



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

APELLIDOS	NOMBRE	NIA
	SOLUCIÓN	

EJERCICIO 1 (2 puntos, 30 minutos):

Diseñe un microprocesador de arquitectura Von Neumann, con una memoria de 8 bits x 64K, indicando:

- Tamaño del bus de datos: _____ bits y del bus de direcciones: _____ bits
- Tamaño del PC: _____ bits y del MAR: _____ bits
- Si necesitamos gestionar 16 instrucciones, ¿Cuál es el tamaño del opcode? _____ bits
- Si contásemos con 4 registros de propósito general que podrían ser utilizados indistintamente como fuente y como destino, y partiendo de la información del apartado c) ¿Cuál sería el tamaño mínimo del IR? _____ bits
- Si el Registro de Instrucción tuviera una capacidad múltiplo de 8 bits, y nos pidiesen que implementásemos la siguiente instrucción:

MOV Reg, <dirección>

Reg ← (dirección)

- ¿Qué espacio mínimo ocuparía esta instrucción en memoria? _____ bits / _____ palabras
- ¿Podría ejecutarse en este micro? Sí No.
- Si fuera posible, indique la secuencia de pasos en sentencia RTL al mayor nivel de detalle.



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

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
EXAMEN FINAL (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

- f) Describa la secuencia de pasos en RTL con el mayor nivel de detalle para ejecutar la siguiente instrucción:

ADD Reg, <dirección_8bits>
Reg ← Reg + <dirección_8bits>

- g) ¿Qué limitación clara tendría la instrucción anterior? (responda en 1 línea)
-



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:

- Tamaño del bus de datos: 8 (memoria de 8bits) bits y del bus de direcciones: 16 (direccionamiento de 64K) bits
- Tamaño del PC: 16 (igual que el bus de direcciones) bits y del MAR: 16 (idem) bits
- Si necesitamos gestionar 16 instrucciones, ¿Cuál es el tamaño del opcode? 4 (para codificar 16 instrucciones) bits
- Si contásemos con 4 registros de propósito general que podrían ser utilizados indistintamente como fuente y como destino, y partiendo de la información del apartado c) ¿Cuál sería el tamaño mínimo del IR? 8 (si solo usase un registro y el opcode, se necesitarían 6 bits, que se completa hasta el tamaño de una palabra de 8 bits; si son dos registros y el opcode, entonces serían 8 bits; por lo tanto el tamaño mínimo debería ser el de una palabra de 8 bits) bits
- Si el Registro de Instrucción tuviera una capacidad múltiplo de 8 bits, y nos pidiesen que implementásemos la siguiente instrucción:

MOV Reg, <dirección>
Reg ← (dirección)

- ¿Qué espacio mínimo ocuparía esta instrucción en memoria? 24 (la dirección son 16 bits, por lo tanto al menos necesitaba otra palabra para ubicar el opcode, por lo tanto salen 24) bits / 3 palabras
 - ¿Podría ejecutarse en este micro? Sí No.
 - Si fuera posible, indique la secuencia de pasos en sentencias RTL al mayor nivel de detalle
 - Fetch: $MAR \leftarrow PC; MBR \leftarrow (MAR); IR_{23.16} \leftarrow MBR; PC \leftarrow PC + 1$
 - Decodificación:
 - Captura dirección 1: $MAR \leftarrow PC; MBR \leftarrow (MAR); IR_{15.08} \leftarrow MBR; PC \leftarrow PC + 1$
 - Captura dirección 2: $MAR \leftarrow PC; MBR \leftarrow (MAR); IR_{07.00} \leftarrow MBR; PC \leftarrow PC + 1$
 - Captura dato: $MAR \leftarrow IR_{15.00}; MBR \leftarrow (MAR); Reg \leftarrow MBR$
- f) Describa la secuencia de pasos en RTL con el mayor nivel de detalle para ejecutar la siguiente instrucción:

ADD Reg, <dirección_8bits>



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

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
EXAMEN FINAL (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

$\text{Reg} \leftarrow \text{Reg} + \langle \text{dirección_8bits} \rangle$

1. Fetch: $\text{MAR} \leftarrow \text{PC}; \text{MBR} \leftarrow (\text{MAR}); \text{IR}_{23..16} \leftarrow \text{MBR}; \text{PC} \leftarrow \text{PC} + 1$
2. Decodificación:
3. Captura dirección 1: $\text{MAR} \leftarrow \text{PC}; \text{MBR} \leftarrow (\text{MAR}); \text{IR}_{15..08} \leftarrow \text{MBR}; \text{PC} \leftarrow \text{PC} + 1$
4. Captura dato: $\text{MAR} \leftarrow \text{OO}; \text{IR}_{15..08}; \text{MBR} \leftarrow (\text{MAR})$
5. Ejecución: $\text{Reg} \leftarrow \text{Reg} + \text{MBR}$

g) ¿Qué limitación clara tendría la instrucción anterior? (responda en 1 línea)

Imposibilidad de llegar a cualquier posición de memoria (dirección < 0x0100).



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 2 (4 puntos, 90 minutos):

Basándose en el RTC, en dos GPIO y en el display, diseñe un reloj que muestre: HH:MM, con las siguientes funcionalidades:

1. Dos modos de funcionamiento:

- *Normal*: El reloj mostrará la hora en el formato indicado
- *Programación*: El reloj permitirá que se modifique horas y minutos.

2. El cambio de modo “programación” a “modo normal” y viceversa, se producirá pulsando los dos botones a la vez.

3. Uno de los botones servirá para moverse de forma circular entre horas y minutos cuando el reloj se encuentre en modo programación. Mientras permanezca en este modo, la hora no cambiará de forma automática.

4. El otro botón servirá para incrementar el parámetro seleccionado (horas o minutos) en modo circular (horas de 0 a 23 y minutos de 0 a 59).

SE PIDE:

a) Desarrolle la función de configuración y arranque del RTC: “void config_RTC(int hora, int minuto)” (hora: 0 - 23 , minuto: 0-59).

Nota: $LSE = 32.768 \text{ Hz}$

b) Escriba la función “void hora_to_str(unsigned char *cadena)”, que pase la hora del registro de tiempo del RTC a un string en formato “HH:MM”.

c) Diagrama de flujo de la función “int get_botones(void)” sabiendo que:

- Devuelve 0 si no se ha pulsado ningún botón.
- Devuelve 1, 2 o 3 si se ha pulsado y soltado el botón 1, el botón 2, o ambos botones a la par.

Nota: *Tenga en cuenta que pulsar 2 botones “exactamente a la vez” es casi imposible.*

d) Escriba la función “int get_botones(void)” con la siguiente condición de diseño:

- Utilice los pines PA11 y PA12, sabiendo que si el botón está pulsado entrega 0 voltios y si no lo está entrega 3 voltios.



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

a)

```
void config_RTC(int hora, int minuto){

    int valor = 0;

    // cojo minutos
    valor = (((minuto / 10) << 4) + (minuto % 10)) << 8);
    // cojo horas
    valor |= (((hora / 10) << 4) + (hora % 10)) << 16);

    // configuro RTC
    RTC->WPR=0xCA;
    RTC->WPR=0x53;
    RTC->ISR |= (1<<7);
    while ((RTC->ISR & (1<<6))==0);
    RTC->PRER=255;
    RTC->PRER|=127<<16;
    RTC->TR = valor;
    RTC->CR = 0x00000000;
    RTC->ISR &= ~(1<<7);
    RTC->WPR=0;
    while ((RTC->ISR & (1<<6))!=0);

}
```

b)

```
void hora_to_str(unsigned char *cadena){

    int valor = 0;

    valor = RTC->TR;

    // tiro los segundos
    valor = valor >> 8;

    // cojo minutos
    *(cadena+4)=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    *(cadena+3)=(valor & 0x00000007)+'0';
    valor = valor >> 4;

    // cojo horas
    *(cadena+1)=(valor & 0x0000000F)+'0';
    valor = valor >> 4;
    *cadena=(valor & 0x00000003)+'0';

}
```

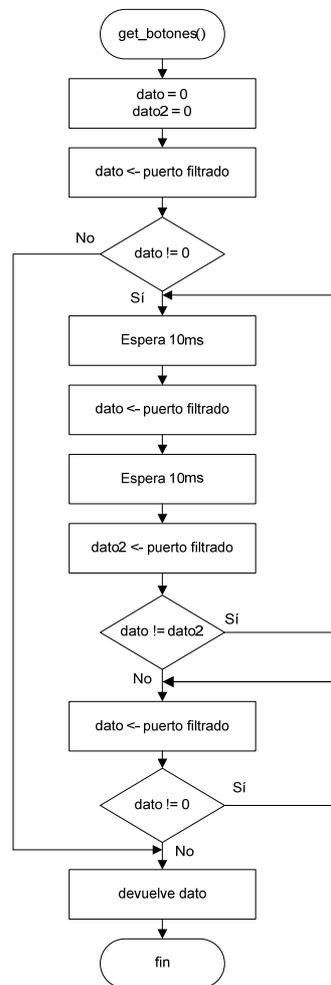


No se permiten ni libros, ni apuntes, ni calculadoras programables. Sólo se permite el manual del microcontrolador
Caja ejercicio se contestará en hojas independientes. Se pueden utilizar tantas hojas por ejercicio como considere oportuno (salvo el ejercicio 1 que se contestará aquí). Las respuestas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

```
valor = valor >> 4;
```

```
}
```

c)





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

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
EXAMEN FINAL (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

d)

```
int get_botton(void){

    int dato = 0;
    int dato2 = 0;

    // Teclas pull-up: pulsado = 0, no pulsado = 1
    // PA11 pulsado -> dato = 0x1
    // PA12 pulsado -> dato = 0x2
    // PA11 y PA12 pulsado -> dato = 0x3
    // PA11 y PA12 no pulsados -> dato = 0x0
    dato = ((~((GPIOA->IDR&(0x3<<11)) >> 11))& 0x3);

    if(dato){

        do{

            espera(100);
            dato = ((~((GPIOA->IDR&(0x3<<11)) >> 11))& 0x3);

            espera(100);
            dato2 = ((~((GPIOA->IDR&(0x3<<11)) >> 11))& 0x3);

        }while( dato != dato2 );

        // si dato es distinto de cero, espera que suelte la tecla
        while( (GPIOA->IDR&(0x3 << 11)) != (0x3 << 11) );
        espera(100);

    }

    return(dato);

}
```



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, **al cual le puede faltar alguna función y/o alguna definición y/o contener 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, con una variación de velocidad menor de 1Hz.
- 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*):

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 vaya 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. (10%)
3. De un significado breve a las siguientes variables del programa. No tienen por qué ser acrónimos de ningún tipo (20%):
 - a. cero
 - b. uno
 - c. dos
 - d. tres
 - e. cuatro
 - f. cinco
 - g. seis
4. El código presenta al menos tres errores. Dos de esos errores está en los nombres de las funciones, mientras que el otro se encuentra en el funcionamiento del mismo. Encuéntrelos, justifíquelos y dé una solución a los mismos. (20%)
5. Analizando la pureza del código, un experto expone una crítica a la línea 35. ¿Podría decir qué crítica es, y cómo la solucionaría? (10%)
6. 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. (30%)



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

ANEXO I

```
1 #include "stm3211xx.h"
2 #include "Biblioteca_SDM.h"
3 unsigned char radio=23;
4 unsigned uno, dos;
5 unsigned char cero;
6 unsigned tres;
7 unsigned cuatro;
8 unsigned cinco;
9 unsigned seis;
10
11 void RAI1 (void) {
12     EXTI->PR = 0x01;
13     NVIC->ICER[0] |= (1 << 6);
14     if ((EXTI->RTSR & 0x01) == 0) {
15         TIM4->CCR2 = TIM4->CNT + 2000;
16         TIM4->CR1 |= 0x0001;
17         EXTI->RTSR |= 0x01;
18         EXTI->FTSR &= ~(0x01);
19     }
20     else {
21         TIM4->CR1 &= ~(0x0001);
22         EXTI->FTSR |= 0x01;
23         EXTI->RTSR &= ~(0x01);
24     }
25     NVIC->ISER[0] |= (1 << 6);
26 }
27
28 void RAI2 (void) {
29     if (TIM4->SR & 0x0004 == 1) {
30         cero = 1;
31         TIM4->CR1 &= ~(0x0001);
32         TIM4->SR = 0x0004;
33     }
34     else {
35         seis = 2 * 3142 * radio;
36         tres += seis;
37         dos = TIM4->CCR1 - uno;
38         if (dos < 0) dos += 0xFFFFFFFF;
39         cinco = seis / dos;
40         uno = TIM4->CCR1;
41         TIM4->SR = 0x0002;
42     }
43 }
44
45 int main (void) {
46     Init_SDM();
47     Init_LCD();
48     cero = 0;
```



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

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
EXAMEN FINAL (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

```
49 GPIOA->MODER |= 0x00000300;
50 GPIOA->MODER &= ~(1 << (0*2 +1));
51 GPIOA->MODER &= ~(1 << (0*2));
52 GPIOA->PUPDR &= ~(11 << (0*2));
53 GPIOB->MODER |= 0x00000001 << (2*6 +1);
54 GPIOB->MODER &= ~(0x00000001 << (2*6));
55 GPIOB->AFR[0] &= ~(0x0F << (4*6));
56 GPIOB->AFR[0] |= 0x02 << (4*6);
57 ADC1->CR2 &= ~(0x00000001);
58 ADC1->CR1 = 0x02000000;
59 ADC1->CR2 = 0x00000472;
60 ADC1->SMPR1 = 0;
61 ADC1->SMPR2 = 0;
62 ADC1->SMPR3 = 0;
63 ADC1->SQR1 = 0x00000000;
64 ADC1->SQR5 = 0x00000004;
65 ADC1->CR2 |= 0x00000001;
66 while ((ADC1->SR&0x0040)==0);
67 ADC1->CR2 |= 0x40000000;
68 EXTI->FTSR |= 0x01;
69 EXTI->RTSR &= ~(0x01);
70 SYSCFG->EXTICR[0] = 0;
71 EXTI->IMR |= 0x01;
72 NVIC->ISER[0] |= (1 << 6);
73 TIM4->CR1 = 0x0000;
74 TIM4->CR2 = 0x0000;
75 TIM4->SMCR = 0x0000;
76 TIM4->PSC = 12000;
77 TIM4->CNT = 0;
78 TIM4->ARR = 0xFFFF;
79 TIM4->CCR2 = 0;
80 TIM4->DCR = 0;
81 TIM4->DIER = 0x0006;
82 TIM4->CCMR1 = 0x0001;
83 TIM4->CCMR2 = 0x0000;
84 TIM4->CCER = 0x0003;
85 TIM4->CR1 |= 0x0001;
86 TIM4->EGR |= 0x0001;
87 TIM4->SR = 0;
88 NVIC->ISER[0] |= (1 << 30);
89 uno = 0;
90 while (1) {
91     if (cero!=0) {
92         uno = 0;
93         tres = 0;
94     }
95     while ((ADC1->SR & 0x0002)==0);
96     cuatro = (unsigned char)(ADC1->DR & 0x000000FF);
97     MuestraVelocimetro(cinco, tres, cuatro);
98 }
99 }
```



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

1) Los elementos que se están utilizando son:

- *Conversión Analógica/Digital (líneas 57-65)*
- *Temporizador (Timer 4) en funcionalidades de TIC y TOC (líneas 73-87)*
- *Interrupción externa 0 (EXTIO) (líneas 68-71)*
- *Controlador de Interrupciones Vectorizadas (líneas 72, 88)*

2) La configuración de los distintos periféricos es:

- *ADC: En la línea 58 se configura a 8 bits, y en la línea 59 se configura en modo burst, a la velocidad más lenta. En las líneas 63-64 se configura un único canal que sea el AIN4. Por lo que se hacen medidas continuas de un valor analógico externo (suponemos que es la temperatura).*
- *Timer 4: En la línea 76 se configura el pre-escalado, para que el temporizador mida unidades de milisegundos. Además se activa la funcionalidad TIC, mediante el flanco de bajada en el canal 1 y pin PB6. En un determinado momento se utiliza la funcionalidad TOC, para medir 2 segundos desde que se pulsa el reset (se activa la EINT1). Todo evento (tanto el TIC como el TOC) provoca interrupción, y en la RAI correspondiente (RAI2), se mira qué servicio ha dado la interrupción y se procede en consecuencia. El temporizador funciona siempre en modo continuo, no reseteándose nunca, por lo que todas las medidas se hacen de forma relativa (en relación al valor anterior o el actual del CNT).*
- *EXTI: Se utiliza el EXTIO por PA0 y se configura para que salte por flanco de bajada inicialmente, y luego se cambia la polaridad, pasando a flanco de subida, o flanco de bajada, según interese (el cambio se hace en la RAI1). La configuración inicial se hace en las líneas 68-71.*
- *NVIC: Se activan interrupciones por dos canales de entrada: la EXTIO (línea 72), y el TIM4 (línea 88). En la RAI1 se desactiva el NVIC del EXTIO y luego se vuelve a activar.*

3) Los significados de cada variable son:

- *cero: es un flag que se activa cuando se recibe una pulsación por EXTIO, y se mantiene durante un tiempo de 2 segundos mínimo. Provoca el poner a cero la distancia y velocidad.*



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

- *uno: valor de tiempo capturado en la anterior medida*
- *dos: tiempo que ha transcurrido entre la anterior medida (paso por vuelta) y la actual. Tiempo de vuelta de rueda.*
- *tres: distancia acumulada*
- *cuatro: temperatura obtenida del ADC*
- *cinco: velocidad instantánea*
- *seis: longitud de la rueda.*

4) Los tres principales errores son:

- *La RAI1 realmente es la RAI de la EXTI0, por lo que habría que sustituir RAI1 por EXTI0_IRQHandler*
- *La RAI2 es la RAI del TIM4, por lo que habría que sustituir RAI2 por TIM4_IRQHandler*
- *La variable cero (el reset), no se vuelve a poner nunca a cero, por lo que, una vez activado, siempre se quedará en esa rama del bucle y por tanto no mostrará ninguna medida real de velocidad y distancia.*

5) Quitando menciones a la precisión de la medida de la longitud de la rueda (la cual se hace así para no hacer cálculos en aritmética de punto fijo, y sólo usar aritmética entera), el problema es que ese cálculo sólo depende del valor de radio, el cual es una constante y por lo tanto no se vuelve a cambiar. Por tanto, no es lógico que esa línea se encuentre ejecutándose cada vez que salte la RAI, sino que debería haber sido calculada la variable "seis" al principio del programa para que sólo se ejecute una vez, o incluso haberlo configurado como una constante, para evitar cualquier tipo de ejecución.

6) Los diagramas de flujo, una vez traducidas las variables, son:



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

