Bloque 2

RESUMEN

El **BLOQUE 2** (***DESARROLLO BASADO EN MICROCONTROLADORES***) se divide en 5 temas. Para varios de dichos temas se adjuntan videos explicativos. Algunos de esos videos corresponden a una versión anterior de las transparencias y utilizando un entorno de desarrollo distinto (el Keil uVision 5 en lugar del STM32CubeIDE). Sin embargo, las explicaciones siguen siendo totalmente válidas, teniendo (para cualquier tipo de incongruencia) validez las transparencias publicadas. Los 5 temas son los siguientes:

En el **tema 4** (***Entorno de Desarrollo***) se enseña cómo instalar y utilizar el entorno de desarrollo necesario para realizar la parte práctica del curso, así como diferentes conceptos necesarios relacionado con él, dividido en varios puntos:

En el primer punto se habla del ciclo de desarrollo de cualquier proyecto.

A continuación se describe qué es un diagrama de flujo, qué tres componentes básicos tiene y se pone un ejemplo muy sencillo de aplicación.

En el tercer punto se explica la placa de desarrollo STM32L-DISCOVERY, con sus componentes, diagrama de bloques y layout.

Una vez vista la placa de desarrollo, se muestran los pasos necesarios para instalar y utilizar el software “STM32CubeIDE”, así como realizar una primera ejecución del software y cuáles son los pasos para trabajar con él.

Se continuará describiendo cómo crear una estructura básica de trabajo para, después, mostrar los pasos necesarios para crear un proyecto con ayuda de un ejemplo.

Tras haber descrito como hacer un proyecto, se muestran los pasos necesarios para depurarlo con ayuda del mismo ejemplo del punto anterior.

Como se podrá ver, el desarrollo se va a hacer en lenguaje C, pero es muy importante tener en cuenta ciertas peculiaridades a la hora de aplicar la programación en C a los microcontroladores. A ello le dedicaremos la penúltima sección del capítulo.

Finalmente se indican varias recomendaciones para el uso de la placa de desarrollo STM32L-DISCOVERY, con un detalle de conexión y un ejemplo de uso.

En el **tema 5** (***Pines de Entrada/Salida de Propósito General***) se enseña cómo utilizar el bloque GPIO del microcontrolador. Esto se hará concentrándose en los siguientes puntos:

En el primer punto se habla de los conceptos previos del bloque GPIO. A continuación se describe la funcionalidad del los pines.

En el tercer punto se explican los diferentes ***registros de control*** del bloque GPIO:

- **GPIOx→MODER**
- **GPIOx→AFR**
- **GPIOx→OTYPER**
- **GPIOx→OSPEEDR**
- **GPIOx→PUPDR**

En el cuarto punto se explican los diferentes ***registros de datos*** del bloque GPIO:

- **GPIOx→IDR**
- **GPIOx→BSRR**

Y en el quinto punto se indica que el bloque GPIO no tiene ***registros de estado***.

A continuación, se muestran las peculiaridades al usar los pines de la placa Discovery y cómo configurarlos con el software STM32CubeIDE, mediante la perspectiva CubeMX. También es muy importante ver cómo poder programar el funcionamiento de esos pines utilizando máscaras en C.

Finalmente se muestra un ejemplo práctico para probar por el alumno, con su programa correspondiente (inicialización y funcionamiento continuo).

En el **tema 6** (***Conversión Analógico/Digital y Digital/Analógico***) se enseña cómo utilizar el bloque ADC y DAC del microcontrolador, dividido en los siguientes apartados:

En el primer punto se habla de los conceptos básicos de la conversión analógico/digital (ADC) y la conversión digital/analógica (DAC).

En el segundo punto se describe el **BLOQUE ADC**, indicando primero sus características básicas y diagrama de bloques, después explicando sus diferentes modos de funcionamiento y finalmente sus diferentes registros de control, datos y estado.

- Los diferentes ***registros de control*** del bloque ADC son los siguientes:
  - **ADC→CR1**
  - **ADC→CR2**
  - **ADC→SQRx**
- El único ***registro de datos*** del bloque ADC es el siguiente:
  - **ADC→DR**
- El único ***registro de estado*** del bloque ADCes el siguiente:
  - **ADC→SR**

A continuación, se muestran las peculiaridades al usar el bloque ADC con el software STM32CubeIDE, en su perspectiva CubeMX.

Finalmente se muestran dos ejemplos prácticos (uno con conversión simple y otro con conversión continua) para probar por el alumno, con su diagrama de flujo, con su programa correspondiente (inicialización y funcionamiento continuo).

En el tercer punto se describe el **BLOQUE DAC**, indicando primero sus características básicas y diagrama de bloques y finalmente sus diferentes registros de control, datos y estado.

- El único ***registro de control*** del bloque DACes el siguiente:
  - **DAC→CR**
- Los diferentes ***registros de datos*** del bloque DAC son los siguientes:
  - **DAC→DHR12R1**
  - **DAC→DHR12L1**
  - **DAC→DHR8R1**
  - Y de la misma forma existen, para el canal 2: **DAC→DHR12R2, DAC→DHR12L2, DAC→DHR8R2**

- El único ***registro de estado*** del bloque ADC es el siguiente:
  - **DAC→SR**

A continuación se muestran las peculiaridades al usar el bloque DAC con el software STM32CubeIDE en su perspectiva CubeMX.

Finalmente se muestra un ejemplo práctico para probar por el alumno, con su programa correspondiente.

En el **tema 7** (***Interrupciones y EXTI***) se enseña cómo utilizar el bloque EXTI y cómo funcionan las interrupciones del microcontrolador. Este es uno de los temas más importantes del curso, ya que el correcto uso de interrupciones es lo que le aporta una gran potencia al desarrollo basado en microcontroladores. Este tema se va a dividir en los siguientes apartados:

En el primer punto se habla de los conceptos previos para la gestión de interrupciones.

A continuación se describe el Controlador de Interrupciones Vectorizadas (NVIC), se muestra la tabla resumen con las utilizadas en el cuso con su nombre, descripción, acrónimo, prioridad y posición en el vector, y por último los registros de 32 bits que se utilizan para controlarlo:

- **NVIC->ISER[x]**
- **NVIC->ICER[x]**

En el tercer punto se describe el **BLOQUE EXTI**, indicando primero sus características básicas y finalmente sus diferentes registros de control, datos y estado.

Los diferentes ***registros de control*** del bloque EXTI son los siguientes:

- **SYSCFG→EXTICR**
- **EXTI→IMR**
- **EXTI→RTSR**
- **EXTI→FTSR**

El bloque EXTI no tiene ***registro de datos.***

El único ***registro de estado*** del bloque EXTI es el siguiente:

- **EXTI→PR**

Finalmente se muestran dos ejemplos prácticos (uno por eventos y otro por IRQ) para probar por el alumno, con su programa correspondiente. El primero

de los ejemplos es utilizando eventos, mientras que el segundo es utilizando el mecanismo de interrupciones y su correspondiente Rutina de Atención a la Interrupción (RAI).

En el **tema 8** (***Temporización: TOC, PWM y TIC***) se enseña cómo utilizar el bloque TIMERS del microcontrolador. Este vuelve a ser un tema de vital importancia para el curso, ya que prácticamente la totalidad de los proyectos a afrontar van a tener que gestionar temporizaciones. Este tema va a estar dividido en varios puntos:

En el primer punto se habla de los conceptos previos del bloque TIMERS y finalmente mostrando dos ejemplos de las dos funcionalidades básicas de un temporizador (TOC y TIC).

En el segundo punto se describe la **arquitectura** del **BLOQUE TIMERS**, indicando primero sus características básicas y diagrama de bloques, después explicando su modo de funcionamiento y finalmente sus diferentes registros de control, datos y estado.

Los diferentes ***registros de control*** del bloque TIMERS son los siguientes:

- **TIMx→CR1**
- **TIMx→CR2**
- **TIMx→SMCR**
- **TIMx→DIER**
- **TIMx→EGR**
- **TIMx→CCMR1**
- **TIMx→CCER**

Los diferentes ***registros de datos*** del bloque TIMERS son los siguientes:

- **TIMx→CNT**
- **TIMx→PSC**
- **TIMx→ARR**
- **TIMx→CCRy**

El único ***registro de estado*** del bloque TIMERS es el siguiente:

- **TIMx→SR – Status Register**

A continuación se muestran las peculiaridades al usar el bloque TIMERS con el software CubeMX.

En el tercer punto se describe la **funcionalidad TOC** del **BLOQUE TIMERS** y finalmente dos ejemplos prácticos (uno sin interrupciones y otro con interrupciones) para probar por el alumno, con su programa correspondiente (inicialización y funcionamiento continuo para el ejemplo sin interrupciones y variables globales/RAI, inicialización y funcionamiento continuo para el ejemplo con interrupciones).

En el siguiente punto se describe la **funcionalidad PWM** del **BLOQUE TIMERS** y finalmente un ejemplo práctico para probar por el alumno, con su programa correspondiente (inicialización y funcionamiento continuo).

En el último punto se describe la **funcionalidad TIC** del **BLOQUE TIMERS** y finalmente dos ejemplos prácticos (uno sin interrupciones y otro con interrupciones) para probar por el alumno, con su programa correspondiente (inicialización y funcionamiento continuo para el ejemplo sin interrupciones y variables globales/RAI, inicialización y funcionamiento continuo para el ejemplo con interrupciones).