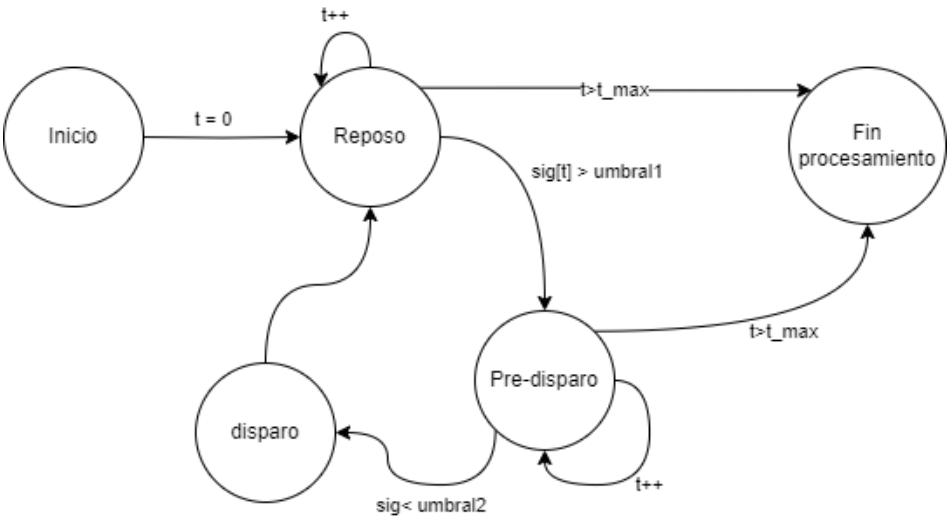
Práctica 0 - 2023

Repaso de C

1. Cree una función para “prender o apagar” bits de una variable. La función debe recibir una variable entera sin signo de 32 bits por referencia, un entero que indique el número de bit que debe modificarse de la variable, y otro entero que indique si debe ponerse en 1 o 0. Implemente un programa para demostrar el uso de la función.

2. Implemente un programa que cumpla con el siguiente procesamiento de un arreglo en forma de máquina de estados:



   Represente los estados con un `enum`, y el tiempo con un entero. El disparo debe producir una impresión en pantalla que muestre el tiempo de ocurrencia del disparo. Puede utilizar los siguientes valores (que deberían producir un único disparo):

```
umbral1 = 30;
umbral2 = 25;
sig[] = {-0.1969, 0.7252, 1.3017, 0.6221, 1.5131, 0.6530, 2.2430,
0.1166, 6.9498, 2.3170, 3.1620,-7.9198,18.5878,64.9743, 0.7815, 7.3498,
2.5495, 9.0523,13.3169,26.7396,19.9011, 6.1930, 3.6618, 2.1840, 2.7444,
1.6324, 0.4408,-0.0509, 0.5014, 0.6186, 0.1427};
```

Repaso de circuitos

3. Dibuje el esquemático de un circuito para conectar un pulsador normal abierto, que en reposo esté a 0 V y al pulsarlo a 3.3 V. Realice una simulación temporal del circuito en cualquier software de simulación (circuitlab.com, TINA TI (Texas Instruments), LTSpice (Analog Devices)) reemplazando el pulsador por una llave controlada por tiempo. Implemente el circuito y realice una medida utilizando el osciloscopio. Registre si hay rebotes y la amplitud de la perturbación resultante.

4. Se desea generar un valor de tensión constante utilizando una salida PWM de 100 kHz, 50% de ciclo de trabajo y 0 a 3.3 V de rango.
   ¿Qué tensión se espera ver a la salida?
   Calcule dos filtros pasabajos RC con frecuencia de corte de (i) 1 kHz y (ii) 16 Hz y realice para cada uno una simulación de AC para evaluar si el cálculo fue correcto y una simulación temporal para observar la forma de onda de salida. Registre la amplitud del "ruido" resultante en cada caso.

Creación de proyectos con el IDE

Recomendaciones para el uso del kit de la cátedra

Cada kit contiene un **microcontrolador STM32F103C8T6** soldado a una placa de desarrollo llamada "Blue Pill". La placa contiene al microcontrolador (µC) y elementos accesorios para lograr una funcionalidad mínima, incluyendo pines que dan acceso al puerto de programación y alimentación para el sistema. **Notar que el conector micro-USB no sirve para programar el µC**, sino que puede usarse en un futuro cuando se quiera implementar una conexión a través de ese puerto.

La programación y depuración del firmware del µC se realiza utilizando el herramienta ST-LINK V2 que se encuentra en cada kit, la cual viene acompañada de 4 cables para conectar a la placa de desarrollo (PD)

La conexión entre el programador ST-LINK y la PD es sencilla ya que la función de los pines se encuentra etiquetada en ambos dispositivos. Sin embargo, **los pines no se encuentran en el mismo orden** y debe prestarse atención al conexionado.

Utilice el **cable extensor USB** para conectar la PD más fácilmente a la PC y evitar ejercer fuerzas sobre los pines de conexión.

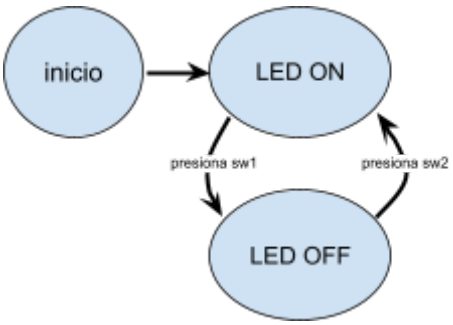
5. Cree un proyecto "Hola mundo" de sistemas embebidos. No avance al siguiente paso antes de realizar la comprobación pedida.

Paso	Descripción	Check
1	Cree un proyecto en el STM32CubeIDE para el microcontrolador STM32F103C8T6 utilizando el asistente Cube MX. Configure el GPIO correspondiente al LED como salida. Genere el código y compruebe poder compilar el código y descargarlo	
2	Programe el código para que el LED permanezca siempre encendido. Verifique que se prende	
3	Programe el código para que el led destelle permaneciendo en cada período medio segundo encendido y dos segundos apagado. Utilice las funciones HAL_GPIO_WritePin para modificar el estado del LED y HAL_Delay para generar los retardos. Verifique su funcionamiento.	

4	A partir del esquemático de la placa, calcule la corriente que consume el LED cuando el GPIO está nivel en alto y en bajo. Asuma que el LED encendido tiene una caída de 2V. Mida los valores.	

6. Agregando pulsadores

Paso	Descripción	Check
1	Seleccione dos pines en el esquemático de la placa tales que no estén conectados a ningún otro componente.	
2	Cree un nuevo proyecto como en el inciso anterior, configurando la salida para el led. Puede “copiar y pegar” el proyecto en el panel “Project Explorer” del IDE, pero debe cambiar manualmente el nombre al archivo .ioc para que coincida con el del nuevo proyecto y borrar el archivo .launch. Compruebe que puede compilar el proyecto.	
3	Configure dos GPIO como entradas. Configure uno con pull-up interno y el otro con pull-down interno. Lea el estado de los pines con la función HAL_GPIO_ReadPin y guarde en dos variables uint8_t globales SW1 y SW2. Utilizando el debug Compruebe que las variables toman los valores correctos según los pull up o down configurados.	
4	Conecte un pulsador a cada uno de los pines, asegurándose que cada pin cambie de estado al presionar el pulsador correspondiente y se corresponda con la respuesta lógica al depurar.	
5	Mejore el código anterior utilizando macros para definir los estados: PRESIONADO y NO_PRESIONADO. Compruebe que sigue funcionando	
6	Implemente la siguiente máquina de estados y verifique su funcionamiento	


inicio
LED ON
presiona sw1
presiona sw2
LED OFF