Prototipo de supervisión del control por medio de una aplicación WEB y adquisición de datos de un sistema domótico

Gilberto López, José Solano y Mario Cruz

G. López, J. Solano y M. Cruz
Universidad Tecnológica de León, Blvd. Universidad Tecnológica 225, San Carlos, 37670 León, Guanajuato
gpadilla@utleon.edu.mx

M. Ramos., V.Aguilera., (eds.) .Ciencias de la Ingeniería y Tecnología, Handbook -©ECORFAN- Valle de Santiago, Guanajuato, 2014.

Abstract

In this paper the design of a portable data acquisition device with an 8-bit I/O port-microcontroller based on commonly used low-cost 16-bit MSP430 from Texas Instruments (TI) G Family Value is proposed. This proposal is divided into three parts: the first part concerns the design of serial communication layer and the configuration of the I/O data acquisition device. A second covering the design of a PC software interface that links to a data acquisition device using the evaluation board MSP- EXP430G2 from TI called "LaunchPad" writing data to MSSQL database server. It may also be of interest to mention that coding of programs described here becomes relevant around the programming languages C and C #, as well as development environments such as IAR Systems from IAR Workbench and Microsoft's Visual Studio. And finally the exploration and development of applications that make them the role of human-machine interfaces (HMI) to promote interaction between the user and the control system. There are closed-source applications (patented) designed specifically for the case, but this time the issue is addressed by exploiting the benefits of WEB applications, to be specific: those that can be developed with Microsoft's .NET.

9 Introducción

Los sistemas SCADA se utilizan ampliamente en la industria para supervisión del control y adquisición de Datos de los procesos industriales. Las empresas que son miembros de comités de normalización están dictando las tendencias en materia de las tecnologías de la información (TI) en general para el desarrollo de estos sistemas.

De hecho, están penetrando en los laboratorios de física experimental en el control de sistemas auxiliares, como la refrigeración, la ventilación, distribución energía eléctrica, etc. Los sistemas SCADA han hecho progresos sustanciales durante los últimos años en términos de funcionalidad, escalabilidad, rendimiento y apertura, de tal manera que son una alternativa en el desarrollo de la casa inteligente, o incluso en sistemas de control muy exigentes y complejos como los de la física experimental. (A. Daneels, W. Salter, 1999)

Justificación

Del sistema de control en su conjunto, la medición del proceso es una de las tareas más importantes. Esto se determina por el hecho de que la precisión de control es completamente dependiente de lo bien que trabaja la cadena de medición. Para medir el proceso, hoy en día se cuenta con un gran número de dispositivos que realizan tareas de adquisición de datos: tarjetas estándar o normalizadas de bus PCI para la PC, sus versions industriales y módulos para la automatización industrial que por lo general cuentan con RS- 485, CAN y otras interfaces.

Muy a menudo se presentan escenarios en los que es necesario medir datos en campo; donde no es posible el uso de equipo estándar o normalizado de adquisición de datos que requiere de entornos de operación muy restrictivos, como temperatura, infraestructura, humedad, etcétera. En estos casos una computadora portátil equipada con un dispositivo portátil de adquisición de datos puede ser muy valiosa.

En el mercado hay disponibles dispositivos equipados con conectividad USB 2.0, que pueden plenamente sustituir a las tarjetas PCI, pero el costo es por demás elevado y aun así se requiere de trabajo adicional para conformar un sistema que se pueda considerar como SCADA.

La contraparte sería la producción de un prototipo de un dispositivo de adquisición de datos con conexión USB de bajo costo y la integración de un sistema que permita realizar labores de supervisión del control y adquisición de datos como lo hace un sistema SCADA.

Antecedentes

Como base del prototipo se ha trabajado en una propuesta hecha en "Desarrollo libre de sistemas SCADA" en la que se establece un modelo simplificado de 4 capas tomando como referencia el modelo de Daneels y Salter, estas cuatro capas definidas como:

Tabla 9

4	HMI. Interfaz humano-maquina.
3	DB. Registro
2	Tx/Rx. Transmisión de datos
1	ADyC. Adquisición y control

(López Padilla, Cruz Alcaraz, Santos Pompa, Rivera Sandoval, Mata Santos, & Solano Medina, 2013)

9.1 Desarrollo

Objetivos:

- Construcción de un circuito digital que permite el control de dispositivos eléctricos a 110 Voltios.

- Programación de microcontroladores usando una tarjeta de evaluación Launchpad MSP-EXP430G2.

- Programación de un servidor SCADA que enlace la base de datos de MSSQL Server y el dispositivo de adquisición de datos.

- Programación de página WEB (HMI) para supervisión del control.

- Prueba de la comunicación inalámbrica desde un dispositivo móvil con la página WEB para controlar el equipo eléctrico.

Diseño del prototipo

Del modelo simplificado mencionado en "Desarrollo libre de sistemas SCADA", se seleccionó para este proyecto el uso de una API, que en este caso es "System.Data.SqlClient" de .NET, la cual vincula mediante una aplicación desarrollada en C# las capas ADyC y Tx/Rx y que se denominó "Servidor SCADA".

Esta integración produce dos módulos:

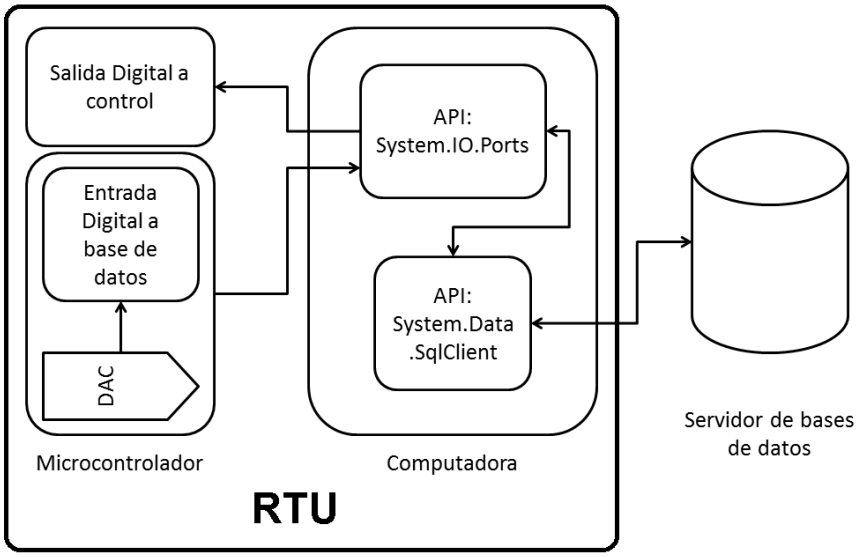
Adquisición de datos/Salida de control: E/S digital de 8 bits, con comunicación serie microcontrolador-PC.

Servidor SCADA: comunicación serie de 8 bits con conexión a base de datos, PC-microcontrolador.

La integración de estos módulos da como resultado la implementación de una Unidad de Transmisión Remota; o RTU, que envía y recibe datos digitales o analógicos a través de un microcontrolador (o varios) y los transmite a una computadora vía comunicación serie de 8 bits, como se muestra en la figura.

Dado que las capas deben trabajar de forma independiente, los módulos deben producirse de manera que la comunicación no "esclavice" a los dispositivos sino que estos se mantengan funcionando independientemente de si existen o no datos en las memorias intermedias (buffers) de comunicación serie tanto de la computadora como del microcontrolador. Estos problemas se solventaron haciendo uno de interrupciones (GPIO) en el microcontrolador y de threads en .NET. Se muestran a continuación los diagramas de flujo propuestos en las figuras.

Figura 9 Pantalla de inicio del software IAR Embedded WorkBench



Para evitar que el servidor SCADA se "cuelgue" mientras espera que una transmisión serie se complete, ".NET Framework" permite la subdivisión del proceso del sistema operativo en dos subprocesos ligeros administrados, llamados dominios de aplicación, representados por "System.AppDomain". Uno o varios subprocesos administrados (representados por "System.Threading.Thread") se pueden ejecutar en uno o cualquier número de dominios de aplicación dentro del mismo proceso administrado. El resultado es que un subproceso administrado puede moverse libremente entre dominios de aplicación dentro del mismo proceso administrado logrando la separación de las capas de Tx/Rx y BD

Figura 9.1 Diagrama de flujo del programa en microcontrolador

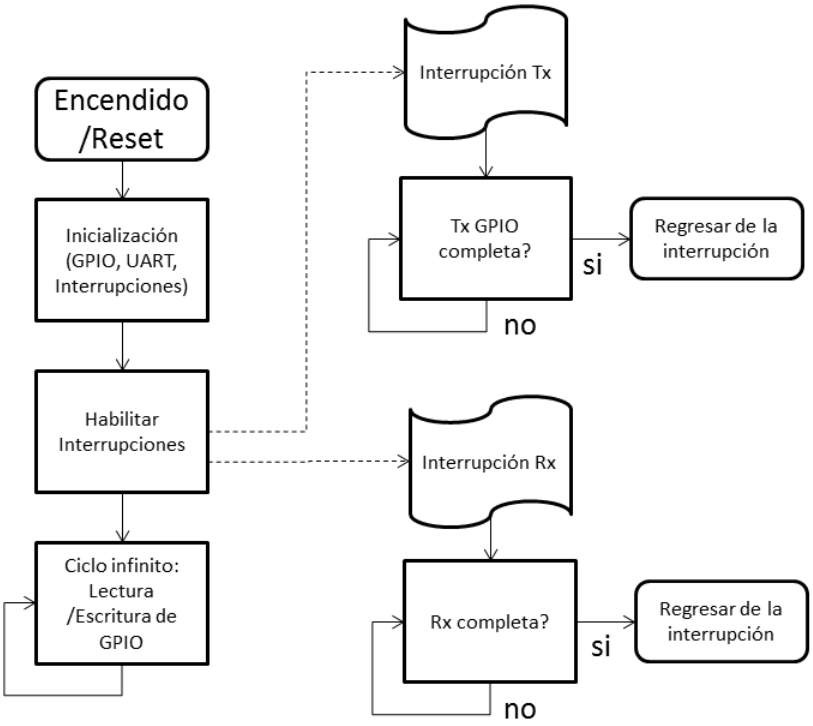
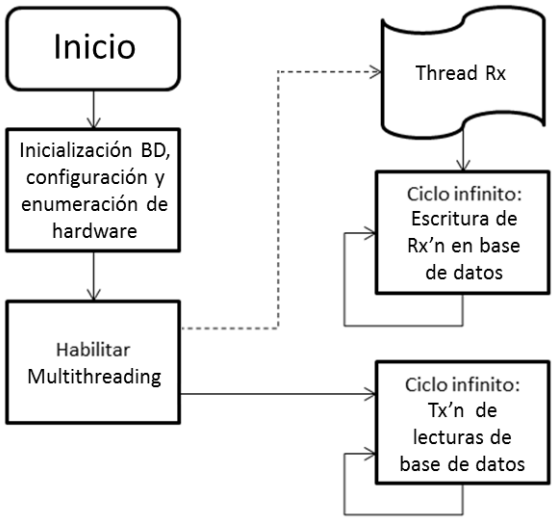


Figura 9.2 Diagrama de flujo del Servidor SCADA



Esquema del prototipo

El prototipo es presentado a continuación en la figura, consta de la tarjeta Launch Pad que está configurada de tal manera que al conectarse por medio de un cable USB hacia la computadora, establece una conexión con la Base de datos de SQL server instalada previamente en el equipo. Esta tarjeta recibe información de la base de datos y la convierte en la salida del circuito.

Para este caso se tienen 4 salidas de potencia independientes y cada una podría supervisar el control de un circuito eléctrico de 110 Voltios. Por otro lado la tarjeta Launch Pad lee y escribe en la Base de Datos y esta información actualiza una página WEB que está alojada en la misma computadora (pero que pudiera estar en cualquier servidor de la red) y que es accesible desde un dispositivo móvil, la actualización de datos se realiza mediante un servidor SCADA que se programó en Visual C#.

Figura 9.3 Diseño de prototipo: conexión

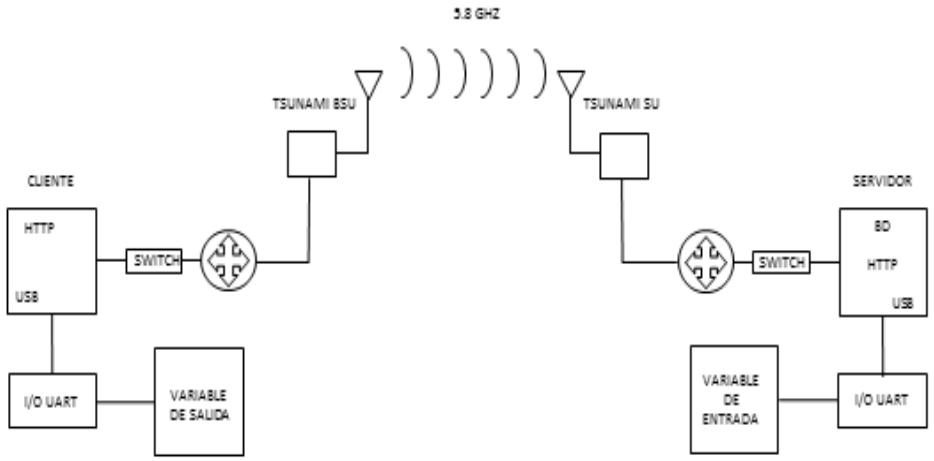
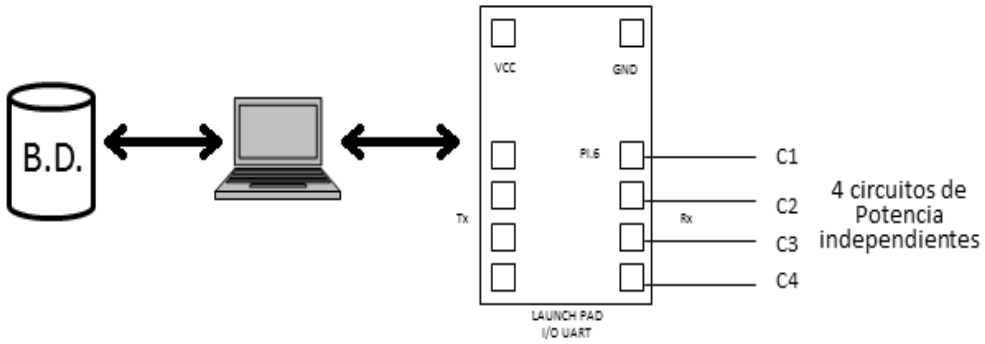


Figura 9.4 Diseño de prototipo: conexión (continuación)



Equipamiento

Para la implementación del diseño en el laboratorio de pruebas se utilizó el siguiente equipo:

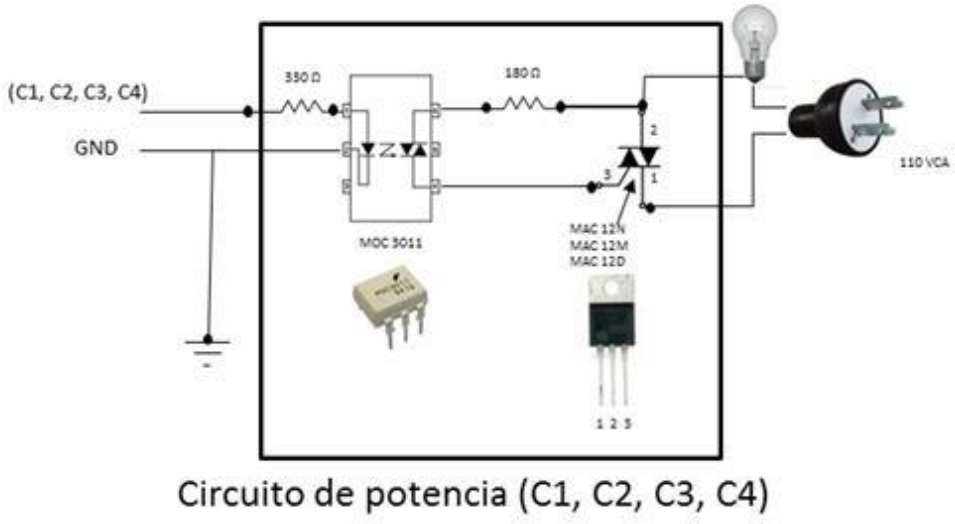
- Equipo inalámbrico
- Tsunami MP 8200 Estación Base de 300 Mbps con power injector.
- Tsunami MP 8200 Estación Suscriptor de 300 Mbps con power injector.
- Computadoras
- Computadora Core 2 Duo HP Compaq (2.66 Ghz), 4 Mbyte de memoria RAM con sistema operativo Windows 7.
- Interfaz Entrada y Salida
- Tarjeta de evaluación Launch Pad MSP-EXP430G2. Dispositivo Móvil
- Samsung Tablet GalaxyTab 3 16 Gb, WiFi 10.1

Ejecución del proyecto

Construcción de un circuito digital que permite el control de dispositivos eléctricos a 110 Voltios.

Etapa 1: Circuito digital para control de dispositivos eléctricos.

Figura 9.5 Diseño de prototipo de potencia



Se realiza la construcción de los cuatro circuitos en protoboard para hacer pruebas y una vez funcionando se pasan al armado en placas soldadas utilizando componente y material descrito en la figura.

Se utiliza un optoacoplador MOC3011 también llamado optoaislador o aislador acoplado ópticamente, es un dispositivo de emisión y recepción que funciona como un interruptor activado mediante la luz emitida por un diodo LED que satura un componente optoelectrónico, normalmente en forma de fototransistor o fototriac. De este modo se combinan en un solo dispositivo semiconductor, un fotoemisor y un fotorreceptor cuya conexión entre ambos es óptica. Estos elementos se encuentran dentro de un encapsulado que por lo general es del tipo DIP. Se suelen utilizar para aislar eléctricamente a dispositivos muy sensibles.

Otro de los componentes utilizados es un TRIAC, dispositivo semiconductor, de la familia de los tiristores. La diferencia con un tiristor convencional es que éste es unidireccional y el TRIAC es bidireccional.

De forma coloquial podría decirse que el TRIAC es un interruptor capaz de conmutar la corriente alterna. Su estructura interna se asemeja en cierto modo a la disposición que formarían dos SCR en direcciones opuestas.

Posee tres electrodos: A1, A2 (en este caso pierden la denominación de ánodo y cátodo) y puerta. El disparo del TRIAC se realiza aplicando una corriente al electrodo puerta.

Una vez armado los cuatro circuitos de potencia, son conectados a las cuatro salidas de datos de la tarjeta Launch Pad MSP-430 que es controlada por servidor SCADA residente en la computadora.

De esta manera cada salida de la tarjeta puede ser uno de dos estados 1 y 0. El 1 hace que el TRIAC se dispare dejando pasar la CA y encendiendo el foco. El 0 hace que el TRIAC no deje pasar la CA, apagando el foco.

El control de cada circuito de potencia es independiente, ajustándose a la salida de la tarjeta Launch Pad que es un número formado por 4 bits, almacenado en la base de datos.

Etapa 2: Programación de microcontroladores de la serie MSP-430

La tarjeta de evaluación Launchpad MSP-EXP430G2 es una herramienta de desarrollo y de evaluación para los dispositivos MSP430 de Texas Instruments. Está enfocada a la línea de dispositivos que ellos denominan como value line.

La tarjeta dispone de un socket de 20 pines que puede albergar uno de los dos microcontroladores de 16 bits de la familia MSP430 que vienen con el kit, dispone además de una conexión USB que permite descargar y depurar programas directamente en el hardware. Fuera de eso, solamente disponemos de dos botones (uno de ellos es de reset), un par de leds y unos headers (hembra/macho) para poder acceder a los pines del microcontrolador, por lo que el hardware específico para la aplicación habrá que implementarlo externamente.

Tiene un cierto parecido con la plataforma Arduino basada en un micro AVR. Sin embargo hay que recordar que el microcontrolador ATMEGA328 dispone de una memoria RAM de 2KB, mientras que los dispositivos MSP430 que vienen incluidos apenas alcanzan los 128 bytes de RAM, aun así, existen microcontroladores MSP430 bastante poderosos y esta tarjeta constituye un punto de entrada excelente para el desarrollo con microcontroladores de TI.

La tarjeta se conecta a la computadora por un puerto USB. Será configurada de tal manera que pueda consultar datos de una base de datos SQL instalada en dicha computadora. Los datos que puede leer y escribir son números decimales que van desde 0 y hasta 255 escritos en un campo de la base de datos en particular.

Estos números son convertidos en un número binario de 8 bits, de los cuales solo 4, seleccionables por medio de programación, son utilizados para controlar los 4 circuitos de potencia. El Launchpad en conjunto con el MSP430G2553 forma el dispositivo de adquisición de datos. Este aparato comunica una palabra de 8 bits (ADQ) que es leída o escrita a un puerto de entrada-salida configurado de esta manera:

Tabla 9.1

GPIO del MSP430	P1.5	P2.0	P2.1	P2.2	P1.6	P2.5	P2.4	P2.3
Palabra ADQ	BIT0	BIT1	BIT2	BIT3	BIT4	BIT5	BIT6	BIT7

Los puertos, aunque son de propósito general (E/S), deben ser programados específicamente dependiendo si se van a usar como entrada o como salida, antes de integrarse al sistema.

Es importante que la comunicación serie del dispositivo se haga usando el sistema de interrupciones del microcontrolador (GPIO en el MSP430), de lo contrario acciones adicionales del microcontroador quedarían en espera hasta que la cuminicación haya terminado, esto es para respetar el trabajo independiente de cada capa del modelo.

El software utilizado para la programación de los microcontroladores de la familia MSP430 es IAR Embedded WorkBench, en él los parámetros que se deben tomar en cuenta para la programación son el número del microntrolador que debe ser el mismo que el MSP430 usado y en el programa que debe ser configurado en FET Debugger para grabar el programa y no dejarlo solo como simulación.

Después se carga el programa que contiene una extensión de lenguaje C. Las siguientes imágenes muestran la tarjeta MSP430 y parte del código en lenguaje C necesario para la configuración del microcontrolador por medio de la tarjeta.

Figura 9.6 Tarjeta Launch Pad MSP430 como se muestra en el servidor SCADA

Figura 9.7 Código del programa main.c

```
include "msp430g2553.h"
include "uart_conf.h"
include "uart.h"
include "uart.c"
include "IO.h"
include "IO.c"

//-------------------------------------------------------------------------
// main()
//-------------------------------------------------------------------------
void main(void)

 unsigned char CEP1=0;
 unsigned char CEP2=0;
   init_msp430();

   // Predeterminadamente todos los puertos son de entrada, indique aqui cuales
   // bits son de salida.
   // Equvalencia:
   //              Palabra ADQ   GPIO del MSP430
   //                bit7              P1.5  -|ultimos 4 pines del lado izquierdo del launchpad
   //                bit6              P2.0   |
   //                bit5              P2.1   |
   //                bit4              P2.2  -|
   //                bit3              P1.6  -|Ultimos 4 pines del lado derecho del launchpad
   //                bit2              P2.5   |
   //                bit1              P2.4   |
   //                bit0              P2.3  -|

   // Ejemplo:
   conf_salida_puertos(bit3+bit2+bit1+bit0); //Habilita como salda la parte alta de la palabra ADQ, (DCE1)
   //conf_salida_puertos(bit7+bit6+bit5+bit4); //Habilita como salida la parte baja de la palabra ADQ, (DCE2


   __enable_interrupt(); //Habilita las interrupciones. No mofificar.

   TimerA_UART_init();                  // Start Timer_A UART.

   for (;;)
   {
       // Espera a rebir un caracter.
       // Wait for incoming character
       __bis_SR_register(LPM0_bits);
```

```
      //Escribe el byte recibido en los puertos
      actualiza_salida_puertos();

      //variable no volatil que recibe la configuracion de entrada
      //CEP1=P1IN&~P1DIR3;  //Deshabilitar en modo eco
      CEP1=~P1DIR;
      CEP1&=P1IN;
      CEP2=~P2DIR;
      CEP2&=P2IN;
      //Experimental, envia la configuracion de entrada al software de pruebas
        //TimerA_UART_tx(~CEP); //No habilitar

      //Transmision Tx
      //TimerA_UART_tx(rxBuffer);   //Modo eco
      TimerA_UART_tx(codifica_salida_puertos(CEP1,CEP2));         //Modo Lectura

  }
```

Figura 9.8 Código de la configuración de la salida de puertos IO.c

```
void conf_salida_puertos(unsigned int byte)
{

  P1DIR |=byte;
  P2DIR |=byte>>8;
}

unsigned char codifica_salida_puertos(unsigned char byte1, unsigned char byte2
{
        unsigned char salida=0x0;
        if (byte1 & 0x20) salida |= 0x01;    // P1.5
        if (byte2 & 0x01) salida |= 0x02;    // P2.0
        if (byte2 & 0x02) salida |= 0x04;    // P2.1
        if (byte2 & 0x04) salida |= 0x08;    // P2.2
        if (byte1 & 0x40) salida |= 0x10;    // P1.6
        if (byte2 & 0x20) salida |= 0x20;    // P2.5
        if (byte2 & 0x10) salida |= 0x40;    // P2.4
        if (byte2 & 0x08) salida |= 0x80;    // P2.3
        return salida;
}

void actualiza_salida_puertos(void)
{
        if (rxBuffer & 0x01) P1OUT |= 0x20; else P1OUT &= ~0x20;    // P1.5
        if (rxBuffer & 0x02) P2OUT |= 0x01; else P2OUT &= ~0x01;    // P2.0
        if (rxBuffer & 0x04) P2OUT |= 0x02; else P2OUT &= ~0x02;    // P2.1
```

Etapa 3: Instalación de manejador de base de datos SQL Server y conexión con la tarjeta MSP-430

Para este proyecto se utilizó SQLEXPRESSEDITION 2005 Y SQL MANAGEMENT ESTUDIO 2008 que pueden ser descargados de www.microsoft.com.

Se instala SQLEXPRESS 2005 tomando en cuenta algunos parámetros como que los componentes de los servicios de base de datos sean accesibles en una red TCP/IP, que la autenticación sea mixta y una contraseña segura para él usuario por defecto “sa”.

Se recomienda la instalación de SQL MANAGEMENT ESTUDIO 2008 para la administración de bases de datos. Durante la instalación se selecciona una nueva instalación independiente de SQL server. Se seleccionan también las características de SQL Management a instalar tales como las herramientas de administración básica y el SDK de conectividad de cliente SQL.

Una vez instalados los programas necesarios se debe crear la base de datos para realizar la conexión con la tarjeta MSP430. Se crea la tabla con el campo “dato” donde se almacenan los datos en forma decimal, entrando al software SQL Server 2008. Una vez creada la tabla con sus campos, se va a configurar la activación de los protocolos TCP/IP.

Figura 9.9 Hoja de verificación de SQL EXPRESS 2005 durante la instalación

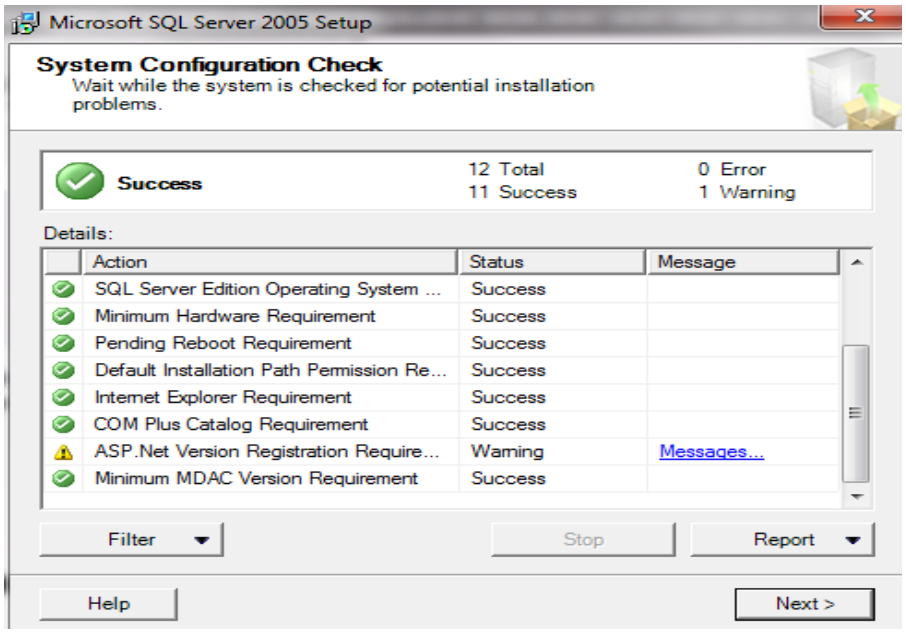
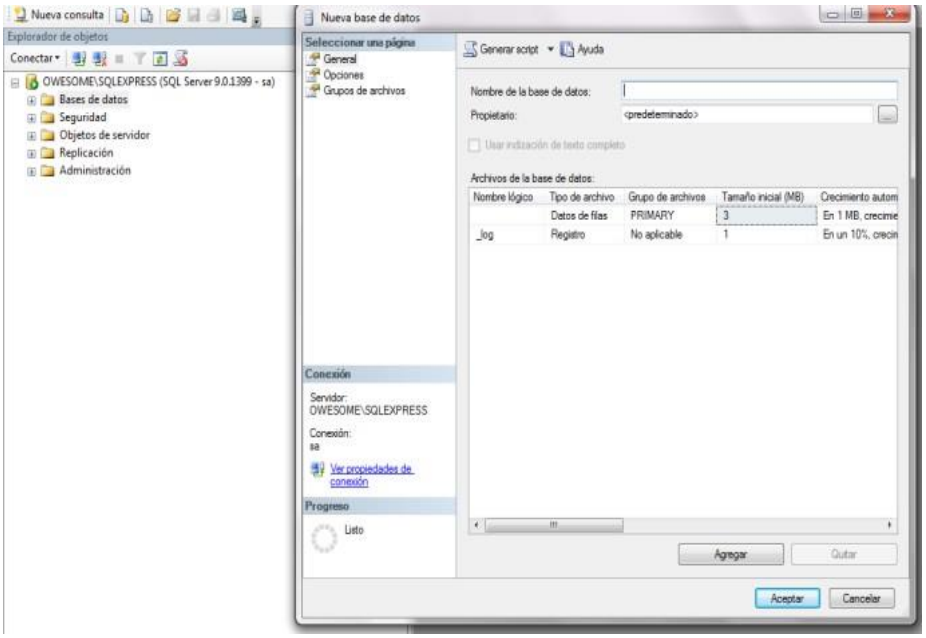


Figura 9.10 Creación de la base de datos



Una vez creada la base de datos y creadas las tablas con sus campos, se realizó la conexión entre la tarjeta MSP430 configurada previamente con la Base de Datos, esto por medio de un programa que se denominó “Servidor SCADA”, que permite probar el funcionamiento de la tarjeta directamente desde el programa o si se requiere se hace la conexión con la base de datos para que la tarjeta pueda consultar los campos que contienen un número decimal y convertirlo a número binario que de valor a las 4 salidas de cada circuito de potencia.

Figura 9.11 Interfaz de conexión entre la tarjeta y la base de datos

Etapa 4: Creación de página WEB con conexión a la base de datos para supervisión del control

Se instaló un servidor de HTTP donde se aloja la página que hace la conexión a la base de datos. El servidor que se uso es IIS con extensiones de .NET, instalado en la misma computadora donde está conectada la tarjeta MSP430 y donde está instalada la base de datos de SQL Server. La programación de la página Web se realiza con el programa Visual Studio para WEB de Microsoft. Esta página muestra los cuatro circuitos de potencia físicamente y permite que el usuario con solo hacer un clic pueda controlar el encendido de cada uno.

La siguiente es una figura que muestra parte del código ejecutable de la página Web, específicamente: se configura la cadena de conexión a la base de datos y se ponen en cero todos los elementos del formulario del usuario. Es un código muy sencillo que tiene por objeto mostrar que la supervisión del control y la adquisición de datos son factibles aún en términos de uso muy básicos. De hecho la simpleza reside en que solo un evento desencadena una lectura, y este es un clic en el botón "submit" del formulario; al hacerlo se convierte el valor booleano de los "checkbox" del formulario a un valor decimal de tamaño byte (char para el microcontrolador), el cual es escrito en, o leído desde la tabla en la que el dispositivo de adquisición de datos lee y escribe para actualizar la salida de sus puertos E/S de acuerdo a la siguiente tabla:

Puertos MSP430	P1.5	P2.0	P2.1	P2.2	P1.6	P2.5	P2.4	P2.3
Valor Byte	1	2	4	8	16	32	64	128

Las 256 combinaciones posibles del Byte se consiguen con una simple tabla de numeración binaria de 8 bits y su correspondiente conversión decimal.

Figura 9.12 Código de conexión a la BD

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Drawing;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        CheckBoxList2.Enabled = false;
        Label1.Text = "No se han transmitido datos al microcontrolador";
        Label2.Text = "No se han recibido datos desde el microcontrolador";
        SqlDataSource1.ConnectionString = "Data Source=CROOKEDPC\\SQLEXPRESS;"+
                                          "Initial Catalog=serialDB;"+
                                          "User ID=gilberto;"+
                                          "Password=1234";
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        byte registro_tx = 0;
        byte registro_rx = 0;
        Label1.Text = "Valor transmitido al microcontrolador: ";
        Label2.Text = "Valor recibido desde el microcontrolador: ";
        foreach (ListItem item in CheckBoxList1.Items)
        {
            if (item.Selected)
            {
                registro_tx += Convert.ToByte(item.Value);
            }
        }
        Label1.Text += registro_tx.ToString();
        SqlDataSource1.UpdateCommand = "UPDATE dce1 SET dato=" + Convert.ToString(registro_tx);
        SqlDataSource1.Update();
        SqlDataSource1.SelectCommand = "SELECT dato FROM dce2";
        DataView dv = (DataView)SqlDataSource1.Select(DataSourceSelectArguments.Empty);
        registro_rx=Convert.ToByte(dv.Table.Rows[0][0]);
        Label2.Text += registro_rx.ToString();
        foreach (ListItem item in CheckBoxList2.Items)
        {
            item.Selected =Convert.ToBoolean(registro_rx % 2);
            registro_rx =Convert.ToByte(registro_rx / 2);
        }
    }
}
```

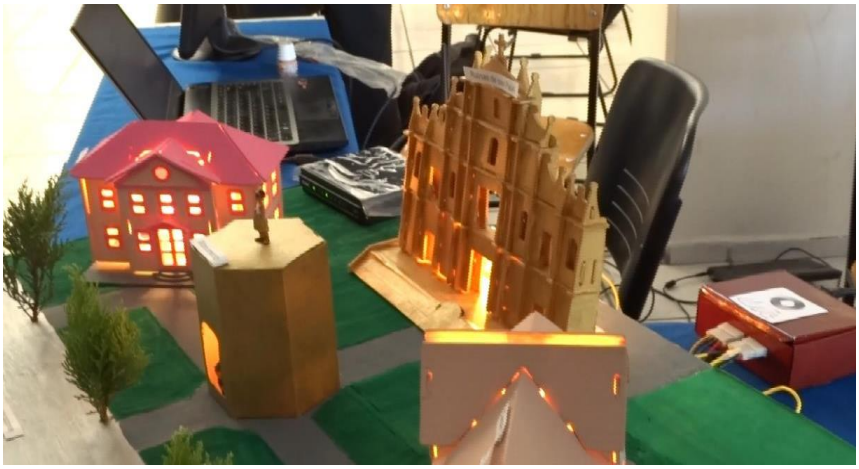
Figura 9.13 Página Web del control automatizado

Esta página Web muestra las opciones de encender y apagar 4 focos en una maqueta que fue fabricada para mostrar el funcionamiento del prototipo. Escribe un número decimal en la BD y la tarjeta MSP430 que contiene el microcontrolador que lee desde la BD y lo convierte en una salida binaria de 4 bits, donde cada bit es utilizado como la entrada de cada uno de los circuitos de potencia haciendo que encienda el foco si el valor del bit es 1 y apagándolo si el valor del bit es 0.

Etapa 5: Prueba de la comunicación inalámbrica desde un dispositivo móvil con la página WEB para controlar el equipo eléctrico.

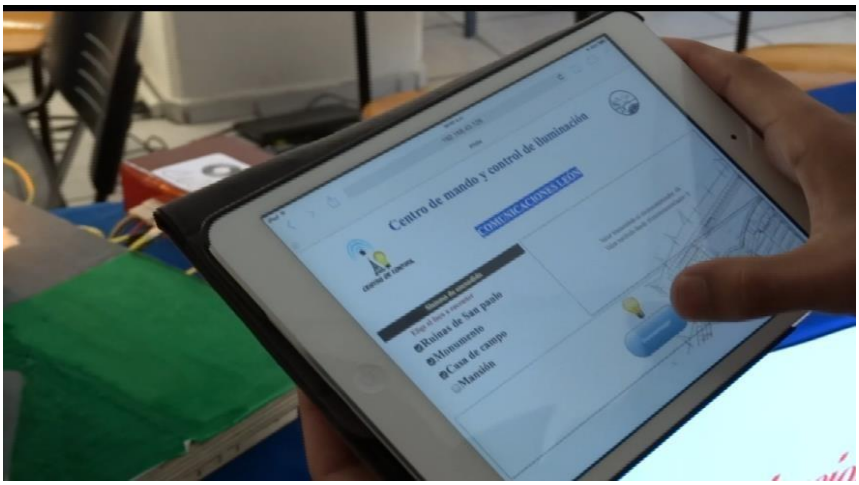
Para realizar las pruebas del prototipo se realizó una maqueta que muestra la utilidad de la aplicación, donde por medio de una Tablet se realiza una conexión inalámbrica y se accede a la página Web del servidor.

Figura 9.14 Maqueta de una ciudad



Cada edificio contiene un foco que se apaga o se prende según lo que se solicite des la página Web. La siguiente figura muestra una Tablet con la página Web y se hacen las pruebas de control.

Figura 9.15 Página Web que controla la iluminación de una ciudad



9.2 Conclusiones

El proyecto se realizó exitosamente de acuerdo a los objetivos acordados, además de concluir con cada una de las partes del proyecto en tiempo y forma. El proyecto ha contribuido a desarrollar un sistema confiable, fácil de usar y accesible en cuestiones económicas.

Además cabe destacar que el sistema de domótica se integró exitosamente con el sistema de iluminación de cada una de las estructuras, lo que permitió manipular el control del sistema eléctrico mediante cualquier dispositivo móvil de manera remota.
Permite simplificar las tareas diarias de las personas permitiendo mejorar su estilo de vida y sobre todo mejorar las condiciones de operación de los dispositivos con los que contamos.

La domótica es sin duda una nueva tecnología que tiene gran potencial no solo en hogares si no en todos los sectores productivos ya que nos permite tener control, seguridad y monitorización de los bienes por medio de servicios de Internet.

Referencias

A. Daneels, W. Salter. (1999). What is SCADA? International Conference on Accelerator and Large Experimental Physics Control Systems (pág. 5). Trieste, Italy: CERN.

López Padilla, G., Cruz Alcaraz, M., Santos Pompa, D. M., Rivera Sandoval, C. A., Mata Santos, M. V., & Solano Medina, J. J. (2013). Desarrollo libre de sistemas SCADA. Congreso Internacional de Investigación ACADEMIA JOURNALS Celaya 2013. 5, págs. 1896-1900. Celaya, México: ACADEMIA JOURNALS.

Microsoft. (s.f.). Threading. Recuperado el 21 de Junio de 2012, de MSDN: http://msdn.microsoft.com/en-us/library/aa720724(v=vs.71).aspx

(3 de Septiembre 1999) Definiciones extraídas de la asociación CEDOM Asociación Española de Domótica. Recuperado el 22 de marzo de 2014, de http://www.domodesk.com/que-es-domotica

Andrew S. (2003). Redes de computadoras (4ª edición). México: Pearson Educación.
Boylestad Robert (2009).Electrónica: Teoría de circuitos y dispositivos electrónicos. México: PEARSON.

Cruz, Claudio. (2010). Domótica. Recuperado EL 25 de marzo de 2014, de http://www.arqhys.com/arquitectura/domotica.html

Milan Verle (2008). Microcontroladores. Recuperado el 22 de marzo de 2014 de http://www.mikroe.com/en/books/picbook/picbook.htm