**Apuntes SBM**

En la clase de hoy vamos a ver la utilidad de la capa **HAL**, después vamos a ver el circuito de gestión del reloj (no podemos tener el reloj trabajando siempre) a esto de le llama **RCC**. Después veremos los **GPIO//Pines**, ya que hay algunos que son multifunción. Terminaremos con las interrupciones.

Hoy en día que existen cientos de microcontroladores, no tiene sentido tener que estudiar cada modelo, por eso se crean software que permiten reutilizar gran parte de de mi código sin necesidad de escribirlo de nuevo (HAL para STM).

Gracias ala capa HAL, no vamos a tener que escribir en lenguaje de bajo nivel. Tenemos un montón de librarías que usaremos (La capa LL es como la HAL de a un nivel mucho más bajo.

**Ventajas e inconvenientes:** Es portable, es fácil cambiar de micro. El código es mucho mas legible. El software, nos da una gran cantidad de librerías para una cantidad muy grande de periféricos.

Durante el curso vamos a usar la librería HAL para gestionar el reloj del sistema, GPIO, interrupciones, timers ,etc. Además, ampliaremos el uso de periféricos gracias a CMSIS Driver.

**Circuito de Reset y Control del reloj.**

Sentencias de Reset: Tenemos lo que llamados resets externos, que serían aplicar un nivel bajo de tensión que provoca un reset en el sistema, Además ciertas sentencias como WWDG reset (hay de muchos tipos) que también producen un reset. En muchas ocasiones el reset depende de la alimentación.

Diagrama, Esquemático

Descripción generada automáticamente

En este curso lo que nos interesa es que tenemos el botón externo de reset. Tenemos un pin NRST que, al pulsar el botón externo negro, ese pin se pone a 0 y produce la condición de reset.

¿Qué pasa cuando se ejecuta la opción de reset? El reloj interno de 16MHz, todos los pines vuelven a sus estados por defecto. Es el usuario que, a través del main() tiene que configurar el procesador y los periféricos. La primera condición que debe ejecutar el main() es *Hall\_init()*.

El consumo en un circuito digital depende de la frecuencia del reloj y el voltaje. Entonces reduciremos la velocidad de conmutación. Nosotros podemos elegir la frecuencia que llega a cada uno de los periféricos.

**Conceptos:**

High Speed Internal Clock (HSI): 16MHz (El micro funciona a esta frecuencia).

High Speed External Clock (HSE): 8MHz

Phase Locked Loop (PLL): Es un circuito cuya misión es multiplicar el reloj, para conseguir la frecuencia que a mí me interese.

Una captura de pantalla de un celular

Descripción generada automáticamente con confianza media(PLL)

Tendremos que escribir los valores de xN y /P para cambiar la frecuencia. Tenemos que jugar con ciertos valores (Hoja de referencia del fabricante).

La parte que aparece RCC en el main del código es la que tenemos que cambiar.

EN la pagina 20 del STM32f429-DATASHEET tenemos una imagen que nos indica que reloj de entrada es el que tenemos que utilizar.

En el programa principal encontramos 2 funciones: SystemClock\_Config() y SystemCoreClockUpdate(), si accedemos a *SystemCoreClock* que, si la visualizamos en decimal, puedo ver la frecuencia a la que está funcionando mi microprocesador en ese momento

**GPIO:**

Los pines son configurables, algunos. Los pines tienen diferentes funciones, entradas de salida digital entrada o salida, funciones especificas para diferentes periféricos (Es decir, diferentes funciones) a esto se le llama Alternate function, pueden ser analógicos y digitales y elegir cual de las 2 queremos usar. ¡Algo muy importante es que un pin nunca se puede quedar flotante! En este caso utilizo pull-up o pull-down (Diapositiva 3 de los GPIO).

En pagina 76 de STM32f429-DATASHEET podemos ver los pines y que timer tenemos que utilizar.

Para programar esto, vamos a utilizar el HAL:

1. Siempre utilizarenmos el HAL\_Init al inicio.
2. \_\_HAL\_RCC\_GPIOB\_CLK\_ENABLE(); Se utiliza para fijar el reloj del puerto

En las transparencias de GPIO tenemos ejemplos y relaciones de los esquemas con el código.

Interrupciones:

Cuando se produce una interrupción, el micro guarda la configuración del programa y el contador, salta a la rutina de servicio de interrupción. Los vectores de interrupción se encuentran en el entorno de Keil, en el fichero de Startup, tiene definido todos los vectores de interrupción