

DISEÑO DE UN CONTROLADOR LÓGICO PROGRAMABLE BASADO EN UN MICROCONTROLADOR MOTOROLA

(Design of a logic and programmable controller based on the Motorola Microcontroller)

Wilson Javier Pérez Holguín, Álvaro Fernández Acevedo

Escuela de Ingeniería Electrónica, UPTC-Sogamoso, Grupo de Investigación GIRA,
wjperzh@yahoo.es, afermac@gmail.com

(Recibido marzo 15 de 2006 y aceptado noviembre 17 de 2006)

| | |
|--|---|
| <p>Resumen: En este artículo se presentan los resultados obtenidos en el diseño de un PLC basado en un microcontrolador, utilizando las normas de compatibilidad electromagnética. Se realizó un diseño tanto de software como de hardware y se implementó en un ambiente industrial, obteniendo muy buenos resultados, en frente al ruido electromagnético propio del ambiente industrial.</p> | <p>Abstract: This article, presents the results achieved in the design of a PLC based on a microcontroller, using the norms of electromagnetic compatibility. A design of software and hardware was carried out and this design was implemented in an industrial atmosphere, obtaining very good results, in front of the electromagnetic noise characteristic of the industrial atmosphere.</p> |
| <p>Palabras Clave: autómata programable, microcontrolador, reprogramación, compatibilidad electromagnética.</p> | <p>Keyword index: programmable robot, microcontroller, reurning, electromagnetic compatibility.</p> |

1. INTRODUCCIÓN

Los autómatas programables son dispositivos diseñados para el control de procesos industriales, su propósito inicial fue el de controlar procesos que involucren Control de eventos discretos, sin embargo dada su robustez y el avance de la tecnología poco a poco se convirtieron en dispositivos potentes que incursionaron en el control continuo. Sin embargo si se observan detalladamente no son dispositivos complejos, todos tienen los mismos bloques principales: entradas aisladas, procesador o tarjeta de control, salidas de mediana potencia, comunicación serial (Balcells, 1997).

En el presente artículo se muestran los resultados del trabajo en la línea de investigación de automatización industrial del grupo GIRA, dentro de la cual se diseñó un PLC o autómata programable que consta de una tarjeta de control que aloja a un microcontrolador (que actúa como procesador) encargado de almacenar el Sistema Operativo y el código de usuario en su memoria de programa. Además, posee tarjetas de entrada y salida (para el caso, digitales). Por último, se desarrolló una interfaz de programación con usuario de tipo visual. La idea era verificar la realidad sobre la posibilidad de implementar sistemas de control tipo PLC basados en PLC, como una alternativa económica de control para procesos industriales en los que existen altos niveles de ruido.

2. METODOLOGÍA

2.1 Descripción de un autómata programable

El autómata programable (en adelante AP) es un equipo de control industrial, es la unidad de control, incluyendo total o parcialmente las interfaces con las señales de proceso. Por otro lado se trata de un sistema con un hardware estándar, con capacidad de conexión directa a las señales de campo (niveles de tensión y corriente industriales, transductores y periféricos electrónicos).

2.1.1 Hardware del autómata:

- CPU: sistema digital basado en microprocesador o microcontrolador.
- Reloj: marca el funcionamiento de todo el sistema.
- Memoria no volátil (ROM): parámetros fijos, sistema operativo, drivers para el hardware.
- Memoria volátil (RAM): datos y programa de usuario.
- Interfaces de entrada/salida: adaptadas para enviar y recibir señales desde la planta.
- Fuente de alimentación.
- Comunicaciones: serie.
- Temporizadores y contadores.

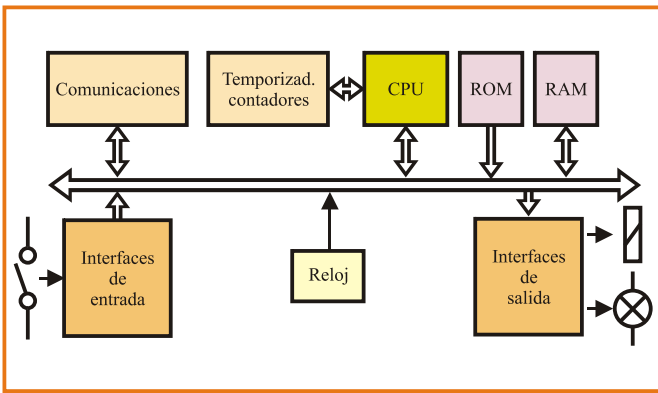


Figura 1. Diagrama de bloques de un autómata

2.1.2 Software de un autómata:

- Sistema operativo: es un software residente en la memoria ROM, esta encargado de gobernar el hardware del autómata, permitir su programación y facilitar la comunicación del programa de control con el hardware.
- Programa específico de control: es la secuencia de operaciones a realizar sobre la planta.

2.1.3 Ciclo de funcionamiento de un autómata:

El siguiente diagrama de flujo ilustra un ciclo simple de funcionamiento de un AP.

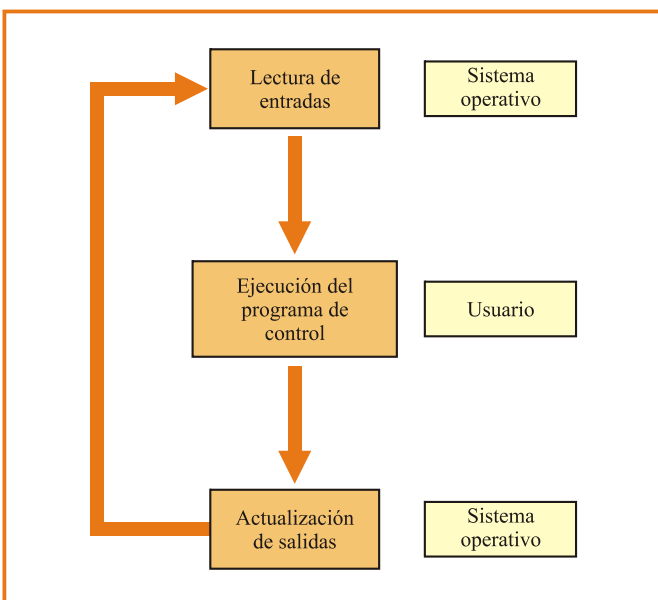


Figura 2. Ciclo de funcionamiento de un autómata.

Esta breve descripción de lo que es un autómata programable y sus partes que lo conforman permite iniciar el proceso de diseño.

2.2 Diseño del autómata

Teniendo en cuenta el hardware que conforma un AP se inicia el proceso de diseño. El diseño del AP se divide en dos partes fundamentales que son: el desarrollo del hardware que incluye el diseño del módulo de control, módulos acondicionadores de señal y backplane (incluye fuente de alimentación); y el desarrollo del firmware donde se integran los programas implementados en el microcontrolador (sistema operativo) así como los programas desarrollados para comunicación con un PC que permiten programación y monitoreo. A continuación se describen cada una de las etapas.

2.2.1 Hardware del autómata:

- *Módulo de control:* el módulo de control consta de un Circuito de comunicación: el cual está compuesto por una adaptador de niveles TTL a RS232 (MAX232), un adaptador de niveles TTL a RS485 (MAX485). La comunicación por RS232 permite la comunicación entre el autómata y el computador ya sea para programación o para monitoreo. La comunicación por RS485 es utilizada para la interconexión a otros autómatas o para el monitoreo remoto.
- *Microcontrolador:* se utiliza un MC68HC908GP32 de Motorola que posee muchas de las prestaciones requeridas por el sistema tal es el caso de la programación en modo usuario, el modulo de comunicación serial enfocado a redes, detección de bajo voltaje, temporizadores para las bases de tiempo de los timers del autómata, amplia cantidad de memoria RAM y flash y un conjunto de instrucciones enfocadas a bits.
- *Circuito de multiplexación:* consta de un decodificador 74LS42 el cual recibe del microcontrolador la dirección de la tarjeta que se desea habilitar y permite el manejo de hasta 8 módulos de entrada y 8 módulos de salida simultáneamente.

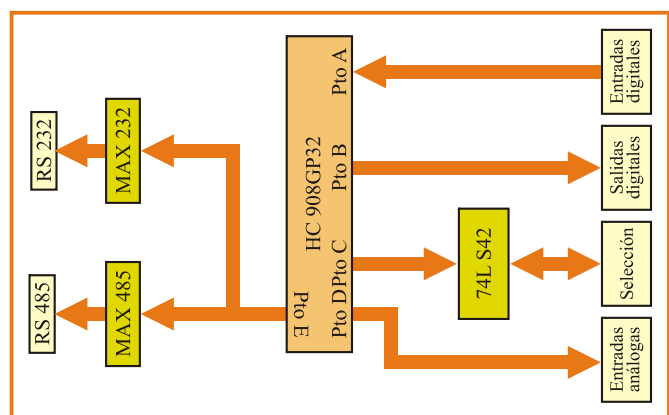


Figura 3. Diagrama de bloques del módulo de control.

- *Módulo de entradas:* el módulo de entradas esta compuesto por una sección de visualización y protección para las señales de 110V en seguida se encuentran ubicado el aislamiento óptico que protege la sección de la tarjeta que trabaja con niveles TTL

por último esta el circuito que permite que los datos presentes en las tarjetas pasen al bus de entradas y en seguida al módulo de control.

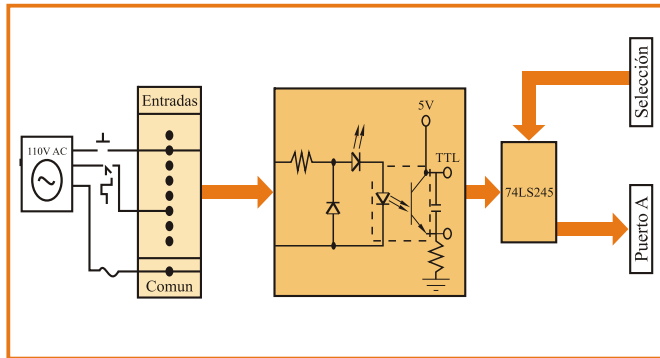


Figura 4. Diagrama del módulo de entradas.

- **Módulo de Salidas:** este módulo está compuesto por una memoria de 8 bits que almacena el estado actual de las 8 salidas correspondientes a la tarjeta, la modificación de dicho estado se realiza mediante la activación de una línea de habilitación, también se encuentra un circuito formado por diodos LED que permite la visualización del estado actual de las salidas. Para aislar los niveles TTL de la etapa de potencia se utilizan aislamiento óptico. La etapa de potencia esta compuesta por triacs que permiten conmutar el voltaje AC, se utilizaron dos protecciones una contra sobre corrientes (fusibles) y otra contra sobrevoltajes (varistores).

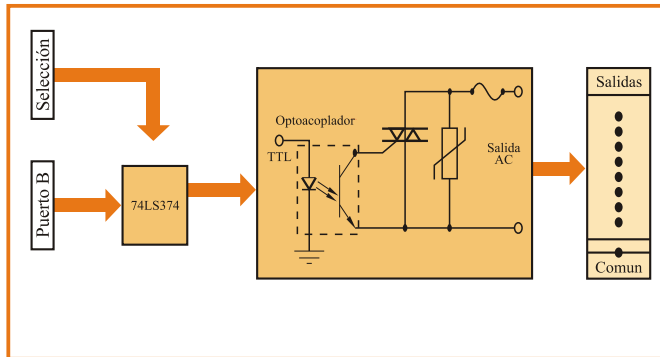


Figura 5. Diagrama del módulo de salidas.

- **Backplane:** mediante este modulo se logra la interconexión entre las diferentes tarjetas que posee el sistema posee tres buses, uno de entradas, uno de salidas y uno de control. También funciona como soporte físico para las tarjetas y de interconexión al punto común de apantallamiento. Debido a su ubicación se opto por incorporar la fuente de alimentación a este modulo, esta tiene una capacidad máxima de 3 amperios y su componente fundamental es un regulador conmutado.

2.2.2 Software del autómeta:

Está compuesto por el software residente en el microcontrolador y la aplicación que permite la programación y el monitoreo desde el computador.

- **Software residente en el microcontrolador:** esta compuesto por una serie de rutinas que permiten realizar todas las tareas propias de un autómeta, y se desglosan de esta manera:

- **Rutinas de programación:** permiten que se pueda llevar a cabo un cambio en el código sin remover el microcontrolador del circuito y sin la necesidad de hardware adicional.

- **Rutinas de comunicación serial:** permite realizar el monitoreo mediante la identificación de la dirección del autómeta y el envío de los estados actuales de los registros de usuario.

- **Rutinas de detección de bajo voltaje:** realiza la detección de una baja en la alimentación del autómeta y de inmediato guarda el valor de los registros que su usan en la lógica de control en la memoria flash.

- **Rutinas de escaneo:** se encargan de actualizar constantemente el estado de las entradas y las salidas, este proceso se realiza aproximadamente cada 20 ms para asegurar la adecuada respuesta de los dispositivos de expansión.

- **Software residente en el computador:** en el computador se encuentra una aplicación desarrollada específicamente para este sistema, posee su pantalla de edición y programación y la pantalla de monitoreo y depuración.

- **Edición y programación:** esta aplicación permite ingresar el código de control en lenguaje AWL el cual es utilizado en los PLCs de bajo nivel, aquí también se puede llevar a cabo el compilado y la programación del autómeta mediante dos aplicaciones enlazadas al programa principal.

- **Monitoreo y depuración:** una vez ingresada la lógica de control al autómeta se puede utilizar esta interfaz, la cual nos permite ver el estado actual de los registros que deseamos siempre y cuando sean utilizados en el programa escrito en AWL.

2.3 Interferencias electromagnéticas

Dado que el autómeta debe trabajar en presencia de ruido industrial y ruido electromagnético se tuvo en cuenta una serie de recomendaciones para el diseño de sistemas con compatibilidad electromagnética (Balcells, 1998).

2.3.1 Consideraciones para circuitos impresos:

- Incrementar la separación entre las pistas para eliminar el crosstalk debido a la capacitancia de acople.

- Maximizar la capacitancia del PCB ubicando Vcc y tierra paralelas (Catherwood, 1999).

- Ubique las pistas sensibles al ruido de alta frecuencia lejos de las líneas de potencia para evitar la inducción.
- Líneas anchas para Vcc y tierra para evitar impedancias en serie y lazos de retorno demasiado largos (Glenewinkel, 2000).
- Separar las tierras para cada una de las tarjetas y para cada tipo de señal (analógica o digital).
- Utilizar condensadores de desacople uno por cada tarjeta para suplir los requerimientos de corriente durante las conmutaciones, su valor debe ser al menos 47uF.
- Utilizar un condensador tan cercano como sea posible a cada circuito integrado preferiblemente de 0.1 uF.
- Se debe utilizar la familia lógica mas lenta posible para evitar ruido de alta frecuencia.
- Como el oscilador es el circuito que mas consume potencia AC debe estar lo mas cerca posible del microcontrolador para evitar la atenuación.
- Se debe cubrir con un plano de masa la sección de la tarjeta donde no existan pistas, esto con el fin de captar todo el ruido inducido y enviarlo a una tierra física.

2.3.2 Consideraciones para software:

La programación defensiva es una técnica de protección contra interferencias que involucra algunas consideraciones en el desarrollo del software.

- *Refreshamiento de puertos:* uno de los ejemplos mas simples de la programación defensiva es estar constantemente actualizando el estado de los registros de dirección de los puertos, ya que por estar mas cerca de los bordes de microcontrolador, son más susceptibles a ser cambiados cuando el ruido supera cierta amplitud y se propaga por la pastilla de silicio hacia dentro del micro.
- *Prueba de entradas:* este tipo de programación corresponde a la prueba de un bit leído varias veces y escoger el resultado que mas veces se repita, sin embargo, es una práctica poco deseada cuando tenemos aplicaciones que tengan tiempo critico de ejecución (T.C. LUN, 2004).
- *Prueba de rutinas:* esta técnica consiste en el uso de algunos registros ubicados en la RAM que permitirán llevar un control de que rutina se esta ejecutando. Una vez dentro de la rutina se verifica si la ejecución es producto de una llamada correcta, en cuyo caso se actualizan los registros de control, de lo contrario, se envía el programa a una rutina adecuada que permita reiniciar la ejecución correctamente.

3. RESULTADOS

Se implementó inicialmente un software de control básico para el Autómata Programable (primer diseño), en el que no se tuvieron en cuenta los aspectos de protección contra

perturbaciones electromagnéticas aquí descritos, ni los relativos a la programación defensiva, obteniendo los siguientes resultados:

Tabla 1. Pruebas y aciertos del sistema.

| Hora | Aciertos | | Errores | |
|------|----------|---------|----------|---------|
| | Entradas | Salidas | Entradas | Salidas |
| 1 | 2 | 3 | 3 | 2 |
| 2 | 1 | 4 | 4 | 1 |
| 3 | 4 | 3 | 1 | 2 |
| 4 | 3 | 2 | 2 | 3 |
| 5 | 2 | 1 | 3 | 4 |
| 6 | 5 | 3 | 0 | 2 |
| 6 | 1 | 4 | 4 | 1 |
| 8 | 2 | 2 | 3 | 3 |

Se observa que el sistema no produjo los mejores resultados, debido a que las conmutaciones de las tarjetas hicieron fluctuar la fuente de alimentación, además se presentó un problema de interferencias entre las tarjetas que producía que se fijaran valores en el puerto sin haber ingresado datos en el mismo. Es de aclarar que esta prueba se llevo a cabo en un ambiente industrial, mas específicamente en el tren de transporte de alambre en rollo de una empresa Siderúrgica.

Una vez observada las causas del mal funcionamiento se implementaron las NIE, realizando algunas modificaciones en los circuitos, que consistieron básicamente en: separar las líneas de tierra, utilizar conductores mas gruesos para las líneas de alimentación y la ubicación de condensadores de desacople en cada tarjeta, así como modificaciones al software del sistema operativo del Autómata. Una vez realizados estos cambios se procedió a realizar el ensamble de todas las tarjetas de prueba sobre el backplane (1 de control, 2 de salida, 3 de entrada), obteniendo los resultados mostrados en la Tabla 2, para la misma prueba realizada anteriormente.

Tabla 2. Pruebas y aciertos del sistema.

| Hora | Aciertos | | Errores | |
|------|----------|---------|----------|---------|
| | Entradas | Salidas | Entradas | Salidas |
| 1 | 5 | 5 | 0 | 0 |
| 2 | 5 | 5 | 0 | 0 |
| 3 | 5 | 5 | 0 | 0 |
| 4 | 5 | 5 | 0 | 0 |
| 5 | 5 | 5 | 0 | 0 |
| 6 | 5 | 5 | 0 | 0 |
| 6 | 5 | 5 | 0 | 0 |
| 8 | 5 | 5 | 0 | 0 |
| 9 | 5 | 5 | 0 | 0 |
| 10 | 5 | 5 | 0 | 0 |

4. CONCLUSIONES

El estudio de interferencias electromagnéticas permitió la construcción de un equipo altamente inmune al ruido industrial.

Se detectó que la conmutación de los circuitos digitales provocaba interferencias electromagnéticas y errores de lecturas.

Las líneas de retorno de corriente alterna en los circuitos impresos cuando son muy delgadas cusan tensiones parásitas originando lecturas falsas.

Existe un alto nivel de soporte de información para el trabajo con microcontroladores Motorola.

Es posible la utilización de sistemas microcontrolados en ambientes industriales siempre y cuando se tengan en cuenta las normas para compatibilidad electromagnética.

El entorno de desarrollo posee la capacidad de monitoreo en tiempo real lo que facilita la configuración del microcontrolador.

Se demostró que es posible desarrollar un lenguaje para la programación de microcontroladores Motorola compatible con el lenguaje de PLCs de bajo y medio nivel.

El MICROMOTPLC puede ser implementado en cualquier sistema que involucre el control de eventos discretos, esto lo convierte en un sistema flexible.

El desarrollo de un lenguaje de medio nivel para microcontroladores facilita el desarrollo de aplicaciones por parte usuarios con bajo nivel de entrenamiento.

Aunque es difícil la consecución de herramientas adecuadas es posible desarrollar tecnología propia en la región.

5. REFERENCIAS

- Catherwood, Mike. (1999). Designing for Electromagnetic Compatibility (EMC) with HCMOS Microcontrollers. Obtenido el 22 de abril de 2005, de www.freescale.com
- Glenewinkel, Mark. (2002) System Design and layout Techniques for noise Reduction in system MCU. Application Note AN1259. Obtenido el 10 de abril de 2005, de www.freescale.com
- T. C. Lun (2002). Designing for Board Level Electromagnetic Compatibility. MOTOROLA Application Note AN2321. Obtenido el 20 de noviembre de 2004, de www.freescale.com
- Balcells, Joseph, (1997). Autómatas Programables, Editorial Marcombo, Barcelona.
- Balcells Joseph, (1998). Interferencias Electromagnéticas en Sistemas Electrónicos. Editorial Alfaomega, Barcelona, pp. 85-166.