UNIVERSIDAD DE ALCALÁ ESCUELA POLITÉCNICA SUPERIOR DEPARTAMENTO DE ELECTRÓNICA		GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA INDUSTRIAL	
ASIGNATURA	SISTEMAS ELECTRÓNICOS DIGITALES	FECHA	ENERO 2016
APELLIDOS, NOMBRE	SOLUCIÓN	GRUPO	

PRUEBA DE EVALUACIÓN GLOBAL

Ejercicio 1 (60 puntos)

Se propone diseñar un SE basado en microcontrolador LPC1768 (Cortex-M3) capaz de realizar medidas de distancia en un plano y sobre un entorno de 180º, con posibilidad de ser controlado de forma manual mediante un potenciómetro o desde un ordenador mediante una interfaz serie asíncrona. El sistema está basado en el sensor de distancia por ultrasonidos SRF04 (ver anexo 1) y en un servomotor. El diagrama de bloques del sistema se representa en la figura 1.

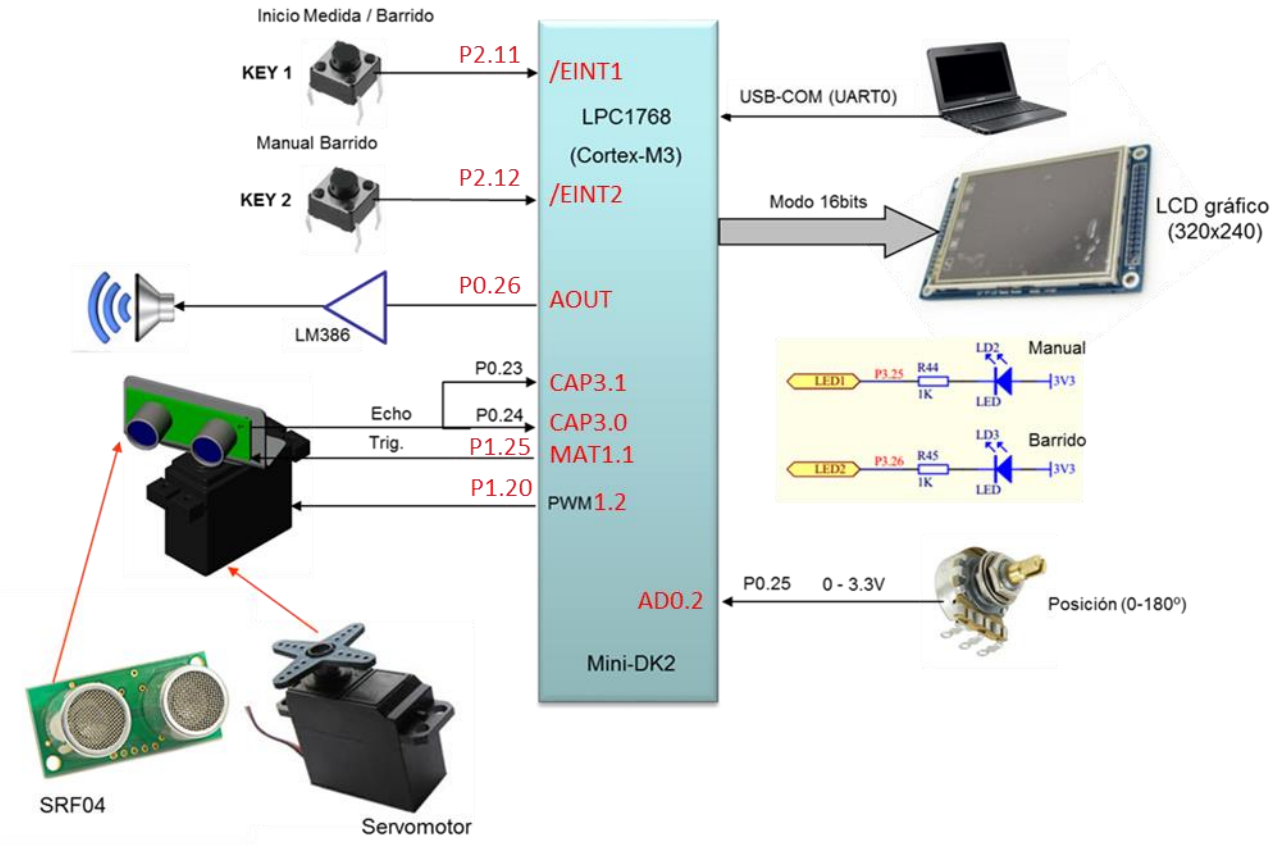


Figura 1. Diagrama de bloques del escáner ultrasónico

El sistema tiene dos modos de funcionamiento. Mediante KEY2 seleccionamos entre el modo **manual** y de **barrido**. Los LEDs 2 y 3 señalizan el modo configurado.

Estando en el **modo barrido** y al pulsar KEY1, el sonar comienza a realizar un barrido (0-180º) con una determinada resolución (en grados y configurable) deteniéndose en cada posición durante **medio segundo** para realizar la medida de distancia correspondiente. Al final del barrido se detendrá automáticamente esperando actuar de nuevo sobre el pulsador si se desea para iniciar de nuevo el barrido.

En el **modo manual**, el potenciómetro permite llevar al servo a una determinada posición (0-180º) de manera proporcional a su recorrido. Una vez situado en la posición deseada, mediante **KEY1** realizamos la medida de distancia al provocar el disparo del sensor SRF04 como consecuencia de aplicar un pulso a nivel alto en la entrada **TRIG**. Dicha medida se obtiene a partir de la duración del pulso en la salida **Echo**, cuyo valor es el tiempo que tardan

los ultrasonidos en ir y la volver tras rebotar en el obstáculo. El valor de la distancia también se ha de sacar por el puerto serie.

Tanto en el modo manual como de barrido, cada vez que se obtiene una medida se ha de enviar su valor, en centímetros, por el puerto serie en ASCII.

Mientras la distancia medida sea menor que la de un umbral (programable) se ha de activar una señal de alarma consistente en un tono de **1 kHz** (señal senoidal con **20 muestras/ciclo**).

Considere el SysTick habilitado y que interrumpe periódicamente cada 50ms. Utilice esta función de interrupción para hacer medidas de tiempos grandes y para leer el valor de la tensión analógica que proporciona el potenciómetro cada **50ms** en el **modo manual.**

a) Complete sobre el diagrama de la Figura 1, el pin **Pn.x** (sobre la línea de conexión) y el nombre del recurso utilizado dentro del bloque que representa la Mini-DK2 (ej. **MAT1.0**). (5 ptos)

b) Complete la función de configuración del ADC e indique la **frecuencia de muestreo** del canal seleccionado considerando que la **Fclk del ADC sea la mínima posible**. (5 ptos)

```
void init_ADC(void)
{
LPC_SC->PCONP|= (1<<12);              // Power ON
LPC_PINCON->PINSEL1|= (0x02<<18);     // P0.25 es AD0.2
LPC_PINCON->PINMODE1|=(0x02<<18);     // Deshabilita pullup/pulldown
LPC_ADC->ADCR= (    1<<2)|           // Canal 2
               (0xFF<<8)|           // CLKDIV=255   FclkADC=25MHz/256=97,656kHz
                 (1<<16)|           // Modo Burst
                 (1<<21);           // PDN=1
NVIC_DisableIRQ(ADC_IRQn);          // ADC no interrumpe!!!
}
```

$$F_{muestreo} = \frac{97{,}656\ \text{kHz}}{65} = 1502{,}4\ \text{Hz}$$

c) Complete la función de configuración de la señal PWM, y de actualización de la posición del servo. Considere que el periodo sea de **15ms**, y el tiempo a nivel alto varíe entre **0.6-2.4ms**, para un movimiento de su posición entre 0º y 180º. (10 ptos)

```
void config_pwm(void)
{
LPC_PINCON->PINSEL3|=(2<<8);
LPC_SC->PCONP|=(1 <<6);  //Power PMW module
LPC_PWM1->MR0=F_pclk*15e-3 -1; // frecuencia PWM
LPC_PWM1->PCR|=(1<<10); //ENA2=1
LPC_PWM1->MCR|=(1<<1);  //Reset on Match
LPC_PWM1->TCR|=(1<<0)|(1<<3); //Start
}

void set_servo(uint8_t grados)
{
LPC_PWM1->MR2 = F_pclk*0.6e-3 + F_pclk*1.8e-3*grados/180;
LPC_PWM1->LER|= (1<<2)|(1<<0);
}
```

d) Complete la función de interrupción que permite realizar la medida de la distancia en centímetros, a partir del pulso que entrega el sensor SRF04. Considere que la velocidad de los ultrasonidos es **342m/s** y que la variable global ***distancia***, se modifica con la distancia al obstáculo en centímetros. Explique la configuración del recurso utilizado (TimerN) sin escribir el código (puede utilizar pseudocódigo). (15 ptos)

NOTA 1: Considere que el timer cuenta con una resolución de 1 microsegundo.
NOTA2: Tenga en cuenta que se ha de sacar por el puerto serie la medida obtenida (un * si la medida no es válida). En el Anexo 2 están las funciones de la UART. Considere si fuese el caso la activación de la señal de alarma.

```
void TIMER3_IRQHandler(void)
{
   LPC_TIM3->IR|=(1<<5);                              // borrar flag CR1
   pulso_duracion= LPC_TIM3->CR0-LPC_TIM3->CR1;     // en microseg.
   distancia= pulso_duracion*342e-6*100/2;          // en centímetros
   if(distancia > umbral)LPC_TIM1->TCR=0x02;         // stop timer
   else LPC_TIM1->TCR=0x01;                         // start timer

   if(pulso_duracion>24000)   // 24ms
   {
   medida_OK=0;          // medida erronea
   tx_cadena("*\n\r");
   }
   else
   {
   medida_OK=1;
   sprintf(cadena,"Distancia= %3.2f \n\r",distancia); // 8 caracteres
   tx_cadena(cadena);
   }

   if(modo&&Start) set_servo(grados_barrido);  // garantizamos que el servo
                                               // alcanza su posición antes del
                                               // disparo en el modo barrido
}
```

Configuración Timer:

Timer 3 modo captura:
- CAP3.0 → flanco subida (NO interrumpe)
- CAP3.1 → flanco bajada (SI interrumpe)

Preescaler = 24 (Ftick = Fpclk/25 = 1Mhz)
Modo running (cuenta continuamente)

e) Escriba las funciones de interrupción de los pulsadores KEY1 y KEY2 de la tarjeta Mini-DK2. Considere que existe la variable global ***modo***, que define el modo de funcionamiento (*modo*=0, manual; *modo*=1, barrido) así como la variable(s) que considere necesarias. Active el LED correspondiente en función del modo seleccionado e inicie la medida, o de la orden de inicio del barrido (utilice la variable **Start=1**) según corresponda. (6 ptos)

```
void EINT1_IRQHandler(void) // Key 1
{
LPC_SC->EXTINT=(1<<1); //borrar flag
modo^=1;
if(modo)LPC_GPIO3->FIOPIN=(1<<25); //LED3 activo
else LPC_GPIO3->FIOPIN=(2<<25);     //LED2 activo
}
void EINT2_IRQHandler(void) // Key 2
{
LPC_SC->EXTINT=(1<<2); //borrar flag
if(modo==0)  disparo_SRF04();
else{
      Start=1;
      set_servo(0); // posición 0°
      }
}
```

f) Escriba la función de interrupción del Timer que saca las muestras hacia el DAC para generar la señal de alarma, y el valor del registro **MRx** correspondiente para obtener la frecuencia deseada. Considere ya inicializado el array ***muestras[20]*** con los valores discretos de un ciclo. (4 ptos)

```
void TIMER1_IRQHandler(void)
{
LPC_TIM1->IR|=(1<<0);   // borrar flag
LPC_DAC->DACR= muestras[indice++]<<6;
if (indice==20)indice=0;
}
```

El periodo de interrupción queda determinado por tiempo hasta alcanzar el valor del registro Match.
Considerando la frecuencia de salida de 1kHz y 20 muestras/ciclo

MR0=F_pclk/1000/20 – 1 = 1249

h) Complete dentro de la función de interrupción del SysTick el código encargado de realizar el barrido y de actualizar la posición del servo en el modo manual. Considere que existe la variable ***N_pasos*** que configura el número de pasos o medidas a realizar dentro de cada barrido (*N_pasos*=180º/resolucion), la variable ***Start*** que al ponerla a uno inicia el barrido, y cuantas variables auxiliares precise. (10 ptos)

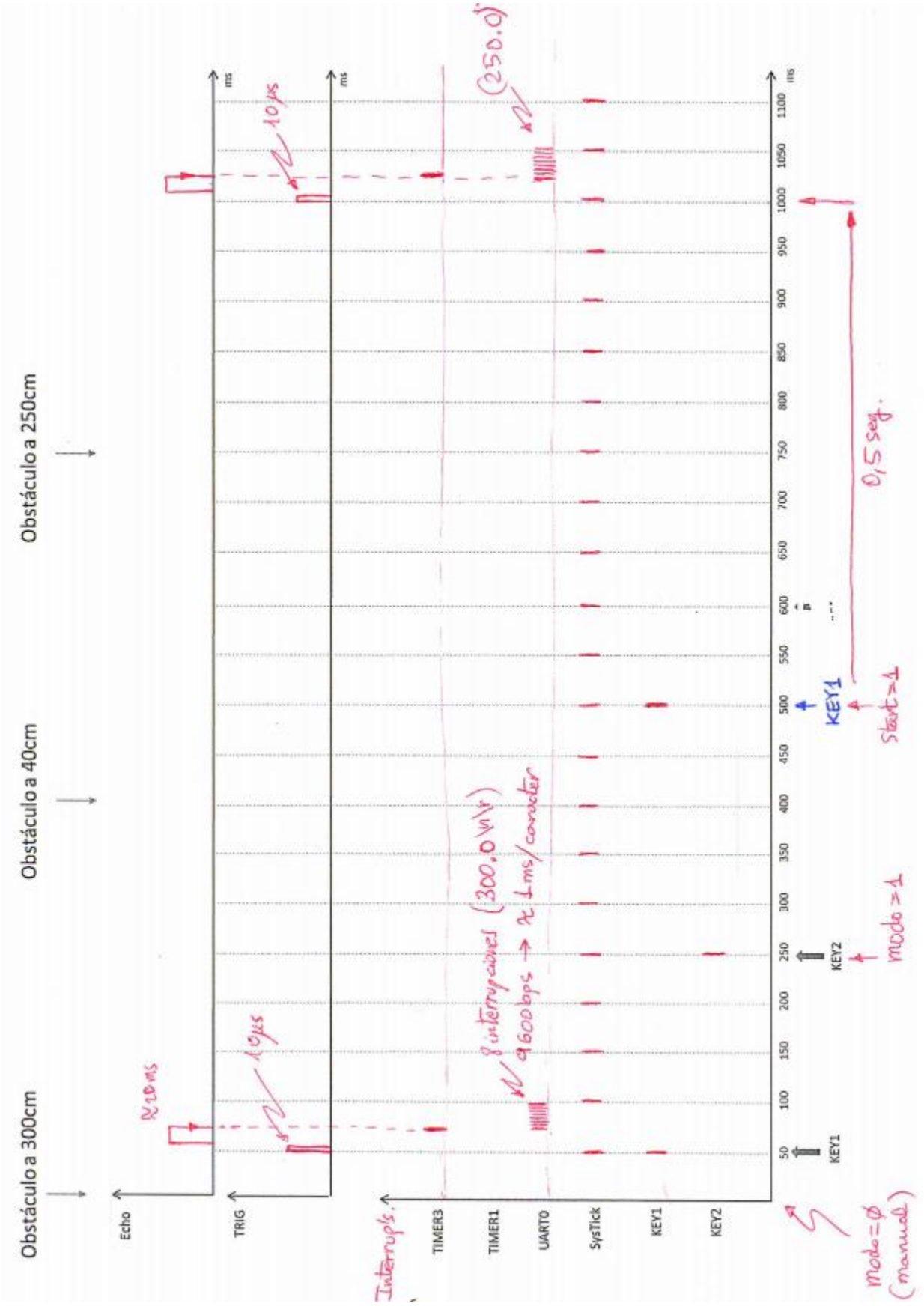
NOTA 1: Tenga en cuenta que el servo avanza su posición cada **0,5 segundos**.
NOTA 2: Considere ya escrita la función ***disparo_SRF04()*** que provoca el pulso de 10us de duración por la entrada TRIG para iniciar una medida.

```
void SysTick_IRQHandler(void)
{
// se ejecuta cada 50mseg
   if (modo&&Start) {
   contador++;
                  if(contador==10){
                                 contador=0;
                                 disparo_SRF04();
                                 pasos++;
                                 grados_barrido+=resolucion;
                                 // ver NOTA
                                 if(pasos==N_pasos){
                                                    Start=0;
                                                    pasos=0;
                                                    fin_barrido=1;
                                                  }

   else if(modo==0){
   grados=((LPC_ADC_ADDR2>>8)&0xFF)*180/255; // valor del potenciómetro
   set_servo(grados);
   }
}
```

NOTA: La orden de mover el servo se da dentro de la interrup. del Timer 3 para garantizar que alcanza su posición antes de realizar la medida. Esto es una de las posibles alternativas.

i) Complete el diagrama temporal de manera aproximada de las tareas asociadas a las distintas fuentes de interrupción y señales de control, en función de la aparición de los eventos de entrada señalados. Tenga en cuenta que después del reset el escáner está en **modo manual**. El umbral de detección se fija en **50cm**. (5 ptos)



Ejercicio 2 (40 puntos)

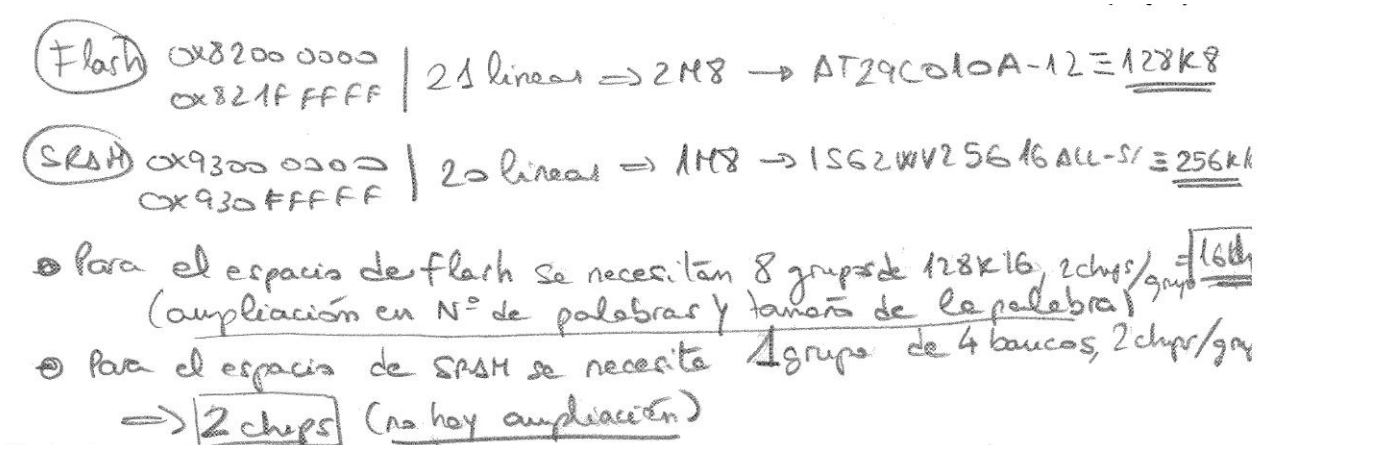
Se desea implementar un espacio de memoria externa para el LPC1788 (CCLK=100MHz) con las siguientes características:

- El espacio incluye una zona de Flash en el rango **0x82000000-0x821FFFFF**, y otro de SRAM a partir de la dirección **0x93000000** de la mitad de capacidad.
- Para implementar el espacio de Flash se usan dispositivos **AT29C010A-12**.
- Para implementar el espacio de RAM se usan dispositivos **IS62WV25616ALL-55**.

NOTA: Las características del EMC y de las memorias se adjuntan en los Anexos 3 y 4.

Se pide:

a) Indique de manera justificada, cuántos dispositivos de memoria Flash y RAM son necesarios para implementar el espacio deseado y con qué tipo de ampliación u ordenación sería necesario conectarlos. Suponga que el espacio de SRAM debe ser direccionable en 8, 16 y 32 bits, mientras que el de Flash sólo debe serlo en 16 bits. (10 ptos.)

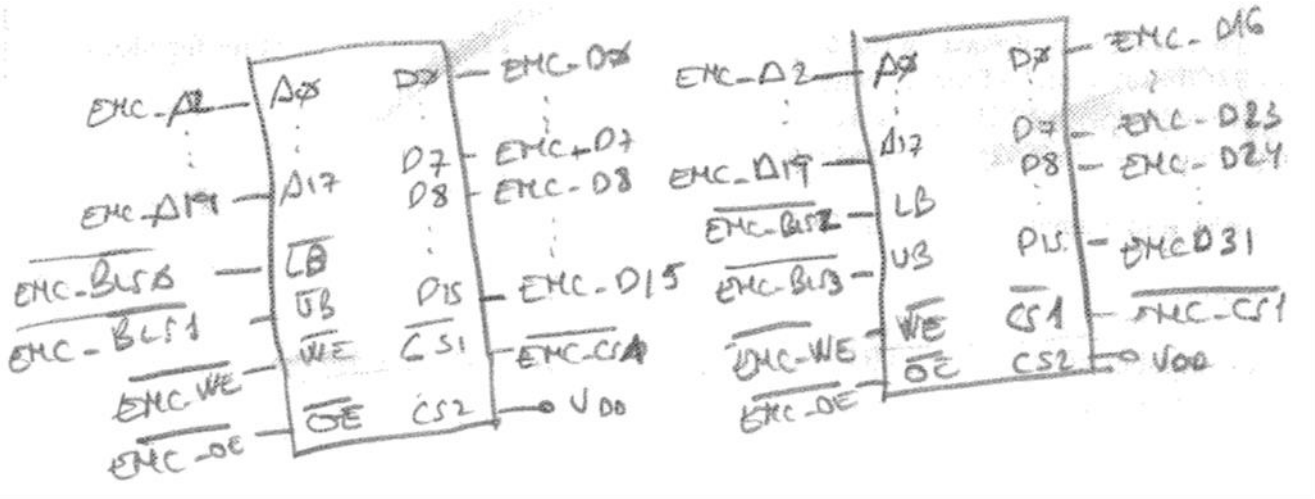


b) Configure, justificadamente, los registros STATICCONFIG necesarios para generar las señales de **chip select** necesarias. (3 ptos.)

Flash → EMC_CS0, STATICCONFIG0 = 0b0000xx0x01
Como se accede a 16 bits a la vez, BLS0 y BLS1 no actúan ⇒ PB = 0 (16 bits)

SRAM → EMC_CS1, STATICCONFIG1 = 0b0100xx0x10
BLS's controlan el acceso a las bancas en lectura y escritura ⇒ PB = 1 (32 bits)

c) Dibuje el diagrama de conexión de la memoria SRAM con el EMC del LPC1788. (10 ptos)



d) Configure justificadamente **el registro STATICWAITRD** necesario para que los tiempos de selección del microcontrolador se adapten a la velocidad de acceso en lectura de las memorias Flash. En los Anexos 5 y 6 se encuentran las características temporales y los cronogramas de acceso del LPC1788. (10 ptos.)

Como solo se modifica STATICWAITRD, la peor circunstancia es $t_{CE} < t_{ROS}|_{min}$ (STATICWAITOEN = 0)

$t_{CE} = 120ns$. $T_{CLK} = \frac{1}{100MHz} = 10ns$

$t_{ROS}|_{min} = (WAITRD - WAITOEN + 1) \times T_{CLK} - 8'6$.

$WAITRD \geq \frac{t_{CE} 18'6}{T_{CLK}} - 1 = \frac{120 + 8'6}{10} - 1 = 11'86$.

WAITRD = 12 = 0x0C

e) Finalmente se desea añadir un espacio de **EEPROM** a continuación del de Flash, de **2Mbytes**. Complete la decodificación de los 3 segmentos con lógica externa que permita obtener las señales $\overline{CSRAM}$, $\overline{CSFLASH}$ y $\overline{CSEEPROM}$, según las especificaciones dadas. (7 ptos.)

Decodificación completa de $\overline{CSRAM}$, $\overline{CSFLASH}$ y $\overline{CSEEPROM}$ a partir de $\overline{EMC-CS0}$ y $\overline{EMC-CS1}$

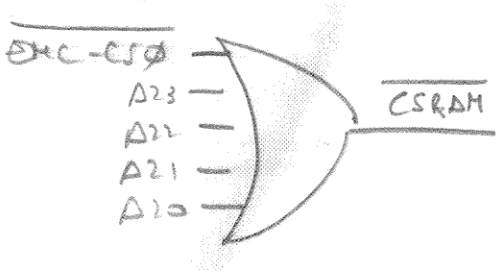
Flash: 0x8200-0000 0x821F_FFFF

EEPROM: 0x8220-0000 0x823F_FFFF

SRAM: 0x9300-0000 0x930f_FFFF

A23 - A20	A19 ... A0	$\overline{CS}$
0 0 0 0	X ... X	$\overline{CSRAM}$

A23 - A20	A19 ... A0	$\overline{CS}$
0 0 0 X	X ... X	$\overline{CSFLASH}$
0 0 1 X	X ... X	$\overline{CSEEPROM}$



ANEXO 1. Descripción del sensor SRF04

El módulo SRF04 (Figura 2) es un sensor de distancias por ultrasonidos capaz de detectar objetos y calcular la distancia a la que se encuentra en un rango de 3 a 300 cm. El sensor funciona por ultrasonidos y contiene toda la electrónica encargada de hacer la medición. Su uso es tan sencillo como enviar el pulso de disparo para iniciar una medida y medir la anchura del pulso recibido a su salida.

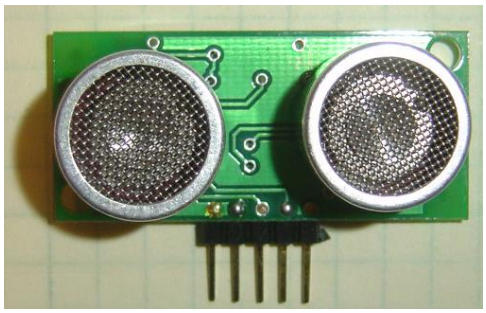


Figura 2. Aspecto del sensor SRF04

Funcionamiento

El sensor SRF04 funciona emitiendo impulsos de ultrasonidos inaudibles para el oído humano. Los impulsos emitidos viajan a la velocidad del sonido hasta alcanzar un objeto, entonces el sonido es reflejado y captado de nuevo por el receptor de ultrasonidos. Lo que hace el controlador incorporado al recibir una señal de disparo, es emitir una ráfaga de impulsos (8 ciclos de 40kHz) y a continuación empieza a contar el tiempo que tarda en llegar el eco. Este tiempo se traduce en un pulso de eco de anchura proporcional a la distancia a la que se encuentra el objeto (Figura 3).

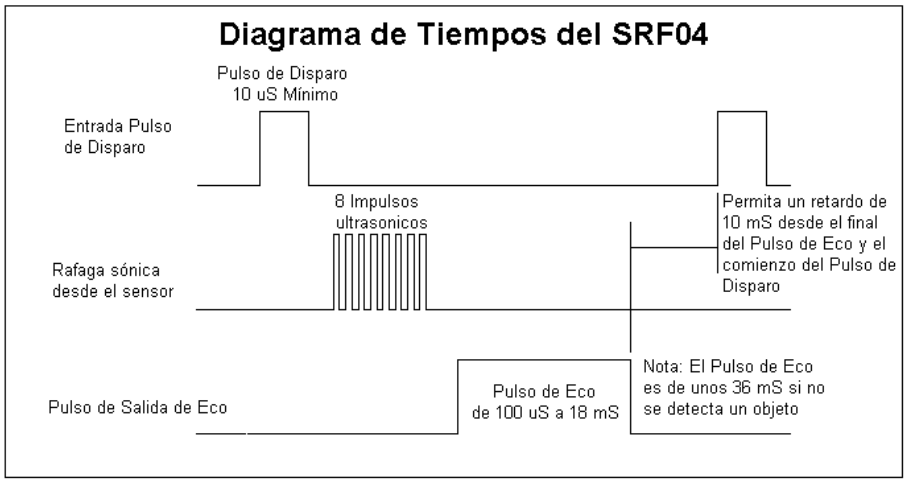


Figura 3. Diagrama de tiempos del sensor ultrasónico SRF04

Desde un punto de vista práctico, lo que hay que hacer es mandar una señal de disparo en el pin 3 del SRF04 y después leer la anchura del impulso que nos proporciona en el pin 2. El pulso de disparo tiene que tener una anchura mínima de **10 µS**. Después leemos el pulso de salida de Eco y medimos su longitud que es proporcional al eco recibido. En caso de que no se produzca ningún eco, porque no se encuentra un objeto, el pulso de eco tiene una longitud aproximada de **36 ms**. Hay que dejar un retardo de 10 ms desde que se hace una lectura hasta que se realiza la siguiente, con el fin de que el circuito se estabilice. En la Figura 4 se muestran las características del sensor y los pines de conexión.

Tensión	5V
Consumo	30 mA Tip. 50mA Max.
Frecuencia:	40 Khz.
Distancia Mínima:	3 cm.
Distancia Máxima:	300 cm.
Sensibilidad:	Detecta un palo de escoba a 3 m.
Pulso de Disparo	10 uS min. TTL
Pulso de Eco:	100 uS - 18 mS
Retardo entre pulsos:	10 mS Mínimo
Pulso de Eco:	100 uS - 18 mS
Tamaño:	43 x 20 x 17 mm
Peso:	10 gr.

Figura 4. Características y conexionado del SRF04

ANEXO 2. Funciones de control del Puerto Serie

```
void UART0_IRQHandler(void) {

   switch(LPC_UART0->IIR&0x0E) {

   case 0x04:                  /* RBR, Receiver Buffer Ready */
     *ptr_rx=LPC_UART0->RBR;   /* lee el dato recibido y lo almacena */
     if(*ptr_rx++ ==13){       /* Caracter return --> Cadena completa */
       *ptr_rx=0;              /* Añadimos el caracter null para tratar los datos recibidos como una cadena*/
       rx_completa = 1;        /* rx completa */
       ptr_rx=buffer;          /* puntero al inicio del buffer para nueva recepción */
     }
     break;

   case 0x02:                      /* THRE, Transmit Holding Register empty */
     if(*ptr_tx!=0)
       LPC_UART0->THR = *ptr_tx++; /* carga un nuevo dato para ser transmitido */
     else
       tx_completa=1;
     break;
   }
}
```

```
void tx_cadena_UART0(char *cadena)
{
   ptr_tx=cadena;
   tx_completa=0;
   LPC_UART0->THR = *ptr_tx;   // IMPORTANTE: Introducir un carácter al comienzo para iniciar TX o
}                              // activar flag interrupción por registro transmisor vacio
```

ANEXO 3. Características EMC

Name	Description
STATICCONFIGn	• **Memory width** (1:0) 0x0: 8 bit (reset) 0x1: 16 bit 0x2: 32 bit • **Page mode** (3). The EMC can burst up to 4 external accesses 0x0: Async page mode disable (reset) 0x1: Async page mode enabled • **Chip select polarity** (6) 0x0: Active LOW chip select 0x1: Active HIGH chip select • **Byte lane state (7).** Enables different types of memory to be connected 0x0: For reads all the bits in BLSn[3:0] are HIGH. For writes all the bits in BLSn[3:0] are LOW 0x1: For reads and writes the respective active bits in BLSn[3:0] are LOW • **Extended wait (8).** Uses StaticExtendedWait register to time both read and write transfers 0x0: Extended wait disabled (reset value) 0x1: Extended wait enabled

Four static memory chip selects:	
0x8000 0000 - 0x83FF FFFF	Static memory chip select 0 (up to 64 MB)
0x9000 0000 - 0x93FF FFFF	Static memory chip select 1 (up to 64 MB)
0x9800 0000 - 0x9BFF FFFF	Static memory chip select 2 (up to 64 MB)
0x9C00 0000 - 0x9FFF FFFF	Static memory chip select 3 (up to 64 MB)

Data bus pins	Address bus pins	Control pins SRAM
EMC_D[31:0]	EMC_A[25:0]	$\overline{\text{EMC\_BLS[3:0]}}$, $\overline{\text{EMC\_CS[3:0]}}$, $\overline{\text{EMC\_OE}}$, $\overline{\text{EMC\_WE}}$

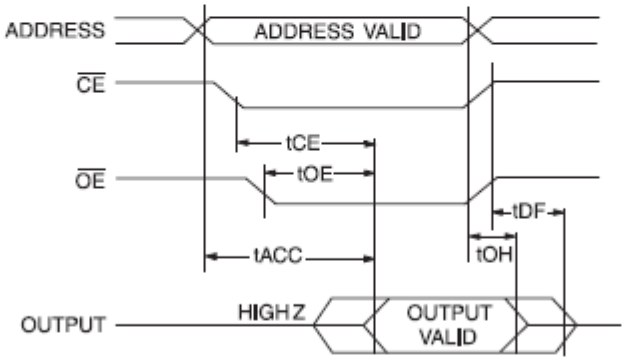
ANEXO 4. Características de Memorias (Ejercicio 2)

AT29C010A-12

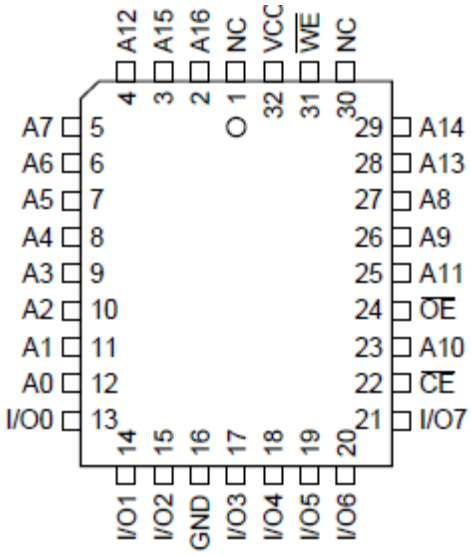
AC Read Characteristics

Symbol	Parameter	AT29C010A-70		AT29C010A-90		AT29C010A-12		AT29C010A-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{ACC}	Address to Output Delay		70		90		120		150	ns
t_{CE} (1)	$\overline{CE}$ to Output Delay		70		90		120		150	ns
t_{OE} (2)	$\overline{OE}$ to Output Delay	0	35	0	40	0	50	0	70	ns
t_{DF} (3)(4)	$\overline{CE}$ or $\overline{OE}$ to Output Float	0	25	0	25	0	30	0	40	ns
t_{OH}	Output Hold from $\overline{OE}$, $\overline{CE}$ or Address, whichever occurred first	0		0		0		0		ns

AC Read Waveforms(1)(2)(3)(4)



Notes:
1. $\overline{CE}$ may be delayed up to $t_{ACC} - t_{CE}$ after the address transition without impact on t_{ACC}.
2. $\overline{OE}$ may be delayed up to $t_{CE} - t_{OE}$ after the falling edge of $\overline{CE}$ without impact on t_{CE} or by $t_{ACC} - t_{OE}$ after an address change without impact on t_{ACC}.
3. t_{DF} is specified from $\overline{OE}$ or $\overline{CE}$ whichever occurs first (CL = 5 pF).
4. This parameter is characterized and is not 100% tested.



Pin Name	Function
A0 - A16	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}$	Output Enable
$\overline{WE}$	Write Enable
I/O0 - I/O7	Data Inputs/Outputs
NC	No Connect

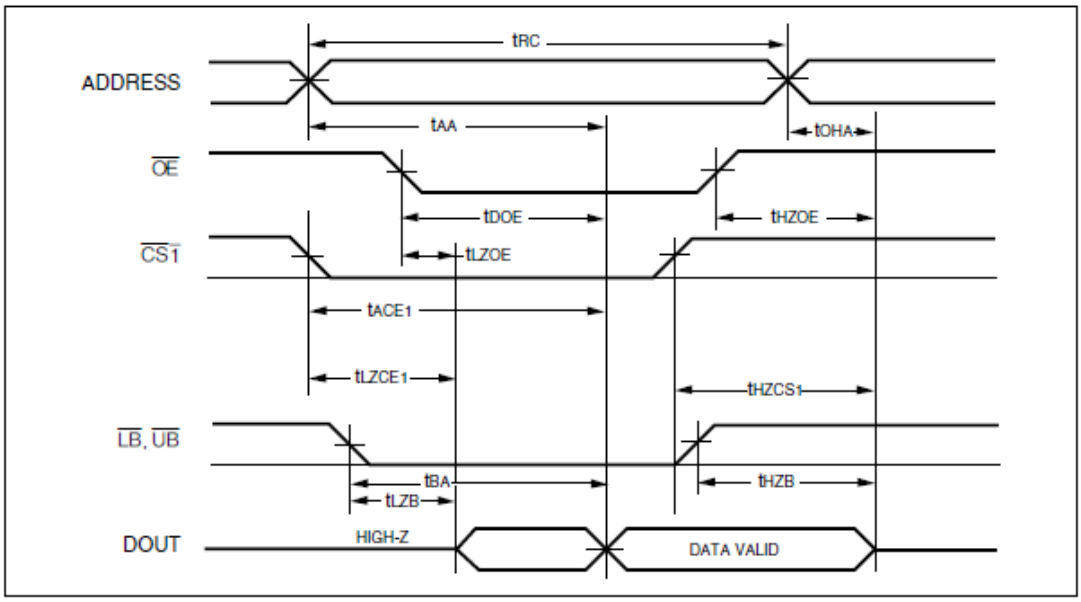
IS62WV25616ALL-55

READ CYCLE SWITCHING CHARACTERISTICS[1] (Over Operating Range)

Symbol	Parameter	55 ns Min.	55 ns Max.	70 ns Min.	70 ns Max.	Unit
t_{RC}	Read Cycle Time	55	—	70	—	ns
t_{AA}	Address Access Time	—	55	—	70	ns
t_{OHA}	Output Hold Time	10	—	10	—	ns
t_{ACS1}	$\overline{CS1}$ Access Time	—	55	—	70	ns
t_{DOE}	$\overline{OE}$ Access Time	—	25	—	35	ns
t_{HZOE}[2]	$\overline{OE}$ to High-Z Output	—	20	—	25	ns
t_{LZOE}[2]	$\overline{OE}$ to Low-Z Output	5	—	5	—	ns
t_{HZCS1}	$\overline{CS1}$ to High-Z Output	0	20	0	25	ns
t_{LZCS1}	$\overline{CS1}$ to Low-Z Output	10	—	10	—	ns
t_{BA}	$\overline{LB}$, $\overline{UB}$ Access Time	—	55	—	70	ns
t_{HZB}	$\overline{LB}$, $\overline{UB}$ to High-Z Output	0	20	0	25	ns
t_{LZB}	$\overline{LB}$, $\overline{UB}$ to Low-Z Output	0	—	0	—	ns

Notes:
1. Test conditions assume signal transition times of 5 ns or less, timing reference levels of 0.9V/1.5V, input pulse levels of 0.4 to V_{DD}-0.2V/V_{DD}-0.3V and output loading specified in Figure 1.
2. Tested with the load in Figure 2. Transition is measured ±500 mV from steady-state voltage. Not 100% tested.

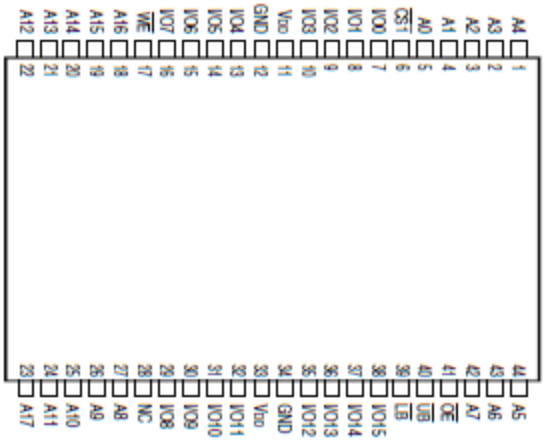
READ CYCLE NO. 2[1,3] ($\overline{CS1}$, $\overline{OE}$, AND $\overline{UB}$/$\overline{LB}$ Controlled)



Notes:
1. $\overline{WE}$ is HIGH for a Read Cycle.
2. The device is continuously selected. $\overline{OE}$, $\overline{CS1}$, $\overline{UB}$, or $\overline{LB}$ = V_{IL}. $\overline{WE}$=V_{IH}.
3. Address is valid prior to or coincident with $\overline{CS1}$ LOW transition.

PIN DESCRIPTIONS

A0-A17	Address Inputs
I/O0-I/O15	Data Inputs/Outputs
$\overline{CS1}$, CS2	Chip Enable Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
$\overline{LB}$	Lower-byte Control (I/O0-I/O7)
$\overline{UB}$	Upper-byte Control (I/O8-I/O15)
NC	No Connection
V_{DD}	Power
GND	Ground



ANEXO 5. Características temporales de los ciclos de lectura-escritura en los accesos a memoria externa estática del LPC178x (Ejercicio 2)

Table 17. Dynamic characteristics: Static external memory interface

C_L = 30 pF, T_{amb} = −40 °C to 85 °C, $V_{DD(3V3)}$ = 3.0 V to 3.6 V. Values guaranteed by design.

Symbol	Parameter[1]	Conditions[1]		Min	Typ	Max	Unit
Read cycle parameters[2]							
t_{CSLAV}	$\overline{CS}$ LOW to address valid time	RD_1		2.7	3.5	4.7	ns
t_{CSLOEL}	$\overline{CS}$ LOW to $\overline{OE}$ LOW time	RD_2	[3]	2.7 + $T_{cy(clk)}$ × WAITOEN	3.4 + $T_{cy(clk)}$ × WAITOEN	4.6 + $T_{cy(clk)}$ × WAITOEN	ns
$t_{CSLBLSL}$	$\overline{CS}$ LOW to $\overline{BLS}$ LOW time	RD_3; PB = 1	[3]	2.8	3.8	5.1	ns
t_{OELOEH}	$\overline{OE}$ LOW to $\overline{OE}$ HIGH time	RD_4	[3]	(WAITRD − WAITOEN + 1) × $T_{cy(clk)}$ − 2.26	(WAITRD − WAITOEN + 1) × $T_{cy(clk)}$ − 2.83	(WAITRD − WAITOEN + 1) × $T_{cy(clk)}$ − 3.7	ns
t_{am}	memory access time	RD_5	[3][4]	(WAITRD − WAITOEN + 1) × $T_{cy(clk)}$ − 8.6	(WAITRD − WAITOEN + 1) × $T_{cy(clk)}$ − 11.9	(WAITRD − WAITOEN + 1) × $T_{cy(clk)}$ − 18.0	ns
$t_{h(D)}$	data input hold time	RD_6	[3][5]	−4.1	−5.8	-	ns
$t_{CSHBLSH}$	$\overline{CS}$ HIGH to $\overline{BLS}$ HIGH time	PB = 1		2.8	3.7	5.1	ns
t_{CSHOEH}	$\overline{CS}$ HIGH to $\overline{OE}$ HIGH time		[3]	2.7	3.5	4.6	ns
t_{OEHANV}	$\overline{OE}$ HIGH to address invalid time		[3]	0.1	0.1	0.16	ns
t_{deact}	deactivation time	RD_7	[3]	-	−3.4	−4.7	ns
Write cycle parameters[2]							
t_{CSLAV}	$\overline{CS}$ LOW to address valid time	WR_1		2.7	3.5	4.7	ns
t_{CSLDV}	$\overline{CS}$ LOW to data valid time	WR_2		2.8	3.9	5.1	ns
t_{CSLWEL}	$\overline{CS}$ LOW to $\overline{WE}$ LOW time	WR_3; PB =1	[3]	2.7 + $T_{cy(clk)}$ × (1 + WAITWEN)	3.5 + $T_{cy(clk)}$ × (1 + WAITWEN)	4.6 + $T_{cy(clk)}$ × (1 + WAITWEN)	ns
$t_{CSLBLSL}$	$\overline{CS}$ LOW to $\overline{BLS}$ LOW time	WR_4; PB = 1	[3]	2.8	3.9	5.1	ns
t_{WELWEH}	$\overline{WE}$ LOW to $\overline{WE}$ HIGH time	WR_5; PB =1	[3]	(WAITWR − WAITWEN + 1) × $T_{cy(clk)}$ − 2.3	(WAITWR − WAITWEN + 1) × $T_{cy(clk)}$ − 2.8	(WAITWR − WAITWEN + 1) × $T_{cy(clk)}$ − 3.8	ns
$t_{BLSLBLSH}$	$\overline{BLS}$ LOW to $\overline{BLS}$ HIGH time	PB = 1	[3]	(WAITWR − WAITWEN + 3) × $T_{cy(clk)}$ − 2.6	(WAITWR − WAITWEN + 3) × $T_{cy(clk)}$ − 3.4	(WAITWR − WAITWEN + 3) × $T_{cy(clk)}$ − 4.9	ns
t_{WEHDNV}	$\overline{WE}$ HIGH to data invalid time	WR_6; PB =1	[3]	2.5 + $T_{cy(clk)}$	3.3 + $T_{cy(clk)}$	4.3 + $T_{cy(clk)}$	ns
t_{WEHEOW}	$\overline{WE}$ HIGH to end of write time	WR_7; PB = 1	[3][6]	$T_{cy(clk)}$ − 2.7	$T_{cy(clk)}$ − 3.4	$T_{cy(clk)}$ − 4.6	ns
$t_{BLSHDNV}$	$\overline{BLS}$ HIGH to data invalid time	PB = 1		2.7	3.6	4.8	ns
t_{WEHANV}	$\overline{WE}$ HIGH to address invalid time	PB = 1	[3]	2.4 + $T_{cy(clk)}$	3.0 + $T_{cy(clk)}$	3.9 + $T_{cy(clk)}$	ns

ANEXO 6. Cronogramas lectura-escritura en los accesos a memoria externa estática del LPC178x (Ejercicio 2)

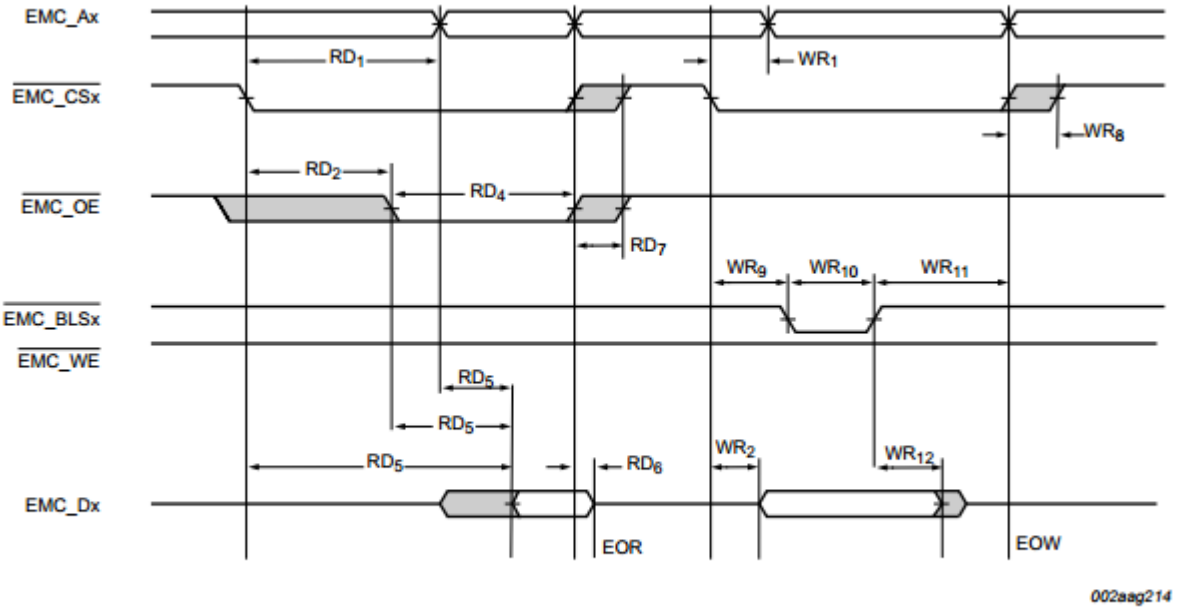


Fig 16. External static memory read/write access (PB = 0)

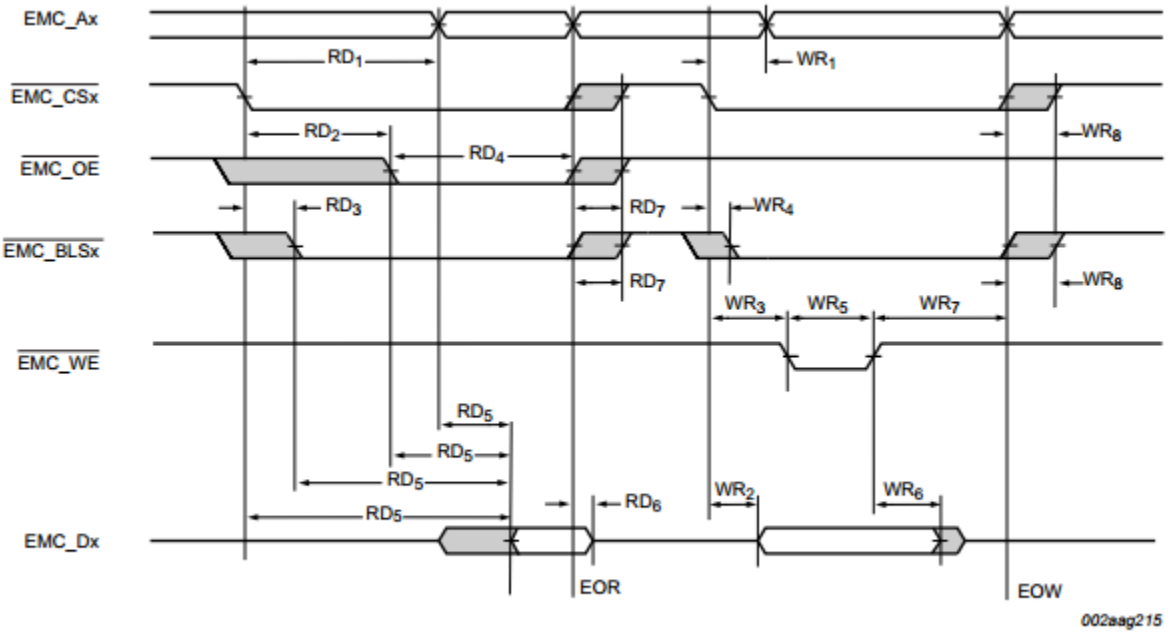


Fig 17. External static memory read/write access (PB =1)