Implementación de software HMI base para controladores lógicos programables DL06 y DL350

G. A. Herrejón P.**, A. C. Segundo S. *
*,**Instituto Tecnológico de Estudios Superiores de Zamora, **Instituto Tecnológico Superior de Ciudad Hidalgo

***Resumen*__ Se discute la enseñanza y utilización de software de interfaz hombre máquina (HMI) para controladores lógicos programables (PLC). Se abordan las bondades y problemáticas de la implementación y utilización de estas interfaces en contraste con las comerciales, tradicionalmente enseñadas a partir de una problemática de automatización real y sencilla.**

I. INTRODUCCIÓN.

La funcionalidad de los controladores lógicos programables (PLC´s) ha evolucionado a lo largo de los años para incluir capacidades más allá del control típico con relevadores: el control de movimiento sofisticado, control de procesos, sistemas de distribución de control y establecimiento de redes complejas han sido añadidos a la lista de funciones del PLC [1].

Dentro de la industria automotriz, aeroespacial, de alimentos, industria de generación de energía como la eléctrica, petrolera, etc, se emplean PLC´s, como base de sus sistemas de automatización, sistemas que pueden ir desde módulos aislados de control, hasta redes de automatización como (Ethernet), red de bus de campo (Profibus), red de dispositivos (Modbus). Tanto los sistemas aislados de control y redes, pueden ser dotados de dispositivos de visualización, que ofrecen una interfaz de medición y mando de instrucciones amigable al ser humano. Estos pueden ser pantallas LCD con botoneras, pantallas táctiles [2] "Interfaz hombre máquina (HMI)"o interfaces para PC [2] "software HMI" En todos los casos de visualización, va implícito un costo relacionado al hardware y software necesario para incorporación de esta característica.
Por otro lado la enseñanza de sistemas de automatización basados en PLC, tradicionalmente ha contemplado el estudio del software de los fabricantes, para la visualización y control.

RVP-AI/2012 - SUB-01 PONENCIA RECOMENDADA POR EL **COMITÉ DE SUBESTACIONES** DEL **CAPÍTULO DE POTENCIA** DEL **IEEE SECCIÓN MÉXICO** Y PRESENTADA EN LA **REUNION INTERNACIONAL DE VERANO, RVP-AI/2012**, ACAPULCO GRO., DEL 8 AL 14 DE JULIO DEL 2012.

II. CARACTERIZACIÓN DEL PROBLEMA

A. Opción Comercial.

Se pretende tener control de accesos, salidas y energización eléctrica de bodegas, controlado todo desde una PC, y guardar todo ello en un historial de eventos.
Una de muchas opciones es la utilización de PLCs, en este caso, por ejemplo por ejemplo un FC 34 que cuenta con la potencia necesaria para ello y crear un historial de lo los eventos registrados y realizados por el PLC, sin embargo esto hace que se necesite también de una licencia de Excel, aparte del software para la HMI propia de la marca. En cualquier otro caso, pasaría algo similar, así todas las marcas de controladores ofrecen estas capacidades, pero ello implica la dependencia en algunos casos de su hardware HMI, de su software HMI y sus licencias, así como acotar al ingeniero a ser un usuario de las herramientas de estos paquetes.
Es una realidad que las aplicaciones de los fabricantes facilitan el trabajo y por lo tanto reducen el tiempo de implementación, sin embargo si se tiene el conocimiento necesario de programación y algunas herramientas básicas ya hechas, los tiempos pueden empatarse. Y si a esto se suma que se requiere crear diferentes cuentas con diferentes permisos, no es descabellado el pensar en el diseño de una interfaz propia.

B. Opción Propia

Se decidió profundizar en los protocolos propios para PLC, esto con la intención de aprender a escribir en sus localidades de memoria desde un programa propio hecho por nosotros en Visual C#, o bien en algún otro como visual Basic etc.
Esto nos abre a la posibilidad de contar con un entorno visual, capaz de enviar y recibir datos desde cualquier PC. Esos datos son las tramas correspondientes al protocolo propio del controlador.
Por lo tanto se debe estudiar el protocolo adecuado para el dispositivo de control a utilizar, en este caso se

utiliza un DL 06, se eligió este PLC, por las siguientes razones.
1.- Su relación precio-capacidades.

Algunos PLC`s cuentan con modelos de bajo costo y con capacidades semejantes a las deseadas, sin embargo con la mitad o ¾ de las salidas booleanas con las que el DL06 cuenta y carecen de la flexibilidad de incorporar entradas analógicas [5], como el caso del FC34 [4], que en precios es competidor del DL06.
Hablando de otros PLC´s no se tiene avance sobre sus protocolos.

2.- Se cuenta con avance en su protocolo.

3.- Ya se realizo un software HMI para un PLC DL350 , trabajar con un DL06 permitirá indagar más en el protocolo en puntos como.

a) Respuesta de las tramas de direccionamiento calculadas para un DL350, en un DL06.

Se observo que en un DL06 para las localidades de entradas virtuales C80 y C90 no se permite escritura.
La trama calculada para la localidad que corresponde a la entrada virtual C60 de un DL350, escribe sobre las localidades C60 y C70 de un DL06.
Esto ocurre cada 100 localidades de entrada es decir si se direcciona C160 esta escribirá sobre ella y sobre la del C170 y así consecutivamente.

I.- Utilización del Protocolo.

A continuación se explica la secuencia que se requiere para direccionar una entrada virtual en el PLC, desde la aplicación hecha en visual C#.

1. Se manda una primer trama que sirve para indicarle al PLC que queremos escribir alguna entrada en el, la trama contiene 3 bytes.

Trama 1		
4E	22	05

2. El PLC contestara con una trama de 3 bytes también donde solo cambia el contenido del último byte si este byte es 06 en hexadecimal la respuesta es afirmativa y se puede continuar con el proceso, de no ser así se deberá volver a mandar la primer trama, una cantidad de veces definida por el programador ya que después de cierto número de veces de intento puede desplegarse un mensaje de error en el puerto, esto último es elección del diseñador.

Trama de respuesta		
4E	22	06

3. De ser afirmativa la respuesta. Se procede a mandar la trama número 2 que contiene 17 bytes, en esta trama se direcciona la entrada virtual del PLC donde se quiere mandar un cero o un uno lógico. Son tres bytes de estos diecisiete los que deben calcularse,(bytes en color gris) para cada entrada virtual del PLC donde se quiera escribir.

Trama 2															
01	30	32	38	31	34	31	39	42	30	30	30	32	30	30	17	05

4. Si el PLC responde un byte con un contenido hexadecimal de 06, indica que a direccionado correctamente y está listo para el paso siguiente.

06

De no ser así el proceso debe empezarse desde la trama 1.

5. Se manda la trama número 3, es la que contiene el dato que se va a escribir en la entrada virtual. Esta consta de 5 bytes en hexadecimal de los cuales solo cambian dos. FF y FF para mandar un uno lógico a esa entrada, 00 y 00 para escribir un cero.

Trama 3				
02	FF	FF	03	00

6. Si el PLC pudo escribir el dato en la localidad de memoria indicada responderá un byte con un contenido hexadecimal 06, aquí solo resta indicar al controlador, que ha finalizado su utilización.

06

7. Para terminar el proceso de escritura y lectura con el PLC. Se manda un byte con un dato hexadecimal 04.

Trama 4
04

El programa implementado en visual C# debe ser capaz de realizar la secuencia explicada en los siete puntos antes mencionados.
. El Siguiente diagrama de flujo indica la forma de trabajo con el protocolo para el PLC propuesto.
En la Figura 1 se puede ver el diagrama de flujo que explica el procedimiento antes explicado.

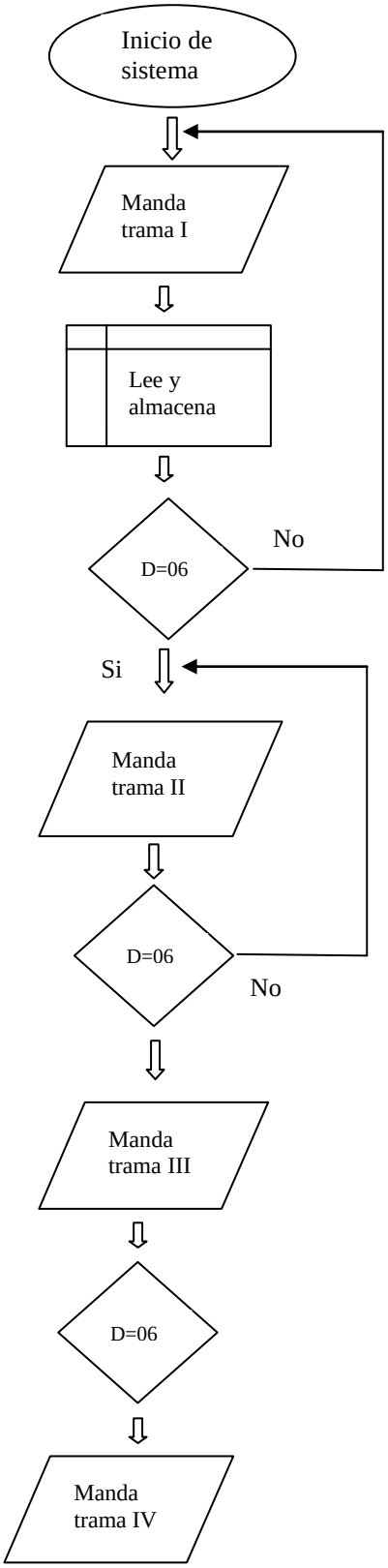


Figura 1. Diagrama de Flujo.

Después de implementar el algoritmo en C#, debe programarse el PLC, es decir colocar las entradas virtuales **C** o **Control relays C** donde la aplicación en C# direccionara y escribirá el dato lógico a través de las tramas del protocolo.

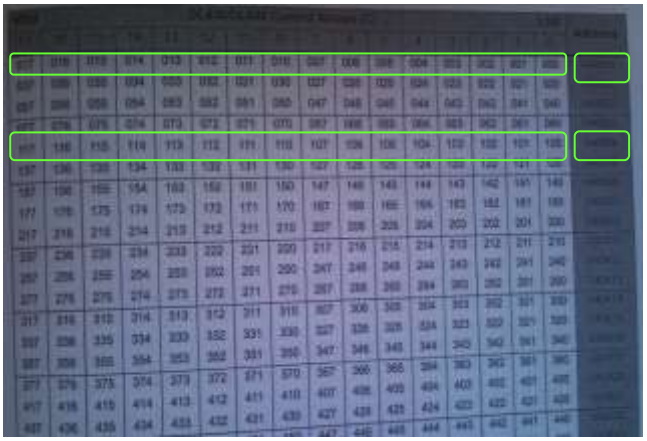
El programa en el PLC se realiza con el software propio de la marca [5], esto constituye una contradicción en lo propuesto en este articulo, ¿para qué escribir una aplicación propia en algún lenguaje de programación si se debe utilizar el software de la marca? En realidad dicha contradicción no existe, ya que la marca ofrece una versión prueba sin costo, limitada en el número de líneas de código, más que suficiente para programar las control Relays. Y la utilización del Software de la marca por si mismo no aliviaría la carencia de la HMI.

Para ello es necesario saber direccionar estas entradas virtuales (C), no es suficiente colocar entradas C1, C2, C3, etc. Ya que el mapa de Bits del controlador contiene direcciones que pueden manipular hasta 18 Bits, si no se direcciona adecuadamente pueden encender todas, muchas o ninguna de las salidas. Y es de interés tener control sobre cada una de las salidas

II. Direccionamiento en el PLC.

En la tabla siguiente se muestra el mapa de Bits de un DL430 o DL440 que da un punto de partida para direccionar las entradas virtuales del PLC.

Tabla 1. Mapa de Bits para un DL405



De la tabla anterior se puede ver que es un error direccionar consecutivamente las entradas virtuales, ya que encenderían todas las salidas del PLC con hacer referencia a uno solo de los control Relays, con la trama calculada, ya que dicha trama tiene como rango de control hasta 18 bits. Por lo tanto se debe escribir el programa en el PLC con direccionamientos separados en más de 18 unidades.

Por ejemplo:

Sí se tienen varios Control Relays y con cada uno de ellos se pretende excitar una bobina (salida) en forma independiente una de la otra. Direccionar como se muestra en la Figura 2 reportará como resultado que

al mandar el juego de tramas correspondientes al primer control Relay C0, los Control Relay del C1 al C5 también ejecuten la instrucción de cambio de estado lógico, provocando así que las salidas de Y0 a Y5 del PLC sean habilitadas o puestas en alto lógico.

Figura 2 Control Relays de C0 a C5 con salidas Y0 a Y5.

Si se atiende la Tabla 1. Y se observan los rectángulos en color verde, se puede ver que al mandar la trama que corresponde a la dirección V40600, los Control Relays desde el 000 hasta el 017 cambiarán su estado lógico.

No se debe olvidar que las tramas calculadas corresponden a un DL305 y el mapa de Bits con el que se cuenta es para un PLC DL430 o un DL440, así que el mapa de Bits puede cambiar para el DL06 ya que tiene menos recursos de memoria y el direccionamiento de la trama 3 puede también cambiar al ser un PLC con 18 Salidas fijas y las tramas pertenecen a un PLC de módulos I/O intercambiables.
Puede parecer muy complicado el direccionar a un PLC de esta forma, si se compara por ejemplo con direccionamientos para entradas virtuales o enlaces internos de señal desde software comercial bastaría con escribir E10.0 o A10.3 por ejemplo [3], en esta misma sección se verá que esto solo es posible con el software del fabricante.

Se procedió entonces a probar y efectivamente existió un ligero cambio entre la trama calculada y su relación con la localidad de memoria a escribir. Este cambio básicamente es en un par de localidades de memoria donde no se puede escribir las correspondientes a C80 y C90, así como la trama relacionada a la localidad C60 escribe sobre dos localidades en lugar de solo una la C60 y C70 y se repite esto cada 100 Control Relays. Aprendido esto la aplicación base de la interfaz de software HMI es completamente funcional.
En las siguientes Figuras, se puede apreciar las únicas opciones con que cuenta el programa prueba y base de la aplicación del software HMI en diseño. De esta interfaz no se genero un ejecutable, para hacer rápido las ediciones necesarias en el código.

Figura 3. Interfaz de base

Figura 4, Interfaz basé para el software HMI

En la Figura 3 puede verse la base de la aplicación HMI que se encuentra en diseño y se puede observar también la salida del PLC que se mando excitar mediante dicha aplicación.
La base del software se encuentra desarrollada en Visual C#, pero ¿qué pasa con el historial de eventos de entrada, procesamiento del PLC así como de sus salidas?
Se propone trabajar con My SQL para la base de datos donde se registrará dicho historial, la recomendación de utilizar My SQL es porque es libre y sus instrucciones prácticamente igual al SQL, que normalmente se enseña en las ingenierías de sistemas o licenciaturas en informática o en Ingeniería en Tecnologías de la información y Comunicación.
Un ejemplo de la funcionalidad de my SQL se puede vivir en la página de Cb Televisión.com.mx, la base de datos de esa página se encuentra hecha en My SQL.

Se creó un ejecutable de una modificación de la base de la aplicación como se puede apreciar en la Figura 4, esta modificación se hiso con el fin de sumarle flexibilidad al programa.

Figura 4. Primera pantalla de la aplicación ya modificada.

En la Figura 4 se nota el cambio del programa de prueba, se ofrecen las opciones.
1.- Abrir
2.- Salir
En la Figura 5 puede verse la segunda pantalla al optar por la opción abrir.

Figura 5. Ventana de apertura de proyecto.

En esta ventana aparecen las opciones:

1.- abrir puerto.
2.- cerrar puerto
3.- trama: sirve para invocar una nueva pantalla.
4.- enciende o mandar datos y apagar.
5.- medición del tiempo que se emplea al mandar las tramas. Y una barra que muestra el envió de datos

En la Figura 6 se ve otra ventana, esta es para traer tramas de un archivo de Excel.

Figura 6. Ventana de importación de tramas.

En La Figura 7 las salidas activadas del PLC que fueron activadas desde la Interfaz.

Figura 7 Interfaz en comunicación con el PLC.

Las ventanas de la aplicación desde donde se energizaron las salidas del PLC se ven en la Figura 8.

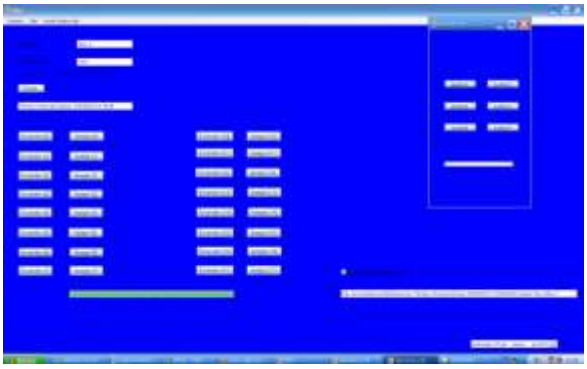


Figura 8. Ventanas de controles de de la interfaz diseñada.

Y contiene los siguientes controles.
1.- Controles de encendido y apagado, de todas las salidas de las que dispone el PLC tanto en forma independiente como por pares. .
2.- Entradas de Usuarios y Contraseña, para tener un reporte de la fecha, hora y que usuario opero el sistema, desde la parte de controles de la interfaz, esos datos solo los puede ver el administrador.
3.- Se incorpora un calendario con el efecto de poder llamar las operaciones del sistema en fechas distintas (esta etapa se encuentra en proceso).

III. Propuesta de enseñanza aprendizaje de PLC´s en ingeniería.

Como se ha argumentado, los proyectos de automatización pueden ser implementados utilizando el software del fabricante, normalmente es lo que se enseña en nivel ingeniería, y una razón poderosa para ello es que mucha industria adquiere sistemas de automatización y los servicios de instalación y puesta en marcha a proveedores que al paso del tiempo se han consolidado como especialistas en diferentes tecnologías, de distintas marcas y un dato importante, es que en México se encuentran pocos centros de desarrollo de ingeniería de este tipo e integración de la mima, Sin embargo y en contraste con esto existen empresas de automatización y control etc. que diseñan su propio software de visualización para el

hardware, aun que este último no sea fabricado por ellos.
A su vez algunos integradores más pequeños hacen lo mismo, ya que esto constituye una forma de generar empleo desde la pequeña industria, al ser dueños del software diseñado y poder así implementarlo y venderlo a la medida de las necesidades y capacidades del cliente, tener una ganancia post venta en el mantenimiento y actualización de los sistemas.
Esto tiene como consecuencia la no necesidad de la inversión en licencias costosas de software HMI y también se puede prescindir en muchos casos del hardware HMI como las pantallas táctiles [6] y así, se consigue mayor flexibilidad de costos, ya que se elimina la necesidad del aumento de precio para la recuperación de la inversión de licencias y hardware.

Se propone entonces:

1. enseñar el software de los fabricantes como tradicionalmente se ha hecho ver Figura 9.
   Se muestra tanto la pantalla táctil, como su software de diseño.

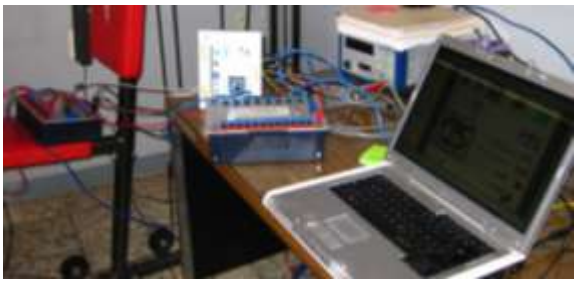


Figura 9. Interfaz HMI Comercial y PLC

2. añadir algún curso que contemple la enseñanza de esta alternativa de diseño e implementación de software HMI, para los Estudiantes de ingenierías Figura 10.

Figura 10. Sistema de integración de un PLC comercial y software HMI propio.

Figura 11. Software HMI en operación.

En la figura 11 se ve un sistema de integración que consta de un PLC DL350 y la interfaz grafica diseñada en C# y ligada a una base de datos.

IV. CONCLUSIONES

- Se Diseñó la base de software HMI
- Se comprobó su funcionalidad
- Se aprendió más del protocolo del PLC
- Se vio que utilizando el mismo protocolo y teniendo la base del Softaware HMI es ágil y práctico implementar interfaces gráficas para diferentes modelos de esta marca,
- Se plantea sumar la enseñanza de estos protocolos y formas de diseño en los cursos de automatización en las ingeniarías

V. REFERENCIAS.

[1] PETRUZELLA Frank D., *"Programmable Logic Controllers"*, 3rd ed., Editorial McGraw-Hill.
[2] Allen Bradley sensors, Rokwell software,.
[3] P. Mengual, STEP 7 Una Manera Facil de programar PLC de Siemens, Marcombo, S.A., 2009
[4] Festo Didactic 2008 GmbH & Co. Kg
[5] D3–350CPU User Manual, *Automationdirect, 2002*
[6] Allen-Bradley, PanelView Plus Terminals 2711p (400,600,1000,1250,1500),

VII. BIOGRAFIAS.

G. Alejandro Herrejón Pintor, nació en Morelia Mich. En 1974. Realizó estudios de ingeniería eléctrica industrial y maestría en ciencias en ingeniería eléctrica en el Instituto Tecnológico de Morelia 2004, México. Su área de interés es el estudio del campo electromagnético y la automatización. g.alex.herrejon.p@gmail.com

Ana Celia Segundo Sevilla, nació en Zamora Mich. En 1980. Realizó estudios de licenciatura en informática en el Instituto Tecnológico de Estudios Superiores de Zamora y Maestría en ciencias de la computación en el Instituto Tecnológico de León.