Soluciones Ejercicios Tema 4-4

- **Pregunta 1**: ¿Cuáles de los siguientes dispositivos utilizados en las prácticas eran accesibles a través del GPIO de la RPi?
    - Pulsadores. Correcto.
    - LEDs. Correcto.
    - Timer. Incorrecto, el timer es interno al procesador y no usamos GPIO.
    - Altavoz. Correcto.

- **Pregunta 2**: Queremos cargar en el registro r1 el comando de control que me permite configurar GPIO22 y GPIO27 como salida, comando que enviaremos al puerto GPFSEL2. Señala cuál de las siguientes instrucciones no da error de compilación:
    - ldr r1, =b00000000001000000000000001000000. Correcto.
    - mov r1, #b00000000001000000000000001000000. Incorrecto. Aunque sintácticamente el código es correcto. Va a producirse un error de constante no válida puesto que mov solo admite inmediatos de 8 bits. Sin embargo, le estamos pasando uno de 32 bits. En otras ocasiones esto funciona porque los puertos usan un rotador de bits para construir valores más grandes a partir de una secuencia de bits pequeña. Sin embargo, en este caso no se puede construir el valor que le estamos pasando y el ensamblador nos avisa del error.
    - ldr r1, #b00000000001000000000000001000000. Incorrecto, con ldr debemos usar = para definir un valor inmediato.
    - mov r1, =b00000000001000000000000001000000. Incorrecto, con mov debemos usar # para definir un valor inmediato.

- **Pregunta 3**: Cuando empiezan a ejecutarse los programas propuestos en el tema 4, el procesador de nuestra RPI 2 se encuentra en el siguiente modo:
    - SVC.

- **Pregunta 4**: Si quiero generar la nota do (256 Hz), el número (entero) de microsegundos que tengo que introducir como tiempo de espera en mi código cada vez que cambie el estado del altavoz es (pon todas las cifras significativas):
    - 1953. Lo primero es calcular el período de la nota (T = 1/F -> T=1/256=0.00390625s). Dado que queremos generar una onda cuadrada con encendido y apagado, dividimos el período en dos partes, una para encendido y otra para apagado, por lo tanto, calculamos el semiperiodo T/2=0.00390625/2=0.001953125s. Finalmente, como queremos microsegundos, multiplicamos por 1.000.000 y obtenemos 1953 microsegundos.

- **Pregunta 5**: ¿Cuál de los siguientes fragmentos de código nos permite inicializar correctamente la pila en modo supervisor para que nuestro programa main pueda hacer uso de la misma?
    - Código A. Incorrecto. Se guarda en r0 la máscara necesaria para activar el modo SVC pero el registro cpsr_c no se puede escribir con un mov normal, tenemos que usar la instrucción msr.

    - Código B. Incorrecto ya que la máscara no es para modo SVC.
    - Código C. Incorrecto porque la dirección de memoria con la que se inicializa el puntero de pila no es la correcta. Tendría que ser #0x8000000. Si usamos la que viene en este ejemplo, entraría en conflicto con la pila para IRQ.
    - Código D. Correcto.

- **Pregunta 6**: Se le ha pedido a alumnos de la asignatura Estructura de Computadores que preparen un código que implemente la siguiente funcionalidad: al arrancar la RPi deberá dejar el led rojo conectado al GPIO 10 parpadeando con una cadencia de 1 segundo. Suponiendo que la definición de las constantes en el fichero "inter.inc" se realiza correctamente, tal y como aparece en los ejemplos de clase: ¿cuál/es de los siguientes códigos es el correcto?.
    - Código A. Incorrecto porque la espera debe ser en microsegundos. Por lo tanto, como el tiempo de espera que usamos es 1, realmente estamos esperando un microsegundo.
    - Código B. Incorrecto porque los saltos a espera son incondicionales (b) por lo que la función no puede retornar por donde iba con bx lr. Tenemos que usar bl para actualizar el registro lr y poder retornar por donde iba el código.
    - Código C. Correcto.