

Sistemas Basados en Microprocesador

Bloque 3. Diseño

Integración y desarrollo de una aplicación de medida de iluminación

Convocatoria extraordinaria

Curso 2021/2022

**Departamento de Ingeniería Telemática y Electrónica
Universidad Politécnica de Madrid**

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 pretende desarrollar es un sencillo medidor de luminosidad que realiza medidas y las almacena en memoria.

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

- Tarjeta NUCLEO STM32F429Zi.
- Tarjeta mbed App Board.
- Módulo AdaFruit con sensor de luz TSL2591.

La tarjeta de aplicaciones y el sensor de luz serán cedidos en préstamo por parte del Departamento. El procedimiento para su reserva y devolución será comunicado por correo electrónico.

La figura 1 muestra un esquema simplificado del sistema.

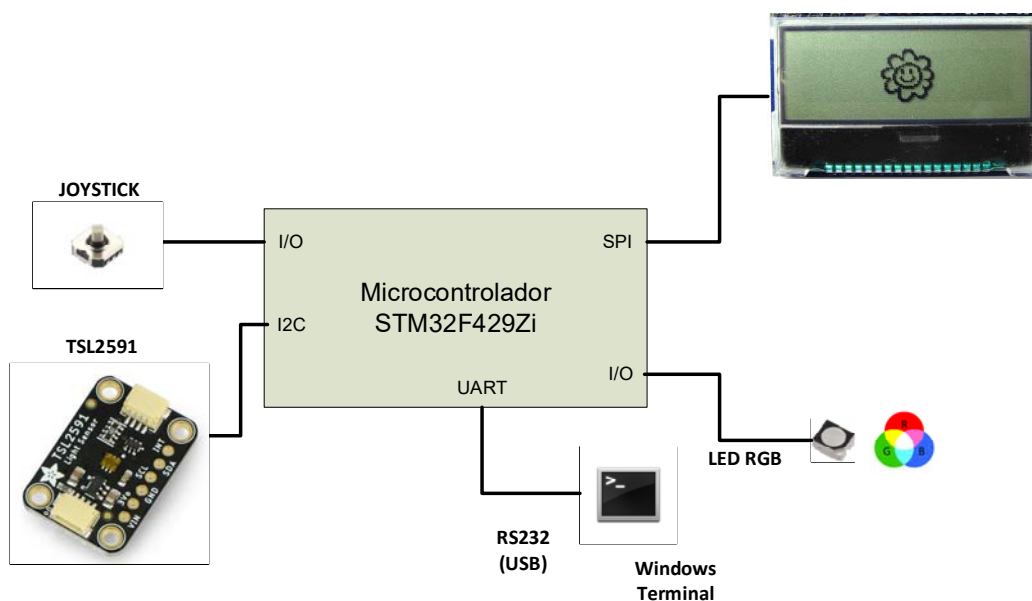


Figura 1

Especificaciones

El funcionamiento del sistema debe responder a las siguientes especificaciones:

- a) Es obligatorio el uso del sistema operativo CMSIS RTOS v2 RTX. La aplicación se construirá en torno a varios *threads/timers* que tendrán que dar soporte a una o varias funcionalidades del sistema.
- b) El sistema implementará por software un reloj que en todo momento mantendrá la hora con una precisión de un segundo y su valor se mostrará en la primera línea del LCD.
- c) El sistema dispondrá de un canal de comunicaciones serie RS232 que permitirá a un ordenador interactuar con el mismo.
- d) Se disponen de dos gestos del joystick (UP/DOWN) (de los que deberá eliminar los posibles rebotes), uno de los cuales se utiliza para realizar el cambio de modo del equipo (UP), mientras el otro se empleará para solicitar una medida (DOWN).
- e) El LED RGB VERDE (mbed APPBoard) deberá estar encendido permanentemente con una luminosidad variable, entre un 10% y un 90% de su valor máximo. Cada barrido desde el 10% al 90 % y viceversa durará dos segundos.
- f) El LED LD2 (NUCLEO STM32F429Zi) se empleará para indicar el estado y la toma de medidas, tal y como se especifica más adelante.
- g) Cada medida de luminosidad que realice el equipo se guardará en memoria RAM.
- h) El funcionamiento del sistema será el siguiente:
 - El sistema tiene tres estados de funcionamiento: INACTIVO, ACTIVO y MEDIDA.
 - Tras un reset o conexión de la alimentación el equipo estará en estado INACTIVO. Para salir de este estado y pasar al estado ACTIVO se deberá pulsar UP durante más de 1,5 segundos. Otra pulsación de más de 1,5 segundos en UP volverá a poner el equipo nuevamente en estado INACTIVO.
 - En el estado ACTIVO el LED LD2 parpadeará con la siguiente cadencia: 1 segundo encendido, 1 segundo apagado. En este estado el sistema estará esperando a que por el canal de comunicaciones RS232 le llegue una trama de configuración de hora. Si la trama es válida el sistema pasará a su estado de funcionamiento MEDIDA y apagará el LED LD2.
 - En el estado MEDIDA, el sistema puede realizar medidas:
 - a. De forma manual mediante el pulsador DOWN. Una pulsación dará comienzo al proceso de una medida manual de iluminación. El proceso de medida arranca con una cuenta atrás en segundos que deberá establecerse entre 2 y 9. Este valor también debe mostrarse en la segunda línea del LCD. La medida se realizará cuando la cuenta llegue a cero. Cada vez que el sistema realice una medida manual su valor se representará en la segunda línea del LCD y se producirá un destello de 350 ms en el LED LD3 (NUCLEO STM32F429Zi).
 - b. De forma automática. Para ello es necesario que el sistema reciba el comando correspondiente¹. En este caso, además de realizar el correspondiente destello de 350 ms en el LED LD3 con cada medida, debe escribirse una C, de ciclo, en la segunda línea del LCD junto al valor de la medida obtenida. Finalizado el ciclo se deberá borrar la segunda línea del LCD. No se atenderán peticiones de medidas manuales hasta acabar el ciclo de medidas automáticas.

¹ Ver tabla de comandos

- Todas las medidas realizadas se guardarán en memoria RAM, junto a su *timestamp*, en un buffer circular de 99 entradas. Toda la información de cada medida, hora y valor deberá estar codificada en ASCII, tal y como se indica en la entrada ejemplo a continuación, para una medida del 86,4% de luminosidad tomada a las 18:00:25.

Buffer	Buffer+1	Buffer+2	Buffer+3	Buffer+4	Buffer+5
18	00	25	8	6	4

- i) Calibración del sistema: Mediante una linterna o móvil se le aplicará luz próxima al sensor, el valor leído será considerado como el máximo a medir y se corresponderá con una luminosidad del 100%. Se tapará el sensor y el valor obtenido se considerará el 0%. El resto de los valores se interpolarán linealmente en el rango obtenido. El día de la verificación del funcionamiento del sistema se deberá utilizar la misma fuente de luz para así poder obtener valores en todo el rango útil de medida una vez calibrado el sistema.

Tramas/comandos.

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 19200 baudios, 8 bits de datos, 1 bit de stop y sin paridad. Las comunicaciones **no** pueden gestionarse por *polling*.

Para el envío de estas tramas/comandos se recomienda el uso del programa Teraterm. Dentro de este programa se pueden utilizar macros, que son ficheros de texto, en los que se definen cada uno de los bytes que componen dicha trama.

El formato de las tramas de comunicación será el indicado en la Tabla 3.

Tabla 3. 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 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, serán ignoradas por el sistema.

El PC podrá enviar diferentes comandos al equipo, especificados en la Tabla 4².

Tabla 4. Comandos del PC al sistema³

Comando / CMD	Ejemplo	Comentario
Puesta en hora [0x20] ⁴	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 cuenta atrás [0x25]	0x01 0x25 0x05 "8" 0xFE	Establece el valor de la cuenta atrás para lectura de la luminosidad en segundos. En este caso, 8 segundos. El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xDA]
Leer valor cuenta atrás [0x35]	0x01 0x35 0x04 0xFE	Lee el valor de la cuenta atrás. El equipo responde con la trama "Respuesta cuenta atrás".
Ciclo Medidas [0x70]	0x01 0x70 0x0A "NxxTyy" 0xFE	Arranca un ciclo de medidas automáticas, donde xx se corresponderá con el número de medidas a realizar (entre 01 y 99) e yy con el tiempo entre medidas expresado en segundos (entre 01 y 99)
Número de medidas almacenadas [0x40]	0x01 0x40 0x04 0xFE	Lee el número de medidas almacenadas. El equipo responde con la trama "Medidas almacenadas"
Ultima medida [0x50]	0x01 0x50 0x04 0xFE	Solicita la última medida almacenada. El equipo responde con la trama "Medida"
Todas las Medidas [0x55]	0x01 0x55 0x04 0xFE	Solicita todas las medidas almacenadas. El equipo responde con tantas tramas del tipo "Medida" como medidas tenga almacenadas.
Borrar medidas [0x60]	0x01 0x60 0x04 0xFE	Borra las medidas almacenadas. Cuando se ha borrado la memoria el equipo responde con la misma trama con el valor CMD en complemento a uno [0x9F]

² No serán tenidas en cuenta las tramas que se reciban mientras se esté realizando una medida manual o un ciclo de medidas.

³ Los datos entre comillas (" ") en esta tabla están representados en ASCII. Las comillas (" ") no forman parte de la trama.

⁴ El comando "puesta en hora" no se atenderá si la memoria de medidas tiene datos.

El sistema podrá responder con las tramas especificadas en la Tabla 5.

Tabla 5. Respuestas del sistema al PC⁵

Respuestas / CMD	Ejemplo	Comentario
Respuesta al comando <i>Puesta en hora</i> [0x20]	0x01 0xDF 0x0C "15:20:15" 0xFE	Si el equipo acepta la hora, devolverá la misma trama con el valor CMD en complemento a uno [0xDF]
Respuesta al comando <i>Establece el valor cuenta atrás</i> [0x25]	0x01 0xDA 0x05 "8" 0xFE	El equipo contesta con la misma trama y el valor CMD en complemento a uno [0xDA]
Respuesta cuenta atrás [0xCA]	0x01 0xCA 0x05 "4" 0xFE	Devuelve el valor de la cuenta atrás. En este ejemplo 4 segundos.
Medidas Almacenadas [0xBF]	0x01 0xBF 0x06 "20" 0xFE	Devuelve el número de medidas almacenadas. En este caso 20 medidas.
Medida [0xAF]	0x01 0xAF 0x12 "18:00:25-86.4%" 0xFE	Devuelve el valor de una medida almacenada. En este caso, 86.4% tomada a las 18:00:25
Respuesta al comando <i>Borrar medidas</i> [0x60]	0x01 0x9F 0x04 0xFE	El equipo responde con la misma trama con el valor CMD en complemento a uno [0x9F]

⁵ Los datos entre comillas (" ") en esta tabla están representados en ASCII. Las comillas (" ") no forman parte de la trama.

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, el joystick y el LED RGB; 3) la tarjeta "TSL2591", que incorpora el sensor de luminosidad y 4) el PC al que llegan las tramas RS232 del sistema (que adicionalmente servirá para la programación y depuración del mismo).

Las conexiones entre todos estos elementos deben realizarse OBLIGATORIAMENTE siguiendo las indicaciones proporcionadas en la **Tabla 1** y la **Tabla 2**.

Tabla 1. 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
	Up	DIP15	PB10	CN10.32 / D36
LED RGB	R	DIP23	PD13	CN10.19 / D28
	G	DIP24	PD12	CN10.21 / D29
	B	DIP25	PD11	CN10.23 / D30
		Pin TSL2591	STM32F429	NUCLEO STM32F429Zi
TSL2591	VCC	1	+5V	CN8.9 / +5V
	GND	3	GND	CN8.11 / GND
	SCL	4	PB8	CN7.2 / D15
	SDA	5	PB9	CN7.4 / D14

Tabla 2. Conexiones entre el módulo TSL2591 y la tarjeta NUCLEO STM32F429Zi

		Pin TSL2591	STM32F429	NUCLEO STM32F429Zi
TSL2591	VCC	1	+5V	CN8.9 / +5V
	GND	3	GND	CN8.11 / GND
	SCL	4	PB8	CN7.2 / D15
	SDA	5	PB9	CN7.4 / D14

Módulos 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 proponen los siguientes módulos, cuya estructura se muestra en la Figura 22.

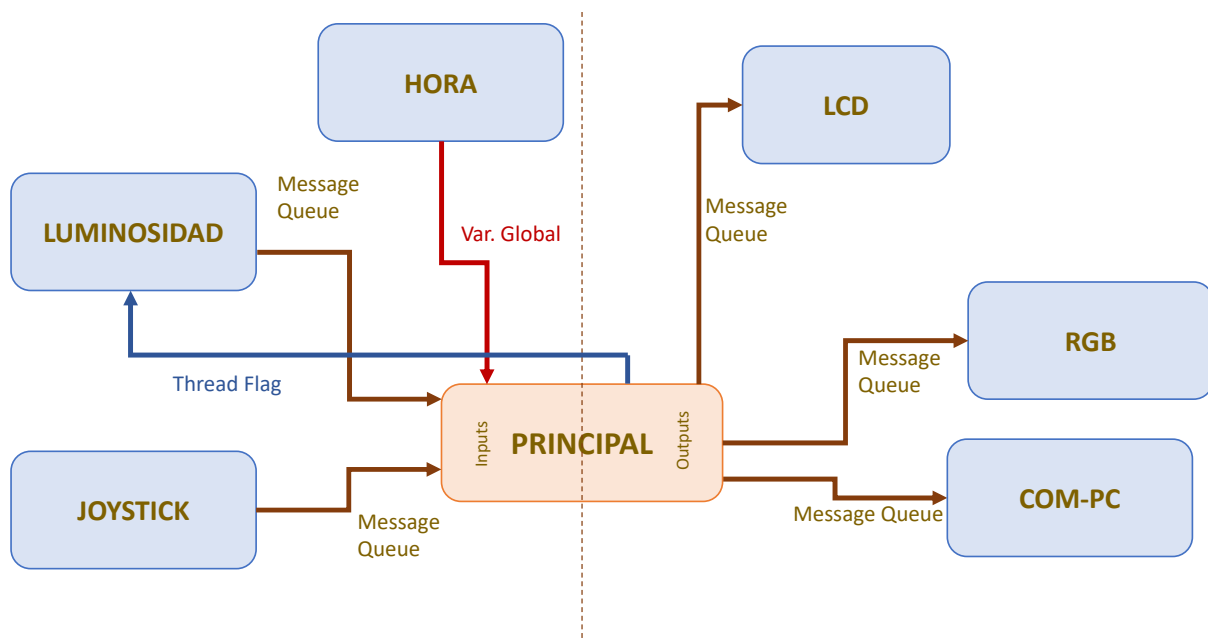


Figura 2. Estructura de los módulos que componen la aplicación

Módulo LCD	Módulo encargado de representación de información por el LCD conectado al bus SPI
Entrada	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
Salida	Ninguna
Ficheros	lcd.c y lcd.h

Módulo LUMINOSIDAD	Módulo encargado de la lectura de la luminosidad proporcionada por el módulo TSL2591
Entrada	Thread Flags
Salida	Método de sincronización: Message Queue Thread leyendo de una cola de mensajes las peticiones del valor de la luminosidad.
Ficheros	TSL2591.c y TSL2591.h

Módulo JOYSTICK	Módulo encargado de detectar e identificar una pulsación en el joystick
Entrada	Ninguna
Salida	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
Ficheros	joystick.c y joystick.h

Módulo COM-PC	Módulo encargado de la comunicación con el PC a través de la línea RS232 integrada en el USB
Entrada	Método de sincronización: Message Queue

	Thread leyendo de cola de mensajes con la información que debe enviarse al PC (Teraterm)
Salida	Ninguna
Ficheros	com.c y com.h

Módulo LED RGB	Módulo encargado de la gestión de los LED RGB (mbed APPBoard) para informar del estado del sistema
Entrada	Método de sincronización: Message Queue Thread leyendo de cola de mensajes con la información del estado del sistema
Salida	Ninguna
Ficheros	rgb.c y rgb.h

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

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

Evaluación del diseño

La evaluación de este diseño se realizará el **jueves, 30 de junio de 2022**, si bien puede realizarse con anterioridad, previa solicitud al coordinador de la asignatura.

Adicionalmente a la implementación del diseño es necesario tener en cuenta que se debe realizar una memoria técnica al finalizar el mismo, siguiendo el modelo que se encuentra en Moodle.

Se recomienda afrontar el diseño del sistema de manera modular e incremental, **generando y almacenando los diferentes proyectos parciales, así como el proyecto de calibración** de Keil µVision que verifiquen el correcto funcionamiento de las diferentes partes del sistema.

La memoria técnica, junto a todos los proyectos parciales y el proyecto final de Keil µVision, deberán subirse a Moodle antes del **lunes, 27 de junio de 2022, a las 20:00 horas**.