



“Banco de pruebas para comunicaciones seriales I2C dedicado al trabajo con microcontroladores Atmel con aplicaciones específicas debidamente documentadas”

(FEBRERO 2012)

Manuel Lenín Mizhquero Ramos
Saúl Ignacio Ashqui Pagalo
Dr. Ing. Carlos Valdivieso

Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del Litoral

Campus Gustavo Galindo Km 30.5, vía Perimetral, Apartado 09-01-5863, Guayaquil, Ecuador
mmizhque@fiec.espol.edu.ec, sashqui@fiec.espol.edu.ec

Resumen

El presente trabajo se proyecta como una guía de referencia para establecer la aplicación del protocolo I2C como estándar de comunicación serial para microcontroladores de la familia ATMEL. La característica principal del sistema resalta por su simplicidad al utilizar dos líneas independientes, necesarias para la transmisión de los datos y sincronización de los mismos. Dicha característica brinda ventajas adicionales como el hecho de poder establecer una transmisión semibidireccional y la correspondiente validación de datos enviados en toda transacción. Para el diseño, análisis e implementación del sistema se ha utilizado herramientas software, tales como, el paquete PROTEUS VSM de Labcenter Electronics y el compilador AVR-Studio de ATMEL. Ambas herramientas son una parte imprescindible para obtener un prototipo final del sistema que cumple las especificaciones establecidas por la norma I2C, por lo cual se verifica que es posible establecer una transmisión de este tipo con cualquier dispositivo fabricado bajo este estándar. El objetivo final de este trabajo es presentar de forma sencilla el uso de uno de los módulos componentes del microcontrolador AVR-ATMEL, específicamente el desarrollado para trabajar con el protocolo mencionado.

Palabras Claves: Protocolo I2C, Comunicación serial, Transmisión semi-bidireccional.

Abstract

This work is projected as a reference guide to establish the application of the I2C protocol as the standard of serial communication for the ATMEL family microcontrollers. The main characteristic of the system stands out for its simplicity by using two independent lines, necessary for the transmission of the data and their synchronization. This feature provides additional advantages such as being able to establish a half-bidirectional transmission and the correspondent validation of data sent in every transaction. For the design, analysis and implementation of the system we have used software tools, such as the PROTEUS VSM package from Labcenter Electronics and the AVR-Studio compiler from ATMEL. Both tools are an essential fragment to obtain a final prototype of the system that meets the specifications established by the I2C standard, so it is verified that it is possible to establish a transmission of this type with any device manufactured under this standard. The ultimate goal of this paper is to easily present the use of one of the component modules of ATMEL AVR microcontroller, specifically the module developed to work with the mentioned protocol.

Keywords: I2C Protocol; Serial Communication; Half-bidirectional transmission.



1. Introducción

El bus I2C fue desarrollado por Philips al inicio de la década de 1980 [6], como respuesta a la necesidad de interconectar dispositivos electrónicos, y ha encontrado una gran aceptación en el mercado. Actualmente los principales fabricantes de dispositivos semiconductores ofrecen circuitos que implementan un bus I2C para su control.

El acrónimo IIC o I2C significa Inter Integrated Circuit; es decir, que cuando se habla del bus I2C, nos referimos a un bus cuyo ámbito de aplicación es la comunicación entre circuitos integrados. A primera vista puede resultar un tanto extraño tal concepto, puesto que posiblemente se piense en la manera natural en la que se interconectan diversos circuitos integrados. Por ejemplo, si se considera un microprocesador y los tipos de circuitos que habitualmente forman parte de su sistema (memorias RAM, EEPROM, y periféricos como conversores A/D y D/A, puertos paralelos, pantallas alfanuméricas de cristal líquido, etcétera) entonces automáticamente se pensará en que el modo normal de conectar dichos circuitos es haciendo uso del bus paralelo a nivel de componente o de sistema.

Efectivamente, al principio el uso del bus paralelo puede permitir conseguir el máximo rendimiento en términos de velocidad de la unidad central de procesos. Pero si se piensa no tanto en un microprocesador y circuitería asociada sino en un sistema de control basado en microcontrolador; en este caso muy frecuentemente el MCU ya contiene toda la circuitería periférica que requiere el sistema, de manera que no se necesitará circuitería externa.

No obstante, también es frecuente que el sistema que se pretenda diseñar requiera circuitería periférica con características de las que no dispone ningún miembro de la familia de controladores que se utiliza. Esto implica la necesidad de generar externamente al MCU los buses de dirección, de datos y de control para así poder conectarle externamente los circuitos oportunos. Esta cuestión, que conceptualmente es verificada y no presenta ningún tipo de dificultad, resulta un inconveniente en ciertos casos. Ciertamente en aquellos en los que es de primordial importancia minimizar el tamaño del diseño. No se pierda de vista que un circuito integrado cuantas más señales contemple mayor será el número de patillas que necesite y por tanto mayor será el tamaño del encapsulado utilizado y mayor espacio de placa de circuito impreso requerirá el trazado de las pistas de interconexión; por no hablar del coste. Por otro lado, cuanto mayor número de señales circulan por una placa de circuito impreso o interconectan sistemas tanto mayor es la posibilidad de emisión o captación de EMI.

La clave del bus I2C está en la manera de ver los circuitos integrados. Aquí el concepto de comunicación entre sistemas se aplica a su nivel más bajo; es decir, cualquier tipo de circuito integrado, sea de la naturaleza que sea, se considera como un sistema cuyo modo de funcionamiento es controlable y que además requiere un canal de comunicación bidireccional con el sistema que lo controla. Este canal, que con ser semi-bidireccional es suficiente, se utilizará por un lado para que el sistema central defina como se comportará tal circuito – es decir, para transmitir órdenes – y por el otro lado para transmitir los datos que sean precisos en uno u otro sentido. Si se piensa en un módulo conversor dual A/D y D/A podrá captarse la idea: el conversor A/D podrá tener varios modos de funcionamiento (conversión única o continua, y en este último caso la frecuencia de la conversión); en lugar de usar señales específicas se utilizaría el canal de comunicación para transmitir una orden de configuración operativa. Por otra parte, una vez realizada una conversión es necesario que el CAD envíe el dato convertido al procesador; y de igual manera éste deberá enviar al CDA el dato a convertir a analógico.

2. Materiales y Funcionamiento

2.1. Materiales.

Detallamos los principales componentes utilizados en la implementación del proyecto:

Hardware:

- Microcontrolador AVR ATmega32 [1].
- Pantalla LCD, 16x2 [5].
- Memoria EEPROM 24LC32 [10]
- Reloj en Tiempo Real DS1307

Software:

- PROTEUS VSM version 7.5 o superior.
- AVR-Studio version 4.18 [9].

Componentes adicionales:

- Capacitor Cerámico de 27pF.
- Cristal Resonador de 8 Mhz.
- Potenciómetro regular de 10KΩ.

2.2. Funcionamiento

Los principales pasos que establecemos al realizar una transferencia de datos por medio del bus I2C, con el uso del módulo TWI [3], se describen a continuación [7]:

- Inicialización

Para inicializar el módulo TWI para operar en modo Maestro, se debe seguir los siguientes pasos:

1. Habilitar la frecuencia de reloj del módulo TWI, configurando los valores del registro TWBR y los bits TWPS [1:0] en el registro TWSR.
2. Habilitar el módulo TWI, configurando a uno el bit TWEN en el registro TWCR.
3. Transmisión de Condición de Inicio

Para iniciar la transferencia de datos en operación de modo Maestro, es necesario transmitir una condición de Inicio (Start). Esto es realizado con la configuración de los registros TWEN, TWSTA, y al poner en uno el bit TWINT del registro TWCR. Configurando el bit TWEN a uno habilita el módulo TWI. Configurando el bit TWSTA a uno permite al módulo TWI enviar una condición de inicio cuando el bus se encuentra libre, y configurando a uno el bit TWINT limpia la bandera de interrupción para iniciar la operación del módulo TWI transmitiendo la condición de Inicio. Además se debe sondear la bandera TWINT en el registro TWCR para conocer si la condición de Inicio ha sido transmitida completamente.

- Transmisión de Datos

Para enviar un byte de datos, después de transmitir una condición de Inicio (Start), se debe realizar los siguientes pasos:

1. Copiar el byte de datos al registro TWDR.
2. Poner a uno el bit TWEN y el bit TWINT del registro TWCR para iniciar el envío del byte de datos.
3. Revisar la bandera TWINT en el registro TWCR para conocer si el 7byte ha sido transmitido completamente.

- Recepción de Datos

Para recibir un byte de Datos, luego de transmitir el byte SLA+R, debemos realizar los siguientes pasos:

1. Configurar a uno los bits TWEN y TWINT del registro TWCR para iniciar la recepción de un byte de datos. Notar que si es necesario retornar un Acuse de Recibo (ACK) después de la recepción de datos es necesario también configurar a uno el bit TWEA del registro TWCR.
2. Revisar la bandera TWINT en el registro TWCR para conocer si un byte ha sido completamente recibido.
3. Copiar el byte de datos recibido desde el registro TWDR a otro registro como respaldo del mismo.
4. Transmisión de Condición de Parada

Para detener la transferencia de datos, debemos transmitir una Condición de Parada (Stop). Esto se realiza configurando los registros TWEN, TWSTO, y configurando a uno el bit TWINT del registro TWCR [9].

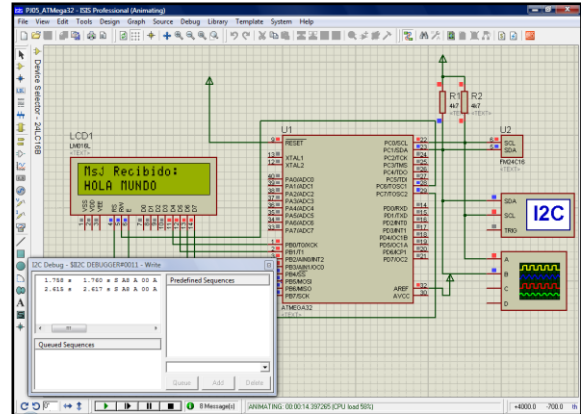


Figura 1. Transmisión y recepción por bus I2C con EEPROM.

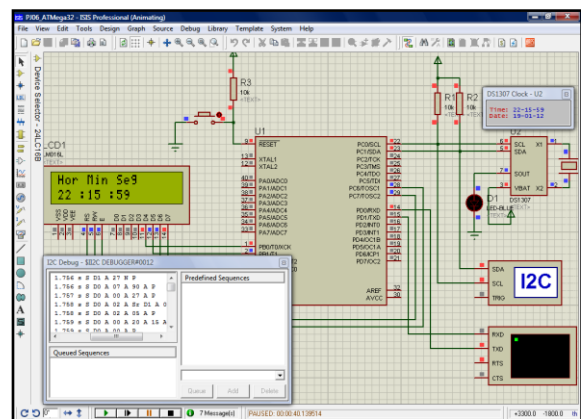


Figura 2. Reloj en tiempo real por bus I2C con DS1307.

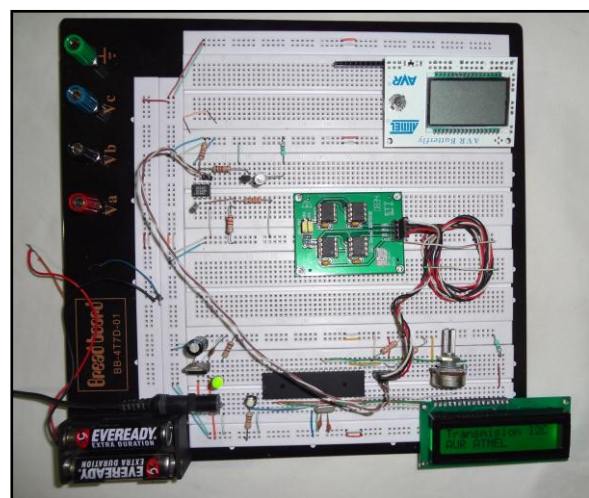


Figura 3. Implementación del proyecto.



3. Resultados

Los resultados obtenidos saltan a la vista. En primer lugar, si se opta por una comunicación serie como la expuesta en el presente trabajo es posible ahorrar un gran número de señales en el bus. De esta manera en este protocolo existe una línea de transmisión y otra de recepción; es decir, se tiene un bus de dos hilos que permite el intercambio de información entre sistemas; y en el plano eléctrico se utiliza, por simplicidad, señales con niveles TTL por lo que la distancia máxima del bus no puede ser nunca excesiva. En el plano lógico se define un protocolo de estructura de mensajes para los distintos tipos de circuitos que se pueden conectar. Pues bien, observamos que esta solución es muy aceptable por múltiples motivos, – siendo uno de los principales – el óptimo rendimiento que ofrece este tipo de enlace serie síncrono en lo que se refiere a la información neta transmitida, sin olvidar que cada carácter debe ir enmarcado por un bit de inicio y otro de parada; eso sin mencionar la posibilidad del uso de topologías multipunto.

4. Conclusiones

Se ha presentado de forma conjunta las especificaciones técnicas y el procedimiento para llevar a cabo una transmisión haciendo uso del módulo TWI, a través de la interfaz I2C, observando así la versatilidad y las distintas ventajas que ofrece este tipo de comunicación por medio de dos hilos. Aunque para dicha comunicación se ha utilizado una EEPROM y el dispositivo RTC-DS1307 en la práctica existen diversos dispositivos que incluyen este tipo de interfaz siempre compatibles por su estandarización.

Ciertamente el nivel de seguridad establecido para la realización de una transmisión debe ser óptimo en toda transferencia, es por esto que se presenta como una alternativa válida un proceso de chequeo o monitoreo del registro de Estado, en el módulo TWI método por el cual se puede establecer procedimientos para control de errores o fallos que se pueden presentar de manera fortuita en nuestra transferencia.

5. Referencias

[1] Atmel, ATmega32A Data Sheet Revision 8155C, <http://www.atmel.com/Images/doc8155.pdf>, fecha de consulta Julio 2011.
[2] AVR, LibC Reference Manual 1.4.4, <http://www.nongnu.org/avr-libc/>, fecha de consulta Julio 2011.

[3] Atmel, AVR315: Using the TWI module as I2C master Application Note. Revision 2564C, <http://www.atmel.com/Images/doc2564.pdf>, fecha de consulta Julio 2011.
[4] Atmel, AVR311: Using the TWI module as I2C slave Application Note. Revision 2565D 2009, <http://www.atmel.com/Images/doc2565.pdf>, fecha de consulta Julio 2011.
[6] Philips Semiconductors, THE I2C-BUS SPECIFICATION V2.1, <http://i2c2p.twibright.com/spec/i2c.pdf>, fecha de consulta Julio 2011.
[7] JMNLAB, Sensor de Ultrasonido I2C, <http://jmnlab.com/i2c/i2c>, fecha de consulta Agosto 2011.
[8] Mann, Richard, How to Program an 8-bit Microcontroller Using C Language, www.atmel.com, fecha de consulta Agosto 2011.
[9] Barnette, M.D., AVR Simulation with the ATMEL AVR Studio 4, www.avr.freaks.com, fecha de consulta Octubre 2011.
[10] Shawn, Johnson, <http://www.cursomicros.com/avr/i2c-eeeprom/eeeprom-24xxx.html>, fecha de consulta Noviembre 2011.