

ESCUELA DE INGENIERÍA INFORMÁTICA

Grado en Ingeniería Informática



PERIFÉRICOS E INTERFACES

PRÁCTICAS DE LABORATORIO 17/18

Módulo 2 – Práctica 2

Bus I2C

“Práctica básica y mejorada”

Ana Isabel Santana Medina

Grupo: 1.18.43

Periodo de realización: 5ª – 9ª semana

Índice

- Objetivos de la práctica y competencias logradas
- Realización de la práctica
 - a. Descripción y herramientas utilizadas
- Mejora de la práctica
 - a. Descripción
- Código de la práctica y la mejoras (compacto)
- Conclusiones
- Comentarios finales
- Recursos utilizados y bibliografía

1. Competencias y objetivos básicos de la práctica

Los siguientes datos e información detallada corresponden a la segunda práctica base de la asignatura Periféricos e Interfaces y la mejora de la misma.

Una breve descripción de esta es que se centra en reforzar y completar los conocimientos teóricos adquiridos en el módulo 2 dedicado a los buses de interconexión. Para alcanzar este propósito lo que se propone es la conexión de un módulo I2C usando un puerto de entrada/salida paralelo de propósito general. Concretamente, se utilizará como primer elemento a conectar el módulo de memoria M24C01 de 128 bytes de capacidad (1 kbit) (o también M24C02 de 256 bytes, 2 Kbit)

Para realizar la práctica hemos hecho uso de una tarjeta “Arduino Mega” y el circuito de memoria M24C01/M24C02 adecuadamente montado en una tarjeta experimental de laboratorio.

En mi caso, además de haber usado el circuito del que disponemos en el laboratorio, he adquirido los diferentes componentes y he montado el circuito por mi cuenta, más adelante daré los detalles.

La práctica se divide en dos etapas:

- La primera es la etapa básica, donde se familiarizará con el entorno de programación del Arduino en general y con la tarjeta “Arduino Mega”. En especial es importante ejercitarse previamente en la comunicación con hardware externo, para poder leer información desde la tarjeta y mostrarla en la consola simulada en el PC y también poder enviarle información leída desde el teclado. Para esto aprovecharemos la herramienta “Serial Monitor” presente en el entorno de desarrollo. Será especialmente importante prestar atención a las conversiones que es necesario realizar de la información que se intercambia.
- La segunda etapa tiene como objetivo darle a las competencias adquiridas

una aplicación práctica lo más cerca posible al mundo real. Así, se desarrollará y experimentará con el protocolo del bus I2C, para profundizar en su conocimiento y conseguir dominar su uso. Para esto se tendrá que implementar a través de los terminales o pines de entrada/salida del Arduino (que se especifican más adelante) el protocolo de comunicación correspondiente al bus síncrono I2C y comprobar que las operaciones de lectura y escritura sobre el dispositivo externo se realizan correctamente. Se utilizarán habilidades de programación y acceso a las señales físicas.

Es tarea del estudiante planificar y desarrollar el software de bajo nivel para programar funcionamiento de estos dispositivos y realizar las transferencias de datos sincronizando la actividad de comunicación y el almacenamiento y lectura en la memoria I2C. Con la realización de esta práctica hemos alcanzado las siguientes competencias:

1. Capacidad para entender los diversos aspectos software y hardware involucrados en la gestión de un bus serie de entrada salida.
2. Capacidad para diseñar e implementar el software necesario para la gestión mediante un procesador de propósito general del acceso a un periférico I2C.
3. Capacidad para hacer uso de la información aportada por un fabricante de un dispositivo periférico real para utilizarlo en la resolución de un problema práctico real.
4. Capacidad para aprender y aplicar nuevos conceptos de forma autónoma e interdisciplinar.
5. Capacidad para emplear la creatividad en la resolución de los problemas.

Gracias a la consecución de los siguientes objetivos:

1. Conocer y usar el bus I2C y su protocolo de comunicación básico.
2. Conocer y entender los aspectos básicos de funcionamiento de los periféricos I2C y, concretamente, la memoria M24C01/M24C02 a partir de la hoja de características proporcionada por el fabricante.

3. Realizar rutinas sencillas que permitan operaciones de escritura y lectura en cualquier dirección de memoria del dispositivo y en los distintos modos soportados.
4. Verificar el correcto funcionamiento del hardware y del software de comprobación desarrollado. Observación de las señales, si fuese necesario.
5. 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.

2. Descripción y realización de la práctica

En primer lugar, comentaremos las funciones básicas implementadas:

Comenzamos con las funciones de protocolo del bus I2C:

- **START:** Capturar el bus, si es posible.
- **STOP:** Liberar el bus, o sea, parar la captura del bus.
- **E_BIT:** Enviar un bit según lo que se recibe.
- **R_BIT:** Leer un bit transmitido por el bus.

Tras esto comencé con la implementación del **byteWrite**, el cual recibe dos parámetros en el que uno es la posición/dirección y el otro el valor a guardar en la susodicha posición.

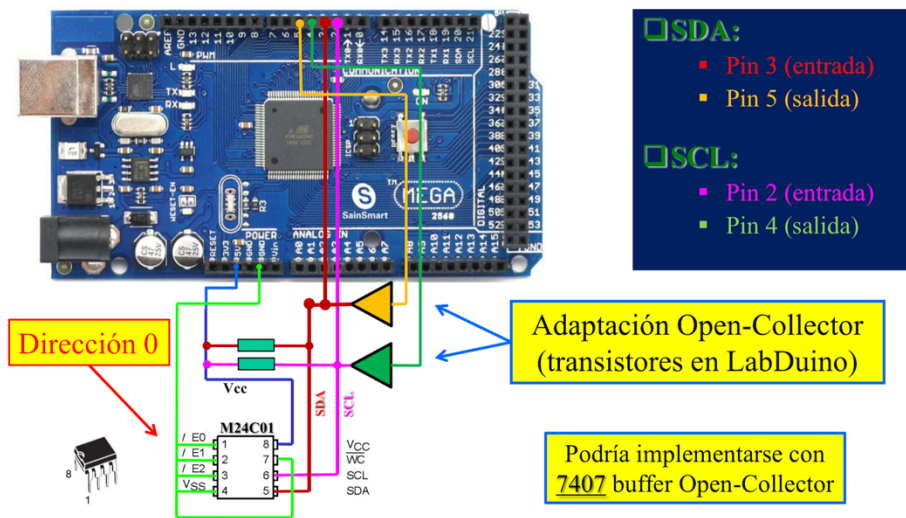
Luego proseguí con la implementación del **byteRead**, funciona de forma muy parecida al anterior, solo que trabajamos con un array en el que la primera posición contiene la posición/dirección a leer y la segunda contiene el valor que ha leído.

Más tarde, desarrollé las funciones que nos pide el enunciado de la práctica, las cuales no indicaré pues ya son conocidas por el profesorado.

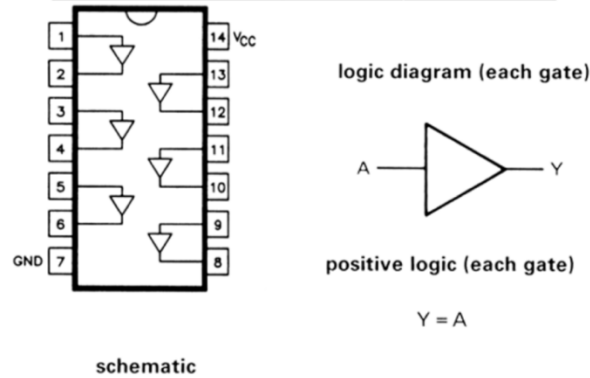
Por último, cabe comentar las dos funciones básicas en Arduino que son el **loop** y el **setup**. El primero solo muestra la interfaz de usuario y puede atender según la opción seleccionada. La segunda activa los pines de E/S.

En segundo lugar, comentaré como realicé la implementación del circuito. Yo comencé realizando la práctica en los laboratorios y más tarde decidí implementar el circuito para trabajar en casa y aprender a hacer el circuito como crecimiento personal.

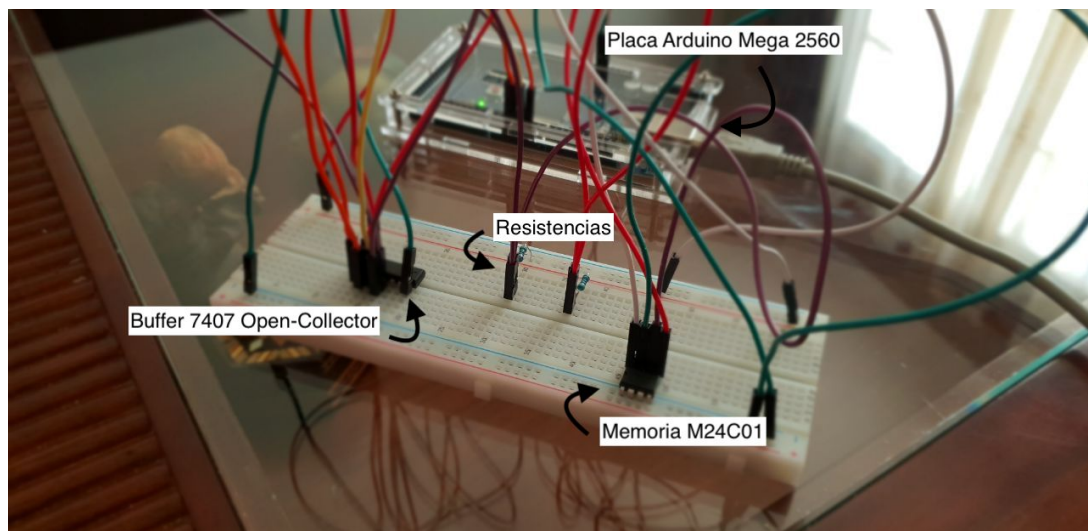
Para la implementación utilicé los esquemas que se ven a continuación:



7407 Buffers/line driver OC HV



Teniendo en cuenta estos dos esquemas, este fue el resultado:



3. Mejora de la práctica

Para la mejora he implementado 7 funciones más, pues decidí no implementar las funciones Page Write y Sequential Read que se proponían en el informe.

Las mejoras implementadas son:

1. **Suma de la diagonal:** Primero se pide al usuario inicializar la diagonal de la memoria a un valor comprendido, se rellena y se almacena la suma en una variable.
2. **Resta de la diagonal:** Primero se pide al usuario inicializar la diagonal de la memoria a un valor comprendido, se rellena y se almacena la resta en una variable, con la limitación de que el valor que se reste no será mayor de -255, si no se saldrá de rango.
3. **La sucesión de Fibonacci:** Tal como el nombre indica, he desarrollado la sucesión de Fibonacci, aunque solo llega hasta el valor 233, ya que el siguiente valor se pasaría de la memoria que es 255.
4. **Inicializar toda la memoria a 0:** Inicializa toda la memoria a 0, para no tener que utilizar la primera función básica.
5. **Ordenar la memoria de forma ascendente:** Inicializa las posiciones de memoria desde 0 hasta 255.
6. **Ordenar la memoria de forma descendente:** Inicializa las posiciones de memoria desde 255 hasta 0.

4. Código de la práctica y mejora de la misma

Se adjunta un archivo .ino con el código de la práctica.

Aquí podemos ver un ejemplo de simulación del menú de la misma:

[illegible]

Y aquí un ejemplo de simulación de una de sus funciones mejoradas:

Introduzca una opcion (0 - 11): 9									
0	0	1	2	3	4	5	6	7	
0	0	1	2	3	4	5	6	7	
8	8	9	10	11	12	13	14	15	
16	16	17	18	19	20	21	22	23	
24	24	25	26	27	28	29	30	31	
32	32	33	34	35	36	37	38	39	
40	40	41	42	43	44	45	46	47	
48	48	49	50	51	52	53	54	55	
56	56	57	58	59	60	61	62	63	
64	64	65	66	67	68	69	70	71	
72	72	73	74	75	76	77	78	79	
80	80	81	82	83	84	85	86	87	
88	88	89	90	91	92	93	94	95	
96	96	97	98	99	100	101	102	103	
104	104	105	106	107	108	109	110	111	
112	112	113	114	115	116	117	118	119	
120	120	121	122	123	124	125	126	127	

5. Conclusiones

Las conclusiones que he obtenido de la realización de esta primera práctica han sido que, en primer lugar, las competencias que se habían planteado han sido logradas gracias al estudio y trabajo de la misma. En segundo lugar, a pesar de no haber programado nunca en arduino ni haber utilizado la placa, ha estado bien utilizar algo que se usa actualmente, y que me servirá para futuros proyectos personales. Por último, me gustaría resaltar, que ha sido un proceso grato de aprender, pues, aunque fue muy complicado al principio, fue gratificante el hecho de que al final conseguí resolver la práctica.

Para la mejora un compañero me dio la idea de la suma de la diagonal, después de esta, pensé en la resta. Tras seguir pensando, me parecían pocas mejoras, muy sencillas, así que desarrollé cuatro ideas más, una opción con la que pudieras inicializar toda la memoria a 0 sin tener que acceder al primer método, una forma de ordenar las posiciones de memoria de forma ascendente y descendentes, para tenerlas inicializadas de esta manera, y más tarde pensé en la sucesión de Fibonacci, pues, las dos anteriores, también habían sido sucesiones.

6. Comentarios finales

Como comentarios finales, he de añadir que, en mi opinión, han resultado suficiente los recursos que nos han facilitado para la realización de la práctica, puesto que aparte de estos, teníamos la capacidad de acceder a internet, y, además, gracias a la construcción de mi propio circuito he sido capaz de probar mi programa fuera del laboratorio.

7. Recursos empleados

Para la realización de esta práctica he hecho uso de los siguientes recursos:

- **Arduino IDE**
- **Enunciado de la práctica 2017 y 2018**
- **Documentos varios relativos al funcionamiento del bus I2C**
- **Documentación de la Memoria M24C01/M24C02**

- **Presentación de la práctica 2017 y 2018**
- **Material para el montaje del circuito:**
 - Cables
 - Resistencias de 470 Ohm (2)
 - Memoria M24C01
 - Buffer 7407
 - Placa Arduino Mega 2560
 - Protoboard
 - Cable USB Tipo B