



UNIVERSIDAD CARLOS III DE MADRID
Sist. Dig. Basados en Microprocesador
3 de marzo de 2014

(Dpto. de Tecnología Electrónica)
(Gr. Ing. Telemática)
1er PARCIAL (60 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 puntos):

Explique en qué se diferencia una arquitectura Harvard, respecto a la Von Neumann. Indique qué ventajas y qué inconvenientes supone frente a la Von Neumann.

La arquitectura Harvard resulta de separar en memorias diferentes el código de los datos del programa. Por lo tanto dicha separación supone crear buses de datos, direcciones y control distintos para la memoria de datos, que para la memoria de programa

Los inconvenientes que conlleva la arquitectura Harvard son, precisamente, procedentes de la complicación del hardware necesario para acceder a dichas memorias, ya que hay que duplicar buses, y por tanto conexiones entre la memoria y el microprocesador.

Las ventajas son varias. Entre ellas se encuentra que, al estar separado el código de los datos, no es posible introducir virus y troyanos en el sistema (que normalmente se ocultan entre los datos) y que se ejecuten (ya que sólo se ejecuta lo que está en la memoria de programa). Además, se puede optimizar más los tamaños de memoria a lo que realmente se necesita, utilizando tamaños distintos para el programa, que para los datos (tanto en número de palabras, como en el tamaño de la palabra).



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.

1
2 **CUESTIÓN 2 (4 puntos):**
3

4 Indique los errores que observa en el siguiente código, tanto de sintaxis como de funcionamiento
5 (justifique su respuesta). Considere que $pclk=15MHz$ y que el código pretende sacar por pantalla, de
6 forma continua la media de las últimas 5 medidas realizadas por el conversor analógico digital.
7

8 Archivo Biblioteca_SDM.h:

```
9     void Init_SDM(void);  
10    void Init_LCD(void);  
11    void LCD_Limpia(void);  
12    void LCD_Texto(unsigned char *texto);  
13    void Bin2Ascii(unsigned short valor, unsigned char *texto);
```

14
15 Archivo Principal.c:

```
16    #include "stm3211xx.h"  
17    #include "Biblioteca_SDM.h"  
18    unsigned short acumulado = 0;  
19    unsigned char cuenta = 0;  
20    unsigned char texto[6];  
21    void EXTI1_IRQHandler(void) {  
22        NVIC->ICER[0]=(1<<18);  
23        while ((ADC1->SR&0x0002)==0);  
24        acumulado += ADC1->DR;  
25        cuenta++;  
26        if (cuenta = 3) {  
27            cuenta = 0;  
28            media = acumulado / 5;  
29            LCD_Limpia();  
30        }  
31    }  
32    int main(void){  
33        unsigned short media = 0;  
34        unsigned short media_anterior=0;  
35        Init_SDM();  
36        GPIOA->MODER |= 0x00000300;  
37        ADC1->CR2 &= ~(0x00000001);  
38        ADC1->CR1 = 0x00000000;  
39        ADC1->CR2 = 0x00000412;  
40        ADC1->SMPR1 = 0;  
41        ADC1->SMPR2 = 0;  
42        ADC1->SMPR3 = 0;  
43        ADC1->SQR1 = 0x00000000;  
44        ADC1->SQR5 = 0x00000004;
```



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.

```
45 while ((ADC1->SR&0x0040)!=0);
46 ADC1->CR2 |= 0x40000000;
47 NVIC->ISER[0] = (1 << 18);
48 while (0) {
49     if (abs(media_anterior - media)>=TOLERANCIA) {
50         media_anterior = media;
51         Bin2Ascii(media,&texto[0]);
52         LCD_Texto(texto);
53     }
54 }
55 }
```



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.

1. Líneas 16-17: debería estar definido *TOLERANCIA* insertando entre esas dos líneas un `#define TOLERANCIA XXX`, siendo *XXX* el valor de dicha tolerancia.
2. Línea 19: la variable `texto[6]` debería ser local, no global.
3. Línea 20: La *IRQ* es la del *ADC1* no la del *EXTI1*, por lo tanto la función debe llamarse `ADC1_IRQHandler`
4. Línea 22: Esto es una espera activa, que está prohibido en una *RAI*, por lo que habría que quitarla de ahí.
5. Línea 25: La igualdad debería ser con `==` en lugar de `con =`
6. Línea 25: Como son las 5 últimas medidas, la comprobación debería ser con `5`, en lugar de `con 3`.
7. Líneas 27-28: Después de hacer la media, el acumulado debería ponerse a `0`, insertando entre las dos líneas la sentencia `acumulado=0`;
8. Línea 28: Esto es un acceso a pantalla, que por tiempos está prohibido en una *RAI*. Habría que eliminarlo de aquí y pasarlo al programa principal.
9. Líneas 29-30: Al principio de la *RAI* se inhibe la *IRQ* por *ADC1*. Se debería volver a activar al final de la *RAI* añadiendo, entre esas dos líneas la sentencia `NVIC->ISER[0] = (1<<18)`;
10. Línea 32: La variable *media* debe ser global, no local.
11. Líneas 34-35: El *LCD* debería inicializarse, insertando entre esas dos líneas, la llamada a `Init_LCD()`;
12. Línea 37: Habría que activar la interrupción por final de conversión en el *CR1*
13. Línea 44: La espera activa debe ser para cuando se haya activado el flag. Por tanto debería ser `==` en lugar de `!=`
14. Línea 47: La condición de bucle infinito es `while (1)`, en lugar de `while (0)`.
15. Línea 48: La función `abs` no es trivial, por lo que debería necesitarse importar alguna biblioteca adicional (por ejemplo `math.h`)



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.

CUESTIÓN 3 (4 puntos):

Se desea realizar un generador de ondas en forma de rampa basándose en un STM32L152RB. El generador cambiará el periodo de la señal generada cada vez que se pulse un botón. Las 4 frecuencias a generar son 10Hz, 100Hz, 1KHz y 10KHz. Dispone de una función suministrada que se llama `void espera(int numero)`, que realiza una espera activa de tantos microsegundos como indica el parámetro `numero`. Partiendo de la placa STM32L-Discovery (), realice (si tiene que hacer alguna consideración adicional, indíquela claramente):

- el diagrama de bloques de la solución
- indique cómo configuraría cada uno de los periféricos (no escriba el código)
- realice el diagrama de flujo.

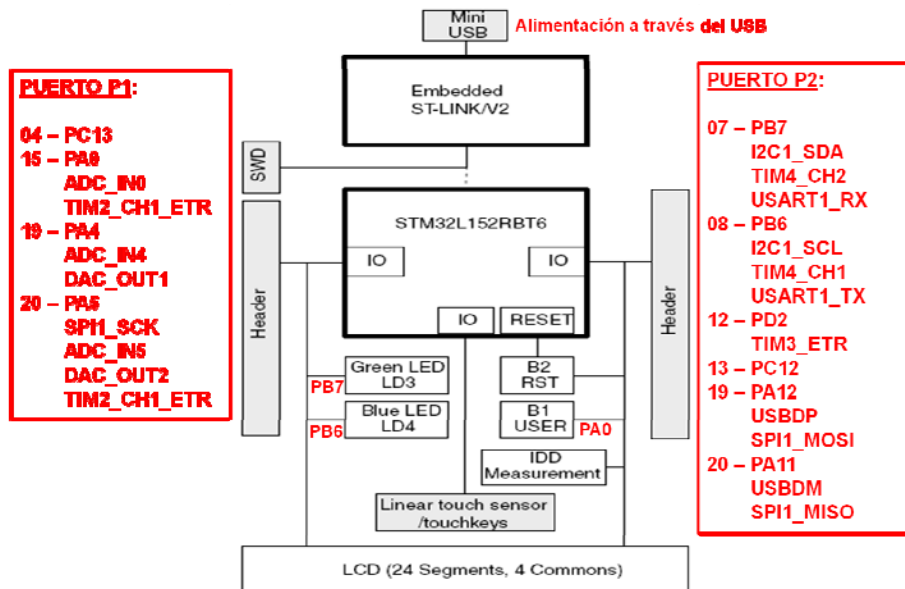
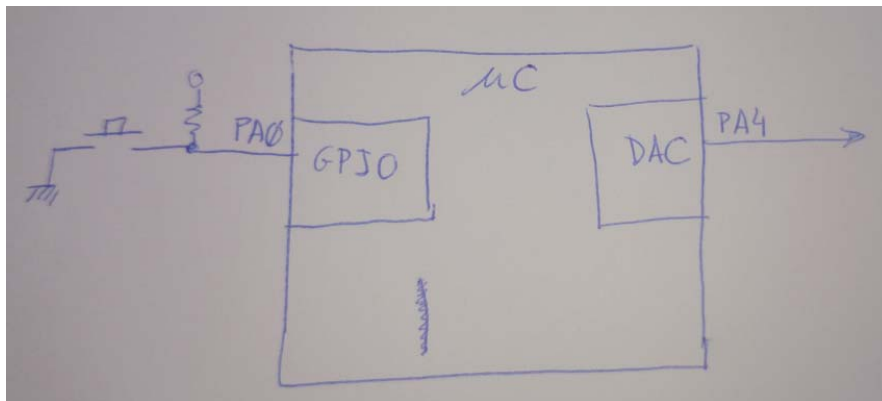


Fig. 1: Diagrama de bloques de la STM32L-Discovery. Puerto P1 se refiere al puerto de expansión de dicha placa, mientras P2 es el segundo puerto de expansión de dicha placa, nada que ver con los puertos GPIO (PA – PD) del STM32L152RB.



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.

a) El diagrama de bloques es el siguiente, donde sólo se necesita un pin de entrada para el botón, conectado, por ejemplo, al PA0, y un pin de salida para el DAC, conectado, por ejemplo, al PA4:



b) No se ve la necesidad de utilizar interrupciones, por lo que los únicos periféricos a configurar son el GPIO en cuanto al puerto A, y el DAC:

GPIO - Puerto A:

- PA0 como entrada y sin resistencias de pull-up ni pull-down
 - Se ponen los bits 0 y 1 del GPIOA->MODER a 0
 - Se ponen los bits 0 y 1 del GPIOA-PUPDR a 0
- PA4 como pin analógico para el DAC:
 - Se ponen los bits 9 y 8 del GPIOA-MODER a 1

DAC:

- Habilitación del canal 1:
 - En DAC->CR, EN1 = 1
- Se utilizará como registro de datos el DAC->DHR8R1

Es importante tener claro el funcionamiento del programa y lo que supone utilizar el DAC con 8 bits en cuanto a los tiempos que nos indican. Si se utiliza en 8 bits, entonces se pueden poner valores desde 0 a 255 en el DAC, formando la rampa. Por lo tanto, un contador para el programa principal sería suficiente. Pero hay que tener en cuenta el cálculo



*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.*

de cuánto tiempo (en microsegundos) habría que esperar entre muestra y muestra. De ahí el siguiente cálculo:

<i>Frecuencia</i>	<i>Periodo (en us)</i>	<i>Tiempo de paso (en us)</i>
<i>10 Hz</i>	<i>100.000</i>	<i>12.500</i>
<i>100 Hz</i>	<i>10.000</i>	<i>1.250</i>
<i>1 KHz</i>	<i>1.000</i>	<i>125</i>
<i>10 KHz</i>	<i>100</i>	<i>12,5</i>

Como se puede ver, para el caso de 10KHz no se tiene un número entero de microsegundos, lo cual se puede salvar haciendo que unas veces cuente 12 us y otras veces 13 us.

c) El diagrama de flujo quedaría así:



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.

