



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador. Cada ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas*

## **SOLUCIÓN**

### **EJERCICIO 1 (2 puntos, 30 minutos):**

Se ha de diseñar un microprocesador bajo los siguientes requisitos:

- Arquitectura Harvard
- Memoria de Programa de hasta 2Kpalabras de 8 bits (sólo direccionable a nivel de palabra)
- Memoria de Datos de hasta 1Kpalabras de 8 bits
- Arquitectura Load&Store: *todas las instrucciones trabajan únicamente con registros internos, salvo las de transferencia de información que mueven datos entre memoria y registros internos*
- 2 registros internos (que pueden ser utilizados tanto para almacenar datos, como direcciones de los datos)
- Las instrucciones pueden estar codificadas en una o en dos palabras de memoria de programa
- Instrucciones a contemplar:
  - 15 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;  $Rd \leftarrow Rd + Rs$
  - 15 instrucciones aritmético/lógicas con direccionamiento inmediato, donde el resultado se almacenará en el registro del otro operando. El dato será siempre del tamaño de 1 palabra.
    - Ejemplo: ADDI Rd, 115;  $Rd \leftarrow Rd + 115$
  - 4 instrucciones de salto absoluto condicional
    - Ejemplo: JMP <direccion>;  $PC \leftarrow \langle \text{direccion} \rangle$
  - 6 instrucciones de transferencia de información, con direccionamiento directo
    - Ejemplo: LD Rd, <direccion>;  $Rd \leftarrow \langle \text{direccion} \rangle$
  - 9 instrucciones de transferencia de información, con direccionamiento indirecto
    - Ejemplo: LDI Rd, Ri;  $Rd \leftarrow (Ri)$

Con todos estos datos, conteste a las siguientes preguntas de forma justificada (todas las preguntas son equivaloradas):

1. Tamaño de todos los buses de la CPU involucrados, salvo el bus de control

*Como es una arquitectura Harvard, hay dos mapas de memoria, por lo que hay cuatro buses, dos de datos y dos de direcciones:*

- *Memoria de Programa:*
  - *Bus de Direcciones: Como son 2K palabras, son 11 bits de bus de direcciones*
  - *Bus de Datos: Como son palabras de 8 bits, el bus de datos es de 8 bits.*



UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas

- **Memoria de Datos:**

- *Bus de Direcciones: Como son 1K palabras, entonces se trata de 10 bits de bus de direcciones*
- *Bus de Datos: como las palabras también son de 8 bits, el bus es de 8 bits.*

2. Tamaño de:

- a. Registro de Instrucción

*Atendiendo a la codificación anterior, se necesitan 16 bits (2 palabras)*

- b. Contador de Programa

*El contador de programa debe tener el tamaño del bus de direcciones de programa, es decir, 11 bits.*

- c. Registros internos

*Los registros internos tienen que tener datos y direcciones de datos. Por lo tanto deberán tener 10 bits*

3. Proponga una codificación cada uno de los tipos de instrucción que se han enumerado. Se valorará la utilización del menor número de palabras para cada tipo de instrucción.

*Como se tienen 2 registros sólo, se necesita sólo 1 bit para codificarlo  
15 o 9 instrucciones se codifican en 4 bits*

*6 instrucciones se codifican en 3 bits*

*4 instrucciones se codifican en 2 bits*

*Las direcciones de programa son 11 bits*

*Las direcciones de los datos son 10 bits*

*Con la codificación que se propone hay instrucciones que sólo ocupan una palabra y otras que ocupan dos palabras. Las que sólo usan una palabra utilizan*



UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
 Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas

*como bit más significativo un 0, y las de dos palabras, el bit más significativo de la primera palabra es un 1.*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
<i>Grupo1</i>									0	0	<i>opcode</i>				<i>Rs</i>	<i>Rd</i>
<i>Grupo2</i>	1	0	1	<i>opcode</i>			<i>Rd</i>	<i>dato</i>								
<i>Grupo3</i>	1	0	0	<i>opcode</i>	<i>Dirección de programa</i>											
<i>Grupo4</i>	1	1	<i>opcode</i>		<i>Rd</i>	<i>Dirección de datos</i>										
<i>Grupo5</i>									0	1	<i>opcode</i>				<i>Ri</i>	<i>Rd</i>



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas*

## **EJERCICIO 2 (4 puntos, 90 minutos):**

Con un microcontrolador STM32L152RB con pclk a 12MHz, sin tener por qué utilizar la placa STM32L\_Discovery, se quiere diseñar un sistema de control de iluminación para una sala de fiestas. Los requisitos que nos pone el cliente son los siguientes:

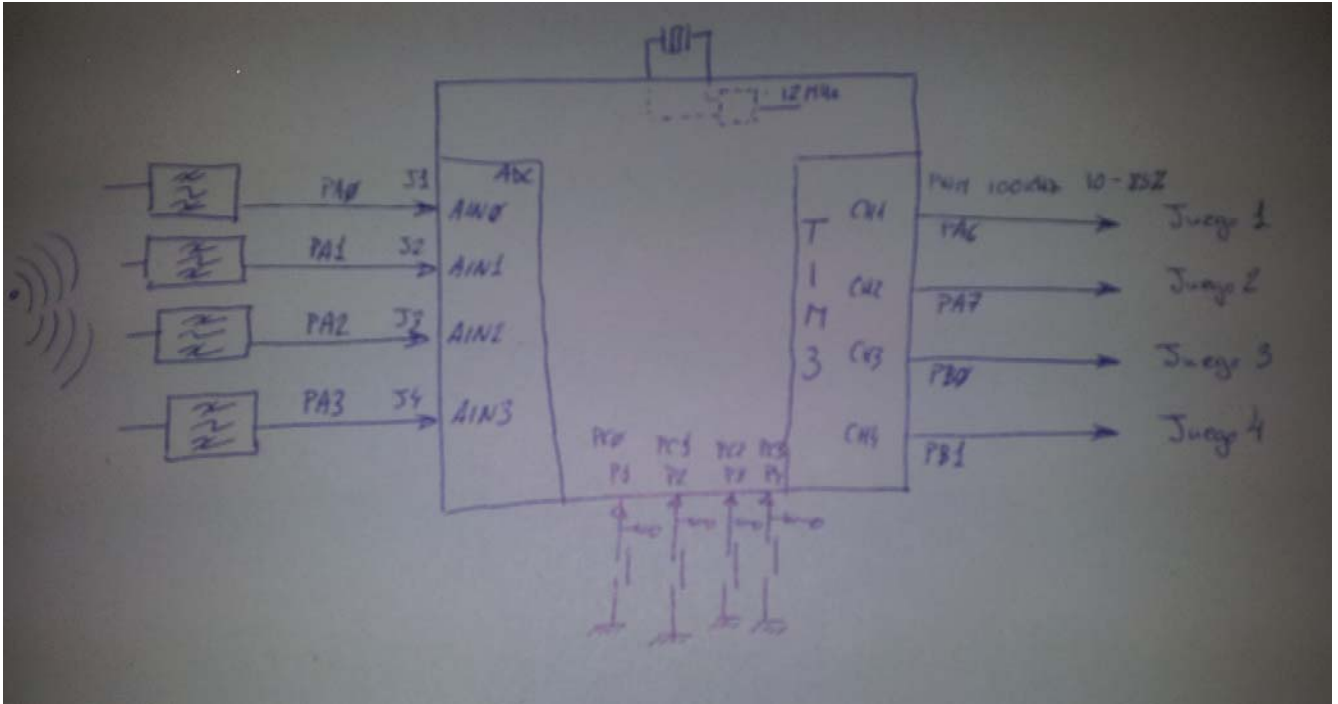
- El control de iluminación estará compuesto por 4 juegos de luces.
- Cada juego de luces se controla con una misma señal PWM de 100KHz, cuyo duty cycle debe estar siempre entre el 10% y el 85%.
- La potencia luminosa de cada juego de luces responderá linealmente la intensidad sonora procedente de una banda de frecuencias:
  - Juego J1: 50 – 800 Hz
  - Juego J2: 800 – 4000 Hz
  - Juego J3: 4000 – 10000 Hz
  - Juego J4: 10000 – 20000 Hz
- Se dispone de unos filtros paso-banda externos que proporcionan a su salida un valor de continua de la intensidad sonora cada banda, siendo dicho valor de continua entre 0 y 3V.
- Adicionalmente se necesita tener un pulsador por cada juego de luces que, durante el tiempo que esté pulsado, ponga el juego de luces pulsado a máxima potencia, y retorne a su valor anterior una vez soltado el pulsador.

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



No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas



b) Indique si utilizaría interrupciones, y en caso afirmativo, para qué y con qué funcionalidad.

*Las salidas por PWM no necesitan interrupciones, por lo que el programa simplemente se reduciría a examinar lo que ocurre en el ADC y en los pulsadores. No hay ninguna actividad adicional prioritaria y que haya que interrumpir, por lo que no son necesarias las interrupciones. Además hay que tener en cuenta que los pulsadores los acciona un ser humano y sólo tienen efecto en una parte visual, por lo que tampoco es realmente necesario considerar transitorios ni fuentes de interrupción externa (siempre se puede hacer una espera activa que atienda las pulsaciones de un usuario).*

c) Configure los periféricos utilizados

*TIM3: Se utilizará en configuración de TOC, con funcionalidad PWM, sacando las señales por sus cuatro canales, a través de PA6, PA7, PBO y PB1. Como*



No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador. Cada ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas

*tiene que ser una señal de 100KHz (10us), y se tiene que tener resolución del 5%, habrá que ser capaz de calcular intervalos menores de 0.5us. Como el pclk es de 12MHz, no se puede obtener una medida exacta del 5%, por lo que se puede apurar lo más posible, y limitar a que el duty-cycle no sobrepase el 85%. Si no se pone pre-escalado, las cuentas serán de 83ns. De esta manera:*

- *Ciclo de la señal (valor máximo del CNT) = 121 (redondeamos hacia arriba por seguridad)*
- *10% de duty-cycle = 13 (redondeamos hacia arriba por seguridad)*
- *85% de duty-cycle = 102*

*Por lo tanto:*

- *TIM3->CR1: ARPE=1, CEN=0, los demás a 0*
- *TIM3->CR2: todo a 0*
- *TIM3->SMCR: todo a 0*
- *TIM3->DIER: todo a 0 (sin interrupciones)*
- *TIM3->CCMR1 y TIM3->CCMR2: CCyS = 00, OCyCE=0, OCyM=110, OCyPE=1, OCyFE=0 (para y desde 1 hasta 4)*
- *TIM3->CCER: CCyNP=0, CCyP=0, CCyE=1 (para y desde 1 hasta 4)*
- *TIM3->PSC = 0 (sin pre-escalado)*
- *TIM3->ARR = 121 (el ciclo de la señal)*
- *TIM3->CCRY = 13 (para y desde 1 hasta 4)*
- *TIM3->SR = 0*
- *TIM3->EGR: UG=1 cuando haya que actualizar los registros (tras la inicialización y tras actualización del PWM)*
- *TIM3->CR1: CEN=1 (activa el contador)*

*Además habría que configurar los distintos pines para que funcionaran como AF, y dentro del AF, como TIM3/4:*

- *GPIOA->MODER|=0x00000001 << (2\*6 +1);*
- *GPIOA->MODER&=~(0x00000001 << (2\*6));*
- *GPIOA->AFR[0]=(0x02 << (6\*4));*



No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas

- $GPIOA \rightarrow MODER |= 0x00000001 \ll (2*7 + 1);$
- $GPIOA \rightarrow MODER \&= \sim(0x00000001 \ll (2*7));$
- $GPIOA \rightarrow AFR[0] = (0x02 \ll (7*4));$
- $GPIOB \rightarrow MODER |= 0x00000001 \ll (2*0 + 1);$
- $GPIOB \rightarrow MODER \&= \sim(0x00000001 \ll (2*0));$
- $GPIOB \rightarrow AFR[0] = (0x02 \ll (0*4));$
- $GPIOB \rightarrow MODER |= 0x00000001 \ll (2*1 + 1);$
- $GPIOB \rightarrow MODER \&= \sim(0x00000001 \ll (2*1));$
- $GPIOB \rightarrow AFR[0] = (0x02 \ll (1*4));$

*ADC: Hay que ponerlo en muestreo continuo, con múltiples canales (en concreto cuatro de ellos, siendo AIN0, AIN1, AIN2, AIN3), que estarán conectados al PA0, PA1, PA2, PA3, y que avise con cada conversión (no con cada secuencia). Las variaciones de frecuencia a medir serán, como mucho de 20KHz (máxima variación que percibe el oído), por lo que no es preciso un control tan exhaustivo de la medición. Habría que ver si se interrumpe en algo el cálculo interno, pero éste será solo una regla de tres para obtener el nuevo valor del PWM. En cuanto a la resolución, la medida va a imponer cambios en el DC del PWM, a nivel de tanto por uno, por lo tanto una serie de 75 valores. Esto implica que con una resolución de 8 bits es suficiente. Por lo tanto:*

- $ADC1 \rightarrow CR2: ALIGN=0, EOCS=1$  (aviso con cada conversión),  $DELS=000$  (sin retardo),  $CONT=1$  (conversión continua),  $ADON=0$  (apagado)
- $ADC1 \rightarrow CR1: OVR1E=0, RES=10, SCAN=0, EOCIE=0$  (sin interrupciones y a 8bits)
- $ADC1 \rightarrow SMPRx=0$
- $ADC1 \rightarrow SQR1: L=00011$  (4 canales)
- $ADC1 \rightarrow SQR5: SQ1=00000, SQ2=00001, SQ3=00010, SQ3=00011$
- $ADC1 \rightarrow CR2: ADON=1$
- $ADC1 \rightarrow CR2: SWSTART=1$  (cuando se quieran iniciar las conversiones)





UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas*

*Previamente se han tenido que configurar todos los pines (PA0 - PA3) como analógicos:*

- *GPIOA->MODER |= 0x0F;*

*GPIO: Se configurarán los pines PC0 - PC3 como pines de entrada digital. Su información será activa a nivel bajo, tal y como se ha hecho el diagrama de bloques, por lo que si está pulsado será un 0, y si no está pulsado será un 1.*

- *GPIOC->MODER &= ~(0x0F);*



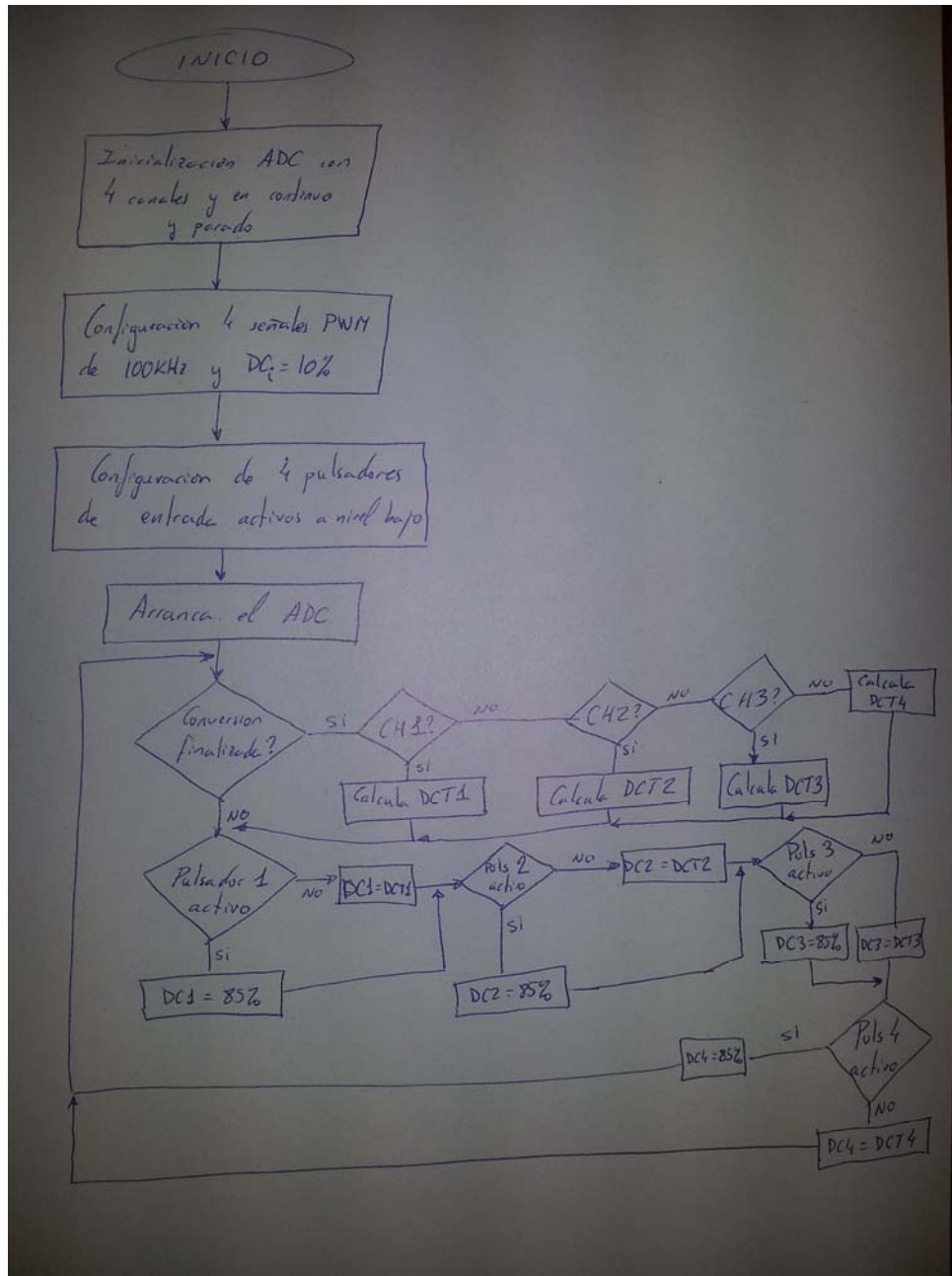


21 de junio de 2012

EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas

d) Realice el diagrama de flujo de la solución





UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador. Cada ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas*

- e) Si además se quisiera que las luces lucieran siempre de forma intermitente, con intervalos de tiempos modificables, ¿Cómo modificaría el diseño realizado?

*Si no se pudiese jugar con el periodo de las señales de PWM, lo mejor sería utilizar un temporizador adicional, en modo TOC, sin salida hardware, que en ciclos alternativos, activase y desactivase la salida del PWM, asegurándose que siempre que está desactivada, se deja a 0. La frecuencia de parpadeo sería la que habría que configurar como ese otro TOC adicional.*



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador. Cada ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas*

### **EJERCICIO 3 (4 puntos, 90 minutos):**

En las próximas hojas aparece el código de un programa a cargar en un sistema basado en un STM32L152RB. Sobre el código del anexo I, indique:

1. Configuración de los periféricos utilizados.

*PB6, PB7, PB2 Como GPIO de salida, con push-pull, velocidad por defecto.*

*PA0, PA11, PA12 Como GPIO de entrada sin pull-up, ni pull-down.*

*Los pines asociados al LCD, como corresponda (esto lo hace la librería Biblioteca\_SDM.lib).*

*Timer 4, TOC1 y NVIC*

- *Timer 4: Preescalado a 1 ms (16MHz), configurando TOC1, sin autoreload, parado.*
- *TOC1: Salta interrupción con match a los 5 segundos, sin salida hardware asociada.*
- *NVIC: Interrupción Timer 4 / TOC1*

2. Elementos que conforman el interfaz de usuario.

*El display, 3 leds y 3 pulsadores.*

3. Si existe alguna diferencia hardware entre los botones del dispositivo, explique cuál.

*Sí: PA0 recibe un 1 cuando se pulsa el botón asociado y un 0 cuando no se pulsa, sin embargo PA11 y PA12 funcionan al revés (0 cuando se pulsa, 1 cuando no se pulsa).*

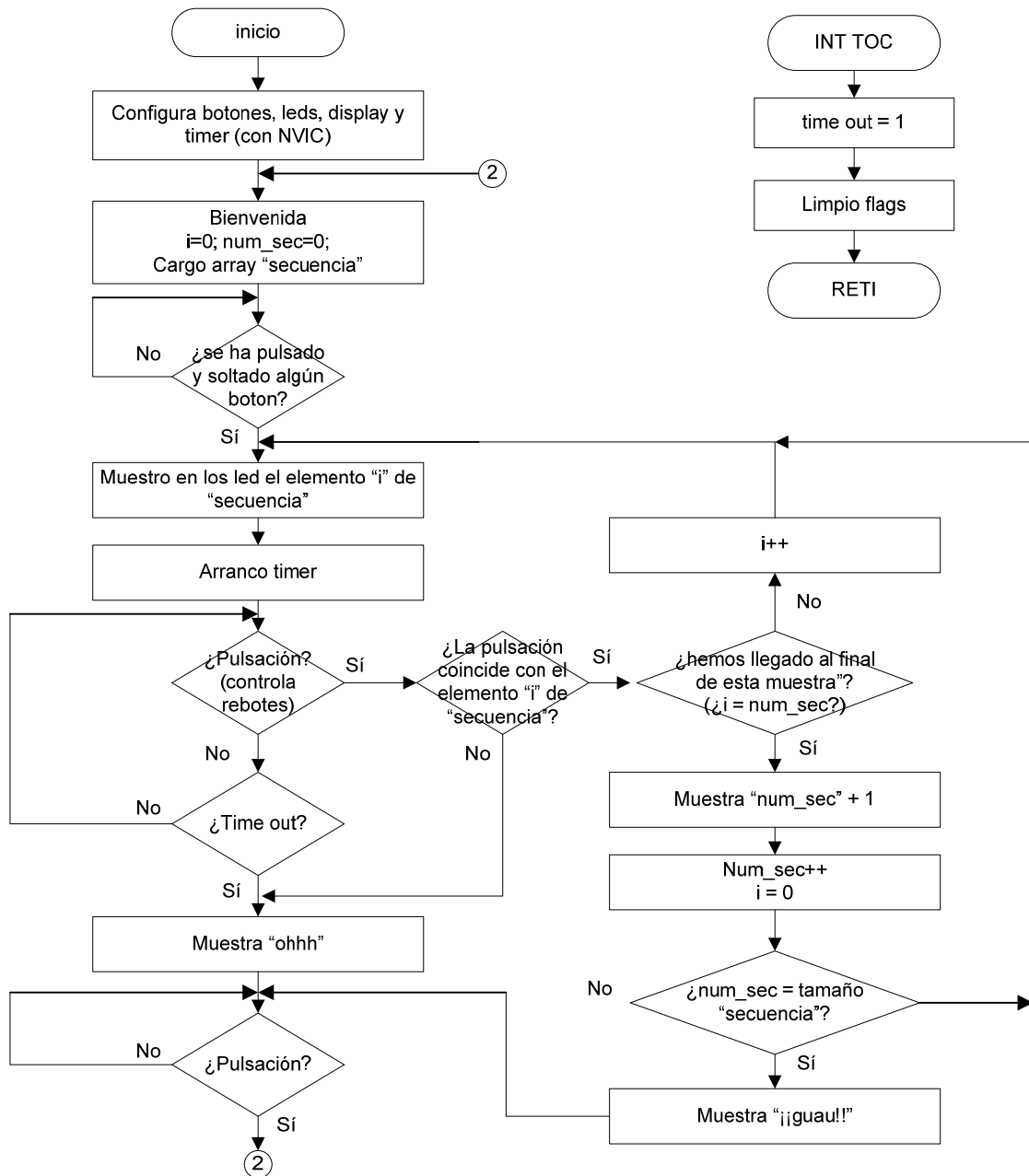
4. Diagrama de flujo.



21 de junio de 2012

EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas



5. Breve descripción general del programa (máximo 15 líneas).

Es un juego tipo "simon", de tres valores posibles (0, 1 y 2), cada uno con un led y un pulsador asociado. Se muestra el último valor de una secuencia (led) y el usuario debe



UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador. Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas*

*recordar todos los anteriores más este último (botones). El final se alcanza mediante fallo, o tras recordar exitosamente los 128 valores de la mayor secuencia (se comienza por una secuencia de 1 pulsación).*

**DETALLE:**

*El programa usa el LCD, 3 leds y 3 pulsadores como interfaz de usuario.*

*Inicialmente guarda una secuencia de 128 números aleatorios (entre 0 y 3), cada número está asociado a un botón y a un led.*

*Cuando dicha secuencia está preparada se muestra un mensaje en el display y se espera a que el usuario pulse un botón para comenzar.*

*El programa muestra una secuencia en los leds (muestra únicamente el último led de la secuencia) y el usuario debe repetir la secuencia completa con los botones.*

*Si acierta la secuencia se incrementa en un nuevo valor que se muestra mediante el led asociado para que el usuario pueda reconocer el siguiente elemento que debe incluir en sus pulsaciones.*

*La primera secuencia tiene un único elemento.*

*La última secuencia tiene 128 elementos, si el usuario la recuerda al completo, le muestra un mensaje indicando que ha ganado y espera que se pulse un botón para iniciar una nueva partida.*



*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere  
oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o  
pluma. Todas las respuestas deben estar justificadas*

## ANEXO I

```
1 #include "stm3211xx.h"
2 #include "Biblioteca_SDM.h"
3 #include <stdlib.h>
4 #include <stdio.h>
5
6 #define TIME_OUT 1
7 int tic = 0;
8
9 /*-----
10
11
12
13 -----*/
14
15 void config_perif(void){
16
17 //-----
18 GPIOB->MODER &= ~(1 << (6*2 +1));
19 GPIOB->MODER |= (1 << (6*2));
20 GPIOB->OTYPER &= ~(1 << 6);
21 //-----
22 GPIOB->MODER &= ~(1 << (7*2 +1));
23 GPIOB->MODER |= (1 << (7*2));
24 GPIOB->OTYPER &= ~(1 << 7);
25 //-----
26 GPIOD->MODER &= ~(1 << (2*2 +1));
27 GPIOD->MODER |= (1 << (2*2));
28 GPIOD->OTYPER &= ~(1 << 2);
29
30 //-----
31 GPIOA->MODER &= ~(1 << (0*2 +1));
32 GPIOA->MODER &= ~(1 << (0*2));
33 GPIOA->PUPDR &= ~(0x3 << (0*2));
34
35 //-----
36 GPIOA->MODER &= ~(1 << (11*2 +1));
37 GPIOA->MODER &= ~(1 << (11*2));
38 GPIOA->PUPDR &= ~(0x3 << (11*2));
39
40 //-----
41 GPIOA->MODER &= ~(1 << (12*2 +1));
42 GPIOA->MODER &= ~(1 << (12*2));
43 GPIOA->PUPDR &= ~(0x3 << (12*2));
44
45 }
46
47 #define AZUL 0
48 #define VERDE 1
49 #define ROJO 2
50
51 /*-----
52
53
54
```



UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o pluma. Todas las respuestas deben estar justificadas

```
55 -----*/
56
57 void set_led(int led){
58
59     // _____
60     GPIOB->BSRRH = (1<<6);
61     GPIOB->BSRRH = (1<<7);
62     GPIOD->BSRRH = (1<<2);
63
64     //_____
65     switch(led){
66
67         case AZUL:
68             GPIOB->BSRRL = (1<<6);
69             break;
70
71         case VERDE:
72             GPIOB->BSRRL = (1<<7);
73             break;
74
75         case ROJO:
76             GPIOD->BSRRL = (1<<2);
77             break;
78
79         default:
80             break;
81     }
82 }
83
84 #define          NO_PULSADO          -1
85 #define          BOTON_1             0
86 #define          BOTON_2             1
87 #define          BOTON_3             2
88
89 /*-----
90
91
92
93 -----*/
94 int get_botton(void){
95
96     int valor = NO_PULSADO;
97
98     //_____
99     if ((GPIOA->IDR&(1<< 0))!=0) valor = BOTON_1;
100    //_____
101    if ((GPIOA->IDR&(1<< 11))==0) valor = BOTON_2;
102    if ((GPIOA->IDR&(1<< 12))==0) valor = BOTON_3;
103
104    return(valor);
105 }
106
107 /*-----
108
109
110
```





UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere  
oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o  
pluma. Todas las respuestas deben estar justificadas*

```
111 -----*/
112
113 void espera_pulsacion(void){
114     while( get_botton() == NO_PULSADO);
115
116 }
117
118 /*-----
119
120 -----*/
121
122
123 -----*/
124
125 void TIM4_IRQHandler(void) {
126     if ((TIM4->SR & 0x0002)!=0) {
127         tic = TIME_OUT;
128         TIM4->CNT = 0;
129
130         //_____
131         TIM4->SR = 0x0000;
132
133         //_____
134         TIM4->CR1 &= ~0x0001;
135     }
136 }
137
138 /*-----
139
140 -----*/
141
142 -----*/
143
144 void config_timer(){
145
146     TIM4->CR1 = 0x0000;
147     TIM4->CR2 = 0x0000;
148     TIM4->SMCR = 0x0000;
149     TIM4->PSC = 32000;    //__t=_____
150
151     // _____
152     TIM4->CNT = 0;
153     TIM4->ARR = 0xFFFF;
154     TIM4->CCR1 = 5000;    //__t=_____
155
156     // _____
157     TIM4->DCR = 0;
158     TIM4->DIER = 0x0002;
159
160     // _____
161     TIM4->CCMR1 = 0x0000;
162     TIM4->CCMR2 = 0x0000;
163     TIM4->CCER = 0x0000;
164
165     // Habilitación de contador
166     // TIM4->CR1 |= 0x0001;
```



UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere  
oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o  
pluma. Todas las respuestas deben estar justificadas*

```
167 // TIM4->EGR |= 0x0001;
168 // TIM4->SR = 0; // Limpia flags
169
170 NVIC->ISER[0] |= (1 << 30); // Habilita IRQ
171 }
172
173 /*-----
174
175
176
177 -----*/
178
179 void start_timer(){
180
181 // _____
182 TIM4->CR1 |= 0x0001;
183
184 TIM4->EGR |= 0x0001;
185 TIM4->SR = 0; //_____
186
187 // resetea variable global
188 tic = 0;
189 }
190
191 int main(void){
192
193 int i=0; // posición del elemento actual
194 int max_secuencia = 0; // numero de elementos de la secuencia actual
195 char secuencia[128]="";
196 char mensaje[6]=""; // mensaje
197 int boton=0;
198 int fallo = 0;
199
200 Init_SDM();
201 config_perif();
202 config_timer();
203 Init_LCD();
204
205 while (1) {
206
207 // sin fallos
208 fallo = 0;
209
210 // carga secuencia (128 valores): 0, 1 y 2
211 for(i=0; i<128; i++){
212     secuencia[i] = (char)(rand() % 3);
213 }
214
215 // dile que empiece
216 LCD_Limpia();
217 LCD_Texto("Start");
218 espera_pulsacion();
219
220 // secuencia actual
221 for(max_secuencia=0; max_secuencia < 128; max_secuencia++){
222
```



UNIVERSIDAD CARLOS III DE MADRID (Dpto. de Tecnología Electrónica)  
**Sist. Dig. Basados en Microprocesador (Gr. Ing. Telemática)**

21 de junio de 2012 EXAMEN CONVOCATORIA EXTRAORDINARIA (3,5 horas)

*No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador  
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere  
oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas han de entregarse escritas en bolígrafo o  
pluma. Todas las respuestas deben estar justificadas*

```
223 // muestro el último valor de la secuencia
224 set_led((int)secuencia[max_secuencia]);
225
226 // _____
227 for(i=0; i<=max_secuencia; i++){
228
229     // arranco timer
230     start_timer();
231
232     do{
233         // _____
234         boton = get_botton();
235         if (boton != -1) break;
236
237     }while(tic != TIME_OUT);
238
239     // _____
240     // _____
241     if(( tic == TIME_OUT ) || (boton != secuencia[i])){
242
243         fallo = 1;
244         break;
245     }
246
247 }
248
249 // si falla en la secuencia
250 if (fallo){
251
252     LCD_Texto("ohh...");
253     espera_pulsacion();
254     break;
255
256 }else{
257
258     LCD_Limpia();
259     sprintf(mensaje,"%d", max_secuencia+1);
260     LCD_Texto((unsigned char *)mensaje);
261
262 }
263
264
265 // Si acierta todas las veces
266 if (fallo == 0){
267
268     LCD_Texto("guau!");
269     espera_pulsacion();
270
271 }
272
273 }
274
275 }
```