

# **Sistemas Basados en Microprocesador**

## **Bloque 1 – Práctica 0**

**Creación de un proyecto para  
NUCLEO-STM32F429 utilizando  
KEIL-ARM-CMSIS**

## **Introducción y objetivos del diseño**

El objetivo de esta práctica es conocer y aprender cómo construir un proyecto en Keil ARM para la tarjeta NUCLEO STM32F429Zi, utilizando CMSIS (Common Microcontroller Software Interface Standard).

Además servirá para conocer las distintas partes de Keil ARM necesarios para obtener dicho proyecto. Este software ya está instalado adecuadamente en los laboratorios de la asignatura y es necesario que cada estudiante utilice exactamente, en sus instalaciones personales, las mismas versiones de cada elemento software.

En las transparencias de las clases del bloque 1 pueden encontrarse las referencias a dichas versiones. También puede encontrarse el procedimiento para crear un proyecto desde cero.

## **Material necesario**

- Tarjeta de desarrollo NUCLEO STM32F429Zi.
- Cable USB a micro-USB.

## **Ejercicio 1**

Complete la siguiente tabla revisando las distintas versiones para cada uno de los elementos referenciados de Keil MDK. Puede encontrar cómo localizarlas en las transparencias de clase.

	Versión puesto laboratorio	Versión equipo personal	Última versión
Keil µVision			
Keil::STM32F4xx_DFP (Device Family Pack)			
ARM::CMSIS			
ARM::CMSIS-Driver			
Keil::ARM_Compiler			
Keil::MDK-Middleware			
ARM C compiler 5			
ARM C compiler 6			
ST-Link Firmware version			

## Ejercicio 2

Cree un proyecto (P0\_0) en Keil uVision para la tarjeta de desarrollo, utilizando los elementos de CMSIS necesarios, y añada el siguiente código dentro y fuera de la función main().

```
static GPIO_InitTypeDef GPIO_InitStructure;

/* Private functions -----*/
/**
 * @brief Main program
 * @param None
 * @retval None
 */
int main(void)
{

    /* Configure the system clock to 168 MHz */
    SystemClock_Config();
    SystemCoreClockUpdate();

    /* Add your application code here
     */

    __HAL_RCC_GPIOB_CLK_ENABLE();

    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_PULLUP;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_VERY_HIGH;

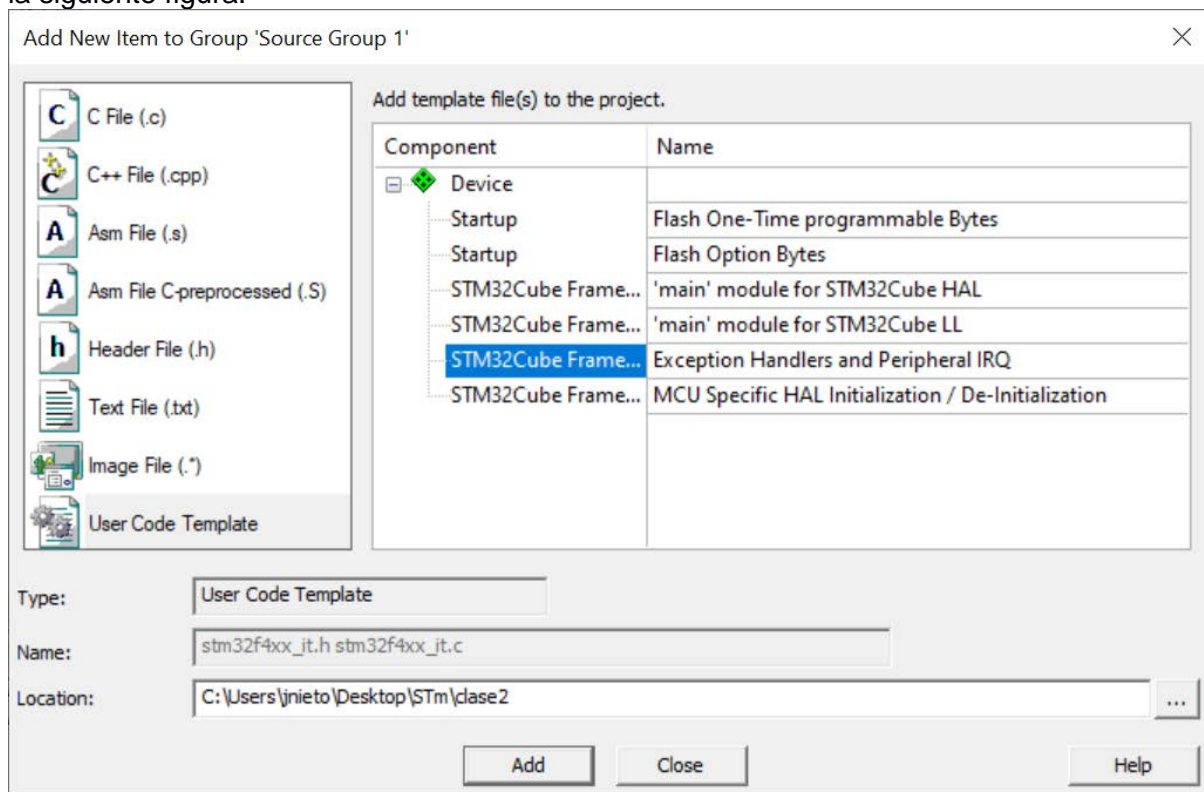
    GPIO_InitStructure.Pin = GPIO_PIN_0;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);

    GPIO_InitStructure.Pin = GPIO_PIN_7;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);

#ifdef RTE_CMSIS_RTOS2

    /* Infinite loop */
    while (1)
    {
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_0);
        HAL_Delay(100);
        HAL_GPIO_TogglePin(GPIOB, GPIO_PIN_7);
        HAL_Delay(100);
    }
}
```

Añada a su proyecto la plantilla básica de manejo de interrupciones, tal y como se indica en la siguiente figura:



Una vez editado, compilado y linkado el proyecto, ejecute la aplicación para determinar su funcionalidad.

Utilizando los documentos y ficheros existentes en su proyecto, justifique y analice cuál es el cometido de cada una de las líneas escritas en el código fuente de fichero principal.c. y añada los comentarios oportunos en cada línea. En este primer ejemplo no se necesita ningún elemento adicional a la tarjeta de desarrollo.

### **Ejercicio 3**

Añada al ejercicio anterior el código necesario para encender/apagar el led ROJO (LD3) de la tarjeta de desarrollo NUCLEO STM32F429Zi. Recuerde que toda la información de esta tarjeta se encuentra en el documento *“UM1974-NUCLEO-F429ZI-USER MANUAL.pdf”* disponible en Moodle.