

Ejercicio 4 Interrupciones

1.6 puntos

Un determinado microcontrolador está basado en un *core* ARM Cortex-M0. Dicho *core* cuenta con un NVIC (*Nested Vectored Interrupt Controller*). Entre los periféricos incluidos en dicho microcontrolador se encuentran: un *timer*, un ADC, un DAC y un GPIO, todos ellos capaces de generar interrupciones mediante sendas señales de petición de interrupción IRQ (IRQ_{TMR}, IRQ_{ADC}, IRQ_{DAC} y IRQ_{GPIO} respectivamente). Además, una circuitería externa al microcontrolador tiene acceso a la señal de interrupción no enmascarable NMI.

El programa corriendo sobre el procesador ha configurado los niveles de prioridad de las interrupciones configurables según la siguiente tabla:

Interrupción	Nivel de prioridad
IRQ _{TMR}	196
IRQ _{ADC}	128
IRQ _{DAC}	64
IRQ _{GPIO}	0

y, además, ha enmascarado todas la enmascarables, excepto las asociadas a IRQ_{TMR} y IRQ_{DAC}.

A su vez, las rutinas de atención a las interrupciones, ISR, asociadas a las distintas interrupciones presentan los tiempos de ejecución dados en la siguiente tabla:

Interrupción	Tiempo de ejecución de su ISR (μs)
NMI	0.50
IRQ _{TMR}	0.75
IRQ _{ADC}	1.25
IRQ _{DAC}	1.75
IRQ _{GPIO}	1.00

Todas estas IRQ y NMI son activas por flanco. Suponga que el tiempo de latencia para todas las interrupciones es despreciable.

El programa principal que el procesador está ejecutando es:

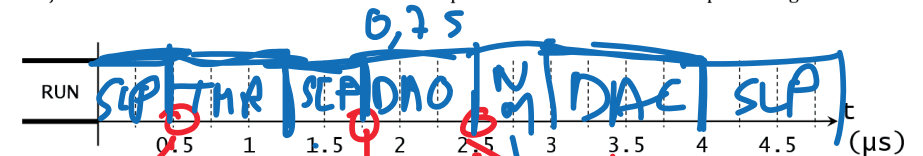
```

Loop
  CPSIE I ; __enable_irq()
  WFI B
Loop
    
```

Apartado A. En estas circunstancias se producen flancos activos en estas líneas IRQ y NMI en los instantes dados en la siguiente tabla:

Interrupción	Instantes en los que ocurren flancos activos (μs)
NMI	2.50
IRQ _{TMR}	0.50
IRQ _{ADC}	3.50
IRQ _{DAC}	1.75
IRQ _{GPIO}	4.00

Complete el gráfico indicando para cada momento qué ISR (NMI, TMR, ADC, DAC o GPIO) está ejecutando el procesador, en qué intervalos de tiempo el procesador está dormido (estos márkuelos como SLP, de *Sleep*) y en qué intervalos de tiempo se está ejecutando el programa principal sin que el procesador duerma (márkuelos como RUN). $t = 0$ s coincide con el instante en el que se ejecuta la instrucción **CPSIE I** del programa principal. Suponga que el tiempo de ejecución de cada instrucción individual es despreciable en la escala de tiempos de la gráfica.

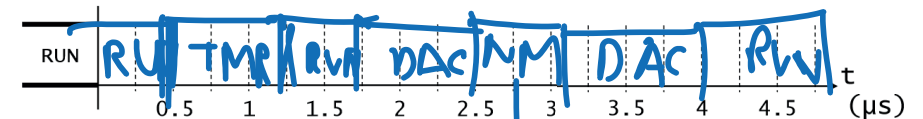


Apartado B. Repita el apartado anterior suponiendo ahora que el programa principal que se ejecuta es:

```

Loop
  CPSIE I ; __enable_irq()
  NOP B
Loop
    
```

De nuevo, $t = 0$ s coincide con el instante en el que se ejecuta la instrucción **CPSIE I** del programa principal.



Apartado C. ¿En cuál de los dos casos anteriores es previsible un menor consumo de energía por parte del procesador? Justifique su respuesta.

Ap. A pq duerme

Ejercicio 5	Periféricos
	1.6 puntos

Sobre el μ controlador de una placa *mbed* LPC1768 (o Nucleo-I432kc) está corriendo el siguiente código.

```
#include "mbed.h"
#include "pinout.h" // para A_PIN, ENA_PIN y PWM_PIN

#define BAUDRATE (9600)
#define US (1000000)

AnalogIn a(A_PIN);
InterruptIn ena(ENA_PIN);
Serial pc(USBTX, USBRX);
PwmOut pwm(PWM_PIN);

bool ena_rise_evnt;
bool ena_fall_evnt;

void ena_rise_isr(void) {
    ena_rise_evnt = true;
}

void ena_fall_isr(void) {
    ena_fall_evnt = true;
}

void main(void) {
    int32_t n = 13;
    pc.putc(BAUDRATE);
    pc.format(7, pc.None, 2);
    ena_rise(ena_rise_isr);
    ena_fall(ena_fall_isr);

    pwm.period_us( (6*US) / BAUDRATE );
    pwm.pulsewidth_us( US / BAUDRATE );

    while(true) {
        if(ena_rise_evnt) {
            ena_rise_evnt = false;
            n = a.read_u16() / 780;
            pwm.pulsewidth_us( (US*(n + 17)) / (21*BAUDRATE) );
        }
        if(ena_fall_evnt) {
            ena_fall_evnt = false;
            if(n >= 46) pc.putc(n);
        }
        __disable_irq();
        if(!ena_rise_evnt && !ena_fall_evnt) __WFI();
        __enable_irq();
    }
}
```

7 →
pc.None

7 → 2b → 1
2b = 0 →

$$CT = \frac{1}{6} \quad \frac{6}{9600}$$

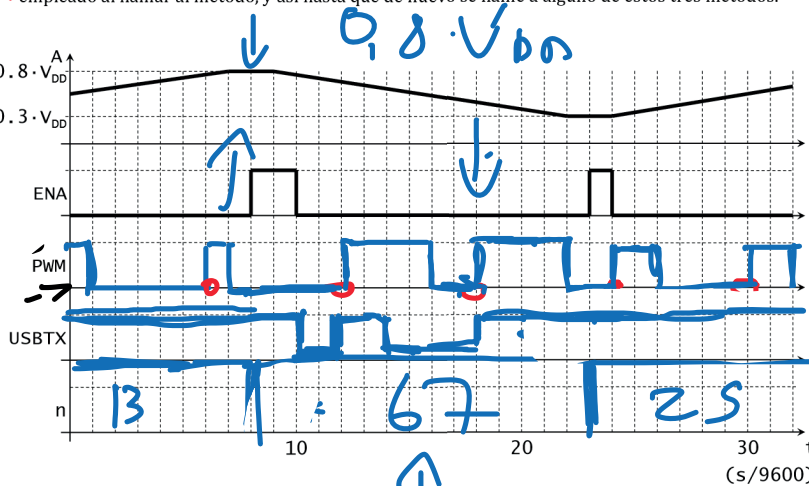
a.r / 780 25 + 17

$$\frac{1s \cdot 84}{21 \cdot 9600} = 4$$

(n >= 46) X
n = 25

En el siguiente cronograma se muestra la evolución de las señales de entrada A y ENA en los pines A_PIN y ENA_PIN. En este cronograma el eje de abscisas está graduado en unidades de $1/9600 \text{ s} \approx 104.67 \mu\text{s}$ y $t = 0$ coincide con el instante en que se entra en `main()`. La tensión de alimentación del μ controlador V_{DD} es 3.3 V y coincide con la de referencia del ADC.

Complete el cronograma para las señales PWM y USBTX (asociadas a los pines PWM_PIN y USBTX) y también para el valor de la variable n a lo largo del tiempo. Suponga que el tiempo necesario para la ejecución de cada una de las líneas de código del programa es despreciable en la escala de tiempos del cronograma.



0 - 3.3 V

0 - 65535

52428/780

n = 67

67 64 13

64 21 = 67

11 11 1000000