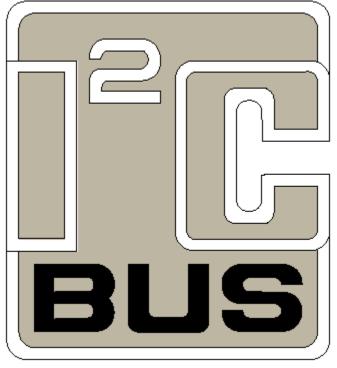


Práctica 3 Introducción al uso del Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar



Actualizado: 27/10/2023



Competencias

☐ Capacidad involucrad	-				-		y	hardware
☐ Capacidad acceso a un	-	· ·				esario para propósito g		
☐ Capacidad dispositivo práctico re	perifér					-		cante de un problema
☐ Capacidad interdiscip	-	prender y	aplic	car nuevo	s concepto	s de form	a a	utónoma e
☐ Capacidad	l para en	nplear la ci	reativ	idad en la	resolución	de los pro	bleı	mas.
☐ Capacidad	l para tr	abajar en e	equipo	o y colabo	rar eficazn	nente con o	tras	s personas.

Objetivos... Conexión de un módulo I2C usando un puerto de entrada/salida paralelo de propósito general



Conocer y usar el bus I2C y su protocolo de comunicación básico.
Conocer y entender los aspectos básicos de funcionamiento de los periféricos I2C y, concretamente, la memoria 24LC64/M24C01/M24C02 a partir de la hoja de características proporcionada por el fabricante.
Realizar rutinas sencillas que permitan operaciones de escritura y lectura en cualquier dirección de memoria del dispositivo y en los distintos modos soportados.
Verificar el correcto funcionamiento del hardware y del software de comprobación desarrollado. Observación de las señales, si fuese necesario.
Planificar el desarrollo software completo de la segunda parte de la práctica. Expresarlo en un organigrama general que refleje la secuencia de acciones necesarias a realizar.



Introducción al Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar

Brevemente ... ¿Qué hay que hacer? Realización de la práctica (Básica):



☐ Programa para hacer un test completo de una memoria I2C 24LC64:

- Guardar un dato (de 0 a 255) en cualquier dirección de memoria del dispositivo 24LC64. Tanto el dato como la dirección se han de solicitar al usuario. (pidiéndolos por teclado/pantalla)
- 2. Leer una posición (de 0 a 8191) del 24LC64 (pidiendo la posición por pantalla)
- Inicializar un bloque de 256 bytes contiguos de la memoria 24LC64 a un valor (la dirección del primer elemento y el valor a escribir se solicitan por pantalla; medir el tiempo el tiempo empleado en acceso a la memoria)
- Mostrar el contenido de un bloque de 256 bytes contiguos del 24LC64, comenzando en una dirección especificada (en una matriz de 8 0 16 columnas, en hexadecimal; medir el tiempo que consume el acceso a la memoria)
- Inicializar <u>usando "Page Write"</u> un bloque de 256 bytes contiguos del 24LC64 a un valor (la dirección del primer elemento y el valor a escribir se solicitan por pantalla; medir el tiempo que consume el acceso a la memoria)
- Mostrar el contenido de un bloque de 256 bytes del 24LC64 (usando <u>Sequential</u>
 <u>Read</u>), comenzando en una dirección especificada (en una matriz de 8 o 16
 columnas, en hexadecimal; medir el tiempo que consume el acceso a la memoria)



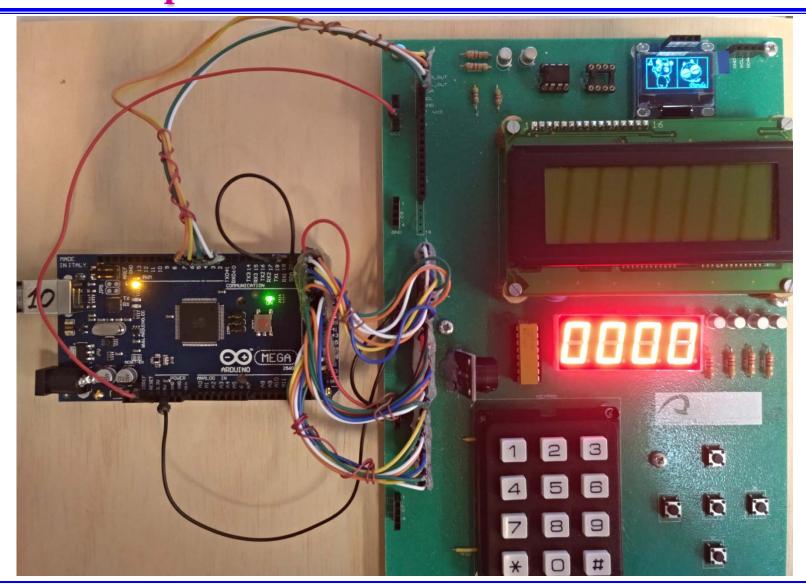
Introducción al Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar

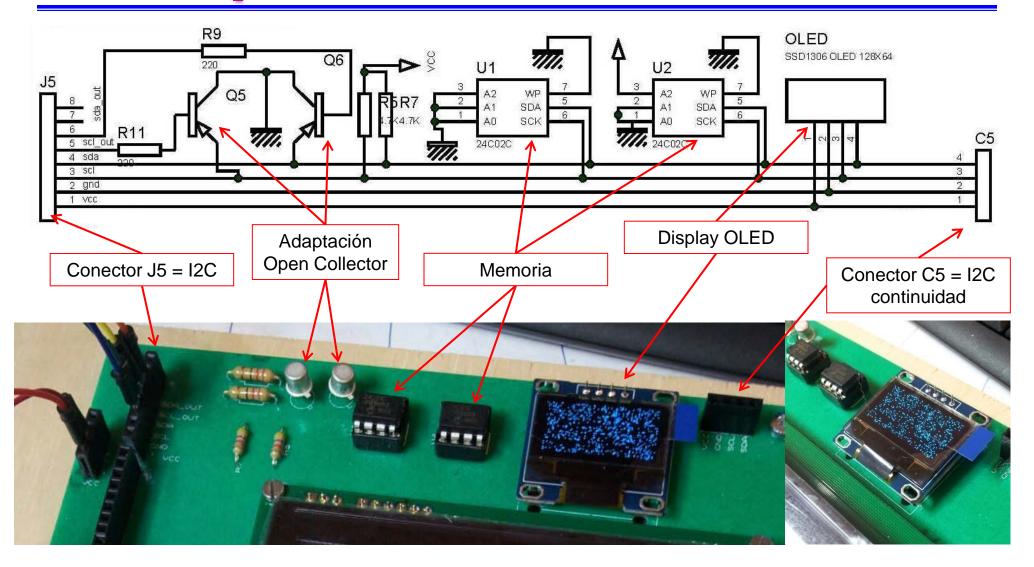
Entorno Experimental de prácticas de laboratorio: Elementos Disponibles





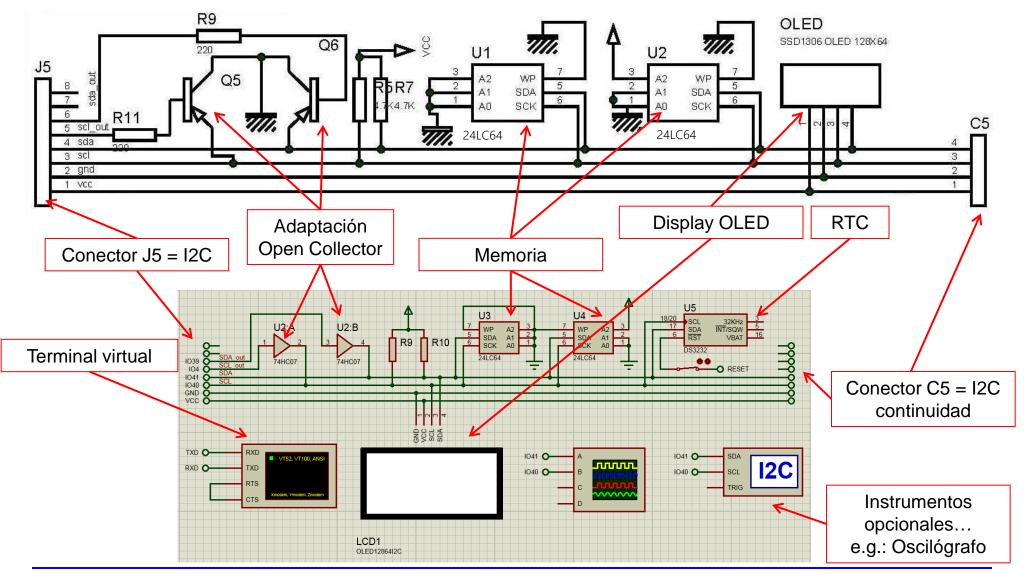


Entorno Experimental: Zona I2C





Entorno Experimental en simulador: Zona I2C





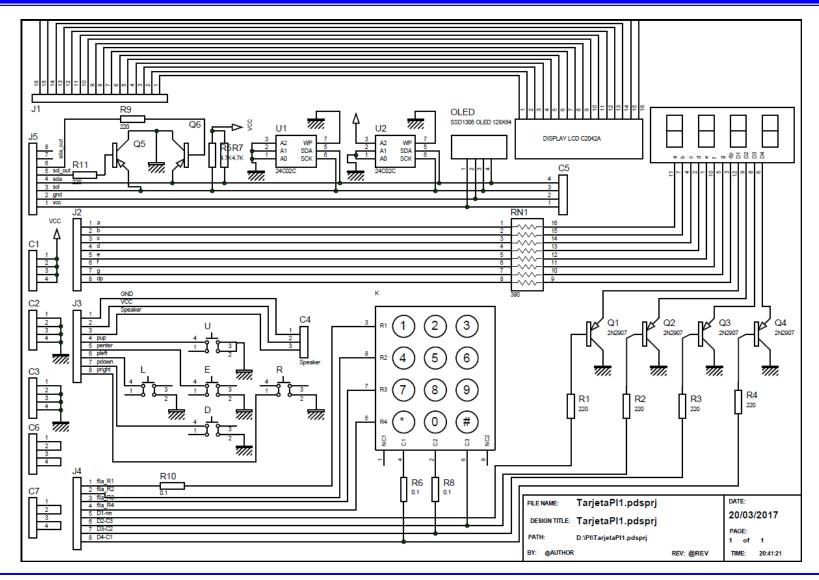
Introducción al Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar

Diseño del Entorno Experimental de prácticas: Esquema del conexionado de los componentes





Preguntas previas....



Utilización del Arduino...

- ☐ El arduino tiene un controlador propio de I2C gestionado por una librería...
 - Si solo usamos esto... ¿qué aprendemos?
 - Nada sobre el protocolo I2C
 - Solo a usar una librería para comunicar con un dispositivo (I2C)
 - Si usamos terminales genéricos e implementamos el protocolo I2C por software...; qué aprendemos?
 - El funcionamiento del protocolo I2C
 - A comunicar con un dispositivo I2C en cualquier entorno
- ☐ ¿Se pueden utilizar terminales genéricos para implementar un I2C? ¿Cómo?
 - El Arduino tiene terminales genéricos, pero
 - No hay garantías de que sea "open-collector"
 - Para asegurarnos, siempre podremos convertir un terminal de salida en open-collector
 - Con un buffer integrado ...
 - Con transistores
 - Etc.

+ Preguntas previas....



Utilización del Arduino...

- ☐ Para usar con la máxima versatilidad y seguridad el Arduino lo más conveniente es:
 - Usar dos líneas de salida genéricas (<u>pin 4 y 39</u>), convirtiéndolas a "opencollector" mediante un buffer adecuado
 - En la tarjeta experimental real de laboratorio utilizamos dos transistores PNP
 - En la tarjeta experimental simulada en Proteus utilizamos dos buffers open-collector
 - Permite cumplir el estándar I2C
 - Eliminamos el riesgo de daños por conflictos de niveles.
 - Usar dos líneas de entrada genéricas (pin 40 y 41), para poder
 - Leer y monitorizar en todo momento la línea SDA
 - Monitorizar en todo momento la línea SCL
 - Es muy importante **programar como "INPUT" los pines 40 y 41** ya en el Setup, sin activar las resistencias de Pull Up

+ Preguntas previas....



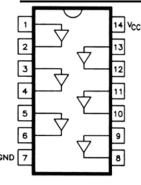
Utilización del Arduino...

□ El 7407:

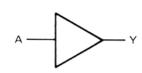
- Es un integrado con seis buffers que poseen salida "open-collector"
- Permite adaptar dos líneas de salida genéricas (pin 4 y 5), convirtiéndolas a "open-collector"
 - En la práctica utilizaremos dos de los hex-buffer estándar contenidos en el TTL 7407
 - Esto permite cumplir el estándar I2C
 - Se ha usado en el circuito para el simulador!



7407 Buffers/line driver OC HV



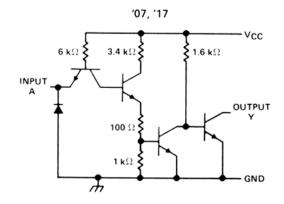
logic diagram (each gate)



positive logic (each gate)

Y = A

schematic

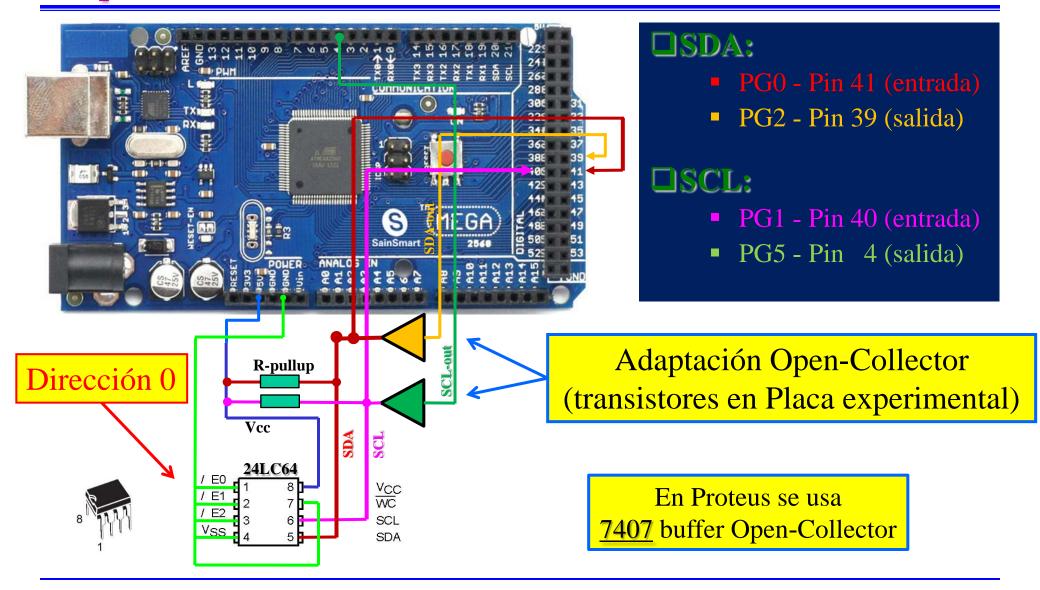


Resistor values shown are nominal not for SN74LS07

(Curso actual)

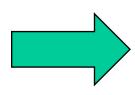
UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

Esquema de conexiones en el Arduino... (v.2021)





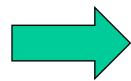
Ejemplo de inicialización del Arduino... (V.2021)



#define LEE_SCL 40 // puerto de entrada para leer el estado de la línea SCL #define LEE_SDA 41 // puerto de entrada para leer el estado de la línea SDA #define ESC_SCL 4 // puerto de salida para escribir el valor de la línea SCL-out #define ESC_SDA 39 // puerto de salida para escribir el valor de la línea SDA-out

void setup() {

// Inicialización del canal serie para comunicarse con el usuario Serial.begin(9600);



```
// Inicialización de los terminales de entrada
pinMode(LEE_SDA, INPUT);
pinMode(LEE_SCL, INPUT);
// Inicialización de los terminales de salida
pinMode(ESC_SDA, OUTPUT);
pinMode(ESC_SCL, OUTPUT);
// Asegurarse de no intervenir el bus poniendo SDA y SCL a "1"....
digitalWrite(ESC_SDA, HIGH);
digitalWrite(ESC_SCL, HIGH);
}
```



Introducción al Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar

El dispositivo =>

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

M24C01: 1kbit = 128 Bytes; 24LC64: 64kbit = 8 KB

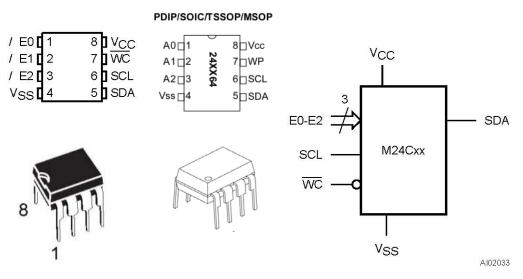


Table 1. Signal Names

E0, E1, E2	Chip Enable
SDA	Serial Data
SCL	Serial Clock
WC	Write Control
Vcc	Supply Voltage
V _{SS}	Ground

Table 3. Operating Modes

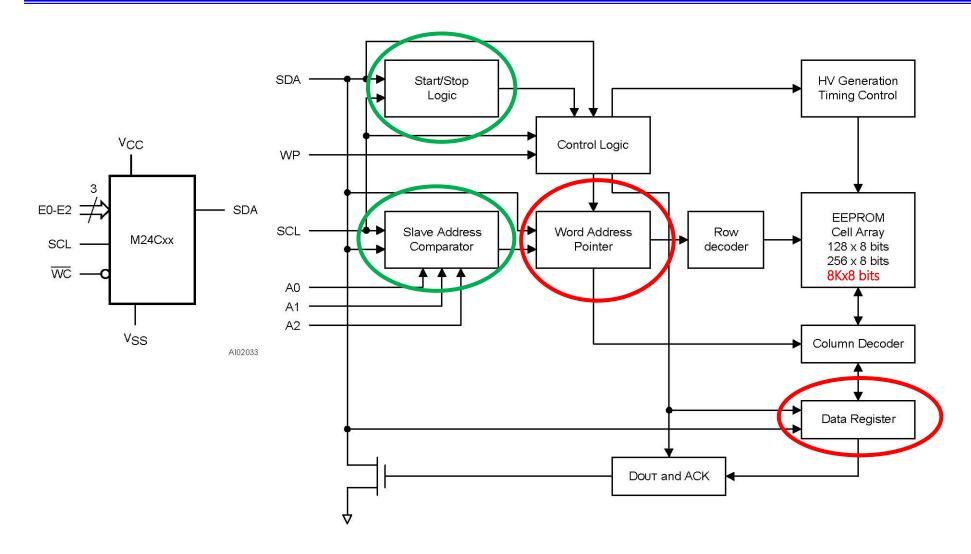
Mode	R₩ bit	WC ¹	Bytes	Initial Sequence
Current Address Read	1	Х	1	START, Device Select, RW = 1
Random Address Read	0	Х	s Ä	START, Device Select, RW = 0, Address
Random Address Read	1	Х		reSTART, Device Select, RW = 1
Sequential Read	1	Х	≥ 1	Similar to Current or Random Address Read
Byte Write	0	V _{IL}	1	START, Device Select, RW = 0
Page Write	0	V _{IL}	≤ 16	START, Device Select, RW = 0

Note: 1. $X = V_{\parallel}$ or V_{\parallel} .

Estudio del dispositivo =>

UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA

M24C01: 1kbit = 128 Bytes; <u>24LC64</u>: 64kbit = 8 KB





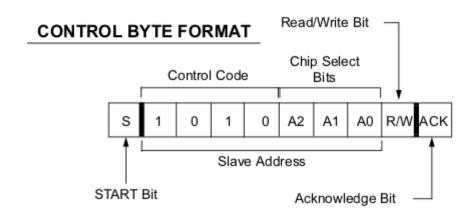
Introducción al Bus I2C

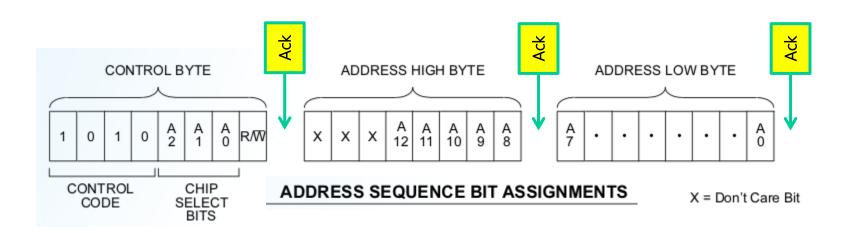
☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar



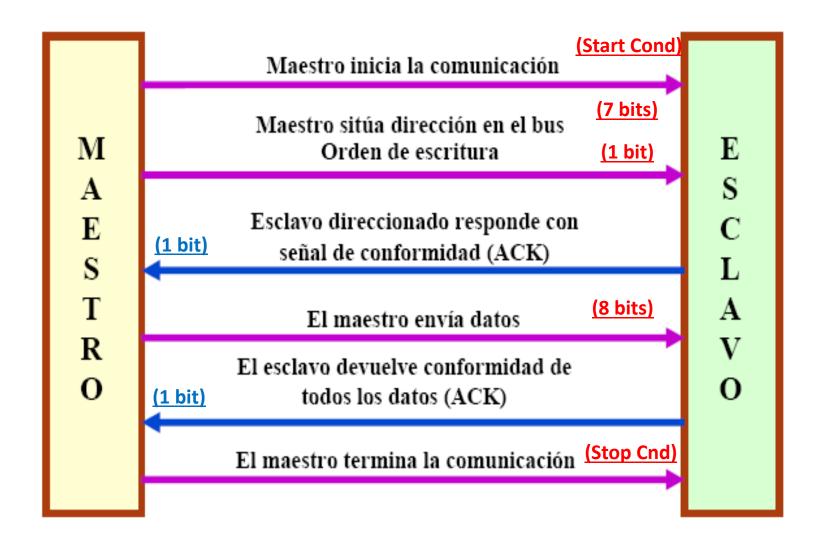
Formato de los mensajes en el 24LC64:







Escritura: repaso del procedimiento general

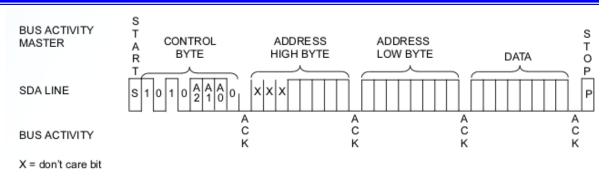


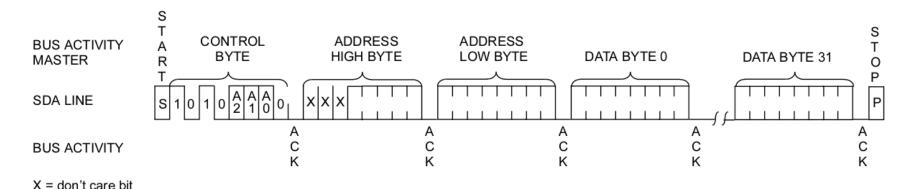


Operaciones de escritura en el 24LC64:

□ Dos modos de escritura:

Byte-Write = almacena <u>solo un byte</u> (data) en una posición (address)





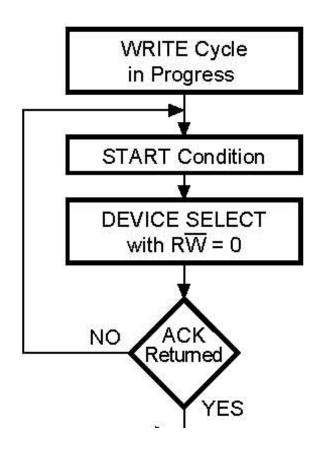
Page-Write = Almacena una secuencia de bytes a partir de una dirección (address H+L)
 Máximo= 32 bytes!!!

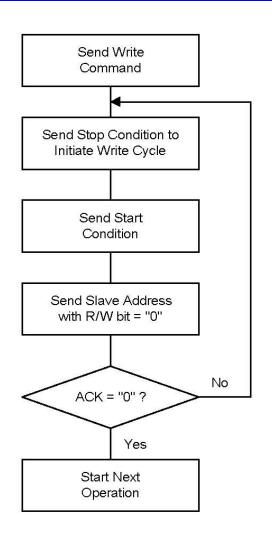
Escritura muy lenta: → ¡5 milisegundos! → Hay que esperar a que esté disponible Se señala con el ACK / NO-ACK después del "Control byte"



Algoritmo de espera: ¡implementación obligatoria!

Prohibido sustituirlo por "Delay"





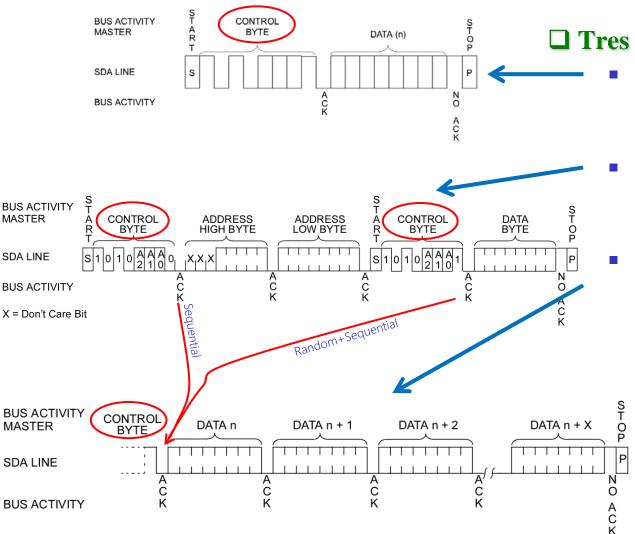


Lectura: Repaso del procedimiento general





Operaciones de lectura en el 24LC64:



- ☐ Tres modos de lectura:
 - <u>Current-Addr.-Read</u> = accedo a la posición a la que apunta el contador interno
 - Random-Addr.-Read = accedo solo a un byte (data-in) en la posición (address H/L)
 - Sequential-Addr.-Read
 - Random → Lectura de una secuencia de bytes a partir de una dirección (address H+L)
 - Current → Lectura de una secuencia de bytes <u>a partir de</u> <u>la dirección del contador</u> interno
 - Si llegamos a la dirección máxima y seguimos leyendo se continua desde la posición cero



Introducción al Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar

Desde el punto de vista de la programación....



(Ejemplo de implementación "naive")

☐ Tenemos que generar <u>solo unas pocas acciones distintas</u> → "<u>5 en total</u>"

Tipo	s de acciones Elementa	iles
Actividad del Maestro	<u>Acrónimo</u>	<u>Cometario</u>
Condición de START	<u>START</u>	
Condición de STOP	<u>STOP</u>	
Envío un bit a "1"	E-BIT-1	Es igual a <u>Envío "NO-ACK"</u>
Envío un bit a "0"	E-BIT-0	Es igual a <u>Envío "ACK"</u>
Envío el "ACK"	<u>E-ACK</u> (= E-BIT-0)	Es igual a <u>Envío bit a "0"</u>
Envío el "NO-ACK"	E-NO ACK (= E-BIT-1)	Es igual a <u>Envío bit a "1"</u>
Recibo el "ACK"	<u>R-ACK</u> (= R-BIT)	Es igual a <u>Recibo un bit</u>
Recibo un bit	<u>R-BIT</u>	Es igual a Recibo "ACK"

Condición de "Start" y "Stop"



(Ejemplo de implementación "naive")

	S	TART		
	Sed	c. temp	oral/ti	ро
Señal	1	2	3	4
	Esc	Lec	Esc	Esc
SDA	1	?1خ	0	0
SCL	1	?1خ	1	0

	9	ТОР		
	Sed	c. temp	oral/ti	ро
Señal	1	2	3	4
	Esc	Esc	Esc	Esc
SDA	0	0	1	1
SCL	0	1	1	1

SDA	1	<u>¿1?</u>	0	0
SCL	1	<u>ئا؟</u>	1	0

Condición de "Stop": Ejemplo de código



void i2c_stop() { // Ejemplo didáctico para empezar ...

// Este algoritmo no es optimo → es muy lento; Pero funciona!

```
digitalWrite(ESC_SCL, LOW); digitalWrite(ESC_SDA, LOW); digitalWrite(ESC_SCL, HIGH); digitalWrite(ESC_SDA, LOW); digitalWrite(ESC_SCL, HIGH); digitalWrite(ESC_SDA, HIGH); // Flanco de subida → Esto causa el stop digitalWrite(ESC_SCL, HIGH); digitalWrite(ESC_SDA, HIGH); // Si se quita sigue funcionando!
```

}

	5	TOP		
	Sed	c. temp	oral/ti	ро
Señal	1	2	3	4
	Esc	Esc	Esc	Esc
SDA	0	0	1	1
SCL	0	1	1	1

Envío de un bit a "0" y de un bit a "1"



(Ejemplo de implementación "naive")

	E-BIT-	0 = E-	ACK	
	Sed	c. temp	oral/ti	ро
Señal	1	2	3	4
	Esc	Esc	Esc	Esc
SDA	0	0	0	0
SCL	0	1	1	0

	E-	BIT-1		
	Sed	c. temp	oral/tip	ро
Señal	1	2	3	4
	Esc	Esc	Esc	Esc
SDA	1	1	1	1
SCL	0	1	1	0

|--|

Lectura de bit



(Ejemplo de implementación "naive")

R-BIT = R-ACK						
Señal	Sec. temporal/tipo					
	1	2	3	4		
	Esc	Esc	Lec	Esc		
SDA	1	1	?*؟	1		
SCL	0	1	?1خ	0		
	* = Valor leído 0,1					

SDA	1	1	٤*؟	1
SCL	0	1	1?	0

Ejemplo de secuencia: 24LC64

☐ Escribir un valor (0x55) en una posición (0x00)

Programar y visualizar en el oscilógrafo

Ejemplo 1: visualizar <u>el envio de START y los</u>
2 primeros bits. (usando digitalWrite y digitalRead) →

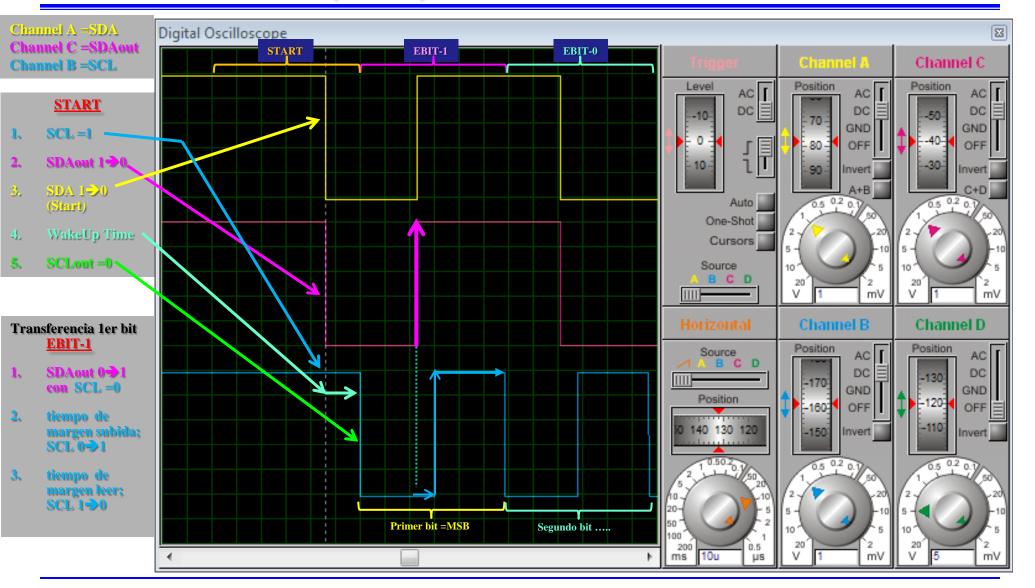
- •¿Cuanto tarda en enviar un bit? ¿cúal es la **frecuencia de transmisión en Kbits/seg**?
- •¿Respecto al estándar ¿es más, igual o menos rápido que alguno de ellos?
- •¿Cómo puede mejorarse el algoritmo "naive?
- •¿Cúal es mayor Kbits/seg que se puede alcanzar con digitalWrite y digitalRead?

"BYTE WRITE" Secuencia Acción START E-BIT-1 E-BIT-0 Los 7 bits de dirección del dispositivo E-BIT-1 1010 000 E-BIT-0 E-BIT-0 E-BIT-0 E-BIT-0 R/W = 0 (W)R-ACK Leer ack del dispositivo; si ok, sigo E-BIT-0 E-BIT-0 Pongo la dirección del byte Hight = E-BIT-0 XXX0 0000 ==> 0000 0000 E-BIT-0 Envío los ocho ceros (esto pondrá la parte E-BIT-0 Hight del contador interno a 0000 0000) E-BIT-0 E-BIT-0 E-BIT-0 R-ACK Leer ack del dispositivo; si ok, sigo E-BIT-0 E-BIT-0 Pongo la dirección del byte Low= E-BIT-0 0000 0000 E-BIT-0 Envío los ocho ceros (esto pondrá la parte E-BIT-0 Low del contador interno a 0000 0000) E-BIT-0 E-BIT-0 E-BIT-0 R-ACK Leer ack del dispositivo; si ok, sigo E-BIT-0 E-BIT-1 E-BIT-0 E-BIT-1 Envío el dato, por ej.: 0101 0101 = 0x55 E-BIT-0 E-BIT-1 E-BIT-0 E-BIT-1 Leer ack del dispositivo; si ok, sigo R-ACK STOP ==> Se produce el almacenamiento (5 msg)

Address

Ejemplos reales de visualización en MC24C01: Start + 1er bit=1 (MSB) + 2do bit=0





¿Cuánto tarda en enviarse cada bit

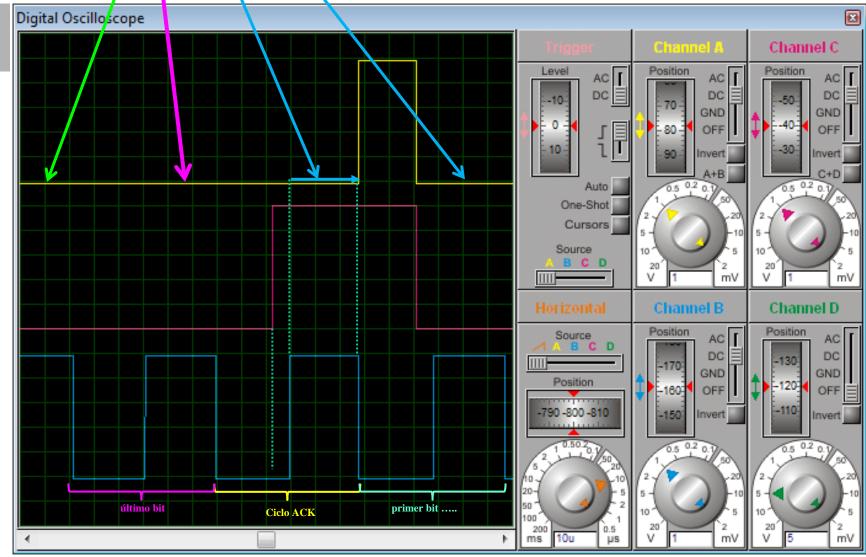


Otros ejemplos

Ejemplos reales de visualización en MC24C01: Escritura: ...0, 0, ACK, 0,



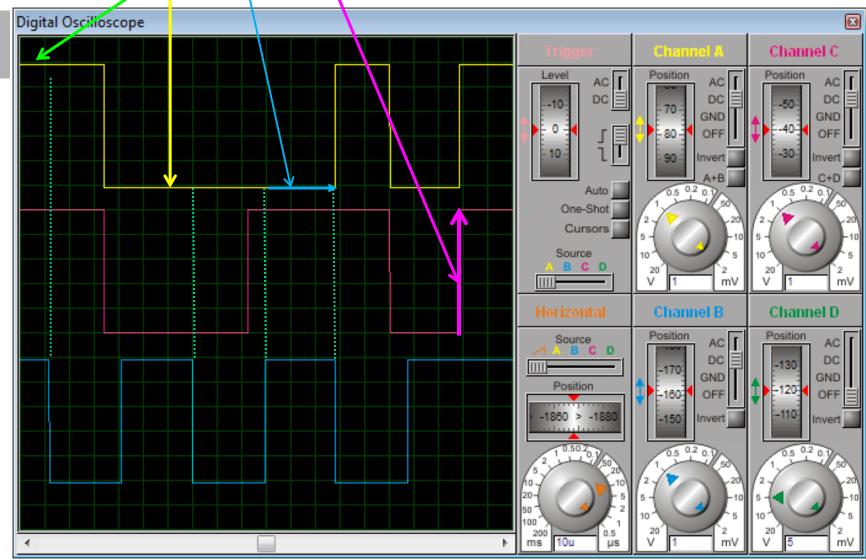
Channel A = SDA
Channel C = SDAout
Channel B = SCL



Ejemplos reales de visualización en MC24C01: Escritura: ...1, 0, ACK, Stop,



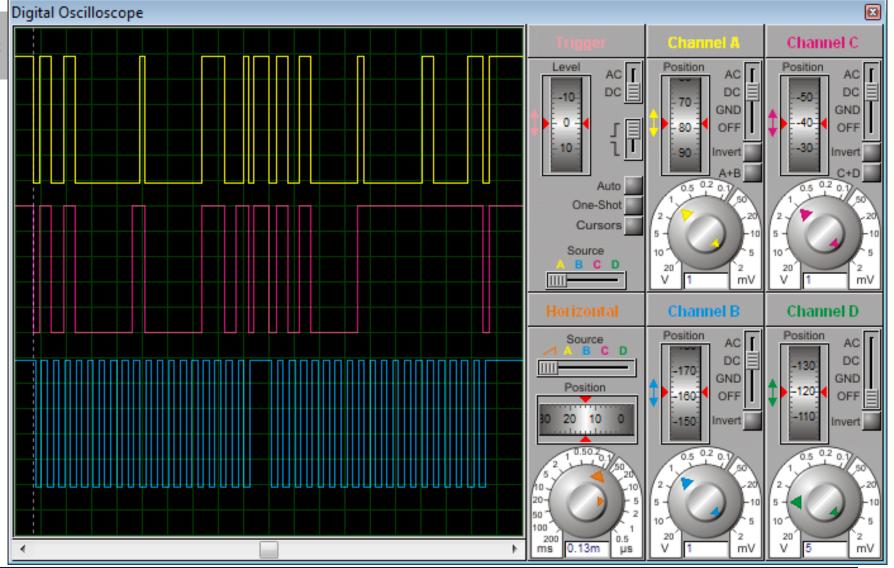
Channel A = SDA
Channel C = SDAout
Channel B = SCL



Ejemplos reales de visualización en MC24C01: Lee la posición 6 que contiene un 8



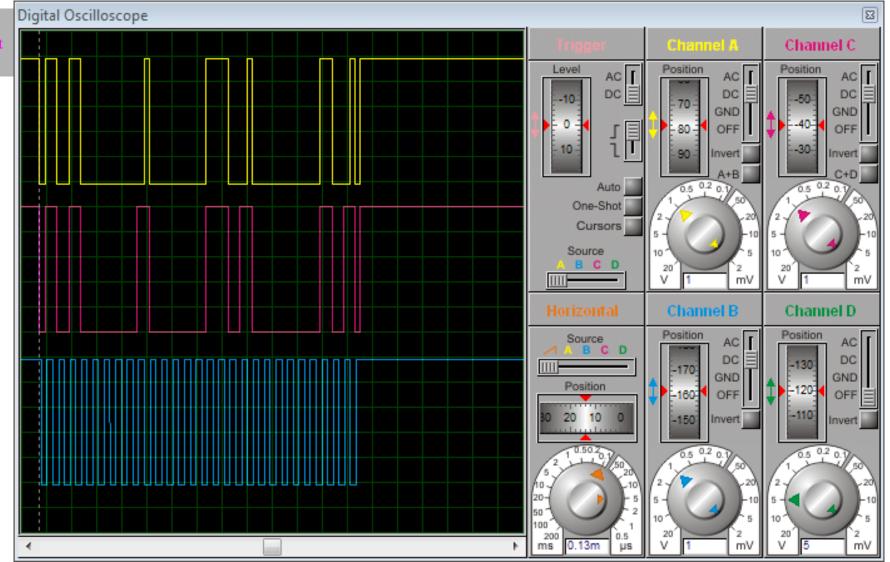
Channel A = SDA
Channel C = SDAout
Channel B = SCL



Ejemplos reales de visualización en MC24C01: Escribe la posición 6 un 2



Channel A=SDA
Channel C=SDAout
Channel B=SCL







☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C

→ Trabajo personal

Tareas a realizar



Introducción al Bus I2C

☐Índice de la presentación

- Competencias y objetivos
- ¿Qué hay que implementar?
- ¿Dónde lo implementamos? Entorno Experimental.
- Esquema de montaje
- Descripción del la memoria 24LC64
- 24LC64: modos de escritura y lectura
- Ejemplo de implementación
- Repaso (muy rápido): Protocolo I2C
- Tareas a realizar

Realización práctica: Tareas a realizar



- ☐ Tarea 1 (previas)
 - Implementación del protocolo I2C: funciones básicas
 - Estudio e implementación de las funciones básicas (start, stop, enviar bit, ...)
- ☐ Tarea 2 (entregable)
 - Menú básico de usuario
- Guardar un dato (de 0 a 255) en cualquier dirección de memoria del dispositivo 24LC64. Tanto el dato como la dirección se han de solicitar al usuario. (pidiéndolos por teclado/pantalla)
- 2. Leer una posición (de 0 a 8191) del 24LC64 (pidiendo la posición por pantalla)
- 3. Inicializar un bloque de 256 bytes contiguos de la memoria 24LC64 a un valor (la dirección del primer elemento y el valor a escribir se solicitan por pantalla; medir el tiempo el tiempo empleado en acceso a la memoria)
- Mostrar el contenido de un bloque de 256 bytes contiguos del 24LC64, comenzando en una dirección especificada (en una matriz de 8 0 16 columnas, en hexadecimal; medir el tiempo que consume el acceso a la memoria)
- Inicializar <u>usando "Page Write"</u> un bloque de 256 bytes contiguos del 24LC64 a un valor (la dirección del primer elemento y el valor a escribir se solicitan por pantalla; medir el tiempo que consume el acceso a la memoria)
- Mostrar el contenido de un bloque de 256 bytes del 24LC64 (usando <u>Sequential</u>
 <u>Read</u>), comenzando en una dirección especificada (en una matriz de 8 o 16
 columnas, en hexadecimal; medir el tiempo que consume el acceso a la memoria)

- ☐ Tarea 3 (mejoras)
 - Dispositivo de Reloj de Tiempo real (opcional, mejora)



Fin de la presentación

¿Preguntas?