

 UNIVERSIDAD DE ALCALÁ ESCUELA POLITÉCNICA SUPERIOR DEPARTAMENTO DE ELECTRÓNICA		 GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN	
ASIGNATURA	SISTEMAS ELECTRÓNICOS DIGITALES AVANZADOS	FECHA	21-12-2017
APELLIDOS, NOMBRE		GRUPO	

PRUEBA DE EVALUACIÓN INTERMEDIA 2

CUESTION 1

Suponiendo un sistema con un LPC1768 con las tareas que se indican a continuación, identifique los parámetros **D** y **T** de cada una de ellas y **justifique el valor explicando el subsistema interno del microcontrolador que utilizaría** y en qué modo se configuraría. Tenga en cuenta que es un único microcontrolador con recursos limitados.

- Recepción de datos a 38400 baudios por la UART 3.
- Generación de una señal cuadrada con un ciclo de trabajo de un 50% de frecuencia variable entre 1kHz y 100kHz.
- Generación de una señal PWM de 10kHz de frecuencia con un ciclo de trabajo que oscila entre un 10% y un 90%.
- Generación de una señal PWM de 5kHz de frecuencia con un ciclo de trabajo que oscila entre un 10% y un 90%.
- Reproducción repetida de un mensaje de audio de 10 segundos de duración con una frecuencia de muestro de 8kHz y muestras de 10 bits utilizando DMA.
- Grabación simultánea de dos canales de audio a una frecuencia de muestreo de 8kHz durante un segundo cuando lo solicite el usuario.

a) La recepción se realiza con 8 bts de datos + start + stop. $\rightarrow T=(8+1+1)/38400 = 260 \text{ us} = D$

b) Una opción es configurar el Match de un temporizador configurado para que conmute con cada interrupción y en la rutina de atención a la interrupción avanzar el Match el tiempo correspondiente a un semiperiodo. En este caso $T = D = 5 \text{ us}$

No se pueden generar dos señales PWM con el mismo módulo PWM de frecuencias diferentes por lo que una señal hay que generarla con un temporizador. Para beneficiar la ejecutabilidad es conveniente que el módulo PWM genere la señal de mayor frecuencia.

c) Si se utiliza un módulo PWM no genera interrupción por lo que no tiene sentido hablar de D y T

d) La señal PWM puede generarse con el mismo temporizador del apartado b utilizando otra salida de Match configurada para que conmute y que en cada interrupción se va incrementando el valor de temporizador primero TH y luego TL repitiendo la secuencia continuamente. En esta situación el tiempo menor entre interrupciones sería el 10% del periodo de la señal. $T(\text{señal}) = 1/5000=200\text{us} \rightarrow T = D = 20\text{us}$

e) En 10 segundos son necesarias 80000 muestras. Con un máximo de tamaño por canal de 4096 transferencias, serían necesarios más de 19 canales. Una opción podría ser crear un array de estructuras de DMA enlazadas de forma que cada una de las estructuras apunte a la siguiente y la última a la primera utilizando un único canal de DMA. Con esta solución no sería necesario utilizar interrupciones.

f) Se podría configurar el ADC en modo ráfaga con un tiempo de conversión de 125/2 us configurando los dos canales. Se podría habilitar la interrupción de uno de los dos canales donde se leerían las dos muestras y se guardarían en memoria. En la interrupción se podría ir contando el número de muestras tomadas y detener el modo ráfaga del ADC al recibir 8000 muestras. $T = 125 \text{ us}$ $D = 125/2 \text{ us}$ ya que si se tarda más se sobrescribiría la muestra del canal que no genera interrupción.

CUESTIÓN 2

Suponiendo que un sistema basado en el LPC1768 tiene tres tareas con los parámetros que se indican en la tabla y el programa principal tiene una región crítica de 1ms

Tarea	Prioridad	Subprioridad	C	T	D
Tarea 1	0	0	1	5	5
Tarea 2	0	1	2	10	5
Tarea 3	1	0	5	30	30
Tarea 4	1	1	10	30	30

Nota: Unidades en ms

Se pide:

- Analice si el sistema es ejecutable.
- Si el sistema es ejecutable:
 - Calcule la duración máxima de una región crítica del programa principal que haría que el sistema dejara de ser ejecutable.

Si el sistema no es ejecutable:

- Identifique los factores que hacen que el sistema no sea ejecutable y proponga una o varias soluciones que hagan el sistema ejecutable.

Tarea 1:

$$R_1 = C_1 + B_1 + I_1 = C_1 + \max(C_2, RC) + 0 = 1 + 2 = 3 \leq D_1 = 5$$

Tarea 2:

$$R_2 = C_2 + B_2 + I_2 = C_2 + RC + \left\lceil \frac{R_2 - C_2}{T_1} \right\rceil \cdot C_1 = 2 + 1 + \left\lceil \frac{R_2 - 2}{5} \right\rceil \cdot 1$$

$$w^0 = 2 + 1 + 1 = 4$$

$$w^1 = 3 + \left\lceil \frac{w^0 - 2}{5} \right\rceil \cdot 1 = 3 + \left\lceil \frac{4 - 2}{5} \right\rceil \cdot 1 = 4 \rightarrow R_2 = 4 \leq D_2 = 5$$

Tarea 3:

$$R_3 = C_3 + B_3 + I_3 = C_3 + \max(C_4, RC) + \left\lceil \frac{R_3}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{R_3}{T_2} \right\rceil \cdot C_2 = C_3 + C_4 + \left\lceil \frac{R_3}{5} \right\rceil \cdot C_1 + \left\lceil \frac{R_3}{10} \right\rceil \cdot C_2$$

$$= 5 + 10 + \left\lceil \frac{R_3}{5} \right\rceil \cdot 1 + \left\lceil \frac{R_3}{10} \right\rceil \cdot 2$$

$$w^0 = 5 + 10 + 1 + 2 = 18$$

$$w^1 = 15 + \left\lceil \frac{w^0}{5} \right\rceil \cdot 1 + \left\lceil \frac{w^0}{10} \right\rceil \cdot 2 = 15 + \left\lceil \frac{18}{5} \right\rceil \cdot 1 + \left\lceil \frac{18}{10} \right\rceil \cdot 2 = 15 + 4 + 4 = 23$$

$$w^2 = 15 + \left\lceil \frac{w^1}{5} \right\rceil \cdot 1 + \left\lceil \frac{w^1}{10} \right\rceil \cdot 2 = 15 + \left\lceil \frac{23}{5} \right\rceil \cdot 1 + \left\lceil \frac{23}{10} \right\rceil \cdot 2 = 15 + 5 + 6 = 26$$

$$w^3 = 15 + \left\lceil \frac{w^2}{5} \right\rceil \cdot 1 + \left\lceil \frac{w^2}{10} \right\rceil \cdot 2 = 15 + \left\lceil \frac{26}{5} \right\rceil \cdot 1 + \left\lceil \frac{26}{10} \right\rceil \cdot 2 = 15 + 6 + 6 = 27$$

$$w^4 = 15 + \left\lceil \frac{w^3}{5} \right\rceil \cdot 1 + \left\lceil \frac{w^3}{10} \right\rceil \cdot 2 = 15 + 6 + 6 = 27 \rightarrow R_3 = 27 \leq D_3 = 30$$

Tarea 4:

$$\begin{aligned}
 R_4 &= C_4 + B_4 + I_4 = C_4 + RC + \left\lceil \frac{R_4}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{R_4}{T_2} \right\rceil \cdot C_2 + \left\lceil \frac{R_4 - C_4}{T_3} \right\rceil \cdot C_3 \\
 &= 10 + 1 + \left\lceil \frac{R_4}{5} \right\rceil \cdot 1 + \left\lceil \frac{R_4}{10} \right\rceil \cdot 2 + \left\lceil \frac{R_4 - 10}{30} \right\rceil \cdot 5 = \\
 w^0 &= 10 + 1 + 1 + 2 + 5 = 19 \\
 w^1 &= 11 + \left\lceil \frac{w^0}{5} \right\rceil \cdot 1 + \left\lceil \frac{w^0}{10} \right\rceil \cdot 2 + \left\lceil \frac{w^0 - 10}{30} \right\rceil \cdot 5 = 11 + \left\lceil \frac{19}{5} \right\rceil \cdot 1 + \left\lceil \frac{19}{10} \right\rceil \cdot 2 + \left\lceil \frac{9}{30} \right\rceil \cdot 5 = \\
 &= 11 + 4 + 4 + 5 = 24 \\
 w^2 &= 11 + \left\lceil \frac{24}{5} \right\rceil \cdot 1 + \left\lceil \frac{24}{10} \right\rceil \cdot 2 + \left\lceil \frac{14}{30} \right\rceil \cdot 5 = 11 + 5 + 6 + 5 = 27 \\
 w^3 &= 11 + \left\lceil \frac{27}{5} \right\rceil \cdot 1 + \left\lceil \frac{27}{10} \right\rceil \cdot 2 + \left\lceil \frac{17}{30} \right\rceil \cdot 5 = 11 + 6 + 6 + 5 = 28 \\
 w^4 &= 11 + \left\lceil \frac{28}{5} \right\rceil \cdot 1 + \left\lceil \frac{28}{10} \right\rceil \cdot 2 + \left\lceil \frac{18}{30} \right\rceil \cdot 5 = 11 + 6 + 6 + 5 = 28 \rightarrow \\
 R_4 &= 28 \leq D_4 = 30 \text{ El sistema es ejecutable}
 \end{aligned}$$

Una región crítica en el programa principal influiría en el bloqueo de la tarea 2 y la tarea 4 principalmente ya que en la tarea 1 y la tarea 3 el bloqueo está influenciado por las tareas del mismo nivel de prioridad (siempre que la RC no sea de mayor duración de estas tareas):

- En la Tarea 2 parece que tiene un margen de 1ms. Hay que comprobar si cambia en algo.

$$\begin{aligned}
 R_2 &= C_2 + B_2 + I_2 = C_2 + RC_{PP} + \left\lceil \frac{R_2 - C_2}{T_1} \right\rceil \cdot C_1 = 2 + 2 + \left\lceil \frac{R_2 - 2}{5} \right\rceil \cdot 1 \\
 w^0 &= 4 + 1 = 5 \\
 w^1 &= 4 + \left\lceil \frac{w^0 - 2}{5} \right\rceil \cdot 1 = 4 + \left\lceil \frac{5 - 2}{5} \right\rceil \cdot 1 = 5 \rightarrow R_2 = 5 \leq D_2 = 5 \rightarrow \text{Sigue siendo ejecutable}
 \end{aligned}$$

- Hay que comprobar si este valor es válido para la tarea 4

$$\begin{aligned}
 R_4 &= C_4 + B_4 + I_4 = C_4 + RC + \left\lceil \frac{R_4}{T_1} \right\rceil \cdot C_1 + \left\lceil \frac{R_4}{T_2} \right\rceil \cdot C_2 + \left\lceil \frac{R_4 - C_4}{T_3} \right\rceil \cdot C_3 \\
 &= 10 + 2 + \left\lceil \frac{R_4}{5} \right\rceil \cdot 1 + \left\lceil \frac{R_4}{10} \right\rceil \cdot 2 + \left\lceil \frac{R_4 - 10}{30} \right\rceil \cdot 5 = \\
 w^0 &= 10 + 2 + 1 + 2 + 5 = 20 \\
 w^1 &= 12 + \left\lceil \frac{20}{5} \right\rceil \cdot 1 + \left\lceil \frac{20}{10} \right\rceil \cdot 2 + \left\lceil \frac{10}{30} \right\rceil \cdot 5 = 12 + 4 + 4 + 5 = 25 \\
 w^2 &= 12 + \left\lceil \frac{25}{5} \right\rceil \cdot 1 + \left\lceil \frac{25}{10} \right\rceil \cdot 2 + \left\lceil \frac{15}{30} \right\rceil \cdot 5 = 12 + 5 + 6 + 5 = 28 \\
 w^3 &= 12 + \left\lceil \frac{28}{5} \right\rceil \cdot 1 + \left\lceil \frac{28}{10} \right\rceil \cdot 2 + \left\lceil \frac{18}{30} \right\rceil \cdot 5 = 12 + 6 + 6 + 5 = 29 \\
 w^4 &= 12 + \left\lceil \frac{29}{5} \right\rceil \cdot 1 + \left\lceil \frac{29}{10} \right\rceil \cdot 2 + \left\lceil \frac{19}{30} \right\rceil \cdot 5 = 12 + 6 + 6 + 5 = 29 \rightarrow \\
 R_4 &= 29 \leq D_4 = 30 \text{ El sistema es ejecutable}
 \end{aligned}$$

La máxima RC en el programa principal aceptable sería de 2 ms

CUESTIÓN 3.

Explique qué es un DSP y sus características fundamentales.

CUESTIÓN 4.

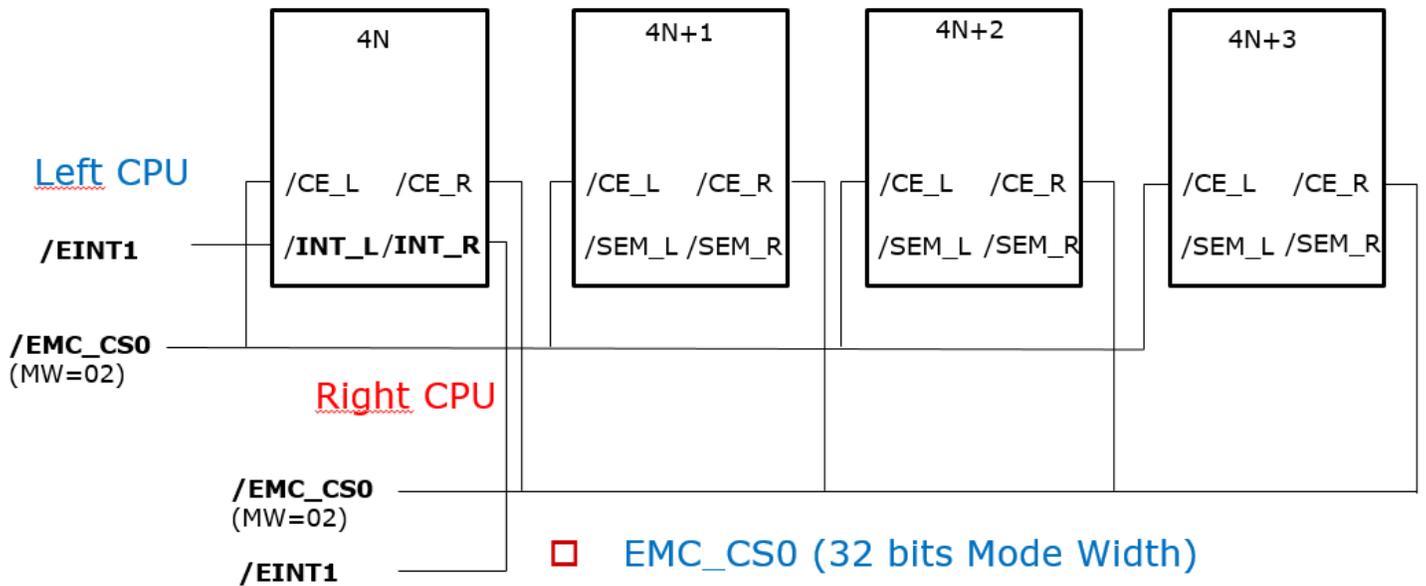
Explique qué se entiende por System on Chip comentando algunos ejemplos.

CUESTIÓN 5.

Explique las ventajas e inconvenientes de utilizar un Sistema Operativo de Tiempo Real en aplicaciones de sistemas empujados similares al proyecto de la asignatura.

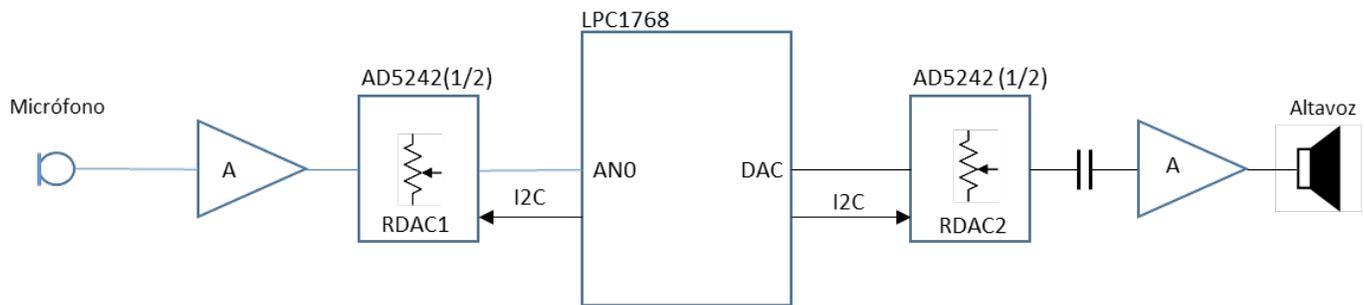
CUESTIÓN 6.

Explique en detalle el método de arbitración por interrupción de las memorias DUAL-PORT y justifícalo sobre el esquema de la figura. Considere que el bloque de memoria está mapeado en la dirección 0x8000.0000 (EMC_CS0) y que cada chip tiene una capacidad de 32Kbytes.



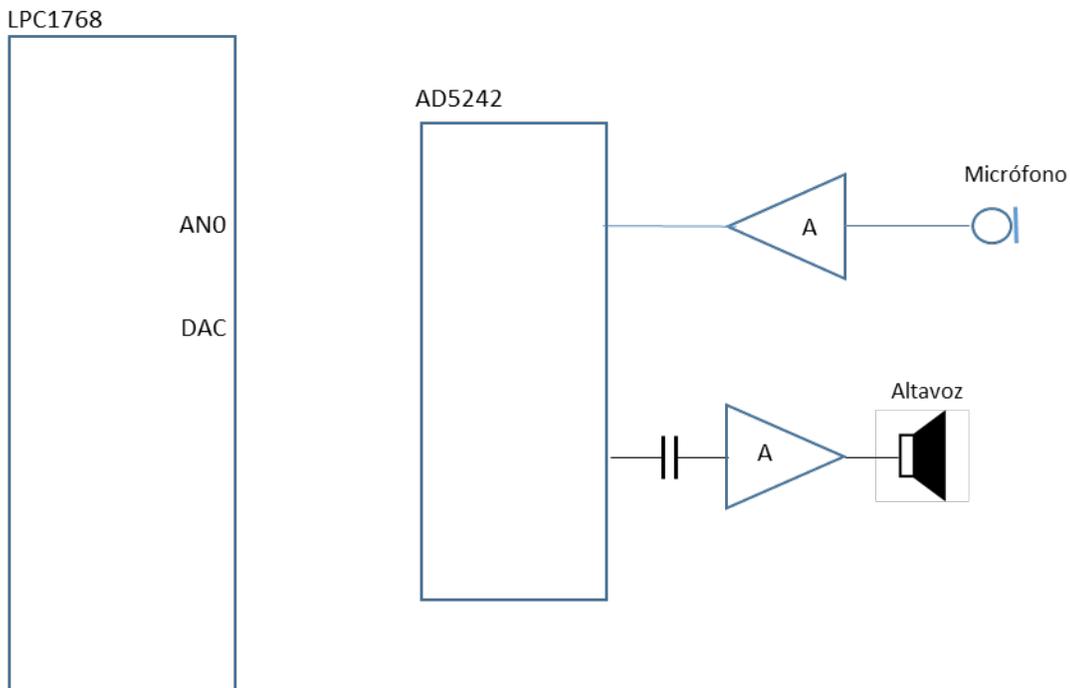
CUESTIÓN 7.

El diagrama de bloques del circuito de la figura representa un sistema de grabación/reproducción de audio en el que se utilizan dos potenciómetros digitales para controlar la ganancia del micrófono y el volumen del altavoz.



Dibuja detalladamente el diagrama de conexión del AD5242 con el LPC1768 teniendo en cuenta la librería de funciones del bus I2C del Anexo 1 y escribe la función `i2C_mic_gain(char porcentaje)`, que permite variar la ganancia del micrófono entre el 0 y 100%.

NOTA: El bit \bar{A}/B del byte de instrucción selecciona el potenciómetro RDAC1 ó RDAC2. El byte de datos (D) modifica el valor de la R de acuerdo a la ecuación mostrada en Anexo1 (Considere $R_w=0$ y $R_{AB}=10k\Omega$)



CUESTIÓN (Si se completa esta cuestión ya no se puede realizar la pregunta equivalente en la PEF)

Se dispone de un sistema formado por un servomotor que hace girar a un sensor que va tomando medidas de distancia y que también dispone de un sensor de temperatura. El sistema tiene dos modos de funcionamiento:

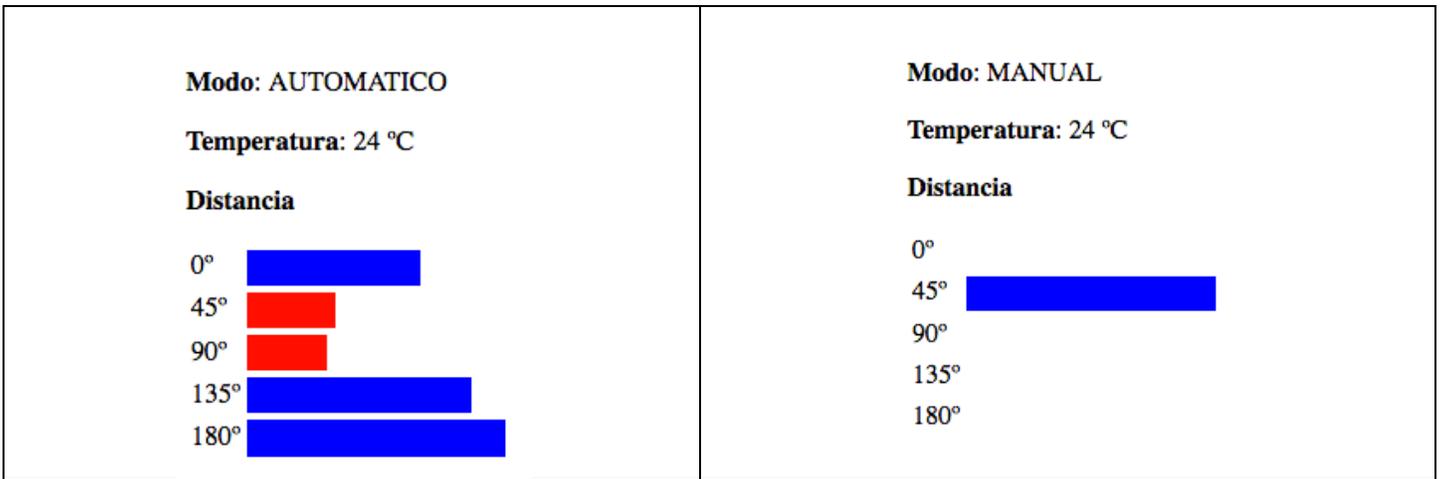
- Automático: está continuamente tomando medidas de distancia en 0°, 45°, 90°, 135° y 180° y las va almacenando en un array: float distancias[5].
- Manual: donde un el usuario puede cambiar manualmente la posición del servomotor (en 0°, 45°, 90°, 135° y 180°) y lanzar una medida. La posición del servomotor se supone almacenada en una variable denominada “posición” que puede tomar valores de 0 a 4. La última medida realizada se almacena en el array de distancias en la posición correspondiente.

Una variable global “modo” indica si el sistema está en modo automático (modo = 0) o manual (modo = 1). También hay una constante denominada UMBRAL que indica la distancia a la que se detectan obstáculos. La temperatura está almacenada en una variable “temperatura”.

Se desea que la aplicación monitorice de forma remota el funcionamiento del sistema (medidas de distancia y temperatura). La distancia se monitorizará en una barra asociada a cada intervalo de ángulo de un tamaño proporcional a la distancia medida siendo el máximo de distancia representable 150cm correspondiente a un ancho de 200 pixels y el mínimo de 15cm, correspondiente a 2 pixels. Si no se detecta obstáculo (Distancia > UMBRAL), la distancia se representa en azul y si se detecta obstáculo se representa en rojo.

Indique el contenido del fichero distancia.cgi al que accedería el navegador y el contenido de la función cgi_func(...) necesario, para que se actualice el valor de la distancia y la temperatura.

En la parte izquierda de la siguiente figura se muestra un ejemplo de página y el código HTML del modo automático y en la parte derecha del modo manual. Se marca con un subrayado la información que debe actualizarse.



```
<!DOCTYPE html>
<html>
<head>
<meta content="text/html; charset=UTF-8" http-equiv="content-type">
<title>index.htm</title>
</head>
<body>
<p><b>Modo</b>: AUTOMATICO</p>
<p><b>Temperatura</b>: 24 &#186;C</p>
<p><b>Distancia</b></p>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">&#186;</td>
<td style="width: 100px; background-color: blue;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">45&#186;</td>
<td style="width: 50px; background-color: red;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">90&#186;</td>
<td style="width: 45px; background-color: red;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">135&#186;</td>
<td style="width: 130px; background-color: blue;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">180&#186;</td>
<td style="width: 150px; background-color: blue;"><br>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<meta content="text/html; charset=UTF-8" http-equiv="content-type">
<title>index.htm</title>
</head>
<body>
<p><b>Modo</b>: MANUAL</p>
<p><b>Temperatura</b>: 24 &#186;C</p>
<p><b>Distancia</b></p>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">&#186;</td>
<td style="width: 0px; background-color: white;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">45&#186;</td>
<td style="width: 150px; background-color: blue;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">90&#186;</td>
<td style="width: 0px; background-color: white;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">135&#186;</td>
<td style="width: 0px; background-color: white;"><br>
</td>
</tr>
</tbody>
</table>
<table border="0">
<tbody>
<tr>
<td style="width: 30px;">180&#186;</td>
<td style="width: 0px; background-color: white;"><br>
</td>
</tr>
</tbody>
</table>
</body>
</html>
```


Anexo 1. AD5242 y librería de funciones del bus I2C

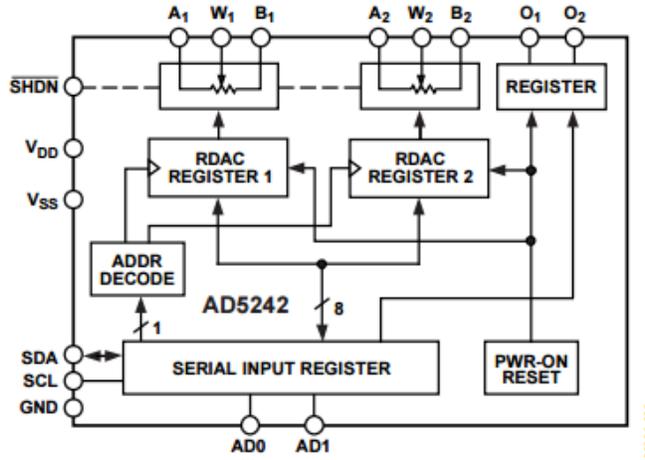


Figure 2. AD5242 Functional Block Diagram

Table 2.

S	0	1	0	1	1	AD1	AD0	R/W	A	\bar{A}/B	RS	SD	O ₁	O ₂	X	X	X	A	D7	D6	D5	D4	D3	D2	D1	D0	A	P
Slave Address Byte								Instruction Byte									Data Byte											

where:

S = start condition

P = stop condition

A = acknowledge

X = don't care

AD1, AD0 = Package pin programmable address bits. Must be matched with the logic states at Pin AD1 and Pin AD0.

R/W = Read enable at high and output to SDA. Write enable at low.

\bar{A}/B = RDAC subaddress select; 0 for RDAC1 and 1 for RDAC2.

RS = Midscale reset, active high.

SD = Shutdown in active high. Same as $\overline{\text{SHDN}}$ except inverse logic.

O₁, O₂ = Output logic pin latched values

D7, D6, D5, D4, D3, D2, D1, D0 = data bits.

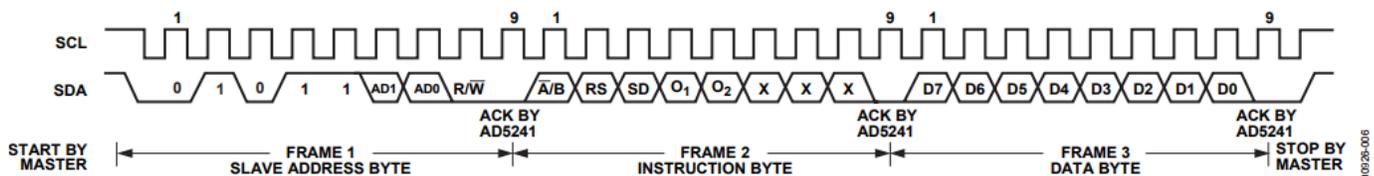


Figure 4. Writing to the RDAC Serial Register

The general equation determining the digitally programmed resistance between W and B is

$$R_{WB}(D) = \frac{D}{256} \times R_{AB} + R_W \quad (1)$$

where:

D is the decimal equivalent of the binary code between 0 and 255, which is loaded in the 8-bit RDAC register.

R_{AB} is the nominal end-to-end resistance.

R_W is the wiper resistance contributed by the on resistance of the internal switch.

FUNCIONES DEL BUS i2c

```

#include <LPC17xx.h>
#define SDA 0
#define SCL 1

void I2Cdelay(void)//retardo minimo de 4.7 us
{
    unsigned char i;
    for(i=0;i<100;i++); //Modificar límite para garantizar los tiempos (Bus standar -->F_max=100kHz)
}

//Genera un pulso de reloj (1 ciclo)
void pulso_SCL(void)
{
    LPC_GPIO0->FIOSET=(1<<SCL); // Genera pulso de reloj (nivel alto)
    I2Cdelay();
    LPC_GPIO0->FIOCLR=(1<<SCL); // Nivel bajo
    I2Cdelay();
}

void I2CSendByte(unsigned char byte)
{
    unsigned char i;
    for(i=0;i<8;i++){

        if (byte &0x80) LPC_GPIO0->FIOSET=(1<<SDA); // envia cada bit, comenzando por el MSB
        else LPC_GPIO0->FIOCLR=(1<<SDA);
        byte = byte <<1; // siguiente bit
        pulso_SCL();
    }

    //Leer ACK que envia el Slave (el Master ha de enviar un pulso de reloj)
    // CONFIGURAR PIN SDA COMO ENTRADA; //espera ACK(config. pin como entrada)
    LPC_GPIO0->FIODIR&=~(1<<SDA);
    pulso_SCL();

    // CONFIGURA PIN SDA COMO SALIDA;
    LPC_GPIO0->FIODIR|=(1<<SDA); // Dejamos SDA de nuevo como salida
}

//Función que envia START + Byte de dirección del Slave (con bit LSB inicando R/W)
void I2CSendAddr(unsigned char addr, unsigned char rw)
{
    //CONFIGURAR PINs SDA, SCL COMO SALIDAS; // Por si se nos olvidada en la conf. general.
    LPC_GPIO0->FIODIR|=(1<<SDA)|(1<<SCL);

    LPC_GPIO0->FIOSET|=(1<<SDA)|(1<<SCL); // SDA y SCL a nivel alto para garantizar el
    // nivel de reposo del bus + tiempo.

    I2Cdelay();
    SDA=0; //condicion de START: Bajar SDA y luego SCL
    I2Cdelay();
    SCL=0;
    I2Cdelay();

    I2CSendByte((addr=addr<<1) + rw); //envia byte de direccion
    //addr, direccion (7bits)
    //rw=1, lectura
    //rw=0, escritura
}

// Función para leer un Byte del Slave. El Master envia al final de la lectura
// el bit ACK o NACK (si es último byte leído) que se pasa como argumento de la función.
unsigned char I2CGetByte(unsigned char ACK)
{
    // ACK = 0, para cualquier byte que no sea el ultimo.
    // ACK = 1 (NACK), despues de leer el ultimo byte
    unsigned char i, byte;
    //CONFIGURAR PIN SDA COMO ENTRADA; //configura pin SDA como entrada
    LPC_GPIO0->FIODIR&=~(1<<SDA);
    for(i=0;i<8;i++){ //lee un bit comenzando por el MSB

```

```

LPC_GPIO0->FIOSET=(1<<SCL); //mientras SCL=1
I2Cdelay();
byte=byte<<1;
if(LPC_GPIO0->FIOPIN&(1<<SDA)) byte++; //Si leemos "1" sumamos para introducir el "1"
LPC_GPIO0->FIOCLR=(1<<SCL); //Si leemos "0" solo desplazamos (se introduce un "0")
I2Cdelay();
}

//CONFIGURAR PIN SDA COMO SALIDA; // Master envía un ACK por cada byte leído.
LPC_GPIO0->FIODIR|=(1<<SDA);

if(ACK)LPC_GPIO0->FIOSET=(1<<SDA); // ACK o (NACK) es funcion del último byte leído
else LPC_GPIO0->FIOCLR=(1<<SDA);

pulso_SCL(); // Pulso de reloj para su envío
return (byte);
}

```

void I2CSendStop(void)

```

{
LPC_GPIO0->FIOCLR=(1<<SDA);
I2Cdelay();
LPC_GPIO0->FIOSET=(1<<SCL); // Subir SCL, y después SDA!! para dejar el bus en reposo
I2Cdelay();
LPC_GPIO0->FIOSET=(1<<SDA);
I2Cdelay();
}

```