestra un esquema simplificado del sistema.



¹ El hardware básico para el desarrollo del diseño (tarjeta STM32F429Zi y tarjeta *mbed App Board*) será cedido en préstamo por el Departamento. El sensor de luz ambiental BH1750 y el LED de alta luminosidad deberán ser adquiridos por los estudiantes.

Figura 1

Especificaciones

El funcionamiento del sistema debe responder a las siguientes especificaciones:

1. Es obligatorio el **uso del sistema operativo CMSIS RTOSv2** (RTX). La aplicación se construirá en torno a varios hilos/timers virtuales que tendrán que dar soporte a una o varias funcionalidades del sistema.
2. **En todo momento el sistema debe mantener la hora con una resolución de un segundo.**
3. **Entradas** al sistema:
 - a. **Sensor de luminosidad BH1750**, que proporcionará **medidas de luminosidad de manera periódica.**
 - b. Joystick, para poder interactuar con el sistema. La **gestión del joystick se realizará por interrupciones, eliminando los posibles rebotes** que éste origine, **distinguiendo entre pulsaciones cortas y largas.**
 - c. **Potenciómetros POT_1 y POT_2** cuya funcionalidad se detallará más adelante.
 - d. **UART para recibir comandos de programación y depuración.**
4. **Salidas** del sistema:
 - a. **LCD.** El usuario podrá conocer los parámetros del sistema a través de la información proporcionada por el LCD.
 - b. **LED RGB que mostrará diferentes estados** del sistema.
 - c. **Señal PWM.** Permitirá gobernar la luminosidad que genera el diodo **LED** de alta luminosidad del sistema.
 - d. **UART para responder a los comandos de programación y depuración.**
5. El sistema tendrá **cuatro modos de trabajo: REPOSO, MANUAL, AUTO, PROGRAMACIÓN.**

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 MANUAL.

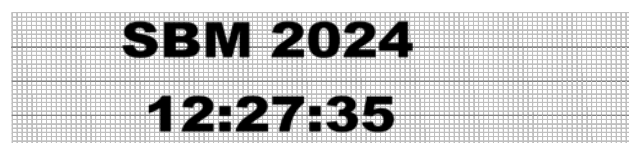


Figura 2. LCD en modo REPOSO

En modo REPOSO el LED RGB VERDE (mbed APPBoard) deberá estar encendido permanentemente con una luminosidad que variará entre un 10% y un 90% de su valor máximo. Cada barrido desde el 10% al 90 % y viceversa durará dos segundos.

- **Modo MANUAL.**

En este modo se estará leyendo la luminosidad del sensor BH1750² cada segundo (Lmed). En modo MANUAL la luminosidad de referencia (Lref) y la iluminación del LED de alta luminosidad se controlarán mediante los potenciómetros POT_1 y POT_2 de la tarjeta de aplicaciones, respectivamente.

El potenciómetro 1 permitirá establecer la luminosidad de referencia entre 1000 y 65000 lux utilizando todo su rango dinámico.

El potenciómetro 2 controlará el ciclo de trabajo de la señal PWM (5%-95%) aplicada al LED de alta luminosidad.

En modo MANUAL debe encenderse el LED ROJO de la mbed APPBoard si la diferencia entre la luminosidad medida por el sensor BH1750 y la de referencia es mayor de 5000 lux.

Cuando el sistema se encuentre en el modo MANUAL, en el LCD debe representarse una información similar a la de la Figura 3. En este caso, la luminosidad medida y la de referencia se representarán en las columnas de la parte derecha del LCD como barras verticales de 4 pixels de grosor cada una, donde una luminosidad de 65000 lux corresponderá a una barra completa y una luminosidad de 1000 lux corresponderá a una barra vacía. El ciclo de trabajo de la señal PWM aplicada al LED de alta luminosidad se mostrará en texto en el LCD.

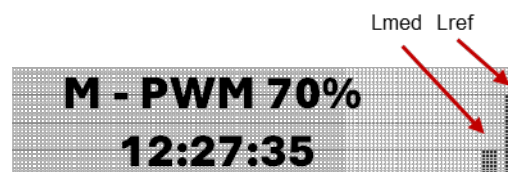


Figura 3. LCD en modo MANUAL

Desde el modo MANUAL podrá pasarse en cualquier momento al modo AUTO mediante una pulsación larga de la posición CENTER del joystick.

- **Modo AUTO.**

En este modo se estará leyendo la luminosidad del sensor BH1750 cada segundo (Lmed). Esta luminosidad se comparará con la de referencia (Lref) que se haya establecido en el sistema³.

En el modo AUTO el sistema debe gestionar de manera automática el LED de alta luminosidad para proporcionar una luminosidad ambiente (medida con el sensor BH1750) lo más cercana posible a la de referencia.

Cuando el sistema se encuentre en el modo AUTO, en el LCD debe representarse una información similar a la de la Figura 4.

² El sensor BH1750 proporciona valores de luminosidad de 1 a 65.535 lux

³ En el arranque del sistema se establecerá Lref = 23.000 lux

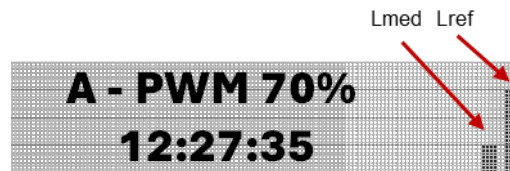


Figura 4. LCD en modo AUTO

Adicionalmente, el sistema debe mantener, cuando se encuentre en este modo, una tabla de 20 medidas almacenadas en un buffer circular. Cada entrada debe mantener en ASCII la siguiente información (hora, luminosidad medida, luminosidad de referencia y ciclo de trabajo de la señal PWM aplicada al LED de alta luminosidad):

HH:MM:SS--Lmed:xx.xxx--Lref:yy.yyy--PWM:zz%

En este modo debe encenderse el LED AZUL de la mbed APPBoard si la diferencia entre la luminosidad medida y la de referencia es mayor de 5000 lux.

Desde el modo AUTO podrá pasarse en cualquier momento al modo PROGRAMACIÓN mediante una pulsación larga de la posición CENTER del joystick.

- **Modo PROGRAMACIÓN.**

En este modo se programará la hora del sistema y la luminosidad de referencia (en el arranque del sistema se establecerá $L_{ref} = 23.000$ lux). La modificación de estos valores se realizará utilizando el joystick (pulsaciones cortas), presentando los cambios realizados en el LCD, tal y como se indica en la Figura 5.



Figura 5. LCD en modo PROGRAMACIÓN

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 deberán validar los valores introducidos para la hora y para la luminosidad de referencia (de 1000 a 65000, en pasos de 1000 lux).

Desde el modo PROGRAMACIÓN podrá pasarse en cualquier momento al modo REPOSO mediante una pulsación larga de la posición CENTER del joystick.

El sistema, cuando se encuentre en modo PROGRAMACIÓN, 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 9600 baudios, 8 bits de datos, 1 bit de stop y sin paridad. Es obligatorio que las comunicaciones serie se gestionen por 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 TOTAL 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 3.

Tabla 3. Comandos del PC al sistema⁴

Comando / CMD	Ejemplo	Comentario
Puesta en hora [0x20]	0x01 0x20 0x0C "15:20:15" 0xFE	Establece la hora a las 15:20:15. Si el equipo acepta la hora, éste devolverá la misma trama con el valor CMD en complemento a uno [0xDF]
Establece el valor de la luminosidad de referencia [0x25]	0x01 0x25 0x09 "35000" 0xFE	Establece el valor de la luminosidad de referencia en 35000 lux. El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xDA]
Todas las Medidas [0x55]	0x01 0x55 0x04 0xFE	Solicita todas las medidas almacenadas en el buffer circular. El equipo responde con tantas tramas del tipo "Medida" [0xAF] como medidas tenga almacenadas.
Borrar medidas [0x60]	0x01 0x60 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 podrá responder con las tramas especificadas en la Tabla 2.

⁴ Los datos entre comillas (" ") en esta tabla están representados en ASCII. Las comillas (" ") no forman parte de la trama.

Tabla 2. Respuestas del sistema al PC

Respuestas / CMD	Ejemplo	Comentario
Respuesta al comando <i>Puesta en hora</i> [0x20]	0x01 0xDF 0x0C "15:20:15" 0xFE	
Respuesta al comando <i>Establece el valor de la luminosidad de referencia</i>	0x01 0xDA 0x09 "35000" 0xFE	
Medida [0xAF]	0x01 0xAF 0x2F "HH:MM:SS-- Lmed:xx.xxx--Lref:yy.yyy--PWM:zz%" 0xFE	Devuelve el valor de una medida almacenada.
Respuesta al comando <i>Borrar medidas</i> [0x60]	0x01 0x9F 0x04 0xFE	

Conexiones del sistema

Como puede comprobarse del análisis de las especificaciones del sistema, el mismo estará compuesto por cinco elementos diferentes:

1. la tarjeta NUCLEO STM32F429Zi, que incorpora el microcontrolador,
2. la tarjeta *mbed app board*, que proporciona el LCD, el joystick, el LED RGB y los potenciómetros,
3. el sensor de luz ambiental BH1750,
4. el LED de alta luminosidad,
5. el PC que transmite/recibe tramas serie del sistema.

Las conexiones entre la tarjeta NUCLEO STM32F429Zi y la tarjeta *mbed app board* deben realizarse OBLIGATORIAMENTE siguiendo las indicaciones proporcionadas en la Tabla 3. El resto de las conexiones se dejan a criterio del estudiante⁵.

Tabla 3. Conexiones entre la *mbed app board* y la tarjeta NUCLEO STM32F429Zi

		Pin mbed app board	STM32F429	NUCLEO STM32F429Zi
VCC	+ 3.3V	DIP40	+ 3.3V	CN8.7 / +3.3V
GND	GND	DIP1	GND	CN8.11 / GND
LCD (SPI)	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
Joystick	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
LED RGB	R	DIP23	PD13	CN10.19 / D28
	G	DIP24	PD12	CN10.21 / D29
	B	DIP25	PD11	CN10.23 / D30
POT_1		DIP19	PA3	CN9.1 / A0
POT_2		DIP20	PC0	CN9.3 / A1

⁵ Verifique cuidadosamente en el manual de la tarjeta NUCLEO si las conexiones adicionales que plantea son compatibles antes de realizar ninguna conexión y encender el sistema. Si tiene dudas, pregunte a su profesor.

Módulos software del sistema

El software del sistema debe desarrollarse, al igual que se ha hecho durante el curso, 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).

Entregables y evaluación del diseño

Los entregables del diseño que deberán subirse obligatoriamente a Moodle antes del **domingo, 23 de junio de 2024, a las 23:59 horas** son:

1. **Memoria técnica** siguiendo el modelo que se encuentra en Moodle.
2. **Proyectos parciales** de Keil μ Vision que verifiquen el correcto funcionamiento de los diferentes módulos del sistema. Es obligatorio afrontar el diseño del sistema de manera modular e incremental, generando y almacenando los diferentes proyectos que tendrá que entregar en Moodle⁶.
3. **Proyecto final.**

La evaluación se realizará el **miércoles, 26 de junio de 2024**. Puede realizarse con anterioridad, previa solicitud al coordinador de la asignatura por correo electrónico. En cualquier caso, es requisito previo haber entregado toda la documentación y el software en Moodle.

⁶ Se proporciona en Moodle un proyecto con un ejemplo de un módulo de salida (*Plantilla ejemplo módulo salida (LEDS)*) y el programa de test que verifica su funcionamiento. Este ejemplo puede servirle de plantilla para el desarrollo de los módulos de su sistema.