

Desarrollo de un Prototipo Didáctico para red CAN

Enrique Gerardo Hernández Vega

Instituto Tecnológico de Chihuahua
ehernand@itchihuahua.edu.mx

Ana Luisa Grijalva Rico

Instituto Tecnológico de Chihuahua
algrijalva@itchihuahua.edu.mx

Jorge Alberto Peñaloza Enríquez

Instituto Tecnológico de Chihuahua
japenaloz@itchihuahua.edu.mx

Oscar Gonzalo Rivera Calderón

Instituto Tecnológico de Chihuahua
ogriverac@itchihuahua.edu.mx

Resumen

Este trabajo presenta los avances en el desarrollo de un prototipo didáctico para la red de comunicaciones conocida como CAN por sus iniciales en inglés (Controller Area Network). El tema de redes CAN es parte del programa de estudios de la materia Microcontroladores del plan de estudios 2010 de la carrera de Ingeniería Electrónica impartida en el Instituto Tecnológico de Chihuahua. Dado que en la actualidad no se cuenta con el equipo necesario para realizar prácticas sobre este tema y que los kits de desarrollo comerciales tienen un costo considerablemente alto, se desarrolló un prototipo didáctico con 6 nodos que permiten verificar la implementación de las diferentes formas funcionales que puede tener un nodo.

Palabras Claves: CAN, microcontrolador, nodo, prototipo, red.

1. Introducción

De acuerdo al plan de estudios 2010 de la carrera de Ingeniería Electrónica, cuya retícula se muestra en la Fig. 1, la materia Microcontroladores contempla 5 unidades, siendo la Unidad IV Interfaces de Comunicación y en la cual se encuentra el tema “Red de comunicación CAN”. Sin embargo, en la actualidad no se cuenta con la infraestructura necesaria para la realización de prácticas de laboratorio para dicho tema. Por esta razón se desarrolló el presente prototipo didáctico, como una forma de auto equipar el laboratorio de sistemas digitales para el soporte de las prácticas correspondientes. Lo que a continuación se presenta es el avance de un prototipo didáctico que será de utilidad para futuras generaciones en la comprensión de este protocolo de comunicación por medio de una tablilla que tenga diversos nodos conectados a la red que transmitan y reciban información entre ellos.



Fig. 1. Retícula de la carrera de Ingeniería Electrónica.

2. Desarrollo.

La red CAN sigue un protocolo serie asíncrono para redes de microcontroladores, basado en la detección de colisiones y acceso múltiple monitoreando la señal portadora (CSMA/CD) y soporta control distribuido en tiempo real con un alto nivel de seguridad [1]. El protocolo CAN fue desarrollado a mediados de la década de los 80's como una solución robusta para aplicaciones automotrices y desde 1990 se utiliza ampliamente tanto en aplicaciones automotrices como en control industrial [2]. La topología es tipo canal (bus), en el cual se conectan diferentes nodos que pueden transmitir y/o recibir información a través de 2 líneas en modo diferencial. Cada nodo de la red debe monitorear el bus y si detecta que no hay actividad, podrá mandar un mensaje. Sin embargo, si dos nodos de la red comienzan a transmitir a la vez un mensaje, se detectará una colisión y por el método de prioridades se resolverá el conflicto. Este protocolo, entonces, se basa en el sistema "Multicast", pues todos los nodos pueden hablar por turnos y ser escuchados. Para transmitir los datos se utiliza un par de cables trenzados llamado bus diferencial que se utiliza para conseguir que la información llegue de manera segura, ya que al estar trenzados tiene alta inmunidad a los efectos electromagnéticos [3].

El protocolo CAN se conforma de una cadena de bits que tiene una longitud de 130 o 154 bits dependiendo del tipo de especificación que queramos usar (CAN estándar o CAN extendido). Consta de 7 campos y cada uno tiene una función específica, siendo de suma importancia el campo identificador con el cual podemos resolver problemas de conexión en el bus [3, 4].

Para iniciar el desarrollo del prototipo, se partió de la tarjeta CCS BUS-CAN, Fig. 2, para tener una base confiable de una red CAN y a la cual se le agregaron 2 nodos. Esta tarjeta CCS BUS-CAN será sustituida por un microcontrolador PIC18F4550 al concluir con este proyecto. La tarjeta CCS BUS-CAN cuenta con cuatro nodos A, B, C y D, implementados con los siguientes dispositivos: microcontrolador PIC18F4580 que contiene controlador CAN, microcontrolador PIC16F876A que no contiene controlador CAN y dos controladores CAN MCP25050, respectivamente. Todos estos dispositivos requieren de

la interfaz a nivel de la capa física, que permite la conexión al bus CAN, por lo que utilizan el transceptor MCP2551. Además la tarjeta cuenta con pines para conectar nodos externos [5, 6, 7].

En cuanto a los 2 nodos desarrollados, uno de ellos contiene un microcontrolador PIC18F4580 y el otro un microcontrolador Arduino Duemilanove, y fueron nombrados como nodo E y nodo F respectivamente.

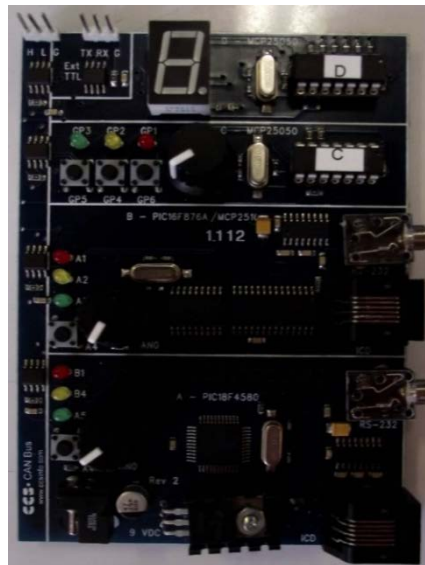


Fig. 2. Tarjeta CCS BUS-CAN.

En la Fig. 3 se muestra el diagrama a bloques del sistema y en la cual se encuentran en tono gris los 2 nodos agregados.

El nodo E fue desarrollado en base al PIC18F4580 ya que este contiene el controlador CAN y puede operar en el modo mejorado, con el cual a nivel de software se cuenta con las librerías necesarias para su programación [8]. El único requerimiento para la utilización de este PIC es la interfaz a nivel de la capa física, que permite la conexión al bus CAN, en este caso se utilizó el circuito integrado MCP2551. En la Fig. 4 se muestra la implementación de este nodo E.

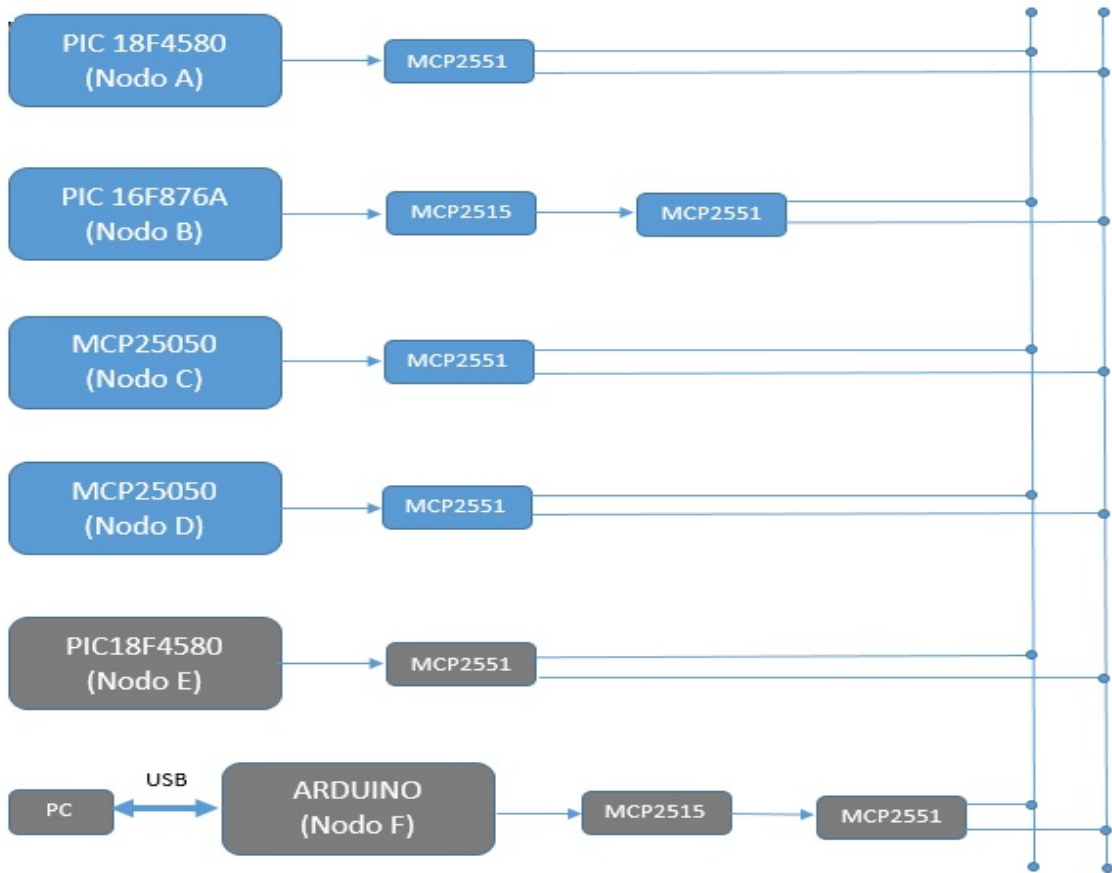


Fig. 3. Diagrama a bloques del proyecto realizado.

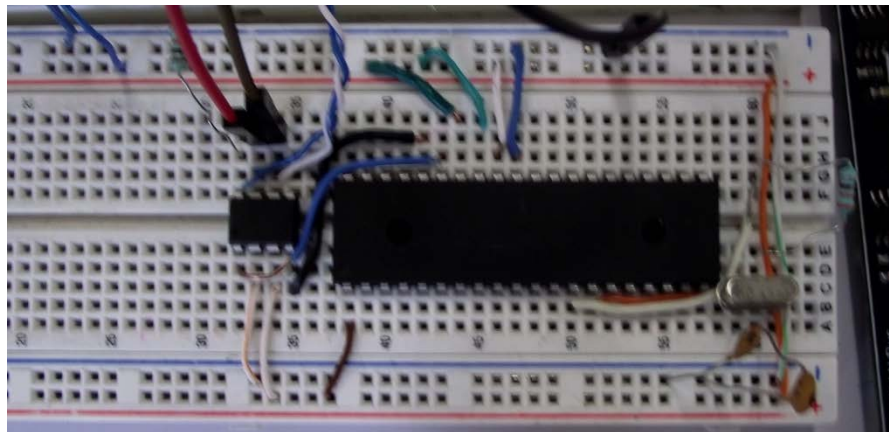


Fig. 4. Nodo E con PIC18F4580.

Para el desarrollo del nodo F se utilizó la plataforma Arduino Duemilanove, que contiene un microcontrolador Atmega 328, el cual no posee un módulo CAN [9]. Por esta razón se requiere el controlador CAN MCP2515 y la interfaz a nivel de la capa física, el transceptor MCP2515. Se debe destacar que a nivel de hardware la conexión que hay entre el Arduino y el controlador CAN MCP2515, es en base al protocolo SPI, lo cual se muestra en la Fig. 5.

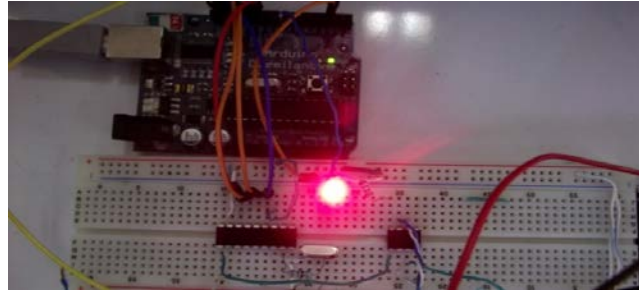


Fig. 5. Tarjeta Arduino manejando SPI a CAN

El nodo F fue desarrollado para tener un analizador del tráfico en la red, el cual permitiera observar en la PC los mensajes que se envían, resaltando los campos más importantes, el identificador del nodo (ID) y los datos enviados [10]. La Fig. 6 muestra la información obtenida con este nodo F que monitorea el tráfico de información en el canal (bus) de la red CAN. La conexión de este nodo con la PC fue por medio de la interfaz USB.

```
COM3
CAN BUS Shield init ok!
MCP2515 Library Receive Example...
ID: 0 Data:
ID: 400 Data: 1E 7F 30
ID: 400 Data: 1E 7F 19
ID: 400 Data: 1E 7F 12
ID: 400 Data: 1E 7F 02
ID: 400 Data: 1E 7F 78
ID: 400 Data: 1E 7F 00
ID: 400 Data: 1E 7F 10
```

Fig. 6. Monitoreo del bus CAN con el nodo F.

3. Resultados

Los resultados obtenidos fueron la implementación de dos nodos capaces de integrarse a la tarjeta CCS CAN BUS. Como ya se mencionó, dichos nodos se basan en el microcontrolador PIC18F4580 y en la plataforma Arduino, uno capaz de transmitir, recibir y procesar datos de la red CAN y el otro con la función de monitorear la actividad del bus y desplegar la información en la pantalla de la PC con la cual se comunica vía la interfaz USB. El sistema completo se muestra en la Fig. 7.

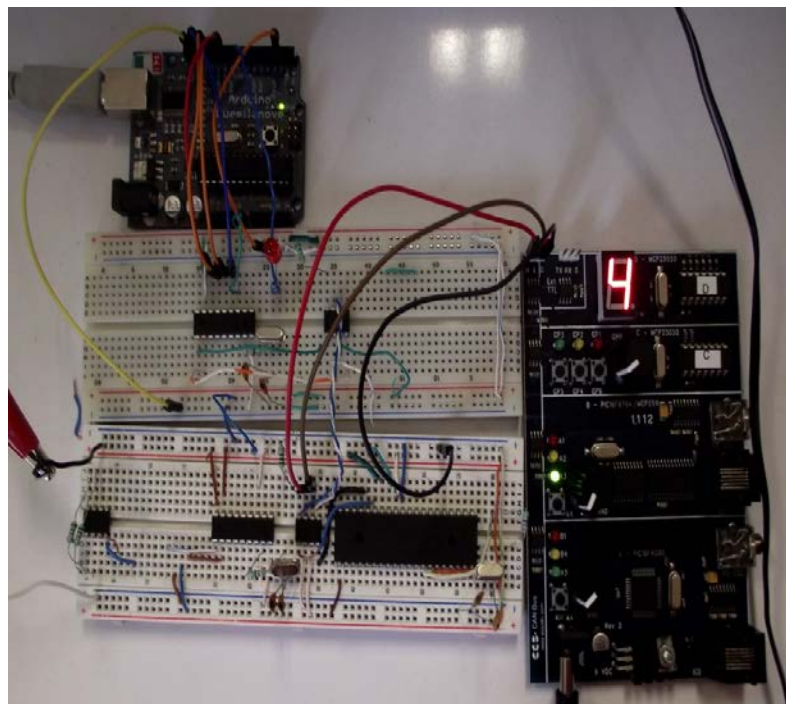


Fig. 7. Prototipo con los dos nodos externos implementados.

4. Discusión

Hay 3 formas de implementación de nodos CAN. Estas son:

1. En base a un dispositivo, regularmente un microcontrolador, que ya contiene el controlador CAN, y al cual se le agrega la interfaz física con el bus, llamada

“transceiver” en inglés. Este tipo de nodo se considera “inteligente” ya que tiene capacidad de procesamiento.

2. En base a un microcontrolador que no contiene el controlador CAN y al cual se le agrega tanto el controlador CAN como la interfaz a la capa física. A este tipo de nodo también se le considera “inteligente”.
3. En base a un dispositivo que no tenga capacidad de procesamiento pero que permita el acceso al bus CAN y el manejo de entradas y salidas digitales. Este tipo de dispositivo es llamado expansor de entradas y salidas. En este caso sólo hay que agregar la interfaz a la capa física.

Las primeras 2 formas fueron implementadas en el prototipo didáctico, quedando pendiente la implementación de la tercera forma y el sustituir el Arduino, en el nodo F, por un PIC18F4550. De tal manera que el prototipo final estará elaborado exclusivamente en base a dispositivos del fabricante Microchip.

5. Conclusiones

De acuerdo a los resultados se puede ver que la implementación de 2 nodos CAN, uno de ellos desarrollado en la forma 1 en base al PIC18F4580 y el otro desarrollado en la forma 2 utilizando la plataforma Arduino, fue satisfactoria, dejando como experiencia el procedimiento para estas 2 formas de desarrollo de nodos CAN y el dominio del protocolo de comunicación. Así mismo se logró desarrollar una interfaz USB-CAN para la implementación de un analizador del tráfico en el bus CAN y que muestra los resultados en la pantalla de la PC.

En trabajos posteriores se implementará un tercer nodo de la forma 3, lo cual nos permitirá implementar una tarjeta de desarrollo de prácticas para el bus CAN en la cual se consideren las 3 formas de implementación de nodos, una interfaz USB-CAN, un analizador del bus y la expansión de nodos. Todo esto en base a productos del fabricante Microchip.

6. Referencias

- [1] R. Bosch. CAN Specification Version 2.0. Manual. 1991.
- [2] W. Lawrenz. CAN System Engineering from Theory to Practical Applications. New York. Springer. 1997. Pp. 26.
- [3] Application Note: A CAN Physical Layer. Microchip Technology Inc. USA. 2002.
- [4] Application Note: Smart sensor CAN node using the MCP2515 and PIC16F876. Microchip Technology Inc. USA. 2010.
- [5] MCP2551 High-Speed CAN Transceiver. Data Sheet. Microchip Technology Inc. USA. 2003-2012.
- [6] MCP2515 Stand-Alone CAN controller with SPI interface. Data Sheet. Microchip Technology Inc. USA. 2003-2012.
- [7] MCP2050 CAN I/O expander. Data Sheet. Microchip Technology Inc. USA. 2003-2012.
- [8] PIC18F2480/2580/4480/4580. Data Sheet. Microchip Technology Inc. USA. 2009.
- [9] Evans B. Arduino programming notebook: A beginner's reference. Creative Commons. 2007
- [10] CAN Capture Overview.
<http://www.cancapture.com/?gclid=COL3sYetocECFY9cMgodFzgAXg>. Mayo del 2014.

7. Autores

M.C. Enrique Gerardo Hernández Vega. Profesor de tiempo completo del Departamento de Ingeniería Eléctrica y Electrónica del Instituto Tecnológico de Chihuahua. Obtuvo su

título de Maestría en Ciencias en Ingeniería Electrónica en 2006 por el Instituto Tecnológico de Chihuahua.

Ana Luisa Grijalva Rico. Alumna de 6° semestre de la carrera de Ingeniería Electrónica del Instituto Tecnológico de Chihuahua.

Jorge Alberto Peñaloza Enríquez. Alumno de 6° semestre de la carrera de Ingeniería Electrónica del Instituto Tecnológico de Chihuahua.

Óscar Gonzalo Rivera Calderón. Alumno de 5° semestre de la carrera de Ingeniería Electrónica del Instituto Tecnológico de Chihuahua.