

SISTEMA SUPERVISORIO DE UNA MICRORED CD BASADO EN COMUNICACIÓN CAN

Ernesto Everardo Tovar Zavala

Instituto Tecnológico de Celaya
neto_omega@hotmail.com

Javier Díaz Carmona

Instituto Tecnológico de Celaya

Elías Rodríguez Segura

Instituto Tecnológico de Celaya

Agustín Ramírez Agundis

Instituto Tecnológico de Celaya

Alejandro Espinosa Calderón

Instituto Tecnológico de Celaya

Resumen

En el presente artículo se describe un sistema que permite supervisar la energía distribuida en una microred CD. Dicha microred forma parte de un proyecto a ser implementado en el Instituto Tecnológico de Celaya. El sistema propuesto permite monitorear el estado actual de celdas fotovoltaicas, bancos de batería y el consumo de energía en cargas. La comunicación a través de los distintos dispositivos que componen la microred se basa en el protocolo de comunicación CAN, el cual permite tener una gran cantidad de nodos, robustez y rapidez en la transmisión de la información. El sistema supervisorio es implementado mediante una interface gráfica en una computadora conectada como nodo a la red CAN, la cual permite visualizar el estado actual en tiempo real de la microred CD.

Palabra(s) Clave(s): Control area network, JAVA, harmony, microred, PIC32.

1. Introducción

En la actualidad nuestro planeta se encuentra en una transición en la que los recursos que ésta nos ofrece se están agotando, debido a la alta demanda de consumo de energía.

Como sabemos la mayor parte de la energía que se consume es la energía eléctrica, ya que nos facilita y mejora nuestra calidad de vida. Sin embargo, la mayoría de los métodos para obtener energía eléctrica son a base de combustibles, carbón, petróleo y derivados; los cuales son elementos no renovables, debido a esto llegará el día en que se terminen estos recursos.

Por esta razón se han buscado formas alternativas para la generación de electricidad y de esta manera reducir el consumo de combustibles. Se busca que estas formas alternativas hagan uso de energías renovables, ya que difícilmente estas fuentes se agotaran. Una de las metodologías para utilizar energías renovables es el empleo de lo que se llama microred.

Una microred es una red eléctrica que utiliza fuentes de energía distribuidas, de las cuales la mayoría son renovables además cuenta con sistemas de almacenamiento de energía. Gracias a esta combinación de energías renovables y dispositivos de almacenamiento es posible el suministro de energía con menores fluctuaciones y así cumplir con la demanda en forma local, a su vez se reducen pérdidas de transmisión de energía y se logra que la combinación de estas sea más eficiente.

Una microred cuenta con una gran cantidad de componentes en cada nodo, entre los cuales se puede mencionar: sistemas fotovoltaicos, convertidores, dispositivos de almacenamiento, entre otros. Derivado de lo anterior una microred requiere que exista transferencia de información eficiente, con la finalidad de administrar la transferencia de energía entre cada uno de los nodos y con ello reducir pérdidas.

Hay diversas maneras de supervisar un sistema de control, lo cual se puede llevar a cabo a través de diversos protocolos de comunicación como lo son el HART, Profibus, Fieldbus, Modbus(RS-232), CAN bus, Ethernet entre otros.

En lo que respecta a microredes de CD se ha reportado el uso de Red de Área local (LAN) para la comunicación entre la microred y un gestor de la red de

distribución [1]. Pero el uso de esta técnica puede no estar disponible entre la microred y el sistema de control. Además se requiere de una gran inversión para poder realizarlo.

La transferencia de información a través de la línea eléctrica (PLC) se ha vuelto popular para el control de sistemas eléctricos AC [2]. Un estudio de la viabilidad de PLC para sistemas DC de bajo voltaje es reportado en [3]. Sin embargo, en [4] se dice que la transmisión de datos a través de PLC provoca atenuaciones en condiciones de cargas capacitivas y cuando varía el tiempo, resultando en una comunicación indeseada.

Un protocolo de comunicación adecuado para la comunicación en una microred es una red de control de área (CAN, Controller Area Network) [5] [6]. Una manera alternativa de emplear CAN para la operación de control de una microred es propuesto en [7], en donde se muestra el potencial de reducir el costo de comunicación, debido a la alta disponibilidad en aplicaciones comunes de electrónica de potencia.

En [8] se utiliza una comunicación a través del protocolo CAN y RS-485. El RS-485 se emplea utilizando el protocolo MODBUS para supervisar el sistema (Transferir datos relacionados con la generación y consumo de energía). Es mencionado que CAN logra una comunicación rápida, confiable y distribuida. Por otro lado la comunicación por medio de RS-485 es lenta.

El proyecto desarrollado en [9] propone una alternativa para implementar un sistema de seguimiento del punto de máxima transferencia de potencia (MPPT) basado en la plataforma FPGA para el control de un módulo fotovoltaico, también en el mismo proyecto se tiene la implementación de los protocolos CAN y SPI usando diferentes plataformas de desarrollo así como lenguajes de programación como VHDL, C++ y ensamblador.

En [10] se desarrolla una interface entre PMBUS y CANBUS, para lograr un control a larga distancia en redes inteligentes de CD.

En el instituto tecnológico de Celaya se tiene proyectado implementar una microred de corriente directa para el suministro de energía eléctrica en las

instalaciones del departamento de Ingeniería Electrónica y de esta manera contar con un laboratorio experimental en el área de energías renovables (figura 1).

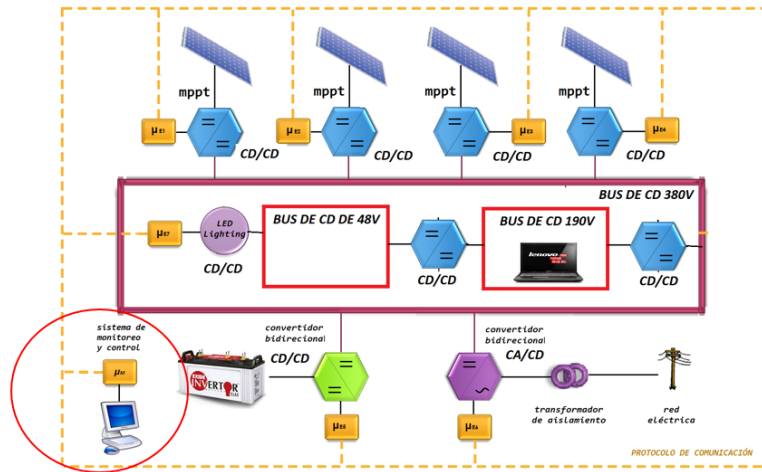


Figura 1 Esquema microred.

La microred CD constará de 40 paneles solares con una capacidad de 250 Watts, los cuales suministrarán energía mediante convertidores CD/CD. Se contará con dos buses de CD uno de 48 Volts y otro de 190 Volts. También habrá un banco de baterías, para en caso de que los paneles no estén generando la suficiente energía éstas entren a cumplir con la demanda de energía. En caso de que los paneles estén generando más energía de la que se está consumiendo, ésta se usará para cargar el banco de baterías. Si por alguna razón tanto los paneles como las baterías no generen la suficiente energía habrá también un convertidor CD a AC bidireccional conectado a la red eléctrica.

Cada uno de los nodos de la microred CD estará interconectada a través del bus CAN. El presente proyecto tiene el objetivo de diseñar e implementar un sistema que supervise el estado de potencia de cada uno de los nodos que componen la microred y en base a la información recibida realizar las acciones pertinentes. Para ello se diseñó un programa en JAVA, con el cual se realiza dicha supervisión mediante una interface USB-CAN basada en un microcontrolador PIC32 de microchip.

En la siguiente sección se definen conceptos relacionados al proyecto y en la tercera sección se describe el sistema propuesto. Los resultados obtenidos al

utilizar el sistema con dos nodos se presentan en la cuarta sección, mientras que la discusión de los mismos se presenta en la última sección.

2. Marco teórico

CAN

CAN es un protocolo de comunicación desarrollado por Robert Bosch en 1968, [11], el cual es un sistema de red de alta velocidad, alcanzando velocidades de transmisión de hasta 1Mbit/s. Los nodos del bus no tienen direcciones específicas ya que en lugar de direccionar la información, se etiquetan los mensajes transmitidos con un identificador, el cual determina la prioridad del mensaje e informa su contenido.

CAN cuenta con varias características como lo es jerarquía de nodos multimaestros, Técnica de acceso CSMA/CD, comunicación de evento impulsado, difusión, respuesta de solicitud remota entre otras. Gracias a estas características CAN proporciona sofisticados mecanismos de detección y tratamiento de errores, así como alta inmunidad a las interferencias electromagnéticas.

CAN cuenta con dos estándares diferentes estándar (CAN 2.0A) y extendido (CAN 2.0B), cuya principal diferencia radica en la longitud del identificador: 11 bits para CAN 2.0A y 29 bits para CAN 2.0B. En comparación con otros protocolos de comunicación serial como lo es el RS 232, CAN puede transmitir datos a altas velocidades y distancias considerables (50 kbits por segundo a 10 km).

Cada mensaje de CAN consiste en un cierto número de bits que son divididos en campos. El significado de estos campos es el ilustrado en la figura 2:

- Bit de inicio = 1 bit = bajo. Marca el inicio de un mensaje.
- Identificador = 11 bits. Dirección lógica y prioridad del mensaje; entre más bajo el valor más alta es la prioridad (0 = Mayor prioridad).
- Bit RTR = 1 bit. El bit RTR (Remote Transmission Request) es usado por un receptor para pedir a un transmisor remoto que envíe su información.
- Campo de control = 6 bits. El primer bit en el campo de control es el bit IDE (Identifier Extension). Si este bit es transmitido como un 0, esto significa que no hay más bits identificadores para enviar (Trama estándar CAN). Bit

r0 es reservado. Los siguientes 4 bits contienen la longitud del dato de los siguientes campos.

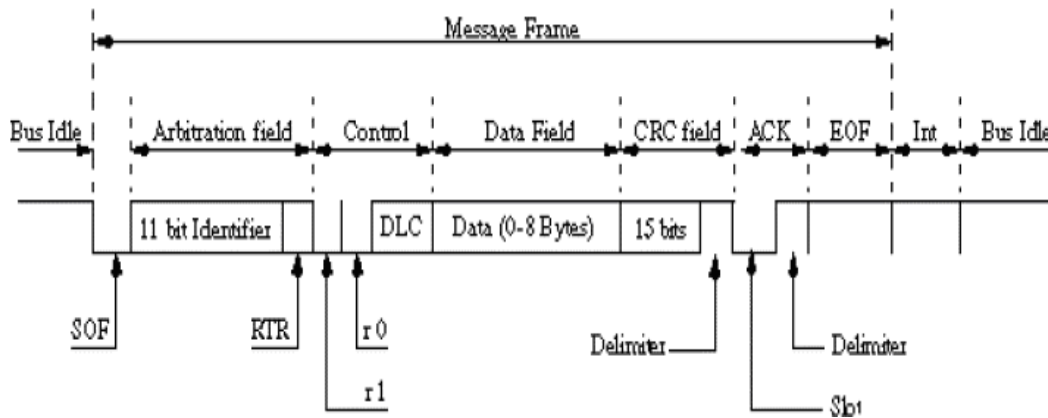


Figura 2 Formato de trama estándar.

- Campo de datos = 0 a 64 bits. Contiene los datos del mensaje de la aplicación.
- Campo CRC = 16 bits. Contiene una suma de comprobación para los bits anteriores del mensaje. La suma de comprobación CRC solo será usada para detección de error.
- Campo ACK = 2 bits. Cada nodo activo en la red que detecte un mensaje correcto en el bus reescribirá el nivel recesivo por un nivel dominante en el campo ACK. Este bit es leído por el transmisor del mensaje. Si lee un nivel recesivo en el campo ACK esto se tratará como un error.
- Campo EOF = 7 bits = Alto. La regla del Bit stuffing es violada cuando el EOF está activo. El EOF marca el final de una trama CAN.
- IFS = 7 bits = Alto. El IFS (InterFrame Space) marca la duración de tiempo necesaria para que el controlador de CAN transfiera una trama correctamente recibida del bus a su correspondiente lugar en el área de almacenamiento de mensajes.
- IDLE = 0 bit = Alto. El bus no es usado.

Plataforma de desarrollo

Existen diversas plataformas comercialmente disponibles para realizar una interface CAN-USB. Se decidió utilizar en el proyecto la tarjeta PIC32MX5 Starter

Kit basada en el microcontrolador PIC32MX570 de microchip debido a que las prestaciones que ofrece son muy adecuadas al objetivo deseado. Entre las que destacan: 512 kB de flash, 64KB de RAM, USB Dvice/Host/OTG, CAN 2.0B, canales ADC, transceiver de alta velocidad.

En cuanto al desarrollo del software se realizó con la herramienta llamada Harmony, la cual cuenta con un conjunto de librerías cubriendo diversos protocolos (CAN, USB, I2C, Bluetooth, SPI, USART, TCP/IP entre otros).

Harmony MPLAB es una plataforma flexible, abstracta, y totalmente integrada para desarrollo de firmware para microcontroladores PIC32. Toma elementos clave del diseño modular y orientado a objetos, añade la flexibilidad de utilizar un sistema operativo en tiempo real (RTOS) o trabajar sin él, y proporciona un marco de módulos de software que son fáciles de utilizar, configurable para las necesidades específicas, y que trabajan juntos en completa armonía. Harmony MPLAB incluye un conjunto de bibliotecas periféricas, drivers y los servicios del sistema que son de fácil acceso para el desarrollo de aplicaciones. El formato de desarrollo de código permite la máxima reutilización y reduce el tiempo de comercialización [12]. De esta manera es más fácil conjuntar ambos protocolos CAN y USB en un solo dispositivo microcontrolador. Al utilizar Harmony y dar de alta las librerías de CAN y USB permite configurar dichos protocolos en USB: Taza de transferencia, cantidad de datos, modos de transferencia (CDC, HID, BULK), número de endpoints, el vendor y product. Por parte de CAN permite configurar el Baud rate, tramas de datos, tramas remotas, cantidad de datos, protocolo standart o extendido, identificadores, máscaras y filtros.

Harmony utiliza el método de estados y subrutinas en su programación. De este modo es más cómodo para el programador visualizar donde puede haber un error, debido a que todo está estructurado. La manera en que Harmony maneja los cambios de estado es el siguiente: Primero se declara el estado donde se iniciará el programa por default. Cuando se inicia el programa éste empieza a hacer lo que se encuentra en el estado inicial. Al final de cada estado se encuentra una línea en la que se indica cual será el estado siguiente. Si por alguna razón no hay un estado siguiente, el programa se mantiene en el estado actual repitiendo su rutina.

Java

La interfaz gráfica y la transferencia de datos se implementó mediante la plataforma JAVA versión 1.8.0_51-b16 utilizando el entorno de desarrollo eclipse mars 2 versión 4.5.2. La transferencia de datos entre computadora y tarjeta se realizó mediante la librería usb4java con la cual es posible el acceso a dispositivos USB.

Java es la base para prácticamente todos los tipos de aplicaciones de red, además del estándar global para desarrollar y distribuir aplicaciones móviles y embebidas, juegos, contenido basado en web y software de empresa.

Java ha sido probado, ajustado y ampliado por toda una comunidad de desarrolladores, arquitectos de aplicaciones y entusiastas de Java. Además está diseñado para permitir el desarrollo de aplicaciones portátiles de elevado rendimiento, para el más amplio rango de plataformas informáticas posible. Así mismo, al poner a disposición de todo el mundo aplicaciones en entornos heterogéneos, las empresas pueden proporcionar más servicios y mejorar la productividad, las comunicaciones, la colaboración del usuario final y reducir drásticamente el costo de propiedad tanto para aplicaciones de usuario como de empresa [13].

3. Desarrollo propuesto

Lo que se propone es el desarrollo e implementación de un sistema supervisorio para una microred de corriente directa. La supervisión de los distintos módulos se realizará en tiempo real a través del protocolo de comunicación CAN.

Los diversos dispositivos que componen la microred contarán con un módulo CAN, para de esta manera lograr la transferencia de datos de manera robusta a través de los nodos CAN. Sin embargo, es indispensable contar con un sistema de supervisión que permita monitorear en tiempo real la potencia disponible o consumida en cada uno de los nodos de la microred.

Por tales motivos la propuesta del sistema supervisorio consta de una interface de usuario gráfica diseñada en JAVA e implementada en una computadora que forma parte de la red CAN. La comunicación se realiza mediante una interface

CAN/USB, para de esta manera poder visualizar y supervisar lo que ocurre en los diversos nodos de la microred (figura 3).

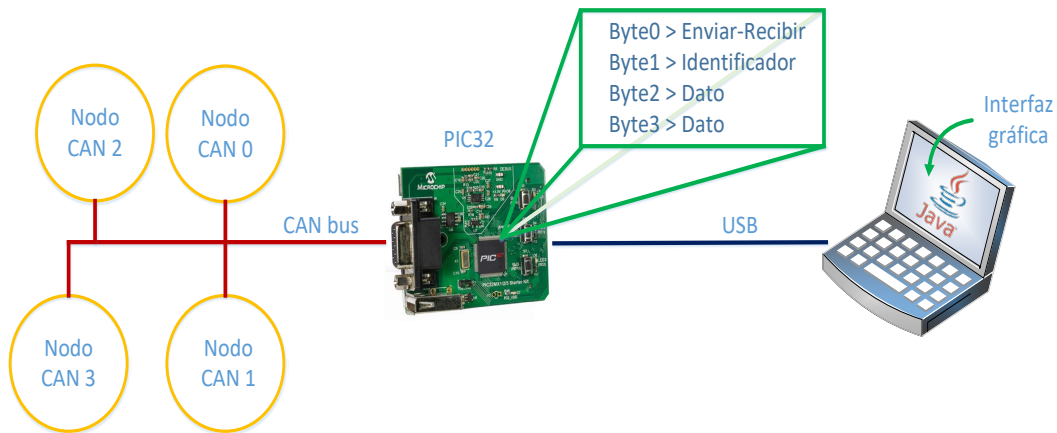


Figura 3 Esquema de sistema propuesto.

Programa tarjeta PIC32

En nuestro caso se configuró el USB en modo de transferencia BULK con una tasa de transferencia de 64 bytes. Para ajustar el tiempo de bit se realizaron los cálculos mostrados en las ecuaciones 1 y 2.

$$T_{BTL} = T_{XOSC} / (2 * (BRP + 1)) \quad (1)$$

$$Bit\ rate = (T_{BTL} / (SJW + PRSEG + SEG1PH + SEG2PH)) \quad (2)$$

$$SJW = 1, PRSEG = 1, SEG1PH = 3, SEG2PH = 3, BRP = 2, T_{XOSC} = 48MHz$$

$$T_{BTL} = \frac{48MHz}{2 * (2 + 1)} = \frac{48MHz}{6} = 8MHz$$

$$Bit\ rate = \left(\frac{8MHz}{1 + 1 + 3 + 3} \right) = \left(\frac{8MHz}{8} \right) = 1Mbps$$

El protocolo CAN funciona de la siguiente manera en la tarjeta: primero se requiere habilitar el modulo CAN para después configurarlo. Para que funcione correctamente se asigna una memoria RAM la cual almacena los mensajes que serán enviados o recibidos. El tamaño de esta depende de la cantidad de canales y el tamaño de los datos. En nuestro caso se configuró para dos canales, uno de transmisión y otro de recepción, de 10 datos ocupando un total de 320 de RAM.

En el buffer de recepción se tienen almacenados los mensajes leídos por el módulo CAN, el cual cuenta con una máscara permitiendo pasar los identificadores del 0 al 3. El diagrama a bloques del software de CAN se muestra en figura 4.

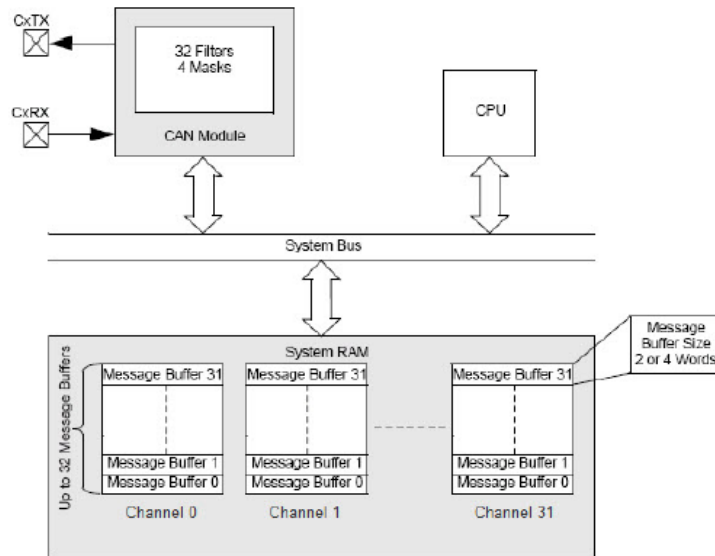


Figura 4 Diagrama de bloques software CAN.

Para acceder a los mensajes de CAN se hace uso de la instrucción `DRV_CAN0_ChannelMessageReceive` la cual tiene como argumentos el canal, el identificador del mensaje deseado, el tamaño del dato y la variable donde será almacenado el mensaje.

En el programa principal se cuenta con 3 estados: estado inicial, espera de configuración USB y estado de tareas principales (ver figura 6). El algoritmo general es el siguiente: De la aplicación en la computadora se envía un primer byte, denominado `byte0`, que representara si se quiere enviar o recibir un mensaje vía CAN. Como segundo byte se envía el identificador del nodo CAN y un tercer byte, que en el caso de que se envíe un mensaje esté representa la acción que se quiere que realice el nodo; en caso de que se quiera recibir un dato habrá un cuarto byte, éste y el tercero representan la lectura de voltaje o corriente del nodo. Para poder apreciar el algoritmo completo observar el diagrama de flujo descrito en figura 5.

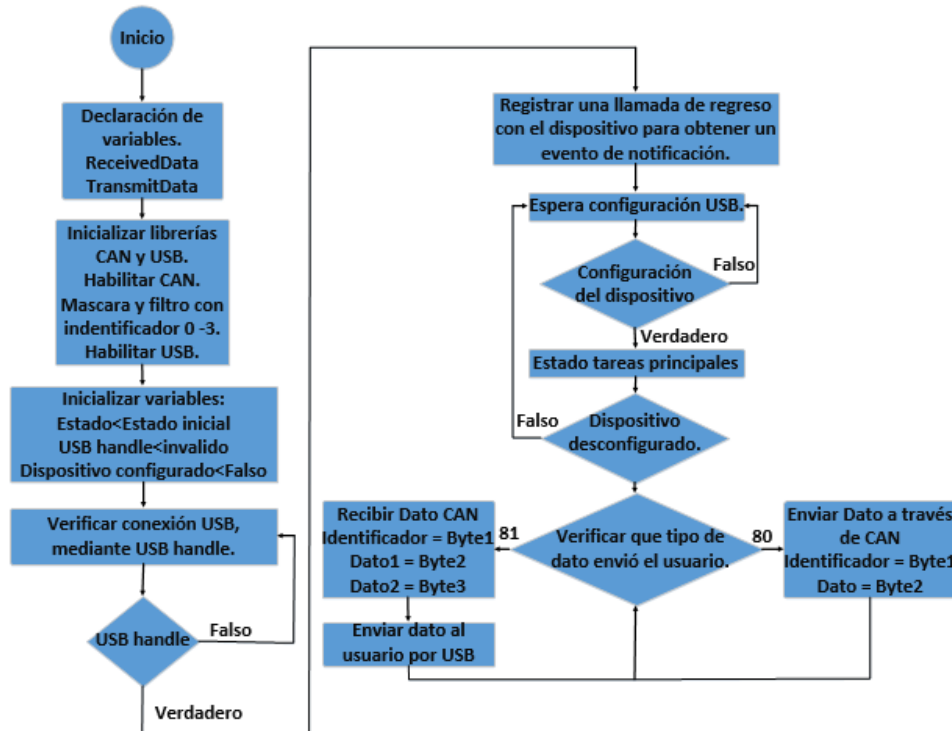


Figura 5 Diagrama de flujo programa de PIC32.

Programa en Java

Lo que se busca con esta interfaz es generar un ambiente de supervisión amigable, de tal manera que se pueda observar lo que ocurre en la microred a través del protocolo CAN y tomar acciones en base al estado de cada uno de los nodos.

La comunicación PC-PIC32 mediante USB se estableció por medio de la librería `usb4java`, la cual es un conjunto de archivos `.jar` que permiten tener una comunicación con diversos dispositivos USB. Para la aplicación deseada se utiliza para enviar y recibir datos de manera BULK al microcontrolador.

Lo primero que realiza el programa es declarar las variables del *product* y *vendor id* del microcontrolador y los *endpoints* de lectura y de envío, que en nuestro caso el de lectura es el `0x81` y el de envío `0x01`. El uso de estas variables hace posible realizar un listado todos los dispositivos USB conectados mediante la instrucción `findDevice`, y con lo cual obtener el descriptor del dispositivo USB de la tarjeta PIC32. Después de se obtiene el descriptor, se realiza una serie de pasos para

obtener su *Handle*, establecer los *endpoints* del dispositivo USB y activar una bandera que indica que la comunicación con el microcontrolador fue un éxito. Una vez establecida la comunicación es posible hacer uso de la instrucción *LibUsb.bulkTransfer(device handle, endpoint, buffer, transfered, timeout)*; la cual dependiendo del *endpoint* asociado envía o recibe un dato del tamaño dado por *transfered* y siendo el dato el *buffer*. Por ejemplo para enviar un dato se usaría el *endpoint* 0x01 y para recibir el 0x81. Definido esto ya es posible tanto enviar como recibir datos desde el microcontrolador PIC32, lo siguiente es diseñar la aplicación.

La interfaz cuenta con información de los nodos principales; esta información va desde el estado de los paneles solares, la cantidad de carga en las baterías, el consumo y generación de energía en la microrred. Este modo es el inicial y cuenta con las opciones de agregar o eliminar nodos (figura 6).

Para la administración de los nodos se utilizó un archivo de texto. Al seleccionar la opción agregar nodo (figura 7) se abrirá la ventana agregar nodo (figura 8) en la cual se pide ingresar el identificador del nuevo nodo así como un nombre con el cual se identificará.

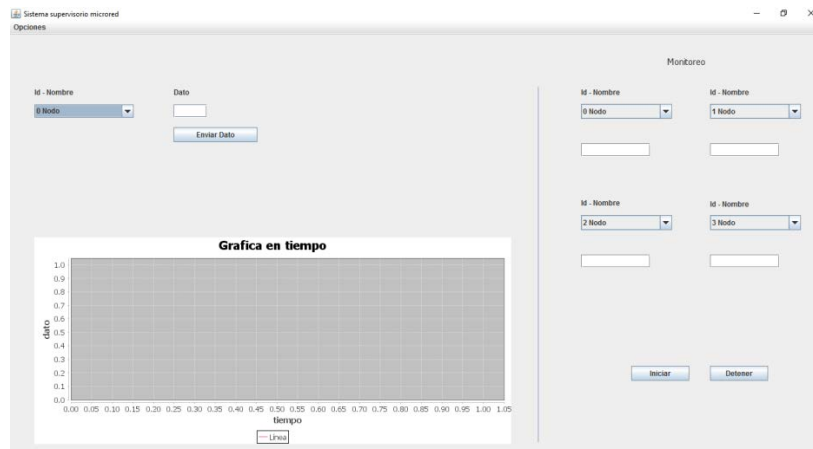


Figura 6 Aplicación sistema supervisorio.

De la misma manera al seleccionar la opción eliminar nodo se abrirá una nueva ventana donde en un *comboBox* se encontrará la lista de los nodos existentes. Se selecciona uno y después se presiona el botón eliminar (ver figura 8).

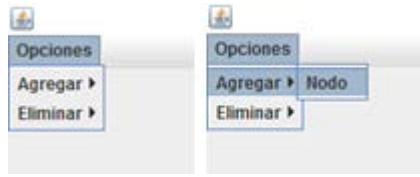


Figura 7 Opciones.

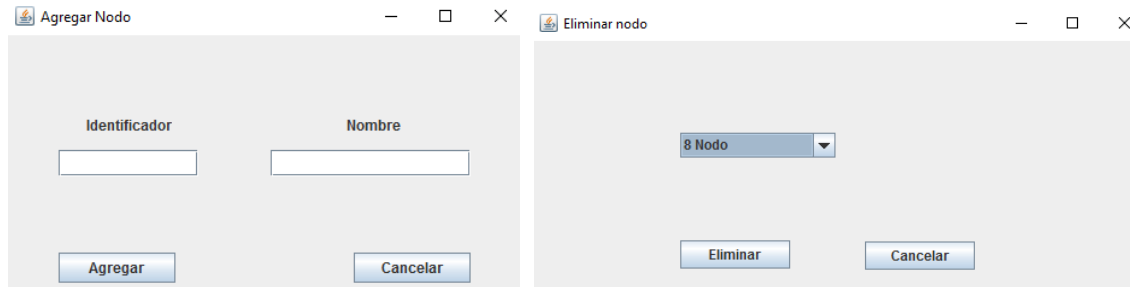


Figura 8 Ventana agregar y ventana eliminar.

La ventana principal se puede dividir en 3 áreas principales: una para enviar un dato a un nodo determinado, una área donde se muestra la lectura de voltaje de 4 nodos diferentes y una área donde se grafican estos voltajes. El objetivo de poder enviar un dato a un nodo específico, es para obtener un control manual de ese nodo y de esta manera tomar la decisión de apagarlo o encenderlo (figura 9).

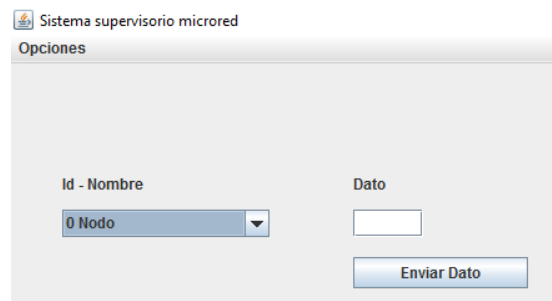


Figura 9 Sección enviar dato.

La segunda sección es la de monitoreo en tiempo real (figura 10). Primero se seleccionan los nodos de los cuales requerimos su información. Al presionar el botón iniciar se inicializaran dos hilos en JAVA los cuales se ejecutan paralelamente. Uno se encarga del envío y lectura de los 4 distintos nodos a

través de USB/CAN. El otro se encarga de realizar lecturas de los 4 nodos, mostrando el comportamiento de voltaje de los nodos (figura 10).

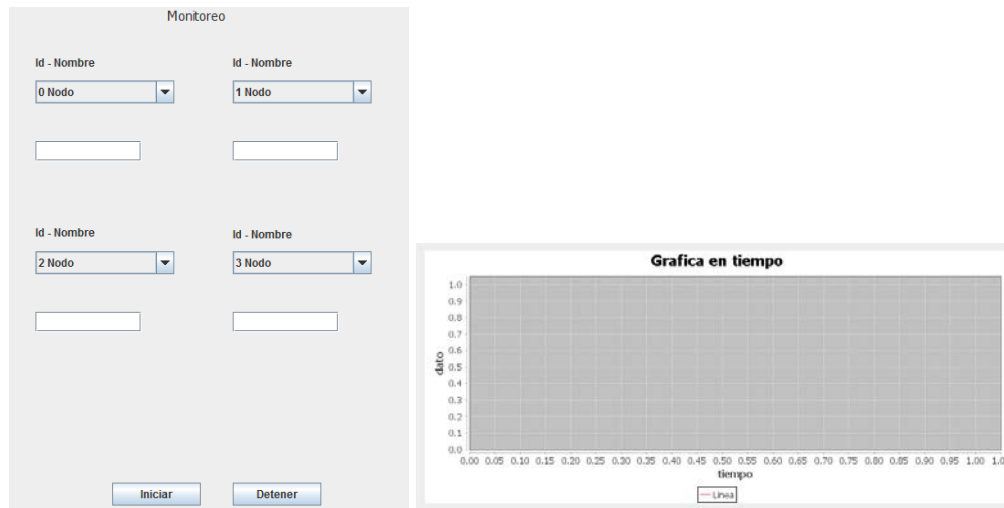


Figura 10 Monitoreo y gráfica de voltajes.

La gráfica se genera utilizando la librería *jfreechart* que al igual que *usb4java* es un archivo tipo *jar*. Dependiendo del tipo de gráfico deseado la estructura de su programación cambia.

4. Resultados

En la parte experimental del sistema se utilizó un simulador modular de panel solar de marca Agilent (figura 11), el cual suministra la energía mediante la simulación del comportamiento de un módulo solar. En este caso se configuró para que genere 124.59 V y una corriente de 1.1922 A, ésta energía es suministrada a dos convertidores CD/CD con una ganancia unitaria (figura 12) los cuales forman parte de una microred de CD de dos nodos, conteniendo cada uno de ellos un módulo CAN (uno con identificador 0 y el otro con identificador 3) con un Baud Rate de 1 Mbps.

Estos dos nodos entregan como dato su voltaje de salida. Con la finalidad de probar el sistema diseñado se varió el voltaje generado por el simulador de panel solar. Iniciando con un voltaje de 180 V, subiendo después a 195 V y después bajando a 123 V. Después se apagó el generador el cual disminuye su voltaje

gradualmente hasta llegar a cero. Al final se volvió a encender y se configuro para un voltaje de 126 Volts.



Figura 11 Simulador modular de panel solar.

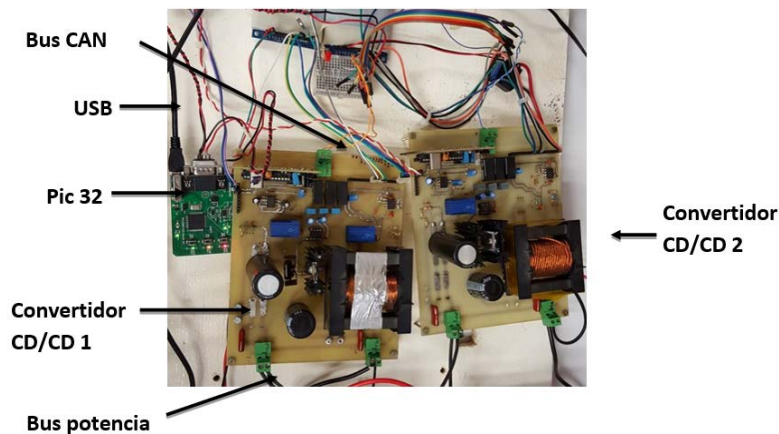


Figura 12 Convertidores conectados a la red CAN.

El voltaje medido a la salida de los convertidores CD/CD fue inicialmente de 195 Volts, a partir del cual éste varió en el rango indicado en la figura 13; con lo cual se verifica el correcto funcionamiento de la comunicación CAN/USB. La línea roja indica el nodo con identificador 0 mientras que la azul el nodo con identificador 3.

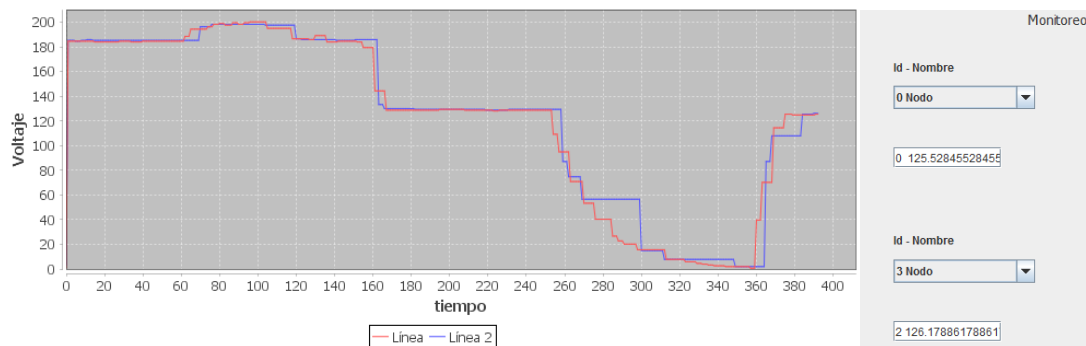


Figura 13 Sistema supervisorio en funcionamiento.

5. Discusión

Los resultados reportados del sistema desarrollado corresponden al comportamiento de voltaje de dos nodos dentro de la microred CD. Pero dicho comportamiento puede ser extrapolado al caso de utilizar un mayor número de nodos. Por otra parte la aplicación desarrollada se probó durante un periodo de tiempo de 18 horas monitoreando de manera correcta el estado de la red. Debido a que ambos convertidores CD/CD son unitarios y que tienen como entrada el mismo voltaje suministrado por el panel solar, idealmente se esperaría que ambos tuviesen un comportamiento de voltaje idéntico. Sin embargo, el sistema supervisorio desarrollado permite observar el comportamiento real de ambos convertidores, los cuales difieren en su respuesta.

Es importante resaltar que esta versión del sistema supervisorio fue puesta a prueba con el monitoreo de valores de voltaje en los dos nodos disponibles. Sin embargo, la medición puede ser realizada sobre valores de corriente e incluso potencia. La razón por la que solo se reporta voltaje es porque esa variable es la disponible en los nodos de los convertidores. El despliegue de información mediante el sistema desarrollado tiene un pequeño retraso, pero éste puede ser reducido mediante el tiempo de muestreo utilizado.

6. Conclusiones

La implementación del protocolo CAN para supervisar el estado de una microred de CD resultó ser una buena opción, ya que al ser el bus de tipo par trenzado la información no se ve afectada por interferencias aparte de que no es muy costoso. Las pruebas experimentales realizadas con dos nodos confirman el correcto funcionamiento del sistema supervisorio. Respecto a la plataforma JAVA ha resultado ser un entorno de programación muy amigable y debido a sus diversas opciones de diseño se puede seguir mejorando el entorno visual para el usuario. Los resultados de las pruebas demostraron que la interfaz CAN/USB es una forma fiable para monitoreo en tiempo real. Ahora que la comunicación ya está establecida, como trabajo futuro queda mejorar aspectos de la aplicación y poder generar un archivo de registro del día a día de la aplicación. Así como poder

implementar el algoritmo de supervisión de la microred ya con más nodos CAN y mejorar los pequeños detalles respecto a la comunicación, mencionados en la discusión.

Particularmente como trabajo futuro a desarrollar es la incorporación de nodos a la microred CD correspondiente a: cargas, bancos de baterías e inversores conectados a la red de distribución AC de CFE, donde cada uno de ellos cuente con una interface CAN con un identificador asignado. Lo anterior permitirá el desarrollo de algoritmos de control que permitan la administración eficiente de la energía disponible. Por ejemplo, un caso podría ser que la energía de los paneles solares sea consumida por una carga de CD y además se cuente con energía remanente. En dicho caso el sistema supervisorio después de sensar la energía disponible en el panel solar y la consumida en carga (ambos como nodos de la red CAN) podría decidir en accionar la carga de un banco de baterías (también conectado como un nodo de la red CAN), o dado el caso que se tengan las baterías en carga completa se podría activar el suministro del sobrante de energía a la red de distribución AC mediante el inversor.

7. Bibliografía y Referencias

- [1] R.Majumder, G. Ledwich, A. Ghosh, S. Chakrabarti, F. Zare, “Droop control of converter-interfaced microsources in rural distributed generation”. *IEEE Trans. Power Del.* Vol. 25. No. 4. Oct. 2010. Pp. 2768–2778.
- [2] D. Guezgouz, D. E. Chariag, Y. Raingeaud, J.-C. Le Bunetel, “Modeling of electromagnetic interference and PLC transmission for loads shedding in a microgrid”. *IEEE Trans. Power Electron.* Vol. 26. No. 3. Mar. 2011. Pp. 747–754.
- [3] A. Pinomaa, J. Ahola, A. Kosonen, “Power-line communication-based network architecture for LVDC distribution system”. in *Proc. IEEE Int. Symp. Power Line Commun. Appl.* Apr. 2011. Pp. 358–363.
- [4] T. A. Papadopoulos, G. K. Papagiannis, P. S. Dokopoulos, “Low-voltage distribution line performance evaluation for PLC signal transmission”. *IEEE Trans. Power Del.* Vol. 23. No. 4. Oct. 2008. Pp. 1903–1910.

- [5] Y. Zhang, H. Ma, "Theoretical and experimental investigation of networked control for parallel operation of inverters". *IEEE Trans. Ind. Electron.* Vol. 59. No. 4. Apr. 2012. Pp. 1961–1970.
- [6] C.-L. Chen, Y. Wang, J.-S. Lai, Y.-S. Lee, D. Martin, "Design of parallel inverters for smooth mode transfer microgrid applications". *IEEE Trans. Power Electron.* Vol. 25. No. 1. Jan. 2010. Pp. 6–15.
- [7] C.-L. Chen, W. Yubin, L. Jih-Sheng, L. Yuang-Shung, D. Martin, "Design of parallel inverters for smooth mode transfer microgrid applications". *IEEE Trans. Power Electron.* Vol. 25. No. 1, Jan. 2010. Pp. 6–15.
- [8] S. S. Thale, R. G. Wandhare, V. Agarwal, "A Novel Reconfigurable Microgrid Architecture With Renewable Energy Sources and Storage". *IEEE Transactions On Industry Applications.* Vol. 51. No. 2. March/April 2015. Pp. 1806–1807.
- [9] A. M. Ramírez Del Ángel, MPPT digital para un conjunto de módulos solares orientado a aplicaciones de micro-redes mediante CAN. Junio 2013. Tesis que para obtener el grado de M.C. en Ing. Electrónica, Instituto Tecnológico de Celaya.
- [10] A. López Arreguín, Control de un sistema de alimentación distribuido basado en pmbus. 2014. Tesis que para obtener el grado de M.C. en Ing. Electrónica, Instituto Tecnológico de Celaya.
- [11] W. Lawrenz, *CAN System Engineering From Theory to Practical Applications*. primera edición. 1997. Ed. Springer-Verlag. New York. Pp. 3–10.
- [12] Harmony. <http://www.microchip.com/mplab/mplab-harmony>. Mayo 2016.
- [13] Java. <https://www.java.com/es/about/>. Mayo 2016.

8. Autores

Ing. Ernesto Everardo Tovar Zavala obtuvo su título de Ingeniero en Electrónica en el Instituto Tecnológico de Piedras Negras, y actualmente está cursando la Maestría en Ciencias en Electrónica en el Instituto Tecnológico de Celaya.

Dr. José Javier Díaz Carmona, obtuvo su título de Ingeniero en Electrónica en el Instituto Tecnológico de Celaya, posteriormente los títulos de Maestro y Doctor en Ciencias en Electrónica en el Instituto Nacional de Astrofísica Óptica y Electrónica, actualmente pertenece al sistema nacional de investigadores nivel 1 y se desempeña como docente e investigador en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.

Dr. Elías José. Juan Rodríguez Segura: obtuvo en 2003 el grado de Doctor en Ciencias en Ingeniería Electrónica en el Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET) en 2003, actualmente es profesor investigador de tiempo completo en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.

Dr. Agustín Ramírez Agundis, obtuvo sus títulos de Ingeniero en Eléctrica y Maestro en Ciencias en la Universidad de Guanajuato, y su título de Doctor en Ciencias en Electrónica en la Universidad Politécnica de Valencia España, actualmente se desempeña como docente e investigador en el departamento de Ing. Electrónica del Instituto Tecnológico de Celaya.

Alejandro Espinosa Calderón recibió su título de Doctorado de la Universidad Autónoma de Querétaro, en Querétaro, Qro., Mex., en 2012, de donde fue titulado con honores, también recibió la medalla al mérito académico de la UAQ, y ganó el premio a la mejor tesis doctoral de la facultad de ingeniería de la UAQ. Ha participado en eventos científicos internacionales en EEUA, Italia y China. Ha recibido premios internacionales de la OSA y USAID, ambos de EEUA. Desde 2013 trabaja en el Dpto. de Ing. Eléctrica y Electrónica del ITCelaya. Sus áreas de interés incluyen: aplicaciones de instrumentación electrónica y procesamiento en sistemas digitales FPGA/microprocesadores.