



UNIVERSIDAD CARLOS III DE MADRID
Sist. Dig. Basados en Microprocesador
8 de mayo de 2012

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
2º PARCIAL (75 minutos)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.*

APELLIDOS	NOMBRE	NIA

CUESTIÓN 1 (2.5 puntos):

a) (1.5 puntos) Se quiere que un STM32L152RB conectado a una placa STM32L_Discovery, con pclk1 de 16MHz, genere una señal de periodo 1ms, con ciclo de trabajo variable entre 10% y 90%. La señal deberá salir por el pin PB7. Realice la configuración inicial necesaria para empezar la generación de señal con un 10%.



UNIVERSIDAD CARLOS III DE MADRID
Sist. Dig. Basados en Microprocesador
8 de mayo de 2012

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
2º PARCIAL (75 minutos)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Se contestará sólo en el espacio reservado al efecto, pudiendo utilizar la cara posterior de la misma hoja.
No se pueden separar las hojas ni presentar hojas adicionales. Todas las respuestas deben estar justificadas.*

APELLIDOS	NOMBRE	NIA

b) (1.0 puntos) Ahora escriba las sentencias necesarias para que la señal generada cambie a tener un ciclo de trabajo del 90%.



CUESTIÓN 2 (2.5 puntos):

Para una aplicación determinada basada en un STM32L152RB y una placa STM32L_Discovery, funcionando con un pclk de 24MHz, se ha generado el siguiente código, el cual puede tener errores y/o ausencia de código.

```
1. #include "stm32l1xx.h"
2. #include "..\Biblioteca_SDM.h"
3.
4. int main(void){
5.     Init_SDM();
6.     GPIOB->MODER |= (0x01 << (2*9+1));
7.     GPIOB->MODER &= ~(0x01 << (2*9));
8.     GPIOB->MODER |= (0x01 << (2*8+1));
9.     GPIOB->MODER &= ~(0x01 << (2*8));
10.    GPIOB->AFR[0] = 0x77000000;
11.    USART1->CR1 = 0x0000000C;
12.    USART1->CR2 = 0x00000000;
13.    USART1->BRR = 0x00000068;
14.    USART1->CR1 |= 0x01 << 13;
15.    USART1->CR1 |= 0x01 << 5;
16.
17.    while (1) {
18.        while (valor==0);
19.        while ((USART1->SR & 0x0080)== 0);
20.        USART1->DR = valor;
21.        valor=0;
22.    }
23. }
```

Respecto a ese código conteste a las siguientes preguntas:

a) (1,0 puntos) ¿Qué periférico se está utilizando? ¿Cómo está configurado?



b) (0,5 puntos) ¿Hay algún error en el código? Si es así, solúcelo indicando las líneas donde se encuentra dicho error

c) (1,0 puntos) ¿Faltan líneas de código? Si es así, añádalas, indicando entre qué líneas las incluiría.



CUESTIÓN 3 (2.5 puntos):

Respecto al lenguaje ensamblador de un ARM, complete la siguiente tabla (cada fila vale 0,2+0,2+0,1 puntos):

INSTRUCCIÓN	MODO DE DIRECCIONAMIENTO	DESCRIPCIÓN EN RTL	REGISTROS QUE CAMBIAN SU VALOR
ADDS R2, R1, #0x0800			
MOVS R2, R6 LSL R1			
LDR R5, [R2, -R6]			
STR R3, [R2], #53			
STR R8, [R2, #3]!			



CUESTIÓN 4 (2.5 puntos):

Se tiene un microprocesador con las siguientes características:

- Arquitectura von Neumann
- 8 registros internos que pueden ser utilizados tanto para direcciones como para datos, sin poder acceder a la memoria de programa.
- Memoria de 1KB organizados en palabras de 8 bits
- Filosofía Load & Store

a) (1,5 puntos) Diseñe una codificación de los distintos tipos de instrucciones microprocesador, minimizando en lo posible el tamaño de la palabra de instrucción. Los tipos de instrucciones que debe tener el microprocesador, son:

- 10 instrucciones aritmético/lógicas de operar entre registros, contemplando que el resultado siempre se guarda en uno de los dos registros de los operandos
 - Ejemplo: ADD Rd, Rs; $Rd \leftarrow Rd + Rs$
- 7 instrucciones aritmético/lógicas con direccionamiento inmediato, donde el resultado se almacenará en el registro del otro operando.
 - Ejemplo: ADDI Rd, 115; $Rd \leftarrow Rd + 115$
- 5 instrucciones de salto absoluto condicional
 - Ejemplo: JMP <direccion>; $PC \leftarrow \langle \text{direccion} \rangle$
- 4 instrucciones de transferencia de información, con direccionamiento directo
 - Ejemplo: LD Rd, <direccion>; $Rd \leftarrow \langle \text{direccion} \rangle$
- 7 instrucciones de transferencia de información, con direccionamiento indirecto
 - Ejemplo: LDI Rd, Ri; $Rd \leftarrow (Ri)$



b) (1,0 puntos) Indique el tamaño de:
a) Registro de Instrucción

b) Contador de Programa

c) Registros internos