

# Sistema de Control por I2C

J. Rodríguez Aparicio, K. Reyes Reyes

Departamento de ingeniería en sistemas computacionales, ESCOM IPN  
chucho970407@gmail.com, kreyesr1300@alumno.ipn.mx

**Resumen—** Durante el desarrollo de esta práctica se implementa un sistema de control de ventiladores basado en un voltaje de entrada y la velocidad de estos, se utilizan 3 microcontroladores PIC diferentes, un PIC16F873A, un PIC12F1822 y un PIC16F819, cada uno realiza una tarea diferente y se comunicaran entre ellos a través de la interfaz I2C.

**Palabras Clave —** Control, interfaz I2C, microcontroladores, ventiladores.

**Abstract—** During the development of this practice, a fan control system based on an input voltage and their speed is implemented, 3 different PIC microcontrollers are used, a PIC16F873A, a PIC12F1822 and a PIC16F819, each one performs a different task and will communicate with each other through the I2C interface.

**Keywords —**Control, I2C interface, microcontrollers, fans.

## I. INTRODUCCIÓN

El protocolo I2C es una interfaz desarrollada por Philips, con el fin de comunicar microcontroladores y periféricos en una comunicación Half-duplex, esta forma de comunicación reduce el cableado y el número de conexiones entre los microcontroladores.

Originalmente fue especificado para velocidades de 100Kbps e intencionalmente para control simple y señales de estatus, simplificar los cableados entre los periféricos y aumentar la inmunidad al ruido, y debido a su bajo coste, su versatilidad técnica y su simplicidad ha logrado su popularidad.

El bus de comunicaciones I2C (nombrado a veces como I<sup>2</sup>C) es un protocolo que se efectúa por medio de DOS hilos. A través de estos dos hilos pueden conectarse diferentes dispositivos donde algunos de ellos serán maestros en cuanto muchos otros dispositivos serán esclavos.

Para poder reconocer cada uno de los dispositivos conectados a los dos hilos del bus I2C, a cada dispositivo se le asigna una dirección.

Así en este tipo de comunicaciones el maestro es el que tiene la iniciativa en la transferencia y este es quien decide con quien se quiere conectar para enviar y recibir datos y también decide cuando finalizar la comunicación.

Los dos hilos del bus de interfaz de comunicación I2C PIC son líneas de colector abierto donde una de las líneas lleva la señal de reloj y es conocida como (SCL), y la otra línea lleva los datos y es conocida como (SDA).

Los pines SDA y SCL I2C se encuentran especificados en todos los componentes que usan este tipo de protocolo de comunicación.

Para que la comunicación funcione se deben utilizar unas resistencias pull up (*resistencias conectadas a positivo*) para asegurar un nivel alto cuando no hay dispositivos conectados al bus I2C.

La conexión I2C entre un maestro y varios esclavos se muestra en la figura 1:

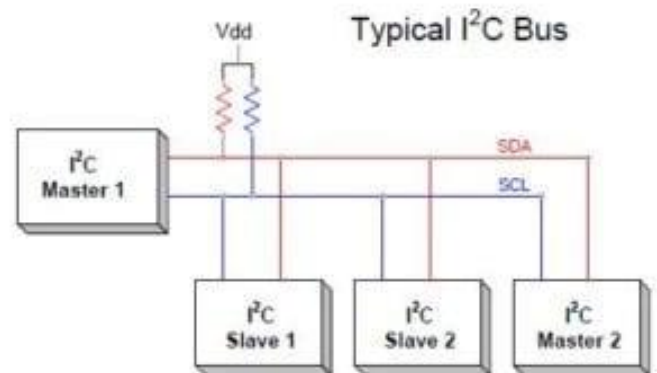


Fig. 1.- Conexión I2C[2].

El número de dispositivos que pueden conectarse de esta forma y además la longitud del BUS es limitada por la capacidad de direccionamiento (de 7 a 10 bits) y por la máxima carga del BUS (400pF). La velocidad máxima estándar es de 100Kbps.

A grandes rasgos en el protocolo de comunicación, el maestro controla la comunicación ya que genera la señal de reloj del bus, inicia y termina la comunicación, direcciona a los esclavos y establece el sentido de la comunicación. Este protocolo requiere que cada byte de información sea confirmado por el destinatario.

## II. METODOLOGÍA/DESARROLLO

### A. Herramientas a utilizar

Para el desarrollo de esta práctica se utilizará un PIC16F873A, un PIC12F1822, un PIC16F819, un teclado matricial, y dos pantallas LCD.

Los programas para los microcontroladores se realizarán en lenguaje C, para esto se hará uso del IDE MPLAB.

Por último, se utilizó el programa Proteus para probar el funcionamiento del circuito final que se muestra en figura 2.

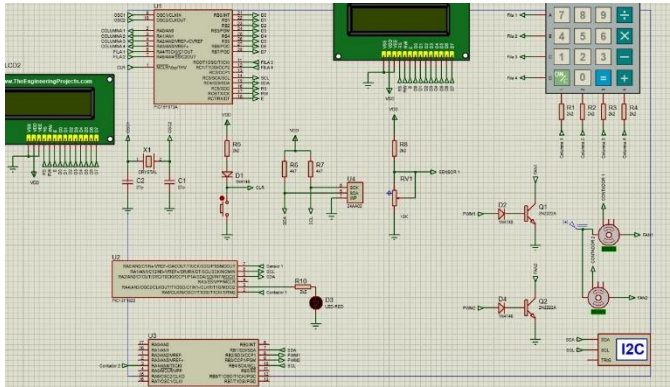


Figura 2.- Esquema del circuito.

### B. PIC16F873A

Este microcontrolador estará actuando como maestro, por lo que se deben hacer las configuraciones pertinentes en el módulo MSSP, así como las entradas SCL y SDA para poder establecer la comunicación con I2C.

Con un teclado matricial de 4x4 se estarán controlando las diferentes opciones a ver:

Comenzando se debe presionar una vez la tecla A y se mostrará en la pantalla LCD el voltaje que se estará leyendo del segundo PIC y la velocidad del primer ventilador.

Si se presiona una segunda vez la tecla A se desplegará el voltaje y la velocidad del segundo ventilador.

Si se presiona la tecla B, C o D se podrán cambiar las condiciones establecidas para el encendido del led y de los ventiladores, los valores ingresados serán guardados en la memoria EEPROM, vemos un ejemplo en la figura 3.

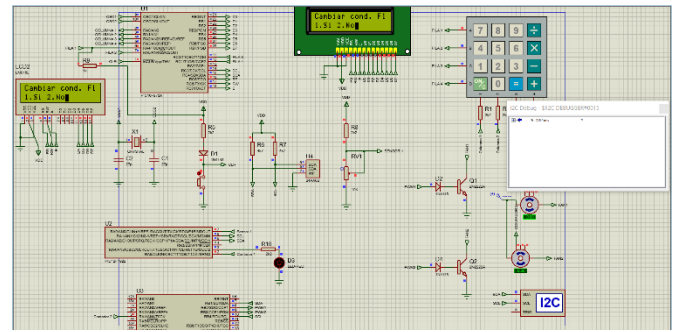


Figura 3.- Opción C.

El voltaje y las velocidades de los ventiladores estarán siendo recibidos a través de las entradas SCL y SDA.

### C. PIC12F1822

Para este PIC, el módulo MSSP se configura como esclavo, continuamos con la configuración las entradas y salidas, que en este caso RA0 será nuestra entrada analógica y RA1 y RA2 son nuestras entradas SCL y SDA para la comunicación I2C y por el puerto RA5 estará recibiendo la velocidad del ventilador 1.

Se deberá configurar al convertidor analógico-digital que viene en el PIC para poder manejar los valores que este detectando el sensor.

Tanto el voltaje y la velocidad del ventilador 1 estarán siendo enviados a través de las entradas SCL y SDA.

### D. PIC16F819

Finalmente, este PIC de igual forma que el anterior será configurado como esclavo con el módulo MSSP, y se configuraran los puertos RB1 y RB4 como las entradas SDA y SCL.

A través de los puertos RB2 y RB3 se estarán controlando los PWM para los ventiladores 1 y 2, estos deben ser configurados como salidas por el puerto RA4, el cual debe ser configurado como entrada, se estará obteniendo la velocidad del ventilador 2, dicha velocidad se estará mandando al PIC maestro a través de la interfaz I2C.

La velocidad del ventilador 1 depende del voltaje leído por el PIC anterior, entre mas pequeño sea el voltaje más velocidad tendrá el ventilador y entre más grande sea el voltaje mas lento irá el ventilador.

Por otra parte, la velocidad del ventilador 2 dependerá de la velocidad del ventilador 1, el ventilador 2 se encenderá cuando la velocidad del ventilador 1 este al 50% de su velocidad máxima.

### III. RESULTADOS Y DISCUSIÓN

A lo largo del desarrollo de esta práctica nos encontramos con algunos problemas, comenzando con el teclado matricial ya que el puerto RA4 estaba como tipo pull up, por lo que se tuvo que realizar una configuración extra dentro del diseño del circuito para el correcto funcionamiento, dicha configuración extra fue agregar una resistencia y conectarla a positivo enmendando así el error.

Por otro lado, notamos que en todos los PICs se debían establecer como entradas todos los puertos para la comunicación I2C, en el diagrama proporcionado esto no estaba así.

También, la velocidad del PIC maestro estaba establecida como 1MHz y esto hacía que nuestra simulación fuera demasiado lento, una vez lo notamos cambiamos la velocidad a una mínima de 4MHz como vemos en la figura 4.

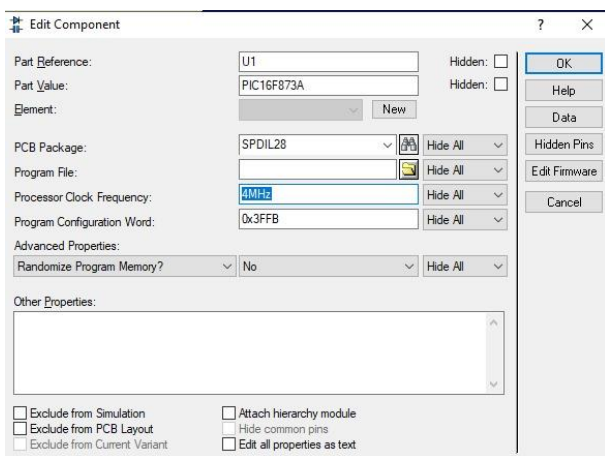


Figura 4.- Ventana de configuración del PIC maestro

Finalmente debido a que los resultados son simulados, no se pudieron apreciar con exactitud algunos parámetros como el voltaje medido por el sensor.

### IV. CONCLUSIONES

Para el desarrollo de esta práctica es necesario tener muy claros los conceptos teóricos previos, así como las prácticas realizadas anteriormente, así como tener cierto conocimiento de

los componentes utilizados ya que algunos tienen características muy específicas.

El diagrama de la simulación fue un punto clave ya que teníamos que acoplarnos a este y tomar en cuenta cualquier detalle por más pequeño que fuera. Se concluye también que los resultados pueden variar dependiendo de la computadora o versión del simulador que se utilice.

También aprendimos que se pueden crear sistemas de control complejos con pocos componentes, así como el impacto que estos pueden tener por las ventajas que estos brindan.

Finalmente, aprendimos otra forma de comunicación entre PICs, y a utilizar más microcontroladores PIC con diferentes dispositivos de entrada y salida.

### AGRADECIMIENTOS

Para finalizar este documento nos gustaría agradecer a todos aquellos que día a día se esfuerzan por compartir su conocimiento a través de diferentes medios, ya sea por videos, publicaciones en blogs, libros y artículos, gracias a ellos el desarrollo y entendimiento de esta práctica fue posible.

También agradecemos a nuestros compañeros de grupo que nos brindaron su apoyo en las partes que no entendíamos o no estaban totalmente claras.

Finalmente, gracias a nuestros familiares y amigos que siempre nos apoyan para seguir avanzado en nuestros estudios.

### REFERENCIAS

- [1] Sergio A. Castaño Giraldo [ Sergio A. Castaño Giraldo]. (26 junio 2016). Tutorial 17.1 Comunicación I2C Parte1- Programación de PIC en CCS C (PIC C). [Video]. Recuperado de [https://www.youtube.com/watch?v=iFbTRpcA-aQ&ab\\_channel=SergioA.Casta%C3%B1oGiraldo](https://www.youtube.com/watch?v=iFbTRpcA-aQ&ab_channel=SergioA.Casta%C3%B1oGiraldo)
- [2] Sergio A. Castaño Giraldo (2016), Comunicación I2C ,Control Automatico Educación , <https://controlautomaticoeducacion.com/microcontroladores-pic/comunicacion-i2c/>
- [3] Natalia Parody López (Enero 2006), Prácticas del uso del bus I2C para PIC16F876 de Microchip, Universitat Técnica Superior Enginyeria