



**EJERCICIO 1 (2.0 puntos):**

Se tiene una arquitectura Von Neumann de 32 bits, con una capacidad de direccionar únicamente palabras, hasta un máximo de 64MB. Dicha arquitectura contiene 16 registros internos propósito general y filosofía Load & Store. Con esta información, conteste a las siguientes preguntas:

1) (70%) Diseñe una codificación de los distintos tipos de instrucciones microprocesador, minimizando en lo posible el tamaño de la instrucción ajustando a números enteros de palabras. Los tipos de instrucciones que debe tener el microprocesador, son:

- 5 instrucciones de transferencia de datos entre memoria y registros internos, con direccionamiento absoluto.
- 5 instrucciones de transferencia de datos entre memoria y registros internos, con direccionamiento indexado.
- 14 instrucciones aritmético/lógicas de operar entre registros, indicando en la instrucción tanto los registros operandos, como el registro que almacena el resultado.
- 14 instrucciones aritmético/lógicas de operar entre registros, con uno de los operandos limitado a 16 bits y dado por direccionamiento inmediato.
- 5 instrucciones de control con direccionamiento inherente
- 7 saltos condicionales con direccionamiento relativo a contador de programa, siendo el desplazamiento relativo de más/menos 1MB

2) (30%) Indique una respuesta justificada a cada una de las siguientes preguntas:

- a) Número mínimo de palabras que usa una instrucción
- b) Número máximo de palabras que usa una instrucción
- c) Tamaño del Registro de Instrucción
- d) Tamaño del Contador de Programa
- e) Tamaño de los Registros internos



## EJERCICIO 2 (4.0 puntos):

En las próximas hojas aparece el código de un programa a cargar en un sistema basado en un STM32L152RB, **el cual puede algún error de programación**. El programa corresponde a la implementación de un velocímetro de bicicleta, capaz de mostrar por la pantalla la información sobre la distancia recorrida, la velocidad instantánea y la temperatura. De dicho sistema se conocen las siguientes características:

- Tiene dispositivos conectados cuya señal de entrada al STM32L152RB, es un valor cualquiera entre 0 y 3,3V.
- Puede tener algún otro dispositivo más conectado.

Se pretende que el alumno analice dicho código y, a partir de ahí conteste **razonadamente** a las siguientes preguntas (*algunas de las justificaciones se pueden realizar indicando las líneas de código donde se encuentra la evidencia*). Se supone que las funciones "LCD\_Mostrar" y "MuestraVelocímetro" están programadas en la librería "Biblioteca\_SDM.h":

1. ¿Qué elementos (periféricos) del STM32L152RB se están utilizando? (10%)
2. Teniendo en cuenta que tiene una configuración de reloj que hace que el pclk de todos los periféricos vayan a 12MHz, indique la configuración de cada uno de los periféricos del microcontrolador cuando se encuentren en funcionamiento. Absténgase de simplemente decir el valor de cada registro de configuración; lo que se pide es la funcionalidad que se obtiene. Es imprescindible detallar la escala temporal (caso de que exista) que utilizan los periféricos. (20%)
3. De un significado a la variable "valor 3" del programa. (10%):
4. El código presenta un error muy sutil relacionado con la función "hora\_to\_str" (esto no tiene porqué significar que el error esté dentro del programa de la función). Encuéntrelo y dé una solución al mismo. (10%)
5. Explique brevemente lo que sucede en las líneas 47-48 y dibuja cómo van evolucionando los bits de la variable "valor 1" según se van ejecutando las instrucciones (10%)
6. Explique brevemente lo que sucede en las líneas 64-72 y dibuja cómo van evolucionando los bits de la variable "valor 2" según se van ejecutando las instrucciones (10%)
7. Realice el diagrama de flujo de todas y cada una de las funciones utilizadas, así como del programa principal. No haga siempre una transposición directa del código en un diagrama de flujo, sino represente la funcionalidad obtenida, mediante dicho diagrama de flujo. (20%).
8. ¿Qué configuraría adicionalmente en los registros de configuración de la USART del sistema para poder enviar cada 30 minutos por el PB6 y PB7 el valor medio de la velocidad a una memoria USB conectada al velocímetro y entre qué líneas las colocaría del programa actual (9600,8,N,1 y sin interrupciones). ¡¡¡No el programa de transmisión!!! (10%)



## ANEXO

```
1  #include "stm32l1xx.h"
2  #include "Biblioteca_SDM.h"
3  unsigned char radio=23;
4  unsigned tiempo_ant, tiempo;
5  unsigned char valor3;
6  unsigned distancia;
7  unsigned temperatura;
8  unsigned velocidad, diámetro;
9  int hora, minuto, cadena;
10 int diámetro = 2 * 3142 * radio;
11
12 void EXTI0_IRQHandler (void) {
13     EXTI->PR = 0x01;
14     NVIC->ICER[0] |= (1 << 6);
15     if ((EXTI->RTSR & 0x01) == 0) {
16         TIM4->CCR2 = TIM4->CNT + 2000;
17         TIM4->CR1 |= 0x0001;
18         EXTI->RTSR |= 0x01;
19         EXTI->FTSR &= ~(0x01);
20     }
21     else {
22         TIM4->CR1 &= ~(0x0001);
23         EXTI->FTSR |= 0x01;
24         EXTI->RTSR &= ~(0x01);
25     }
26     NVIC->ISER[0] |= (1 << 6);
27 }
28
29 void TIM4_IRQHandler (void) {
30     if (TIM4->SR & 0x0004 == 1) {
31         valor3 = 1;
32         TIM4->CR1 &= ~(0x0001);
33         TIM4->SR = 0x0004;
34     }
35     else {
36         distancia += diámetro;
37         tiempo = TIM4->CCR1 - tiempo_ant;
38         if (tiempo < 0) tiempo += 0xFFFFFFFF;
39         velocidad = distancia / tiempo;
40         tiempo_ant = TIM4->CCR1;
41         TIM4->SR = 0x0002;
42     }
43 }
44
```



```
45 void config_RTC(int hora, int minuto) {
46     int valor1 = 0;
47     valor1 = (((minuto / 10) << 4) + (minuto % 10)) << 8);
48     valor1 |= (((hora / 10) << 4) + (hora % 10)) << 16);
49     RTC->WPR=0xCA;
50     RTC->WPR=0x53;
51     RTC->ISR |= (1<<7);
52     while ((RTC->ISR & (1<<6))==0);
53     RTC->PRER=255;
54     RTC->PRER|=127<<16;
55     RTC->TR = valor1;
56     RTC->CR = 0x00000000;
57     RTC->ISR &= ~(1<<7);
58     RTC->WPR=0;
59     while ((RTC->ISR & (1<<6))!=0);
60 }
61
62 void hora_to_str (unsigned char *cadena) {
63     int valor2 = 0;
64     valor2 = RTC->TR;
65     valor2 = valor2 >> 8;
66     *(cadena+4)=(valor2 & 0x0000000F)+'0';
67     valor2 = valor2 >> 4;
68     *(cadena+3)=(valor2 & 0x00000007)+'0';
69     valor2 = valor2 >> 4;
70     *(cadena+1)=(valor2 & 0x0000000F)+'0';
71     valor2 = valor2 >> 4;
72     *cadena=(valor2 & 0x00000003)+'0';
73 }
74
75 int main (void) {
76     Init_SDM();
77     Init_LCD();
78     valor3 = 0;
79     GPIOA->MODER |= 0x00000300;
80     GPIOA->MODER &= ~(1 << (0*2 +1));
81     GPIOA->MODER &= ~(1 << (0*2));
82     GPIOA->PUPDR &= ~(11 << (0*2));
83     GPIOB->MODER |= 0x00000001 << (2*6 +1);
84     GPIOB->MODER &= ~(0x00000001 << (2*6));
85     GPIOB->AFR[0] &= ~(0x0F << (4*6));
86     GPIOB->AFR[0] |= 0x02 << (4*6);
87     ADC1->CR2 &= ~(0x00000001);
88     ADC1->CR1 = 0x02000000;
89     ADC1->CR2 = 0x00000472;
90     ADC1->SMPR1 = 0;
```



```
91  ADC1->SMPR2 = 0;
92  ADC1->SMPR3 = 0;
93  ADC1->SQR1 = 0x00000000;
94  ADC1->SQR5 = 0x00000004;
95  ADC1->CR2 |= 0x00000001;
96  while ((ADC1->SR&0x0040)==0);
97  ADC1->CR2 |= 0x40000000;
98  EXTI->FTSR |= 0x01;
99  EXTI->RTSR &= ~(0x01);
100 SYSCFG->EXTICR[0] = 0;
101 EXTI->IMR |= 0x01;
102 NVIC->ISER[0] |= (1 << 6);
103 TIM4->CR1 = 0x0000;
104 TIM4->CR2 = 0x0000;
105 TIM4->SMCR = 0x0000;
106 TIM4->PSC = 12000;
107 TIM4->CNT = 0;
108 TIM4->ARR = 0xFFFF;
109 TIM4->CCR2 = 0;
110 TIM4->DCR = 0;
111 TIM4->DIER = 0x0006;
112 TIM4->CCMR1 = 0x0001;
113 TIM4->CCMR2 = 0x0000;
114 TIM4->CCER = 0x0003;
115 TIM4->CR1 |= 0x0001;
116 TIM4->EGR |= 0x0001;
117 TIM4->SR = 0;
118 NVIC->ISER[0] |= (1 << 30);
119
120 config_RTC (16,30);
121 hora_to_str (cadena);
120 valor3 = 0;
121 while (1) {
122     if (valor3!=0) {
123         tiempo_ant = 0;
124         distancia = 0;
125         valor3 = 0;
126     }
127     while ((ADC1->SR & 0x0002)==0);
128     temperatura = (unsigned char) (ADC1->DR & 0x000000FF);
129     MuestraVelocimetro (velocidad, distancias, temperatura);
130     LCD_Texto (cadena);
131 }
132 }
```



### EJERCICIO 3 (4.0 puntos):

Con un microcontrolador STM32L152RB alimentado con 3V y pclk a 32MHz, se quiere diseñar un sistema de control de la depuradora de una piscina. Es importante tener en cuenta que la piscina es de  $50\text{m}^3$ , y que el flujo de agua a través de la depuradora es de  $10\text{m}^3/\text{h}$ . Por simplicidad del ejercicio se va a suponer que las condiciones del agua son homogéneas en todo su volumen (es decir, sus características son idénticas en la propia piscina, que en el caudal que está pasando por la depuradora). La depuradora de la piscina consta de las siguientes características:

- Control de encendido y apagado de la depuradora, con posibilidad de programar las horas de funcionamiento y la hora de encendido, a través de una parámetros que se le pueden dar al microcontrolador.
- Posibilidad de encendido y apagado manual de la depuradora a través de un interruptor que proporciona Vcc para encenderla y GND para apagarla.
- Control de los niveles de cloro:
  - Se cuenta con un sensor que indica el nivel de cloro en el agua mediante una tensión entre 0 y Vcc. El nivel óptimo de cloro vendrá dado por el parámetro OBJETIVO\_CLORO, y se considerará normal siempre que se encuentre en un margen del 10% arriba o abajo.
  - Se cuenta con un clorador salino que se activa mediante un tren de pulsos de periodo 1ms, el cual, cuanto mayor es el tamaño del pulso, más fuerza de cloración ejerce. El ancho del pulso nunca puede ser inferior 0,1ms ni mayor de 0,9ms. Se supone que si el pulso es de 0,5ms, se mantiene el nivel de cloro actual.
  - Si el cloro se desvía en más de un 50%, entonces se activará la alarma.
- Control del pH:
  - Se cuenta con un sensor de pH que da, mediante una tensión entre 1V y 2V, el valor de pH en decimales, entre 1,0pH y 12,0pH. El nivel óptimo de pH vendrá dado por el parámetro OBJETIVO\_PH, y se considerará normal siempre que esté dentro de las 2 décimas anteriores o posteriores al nivel de OBJETIVO\_PH.
  - Como el pH tiende siempre a subir, se dispone de un reductor de pH que se activa mediante una señal periódica de 1KHz. Esta señal se activará cuando haya que poner en marcha el reductor, manteniéndola a GND cuando no haya que actuar. Cuando se active, se podrá hacer a velocidad lenta (donde la señal periódica tendrá un ciclo de trabajo del 30%) o a velocidad alta (donde la señal periódica tendrá un ciclo de trabajo del 75%).
  - Si el pH baja de 5,5 o sube de 8,5, se activará la alarma.
- Control del nivel del agua:
  - Se detectará el nivel del agua a través de 3 sensores digitales (S1, S2 y S3) colocados en la pared de la piscina, en vertical, estando el S1 el más profundo y S3 el más cercano a la superficie. Cada uno de esos sensores, si está cubierto de agua (es decir, el nivel llega a la altura del sensor) da un 0 a su salida, mientras que si no está cubierto de agua, dará un 1.
  - Se podrán activar 2 llaves de paso de agua, mediante un par de salidas digitales. Si la salida digital es un 1, la llave se abre, dejando pasar el agua. Si es un 0, la llave se cierra, cortando el agua.
  - Si los tres sensores están cubiertos, las llaves deben estar cerradas. Si sólo S1 y S2 están cubiertos, se debe abrir una de las llaves de paso. Si tanto S2 como S3 se encuentran sin



nivel de agua, entonces se deben abrir las dos llaves de paso. Si los tres sensores se encuentran sin agua, entonces, se lanzará una alarma.

- Comunicación vía serie (a 19200, 8, N, 1) con una consola central, la cual le podrá enviar mensajes de configuración (de cada uno de los valores objetivos, de la hora actual, y de la hora de inicio y duración de la activación de la depuradora), y a la que el microcontrolador le podrá mandar el mensaje de alarma, compuesto por la hora, y el valor de todos y cada uno de los sensores. Ese mensaje se repetirá cada minuto, en el caso de que la situación de alarma no se corrija.
  - Tendrá Vd. que definir las tramas de comunicaciones.
- En la situación de alarma, deberá hacer sonar un pitido de 800Hz, intermitente cada segundo, mediante una salida digital entre 0 y Vcc. Este tono debe mantenerse hasta que se salga de la situación de alarma.
  - Al mismo tiempo, y tal como se ha comentado anteriormente, deberá enviar cada minuto un mensaje a la consola central.

Con esta información, y aquellas aproximaciones o decisiones que Vd. considere necesario (**siempre justificándolas**) conteste a las siguientes preguntas:

- a) Realice el diagrama de bloques de la solución (20%)
- b) Indique si utilizaría interrupciones, y en caso afirmativo, para qué y con qué funcionalidad. (10%)
- c) Diseñe las tramas de comunicaciones. (10%)
- d) Diga como configuraría los periféricos del microcontrolador utilizados, detallando el valor de los bits relevantes de los registros de control. (25%)
- e) Realice el diagrama de flujo de la solución (35%)