



**ESCUELA POLITÉCNICA SUPERIOR DE ALCOY**

Ingeniería Técnica Telecomunicación  
Especialidad Telemática



# **Memoria 24LC512 con Interface I<sup>2</sup>C**

Grupo A01-A04  
Proyecto PAEEES 04/993 UPV.  
Escuela Politécnica Superior de Alcoy  
Marzo 2005

*Javier Antolí González*  
*F. Manuel Díaz-Madroñero Boluda*  
*Javier Moscardó Salido*  
*Jorge Peiró Marqués*  
*Sergio Soler Martín*  
*David Tomás Jover*

## ÍNDICE

1. Descripción del circuito.....	1
2. Características del circuito .....	1
3. Características eléctricas.....	2
• Características DC .	
• Características AC	
4. Consumo del Circuito.....	3
5. Descripción de Pines.....	3
6. Registros Internos del 24LC512.....	5
7. Descripción y temporización del I2C.....	5
8. Modos de funcionamiento del 24LC512 .....	10
• Direccionamiento de Dispositivos	
• Operaciones de Escritura	
• Operaciones de Lectura	
9. Ejemplo de Utilización de 24LC512.....	17
• Función escritura de un byte	
• Función lectura de un byte	
10. Bibliografía.....	21

## 1. DESCRIPCIÓN DEL CIRCUITO

El C.I. 24LC512 es una memoria serie del tipo 64K x 8 E<sub>2</sub>PROM, es decir, una memoria eléctricamente borrable de 512 Kbits, capaz de operar en un amplio rango de tensiones (2.5 a 5.5v) y a una frecuencia máxima de reloj de 400KHz.

Este dispositivo permite realizar lecturas aleatorias y secuenciales de hasta 512 Kbits, es decir, el espacio total direccionable que puede ser ampliado hasta 4 Mbits mediante la utilización de tres líneas adicionales de direccionamiento que permiten conectar hasta ocho dispositivos en un mismo bus.

Está disponible en estándar de ocho pins plástico DIP, SOIC, DFN y 14 pins TSSOP.

## 2. CARACTERÍSTICAS DEL CIRCUITO

Entre las características del 24LC512 podemos enumerar las siguientes:

- Tecnología CMOS de baja potencia:
  - La corriente máxima en escritura es de 5 mA a 5.5V
  - La corriente máxima en lectura es de 400  $\mu$ A a 5.5V
  - La corriente de espera (standby) a 5.5V son 100 nA
- El bus formado por dos hilos con una interfaz serie, compatible con el estándar I2C
- Protección de escritura del hardware
- En un mismo ciclo pueden ser realizadas operaciones de borrado y escritura.
- La capacidad de cada página de escritura es de 128 bytes.
- Circuito interno de supresión de ruido.
- Hasta 1.000.000 de ciclos de borrado y escritura
- El tiempo máximo de escritura por ciclo es de 5ms.
- El dispositivo cuenta con una protección contra las descargas de electricidad estática de 4.000V.
- Acabados disponibles para soldaduras estándares y libres de plomo. (obligatorio a partir de Julio de 2005)
- Tiempo máximo de permanencia de datos de 200 años.
- Rangos de temperaturas de funcionamiento:
  - Industrial (I): -40°C a +85°C.
  - Automoción (E): -40°C a +125°C.

### 3. CARACTERÍSTICAS ELÉCTRICAS

- Tensión de alimentación (Vcc).....6,5V
- Temperatura de almacenaje.....-65°C a +150°C.
- Temperatura de uso.....-40°C a +125°C.
- ESD protección de los pines.....4 KV.

#### CARACTERÍSTICAS DC

Parámetro	Símbolo	Características	Min	Max	Unidades
D1	--	A0,A1,A2,SCL, SDA, wp pins:	--	--	--
D2	Vih	Entrada voltaje Nivel alto	0,7	--	V
D3	Vil	Entrada voltaje Nivel bajo	--	0,3 0,2	V V
D4	Vhys	Entradas histéresis SDA,SCL	0,05	--	V
D5	Vol	Salida voltaje nivel bajo	--	0,40	V
D6	Ili	Corriente de fuga --	+1	μ	A
D7	Ilo	Corriente de salida -	+1	μ	A
D8	Cin, Cout	Pin capacitancia	--	10	pF
D9	Icc Read Icc Write	Corriente de funcionamiento	-- --	400 5	μA mA
D10	Iccs	Corriente de activado --		1	μA

#### CARACTERÍSTICAS AC

Parámetro	Símbolo	Características	Min	Max	Unidades
1	Fclk	Frecuencia de reloj	--	400	Khz
2	Thigh	Reloj a nivel alto	600	--	ns
3	Tlow	Reloj a nivel bajo	1.30 0	--	ns
4	Tr	SDA y SCL tiempo de subida	--	300	ns
5	Tf	SDA y SCL tiempo de bajada	--	300	ns
6	Thd:Sta Condiciones	iniciales tiempo de espera 600		--	ns
7	Tsu:Sta	Condiciones iniciales del tiempo retraso	600		

8	Thd:Dat	Tiempo de espera para datos de entrada	0	--	ns
9	Tsu:Dat	Tiempo de retraso de dato de entrada	100	--	ns
10	Tsu:Sto	Condiciones de parada tiempo de retraso	600	--	ns
11	Tsu:Wp	Tiempo de retraso de protección de escritura.	600	--	ns
12	Thd:wp	Tiempo espera de protección de escritura.	1300	--	ns
13	Taa	Salida de reloj	--	900	ns
14	Tbuf	Tiempo libre que debe estar el bus antes de una nueva transmisión	1300	--	ns
16	Tsp	Supresión del pico de entrada (SDA y SCL pins)	--	50	ns
17	Twc	Tiempo de escritura (byte o pagina)	--	5	ms
18	-- Resistencia		10 <sub>6</sub>	--	Ciclos

#### 4. CONSUMO DEL CIRCUITO

A partir de los valores de tensiones y corrientes suministrados por el fabricante podemos calcular el consumo del circuito para cada uno de los siguientes casos:

- Consumo máximo en escritura:  $P= 27.5 \text{ mW}$
- Consumo máximo en lectura:  $P= 2.2 \text{ mW}$
- Consumo en reposo:  $P= 0.055 \mu\text{W}$

#### 5. DESCRIPCIÓN DE PINES

**Tabla de funcionamiento de los pines para los diferentes encapsulados**

Nombre	PDIP	SOIC	TSSOP	DFN	Función
A0	1	1	1	1	Selección de chip
A1	2	2	2	2	Selección

## MEMORIA 24LC512 CON INTERFACE I2C

Grupo A01-A04/ Curso 04- 05

					de chip
(NC)	--	--	3, 4, 5	--	No conectados
A2	3	3	6	3	Selección de chip
Vss	4	4	7	4	Masa
SDA	5	5	8	5	Datos serie

### Entradas A0, A1 y A2

Las entradas A0, A1 y A2 son utilizadas por el 24LC512 para las operaciones con múltiples dispositivos. Los niveles lógicos en estas entradas se comparan con los bits correspondientes de la dirección esclavo. El chip es seleccionado si coinciden.

Pueden conectarse hasta ocho dispositivos en el mismo bus usando combinaciones de bits del Chip Select. Si estos pines se dejan sin conectar, las entradas serán conectadas internamente a Vss mediante un circuito interno de pull-down.

En la mayoría de las aplicaciones, la direcciones de entrada A0, A1 y A2 son cableadas a nivel lógico '0' o '1'. Para los usos en los cuales estos pines son controlados por el microcontrolador u otro dispositivo de lógica programable, los pines de selección de chip deben ser puestos a nivel lógico '0' o '1' antes de que la operación normal del dispositivo pueda proceder.

### Datos Serie (SDA)

Este es un pin bidireccional que se usa para transferir direcciones y datos de entrada y de salida de los dispositivos. Se trata de un pin en colector abierto, por lo tanto, el bus SDA requiere una resistencia de pull-up para conectarse a Vcc (típicamente 10 k. para 100 khz, 2 k. para 400 khz y 1 Mhz)

Para una transferencia de datos normal, SDA permite el cambio solo durante el nivel bajo de SDL. Los cambios durante el estado alto de SCL son reservados para indicar las condiciones de Start y Stop.

### Reloj Serie (SCL)

Esta entrada se utiliza para sincronizar la transferencia de datos con el dispositivo.

### Protección de Escritura (WP)

Este pin puede ser conectado a Vss o a Vcc. El circuito interno de pull-down de este pin mantendrá el dispositivo en un estado de no protección si permanece flotante, sin embargo, ésta no es una

práctica recomendable. Si WP se conecta a Vss, las operaciones sobre la memoria pueden ser realizadas (lectura/escritura de la memoria completa, desde 0000 a FFFF)

Si se conecta a Vcc, la operación de escritura se inhibe mientras que la operación de lectura no se ve afectada.

## 6. REGISTROS INTERNOS DEL 24LC512

Dentro de los registros internos del circuito 24LC512 podemos encontrar dos tipos de registros. Por un lado los registros de almacenamiento y por otro el registro de direcciones. Los registros que podemos encontrar son los siguientes:

- Principal registro de almacenamiento: Tiene una capacidad de almacenamiento de 512 Kbits, distribuidos en 64K celdas de 8 bits cada una.
- Buffer de página: Es utilizado en el modo de escritura de página como buffer intermedio de almacenamiento, previo a la escritura en memoria de los datos. Su tamaño es de 128 bytes
- Registro de direcciones: Se trata de un puntero de dirección de 2 bytes, pues apunta a direcciones de memoria de ese mismo tamaño.

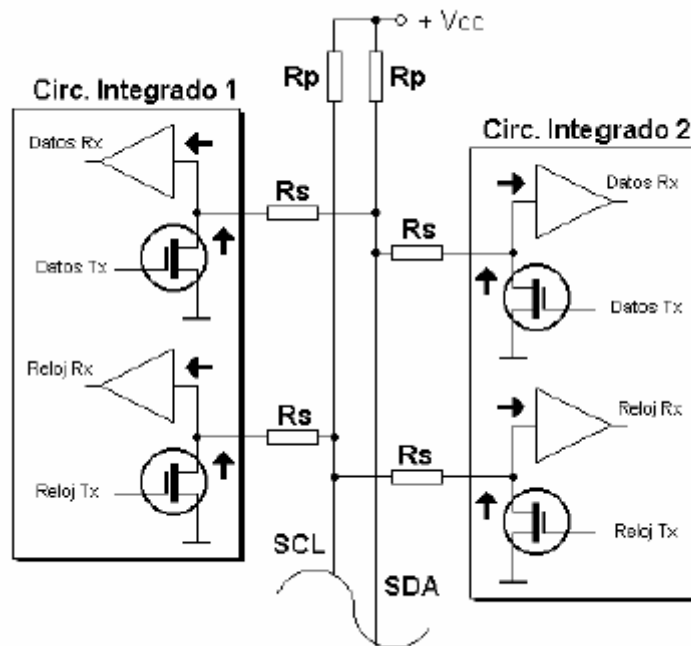
## 7. DESCRIPCIÓN Y TEMPORIZACIÓN DEL I<sup>2</sup>C

El bus I<sup>2</sup>C fue diseñado principalmente para la implementación de un camino de datos a corta distancia. Diseñado por Philips, este sistema de intercambio de información a través de tan solo dos cables permite a circuitos integrados y módulos OEM interactuar entre sí a velocidades relativamente lentas. Emplea comunicación serie, utilizando un conductor para manejar el timing (pulsos de reloj) y otro para intercambiar datos.

Este bus se basa en tres señales:

- **SDA (System Data)** por la cual viajan los datos entre los dispositivos.
- **SCL (System Clock)** por la cual transitan los pulsos de reloj que sincronizan el sistema.
- **GND (Masa)** Interconectada entre todos los dispositivos "enganchados" al bus.

Las líneas SDA y SCL son del tipo drenador abierto, similares a las de colector abierto pero asociadas a un transistor de efecto de campo (ó FET). Se deben poner en estado alto (conectar a la alimentación por medio de resistores Pull-Up) para construir una estructura de bus tal que se permita conectar en paralelo múltiples entradas y salidas.

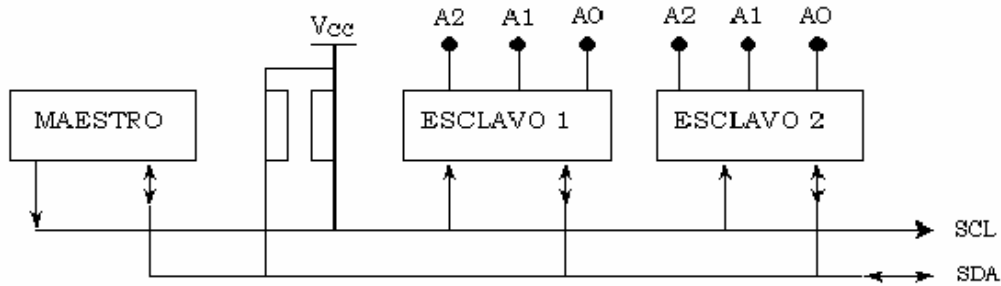


En el diagrama se observa la configuración eléctrica básica del bus. Las dos líneas de comunicación disponen de niveles lógicos altos cuando están inactivas. Inicialmente el número de dispositivos que se puede conectar al bus es ilimitado, pero obsérvese que las líneas tienen una especificación máxima de 400pF en lo que respecta a capacidad de carga. La máxima velocidad de transmisión de datos que se puede obtener en el bus es de aproximadamente 100Kbits por segundo, lo que limitará, como se verá posteriormente, la velocidad de lectura y escritura sobre la memoria a estudiar.

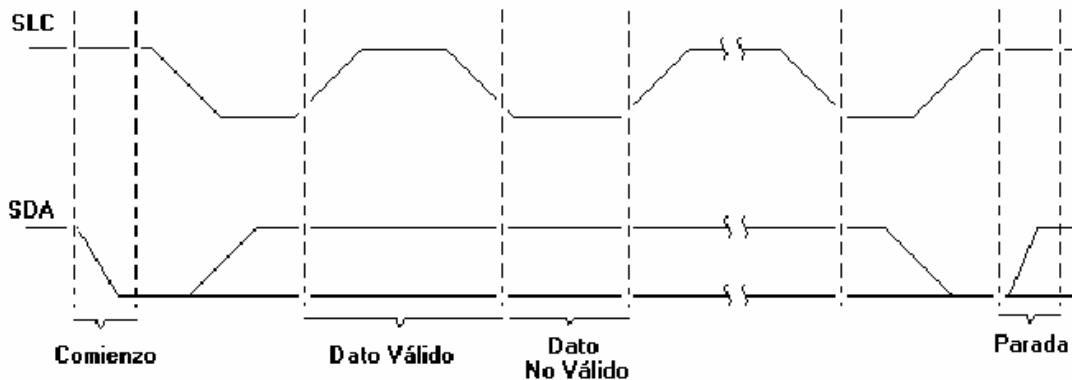
Las definiciones o términos utilizados en relación con las funciones del bus I<sup>2</sup>C son las siguientes:

- **Maestro (Master):** Dispositivo que determina la temporización y la dirección del tráfico de datos en el bus. Es el único que aplica los pulsos de reloj en la línea SCL. Cuando se conectan varios dispositivos maestros a un mismo bus la configuración obtenida se denomina "multi-maestro".
- **Esclavo (Slave):** Cualquier dispositivo conectado al bus incapaz de generar pulsos de reloj. Reciben señales de comando y de reloj proveniente del dispositivo maestro. Se pueden conectar hasta 128 dispositivos, que son los dispositivos seleccionados por el maestro mediante 7 bits (dirección del esclavo)



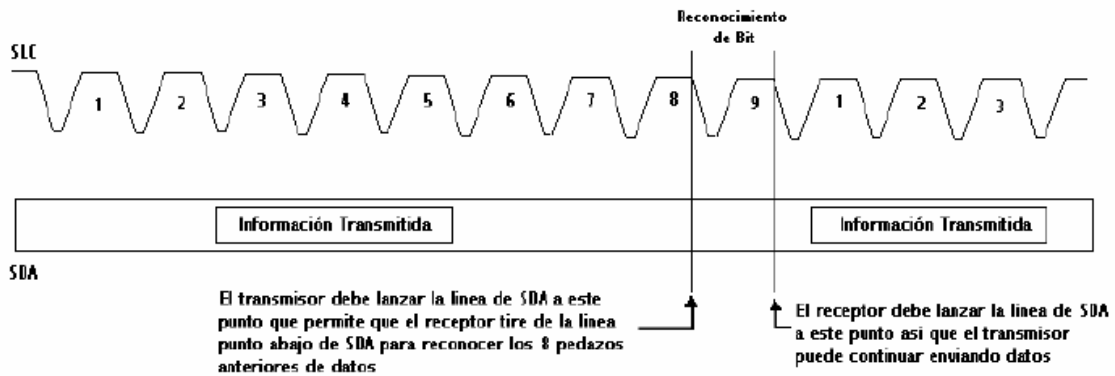


- **Bus Desocupado (Bus Free):** Estado en el cual ambas líneas (SDA y SCL) están inactivas, presentando un estado lógico alto. Únicamente en este momento es cuando un dispositivo maestro puede comenzar a hacer uso del bus.
- **Comienzo (Start):** Sucede cuando un dispositivo maestro hace ocupación del bus, generando esta condición. La línea de datos (SDA) toma un estado bajo mientras que la línea de reloj (SCL) permanece alta.
- **Parada (Stop):** Un dispositivo maestro puede generar esta condición dejando libre el bus. La línea de datos toma un estado lógico alto mientras que la de reloj permanece también en ese estado.
- **Dato Válido (Valid Data):** Sucede cuando un dato presente en la línea SDA es estable mientras la línea SCL está a nivel lógico alto.
- **Dato No válido (Invalid Data):** Sucede cuando un dato presente en la línea SDA es estable mientras en la línea SCL está a nivel lógico bajo.



- **Formato de Datos (Data Format):** La transmisión de datos a través de este bus consta de 8 bits de datos (ó 1 byte). A cada byte le sigue un noveno pulso de reloj durante el cual el dispositivo receptor del byte debe generar un pulso de reconocimiento, conocido como ACK (del inglés Acknowledge).

Esto se logra situando la línea de datos a un nivel lógico bajo mientras transcurre el noveno pulso de reloj.

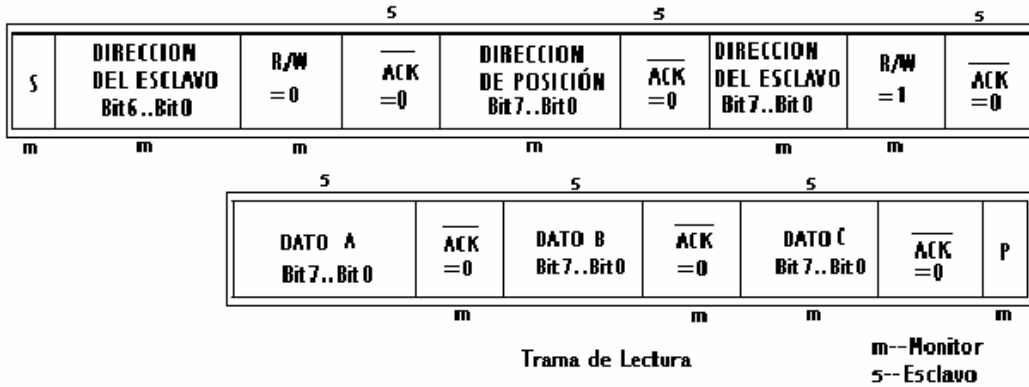


- Dirección (Address):** Cada dispositivo diseñado para funcionar en este bus dispone de su propia y única dirección de acceso, que viene pre-establecida por el fabricante. Hay dispositivos que permiten establecer externamente parte de la dirección de acceso. Esto permite que una serie del mismo tipo de dispositivos se puedan conectar en un mismo bus sin problemas de identificación. La dirección 00 es la denominada "de acceso general", por la cual responden todos los dispositivos conectados al bus.

S	DIRECCIÓN Bit 6..Bit 0	R/W	ACK	DATO A Bit 7..Bit 0	ACK	.....	DATO N Bit 7..Bit 0	ACK	P
---	---------------------------	-----	-----	------------------------	-----	-------	------------------------	-----	---

Trama de Transmisión

- Lectura/Escritura (Bit R/W):** Cada dispositivo dispone de una dirección de 7 bits. El octavo bit (el menos significativo ó LSB) enviado durante la operación de direccionamiento corresponde al bit que indica el tipo de operación a realizar. Si este bit es alto el dispositivo maestro lee información proveniente de un dispositivo esclavo. En cambio, si este bit fuese bajo el dispositivo maestro escribe información en un dispositivo esclavo.



## PROCOLO DEL BUS

Como es lógico, para iniciar una comunicación entre dispositivos conectados al bus I2C se debe respetar un protocolo. Tan pronto como el bus esté libre, un dispositivo maestro puede ocuparlo generando una condición de inicio. El primer byte transmitido después de la condición de inicio contiene los siete bits que componen la dirección del dispositivo de destino seleccionado y un octavo bit correspondiente a la operación deseada (lectura o escritura). Si el dispositivo cuya dirección se apuntó en los siete bits está presente en el bus éste responde enviando el pulso de reconocimiento ó ACK. Seguidamente puede comenzar el intercambio de información entre los dispositivos.

Cuando la señal R/W está previamente a nivel lógico bajo, el dispositivo maestro envía datos al dispositivo esclavo hasta que deja de recibir los pulsos de reconocimiento, o hasta que se hayan transmitido todos los datos.

En el caso contrario, es decir cuando la señal R/W estaba a nivel lógico alto, el dispositivo maestro genera pulsos de reloj durante los cuales el dispositivo esclavo puede enviar datos. Luego de cada byte recibido el dispositivo maestro (que en este momento está recibiendo datos) genera un pulso de reconocimiento.

El dispositivo maestro puede dejar libre el bus generando una condición de parada (Stop). Si se desea seguir transmitiendo, el dispositivo maestro puede generar otra condición de inicio el lugar de una condición de parada. Esta nueva condición de inicio se denomina "inicio repetitivo" y se puede emplear para direccionar un dispositivo

esclavo diferente ó para alterar el estado del bit de lectura/escritura (R/W).

## 8. MODOS DE FUNCIONAMIENTO DE 24LC512

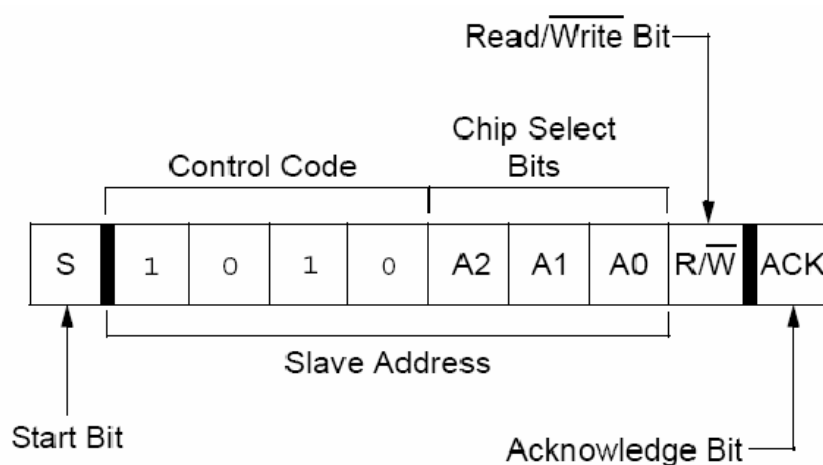
El 24LC512 puede operar de dos modos diferentes en cuanto a transmisión de datos se refiere. Un dispositivo que envía datos sobre el bus se define como transmisor mientras que un dispositivo que recibe datos a través del mismo se define como receptor. El 24LC512 puede funcionar como emisor o bien como receptor.

Típicamente el 24LC512 funcionará como parte de un sistema master-slave, donde el papel de maestro será desempeñado usualmente por un microprocesador y la memoria estudiada será el esclavo. El maestro controlará el bus, a través del cual memoria y microprocesador quedan comunicados, generando la señal de reloj (SCL), controlando el acceso al mismo y generando la señalización necesaria como las condiciones de comienzo y de parada.

### DIRECCIONAMIENTO DE DISPOSITIVOS

Como hemos visto en el apartado anterior, pueden existir varios dispositivos conectados a un mismo bus I2C que funcionarán de una manera correcta siempre que cumplan las especificaciones del citado protocolo y los formatos de trama establecidos. En el caso que nos ocupa, el de la memoria 24LC512, estas tramas toman una forma determinada que es expuesta a continuación.

En primer lugar, es generado y enviado el bit de Start por parte del master que es seguido por el primer byte de la comunicación, el denominado byte de control. En el caso de la memoria 24LC512 el byte de control presenta a siguiente estructura:

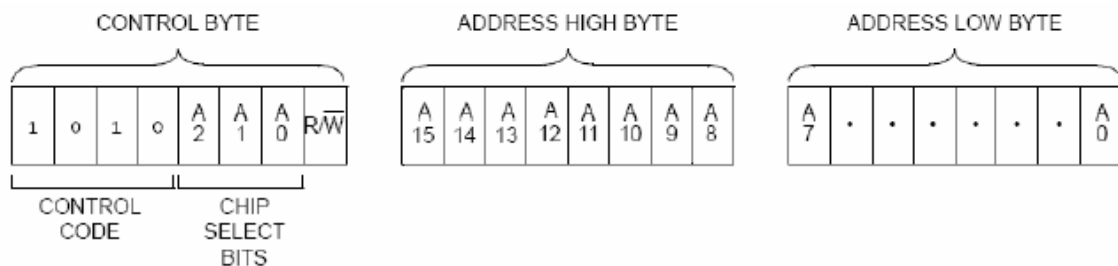


Los cuatro primeros bits conforman el código de control y siempre toman un valor fijo que en el caso del 24LC512 es '1010' tanto para las operaciones de lectura como de escritura. Los tres bits siguientes son los que permiten la selección de chip o dispositivo y se denominan A2, A1 y A0 respectivamente. Estos tres últimos bits pueden tomar cualquier valor por lo que son usados para seleccionar y acceder hasta 8 dispositivos 24LC512 conectados al mismo bus.

El bit siguiente dentro del byte de control anteriormente expuesto define el tipo de operación a realizar. Cuando este bit presenta un nivel lógico 1 se realiza una operación de lectura mientras que si se encuentra a nivel bajo se realizará una operación de escritura.

A continuación son enviados dos bytes que definen la dirección del dato a leer o escribir. Estos 16 bits permiten direccionar hasta 64K posiciones de memoria, lo que en el caso del 24LC512 supone 512 Kbits, pues se trata de una memoria 64K x 8. Esta cantidad de posible memoria a direccionar puede extenderse hasta 4 Mbits haciendo uso de los bits de Chip Select, que permite la conexión de hasta ocho dispositivos de este tipo al mismo bus. (En este caso 4Mbits=512Kbits x 8)

Después de la condición de comienzo, el dispositivo estudiado chequea el bus buscando la existencia del identificador apropiado. Si recibe un '1010' y una combinación de chip select adecuada el dispositivo esclavo genera una señal de reconocimiento o ACK. Dependiendo del valor del bit de lectura/escritura la operación a realizar será una u otra sobre la dirección de memoria determinada por los dos bytes siguientes:



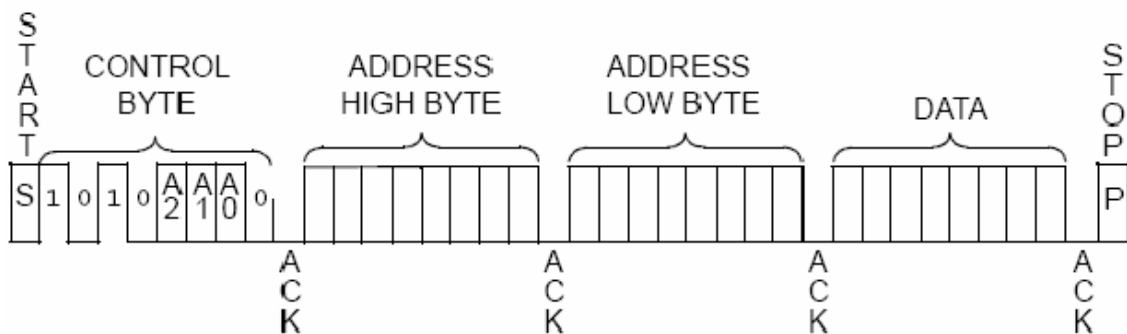
## **OPERACIONES DE ESCRITURA**

El circuito 24LC512 permite diferenciar dos tipos de operaciones de escritura: escritura de un byte y escritura en modo página que permite la escritura secuencial de 128 bytes. Tanto en un caso como en otro, el procedimiento inicial es idéntico y se adapta a las normas del protocolo I2C y al procedimiento anteriormente expuesto.

### Escritura de un Byte

Como hemos visto anteriormente el master es el que genera las señales de control pertinentes dentro del bus para que la comunicación pueda ser llevada a cabo sin ningún tipo de problema. En primer lugar se genera la condición de inicio de la comunicación a continuación de la cual figuran los bits de código de control (fijos a '1010') y los bits de chip select, que permiten elegir el dispositivo esclavo sobre el cual realizar la operación de lectura o escritura (escritura en este caso). Posteriormente es enviado el bit que marca el tipo de operación a realizar (por ser escritura estará en este caso a nivel bajo) y el reconocimiento enviado desde el dispositivo esclavo hacia el master.

Después de transferido el primer byte, se envía la dirección a la que se accederá dividida en dos bytes, después de los cuales será enviada una nueva señal de reconocimiento. Después del segundo ACK, aparecerá sobre el bus el dato, en este caso un byte, a escribir sobre la memoria. Una vez acabada la transferencia y escritura del mismo, la memoria genera un nuevo ACK que es respondido por el master con una condición de parada que pone fin al ciclo de escritura. En este momento la memoria inicia un ciclo de escritura interno, en el cual no genera ningún tipo de reconocimiento sobre ninguna señal. En el caso que el pin WP estuviera a nivel alto la operación de escritura no afectaría a los valores almacenados en la memoria, es decir, no se realizaría escritura alguna. El procedimiento de escritura se muestra en la siguiente figura:



### Velocidad efectiva de escritura de un byte

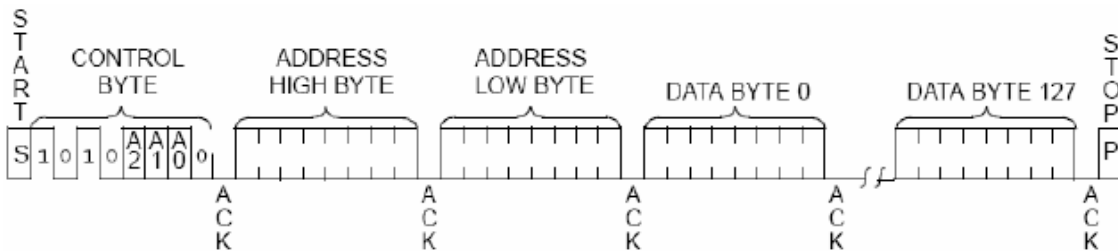
Observando el anterior diagrama y conocido el ciclo de funcionamiento de la memoria 24LC512 podemos calcular la velocidad efectiva de escritura de un byte. La existencia de una gran cantidad de información de control como ACK's, bytes de direccionamiento y de control hacen que la efectividad de la transmisión disminuya y por tanto que disminuya la velocidad efectiva de transmisión. Para calcularla únicamente tendremos que

realizar el cociente entre la información que queremos escribir, un byte u ocho bits, y el tiempo invertido en completar el proceso. Para calcular este tiempo contaremos el numero total de bits emitidos y lo multiplicaremos por el periodo de la señal de reloj que es de 2.5  $\mu$ s

$$Vel.Esc.Byte = \frac{8}{36 \cdot 2.5\mu} \approx 89kbps$$

**Escritura en modo Página**

La secuencia de inicio en este tipo de escritura es idéntica al caso de la escritura de un byte. Después de la condición de comienzo, se transmite el bit de control (con el valor '1010' en el código de control, el valor apropiado de chip select, el bit de lectura/escritura), el primer reconocimiento y la dirección a la que queremos acceder, a la que seguirá el primer byte a escribir en la memoria. Pero en este caso, en lugar de generarse una condición de parada una vez transferido el byte, el maestro transmite hasta 127 bytes adicionales que son almacenados en un buffer del circuito para posteriormente ser escritos en la memoria después de la generación de la condición de parada:



Si el master es capaz de transmitir más de 128 bytes antes de generar la condición de parada, el contador volverá al inicio de la página de escritura sobrescribiendo los datos anteriormente almacenados.

Al igual que en la escritura del byte, una vez recibida la condición de parada se inicia el ciclo interno de escritura durante el cual no se reconoce ningún tipo de señal. En el caso de tener el pin WP a nivel alto no se realizará ninguna operación de escritura sobre el circuito aunque serán reconocidos los intentos de escribir sobre el dispositivo.

**Velocidad efectiva de escritura en modo página**

Para calcular esta velocidad tan sólo hemos de limitarnos a proceder de modo similar al anterior teniendo en cuenta el nuevo procedimiento de escritura. En este caso podemos observar que la

proporción de información de control respecto de la información útil emitida es bastante menor lo que aumentará la efectividad de la comunicación y por lo tanto la velocidad de la misma. Suponiendo que llenamos la página de escritura y con la misma frecuencia en el reloj del dispositivo tendríamos la siguiente velocidad:

$$Vel.Esc.Pagina = \frac{128 \cdot 8}{131 \cdot 8 \cdot 2.5\mu} \approx 390kbps$$

Como podemos observar, obtenemos un valor superior a la velocidad máxima que puede soportar el bus I2C, que según fue documentada en apartados anteriores es de 100kbps. Esta velocidad calculada sólo podría obtenerse en el caso que la velocidad máxima soportada por el bus fuera mayor que este último valor calculado.

## **OPERACIONES DE LECTURA**

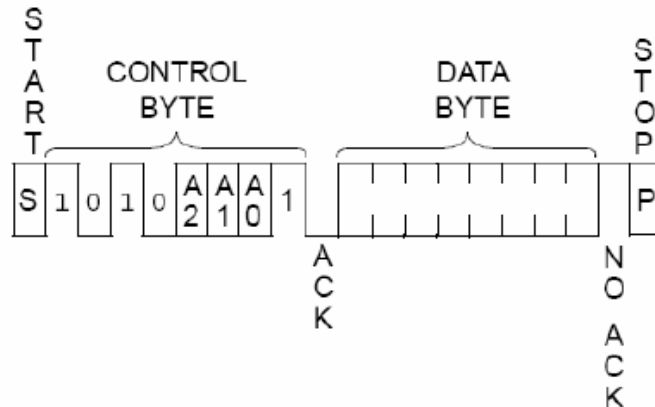
Las operaciones de lectura siguen el mismo patrón que las anteriores con la única diferencia del bit que marca en este caso, estando a nivel alto, que se trata de una operación de lectura. Hay tres tipos de operaciones de lectura: lectura de la dirección actual, lectura aleatoria y lectura secuencial.

### ***Lectura dirección actual***

El 24LC512 dispone de un registro interno que almacena la dirección de la última dirección de memoria accedida. Éste puede incrementarse en una unidad de modo que si la última operación de lectura se realizó sobre la dirección  $n$ , la próxima se realizará sobre la dirección  $n+1$ .

Una vez recibido el bit a nivel alto correspondiente a la indicación de operación de lectura, el 24LC512 envía un ACK al master y envía el byte correspondiente a la dirección apuntada por el registro indicado anteriormente. El master no generará reconocimiento alguno, lanzando la condición de parada como se indica en la siguiente figura:





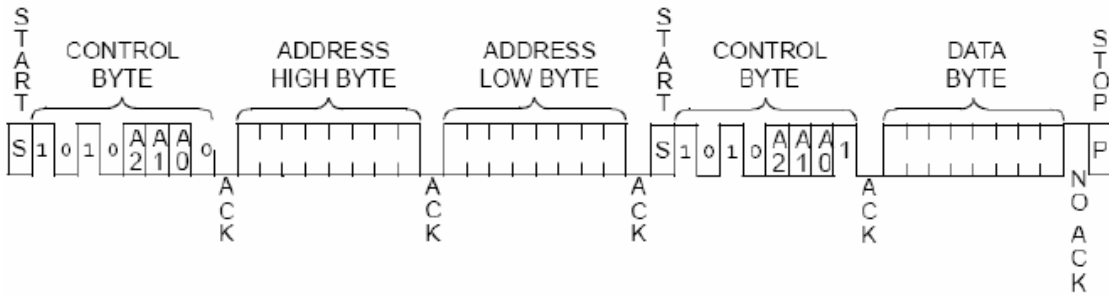
Conociendo la temporización del bus I2C y partiendo del diagrama anterior podemos calcular la velocidad a la que transmitirá la memoria al dispositivo que haya solicitado una lectura de la información que almacena.

$$Vel.Lect.Byte = \frac{8}{18 \cdot 2.5\mu} \approx 178kbps$$

Al igual que en el caso anterior, obtenemos un valor superior a la velocidad máxima que puede soportar el bus I2C. Esta velocidad calculada sólo podría obtenerse en el caso que la velocidad máxima soportada por el bus fuera mayor que este último valor calculado.

### **Lectura Aleatoria**

Las operaciones de lectura aleatoria permiten realizar lecturas sobre las posiciones de memoria que elijamos o que vengan determinadas por el master. Para realizar este tipo de operaciones deberemos indicar al 24LC512, es decir, el esclavo, la dirección de la posición de memoria sobre la que deseamos hacer la lectura. Para ello iniciaremos el ciclo como si se tratara de una escritura. Se envía el byte de control con sus correspondientes valores comentados en apartados anteriores junto con la dirección de memoria a la que se desea acceder en forma de dos bytes separados y con sus correspondientes reconocimientos emitidos por la memoria. Una vez recibido el segundo reconocimiento el master emite una nueva condición de inicio poniendo fin al anterior ciclo escritura e iniciando el de lectura propiamente dicho con el mismo byte de control que en el caso anterior pero cambiando el valor de ese último bit que indica que se trata de una operación de lectura por lo que está a nivel alto. Una vez recibido el 24LC512 emite un ACK y se transmite el byte de datos que no es reconocido por el master, pero al cual responde generando la condición de parada de la comunicación. Se puede apreciar en la siguiente figura:

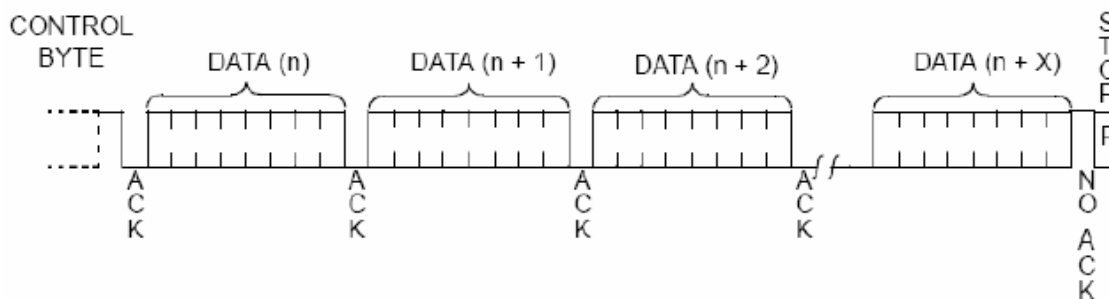


Al igual que en el caso anterior podemos calcular la velocidad de lectura en modo secuencial. En este caso debido a la gran cantidad de información de control adicional obtendremos un resultado significativamente menor:

$$Vel.Lect.Aleat. = \frac{8}{45 \cdot 2.5\mu} \approx 71kbps$$

**Lectura Secuencial**

En este caso la operación se inicia de manera idéntica al caso anterior: se genera la condición de inicio, se transmite el byte de control y la dirección de memoria deseada, para de nuevo enviar el nuevo byte de control que una vez reconocido permite la lectura de la posición de memoria indicada con anterioridad. En este momento se genera la diferencia respecto al caso anterior, pues mientras para el caso de una lectura aleatoria, una vez leído el byte de datos el master generaba la condición de parada, en el caso de una lectura secuencial el master genera un reconocimiento lo que permite el acceso y lectura sucesiva de las siguientes posiciones de memoria que van siendo reconocidas por el master hasta que genera la condición de parada y la lectura secuencial finaliza. Esta operación se realiza gracias a la existencia de un registro interno que se autoincrementa tras la lectura de cada posición de memoria y que permite barrer toda la memoria desde principio a fin, desde la posición 0000 a la FFFF. La siguiente figura muestra el proceso:



Partiendo del anterior diagrama podemos hallar la velocidad de lectura en modo secuencial. En este caso, la velocidad que deseamos conocer dependerá del número de bytes a leer, es por tanto que recurriremos a una expresión general en función de n.

$$Vel.Lect.Sec = \frac{8n}{9(n+1) \cdot 2.5\mu}$$

En el caso de un valor de n significativamente grande obtenemos el siguiente valor de velocidad:

$$Vel.Lect.Sec = \frac{8}{9 \cdot 2.5\mu} \approx 360kbps$$

Al igual que en casos anteriores, obtenemos un valor superior a la velocidad máxima que puede soportar el bus I2C. Esta velocidad calculada sólo podría obtenerse en el caso que la velocidad máxima soportada por el bus fuera mayor que este último valor calculado.

## 9. EJEMPLO DE UTILIZACIÓN DE 24LC512: ESCRITURA DE UN BYTE Y LECTURA DE UN BYTE

### **Suposiciones**

- Tenemos un microcontrolador como MASTER que dirigirá la comunicación.
- Varios dispositivos conectados al mismo bus I2C, entre los cuales tendremos varias memorias (como mucho 8)
- La escritura y la lectura serán sobre una memoria 24LC512 que actuará como ESCLAVO en la comunicación.
- En cuanto a las funciones, suponemos que el reloj se inicializa desde el programa principal, antes de llamar a las funciones, a una frecuencia de 400 KHz, y suponemos también que la variable SDA implementa la línea de datos, y será una variable de 8 bits, definida como variable global fuera de las funciones; lo que valga esta variable será lo que envíe el MASTER a la memoria, o lo que lea de ella cuando funcione como receptor.

### **Función de Escritura de un Byte**

La función tendrá las siguientes características:

- El programa principal llamará a la función cuando quiera escribir un byte, pasándole como parámetros la dirección de la memoria en la que quiere escribir (bits A2 A1 y A0), la dirección en donde quiere escribir dentro de la memoria (2 bytes: ADDRESSH -> los 8 bits más significativos de la dirección y, ADDRESSL -> los 8 bits menos significativos) y por

último, el dato que quiere escribir (será una variable tipo entero, de 8 bits).

- La función devuelve un entero: '0' si no ha podido escribir y '1' si la escritura ha tenido éxito.
- Suponemos que el usuario controla el bit de protección de escritura(WP), y está a '0', permitiendo la escritura.

### **Función "byte\_write":**

```
int byte_write (A2,A1,A0,ADDRESSH,ADDRESSL,DATO)

main()
{
    Start_Condition(); /*función que genera la condición
                        de inicio*/

    SDA=0b 1010 A2A1A0 0; /*MASTER envía el byte de
                           control*/

    if "no detecta ACK"
    {
        return 0; /*la memoria no responde y acaba la
                  función*/
    }
    else /*sí que detecta ACK*/
    {
        SDA=ADDRESSH;

        /*MASTER espera a recibir ACK de la memoria*/
        SDA=ADDRESSH;

        /*MASTER espera a recibir ACK de la memoria*/
        SDA=ADDRESSL;

        /*MASTER espera a recibir ACK de la memoria*/
        SDA=DATO;

        /*MASTER espera a recibir ACK de la memoria*/

        Stop_Condition(); /*función que genera la
                           condición de parada*/
    }
    return 1;
}
}
```

Explicación: siempre que el MASTER envía un byte, tiene que recibir el acuse de reconocimiento (ACK), ya que si no, no puede seguir enviando bytes; en el caso de que no lo recibiera, la función acabaría,

devolvería un 0, y sería el programa principal el que debiera volver a intentarlo llamando de nuevo a la función.

Seguramente falten instrucciones, pero la estructura básica del programa sería ésta.

Por último, comentar que los bits A2 A1 y A0 podrían ser las componentes de un vector, y así pasaríamos a la función únicamente una variable vector. Lo mismo sucede con la dirección, que podría ser otro vector de 16 componentes, de las cuales las ocho primeras conformarían lo que hemos llamado ADDRESSH, y las siguientes 8 la variable ADDRESSL.

### **Función de Lectura de un Byte**

- La lectura será aleatoria, es decir, el micro quiere leer una dirección en concreto que no tiene por qué ser la dirección a la que apunte el registro de direcciones interno de la memoria.
- No habrá que fijarse en el bit de protección porque la operación de lectura siempre está permitida.

La función tendrá las siguientes características:

- Al igual que en la función de escritura, el programa principal llamará a la función cuando quiera leer un byte, pasándole como parámetros la dirección del dispositivo (memoria) al que se quiere acceder, la dirección que se quiere leer dentro de la memoria, y por último, la variable dato, que será un puntero apuntando a una variable de tipo entero (8 bits) que se modificará dentro de la función y pasará por referencia al programa principal(es decir, será una variable de entrada/salida)
- Devuelve un '0' si no se puede leer, y un '1' si la lectura ha sido satisfactoria.

### **Función "byte\_read":**

```
int byte_read (A2,A1,A0,ADDRESSH,ADDRESSL,*DATO)
main()
{
  Start_Condition();
  SDA=0b1010 A2A1A0 0; /*comando de escritura*/
  if "no recibe ACK"
  {
    return 0;
  }
  else
```

```
{
SDA=ADDRESSH;

/*MASTER espera a recibir ACK de la memoria*/

SDA=ADDRESSL;

/*MASTER espera a recibir ACK de la memoria*/

    /*hasta aquí, se ha actualizado la
    dirección del puntero, pero no se
    escribe nada, ya que el MASTER no envía
    ningún dato. Aquí acaba la operación de
    escritura*/

Start_Condition();
SDA=0b1010 A2A1A0 1; /*comando de lectura*/

if "no recibe ACK"
{
    return 0; /*la función acaba y vuelve al programa
    principal*/
}
else
{
    *DATO=SDA; /*copio los 8 bits del dato, que ha
    enviado la memoria, ya que es lo que
    hay en la línea después del ACK*/
    delay_ms(0.0025); /*el MASTER no envía ACK
    después del dato recibido,
    esperándose un
    ciclo de reloj (=2.5
    µs=0.0025 ms)*/

    Stop_Condition;
    return 1;
}
} }
```

**Explicación:** como podemos observar, el MASTER solicita primero una escritura para actualizar el registro interno de direcciones y que se quede apuntando a la dirección que quiero leer, sin enviar ningún dato para escribir; después de enviarle la dirección y recibir el ACK correspondiente, el MASTER genera una nueva condición de inicio indicando a la memoria que ha acabado la operación de escritura, y es ahora cuando le indica a la memoria que quiere leer, poniendo el bit R/W a 1 en el nuevo byte de control que le envía. A partir de aquí es como una lectura corriente, de la dirección actual; el MASTER le indica a la memoria que ya ha acabado

de leer cuando tras el último byte recibido no envía ACK y genera la condición de parada.

## 10. BIBLIOGRAFÍA

- Datasheet 24LC512 de Microchip
- Datasheet PIC 16F877 de Microchip
- [www.comunidadelectronicos.com/articulos/i2c.htm](http://www.comunidadelectronicos.com/articulos/i2c.htm)
- [www.totalphase.com/support/downloads/](http://www.totalphase.com/support/downloads/)