

Software para programar PLC basado en microcontrolador usando lenguaje Basic

Luis Raúl Robles Ramos¹, Gerardo Durán Medrano¹, Manuel Nandayapa¹,
Osslan Osiris Vergara Villegas¹, Ulises I. Ponce¹

¹ Universidad Autónoma de Ciudad Juárez

Resumen

En la industria, las compañías están siempre buscando la manera de incrementar u optimizar la producción. La demanda creciente de productos obliga a los fabricantes a producir más en un menor tiempo. La automatización llegó como solución para incrementar la calidad, reducir costos y tiempo de producción. Este documento describe la elaboración de un software para programar un PLC basado en microcontrolador, usando lenguaje Basic, con lo cual pueden elaborarse secuencias complejas en menos tiempo y de una manera más simple.

Palabras clave: software, programación, PLC, microcontrolador.

Introducción

El PLC (Controlador Lógico Programable, por sus siglas en inglés) es un pequeño dispositivo controlador, que ha sido utilizado para reemplazar los relevadores mecánicos, interruptores y temporizadores de gran tamaño en los paneles de control tradicional. El PLC es básicamente un controlador que es diseñado especialmente para controlar un proceso o maquinaria, basado en el número de entradas/salidas. (Rida, Liu, & Jadi, 2014).

La comunicación entre PLC es dependiente del bus de red. El PLC se ejecuta por el ciclo de escaneo, dividido

en las cinco fases que son: comunicación de programación, lectura de entradas, procesamiento interno, ejecución del programa y procesamiento de salidas. Las fases se completan y se ejecuta el siguiente ciclo de escaneo. (Xiao & Li, 2015).

En la actualidad los PLC's son mayormente programados en los lenguajes del estándar IEC 61131-3, los cuales no están preparados para satisfacer los nuevos retos de los amplios sistemas de automatización distribuida. Actualmente, diferentes soluciones se encuentran bajo desarrollo de la industria

e investigación: una extensión del estándar IEC 61131-3 que incluye programación orientada a objetos; la adopción de un nuevo estándar IEC 61499 que vuelve la ejecución del software basado en eventos, en lugar de basado en ciclo de escaneo como en el estándar IEC 61131-3. (Basile, Chiacchio, & Gerbasio, 2011)

Los diferentes lenguajes de programación han demostrado ser efectivos y la variedad existente permite que cada persona pueda optar por el que le resulte más adecuado para desarrollar sus programas en el caso de existir diversas opciones. En el caso de los PLC's la

variedad de lenguajes es muy limitada y la existencia de software que sirva para tal fin en lenguaje Basic es casi nula, habiendo únicamente un software que lo maneja de manera nativa y de uso exclusivo para el PLC's de la misma empresa.

Con el uso del lenguaje Basic se pueden programar secuencias complejas de una manera más sencilla que en Lenguaje Escalera (LAD, por su abreviación en inglés), el cual es el más común para la programación de PLC's, y a la vez el LAD resulta más extenso para realizar depuraciones debido a su naturaleza estilo diagrama.

Metodología

En la búsqueda de información para el presente artículo, se definieron las palabras claves a utilizar para la investigación.

Se ingresó a la base de datos de la Universidad Autónoma de Ciudad Juárez (UACJ) y a que es la institución que nos brinda el apoyo al proporcionar acceso a diferentes fuentes de información científica. Al estar investigando sobre el tema, se encontró que no existe algún artículo que plantee la problemática y solución que se proponen en el presente documento, por lo que la búsqueda se tuvo que realizar de modo que se pudieran encontrar documentos que trataran temas similares. En este caso sobre software para programación de

PLC's. Aún así, la cantidad de documentos que abordan esta problemática es muy limitada.

La metodología en la elaboración del programa fue dividida en tres etapas. La primera etapa se centra en la creación de una interfaz para comunicarse entre el ordenador y el PLC Basado en Microcontrolador (PLCBM), mostrada en la figura 1. Para la cual primero se tuvo que determinar la cantidad de entradas analógicas, y entradas y salidas digitales, para posteriormente realizar la programación del protocolo de comunicación USB, haciendo uso de las librerías que proporciona Microchip para tal función en conjunto con el Software de programación Visual Basic 2013.

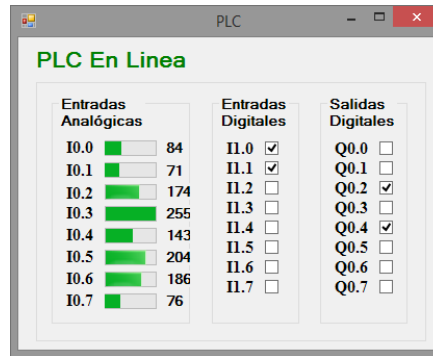


Figura 1. Interfaz gráfica entre el programa y el PLCBM

La segunda etapa es la creación del intérprete para controlar las entradas y salidas del PLC, para lo cual se realizó primero un diccionario de palabras reservadas, y posteriormente la elaboración del algoritmo para la detección de sintaxis en las líneas de código que el usuario introduzca. Una vez hecho lo antes mencionado, el siguiente paso fue programar el algoritmo que muestra al usuario el tipo de error que hace que la sintaxis sea incorrecta y mostrar el número de línea en la cuál se encuentra el error.

En esta misma etapa se consideró adecuado incluir la función que

determinara los colores de las palabras escritas en el código por el usuario, para lo que nuevamente se elaboró un diccionario para palabras reservadas, otro para entradas y salidas y otro para números. Y estando en este punto, también se realizó el algoritmo para determinar si una parte del código es comentario, y de ser así, también aplicarle un color.

Por último en esta etapa se implementó la función de guardar y abrir archivos. El programa en este punto tenía la apariencia mostrada en la figura 2.

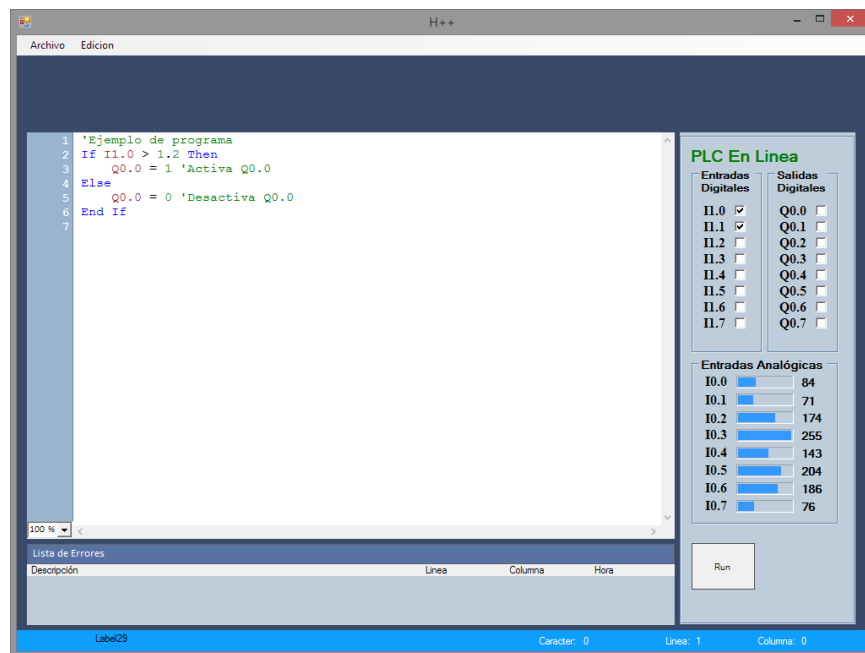


Figura 2. Interfaz gráfica al finalizar la segunda etapa.

En la tercera etapa, es donde se compila el programa para generar un archivo en formato distinto al que se utilizó para programar y que posteriormente es almacenado y leído internamente por el PLCBM, haciendo así que el PLCBM funcione sin la necesidad de estar conectado a un ordenador para ejecutar las secuencias programadas.

En esta misma etapa se agregan más funciones, como un botón para ejecutar el programa en modo intérprete, otro para ejecutar desde el PLCBM y otro para detener la ejecución del código, ya sea en modo intérprete o interno. También se agrega un botón para compilar y otro para programar el PLCBM con la secuencia ya compilada.

Se determinó también que los retardos deberían ser en segundos, en lugar de

milisegundos, esto con el fin de hacer el programa más fácil de utilizar por personas de áreas en las que no es común manejar o programar este tipo de dispositivos.

También se determinó que debería cambiarse el valor de lectura de los puertos analógicos, ya que al principio se mostraba el valor en un rango de 0 a 255, y por el mismo motivo que en el caso de los retardos, se cambió a mostrar la lectura en un rango de 0.0V a 5.0V y a su vez se realizó la modificación en el algoritmo para poder cambiar la comparación de números enteros a números con punto decimal y consecuentemente, también fue necesario realizar cambios en el algoritmo encargado de detectar la sintaxis y mostrar los errores.

Al tener todo lo anterior, se procedió a realizar una modificación visual a la parte visual del programa para que resulte una interfaz más amigable al usuario y se

implementaron botones de acceso directo a funciones básicas que se encuentran en los menús, que son: Nuevo, Abrir, Guardar, Compilar y Programar PLC.

Resultados

La Figura 3. Muestra la apariencia de la interfaz gráfica una vez finalizado el diseño del software de programación, donde se pueden apreciar las diferentes características, en la parte superior se aprecian las funciones de crear un archivo nuevo, abrir un archivo existente, guardar el documento en el que se está trabajando, compilar el programa y programar el PLC. En la parte central, el área de escritura del código, con un encabezado que muestra el nombre del archivo, así como la ruta en la que se encuentra

guardado. En la parte derecha se aprecia el estado del PLC, los valores de entrada tanto analógicos como digitales, y los valores de las salidas. Inmediatamente debajo de estos valores, se encuentran los botones para ejecutar el programa en modo intérprete, desde el PLC y el botón de detener. Por último, en la parte inferior se aprecia la zona de errores, en la que se informará del mismo en caso de haber algún tipo de error en la programación de la secuencia.

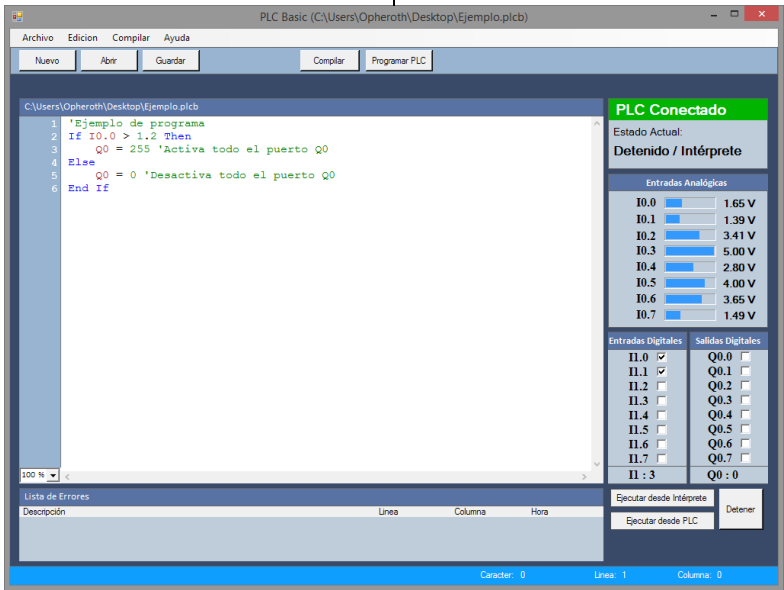


Figura 3. Apariencia gráfica del software de programación.

El software tiene las siguientes características:

Programación en lenguaje Basic.

Ejecución de código en modo Intérprete (Desde el ordenador).

Ejecución de código desde PLCBM (Sin necesidad de ordenador).

Lectura de valores digitales y Analógicos.

Capacidad de almacenar programas.

Capacidad de abrir archivos.

Indentación automática al compilar el código.

Resalta palabras reservadas, números y comentarios en un color distinto.

Muestra la existencia de errores en cuanto a sintaxis y comandos en el programa.

Detección y programación del PLCBM a través de puerto USB.

Programación del PLCBM sin necesidad de reiniciar ni de entrar en modo Bootloader.

Conclusiones

El producto obtenido resulta una herramienta que sirve de apoyo para la enseñanza y que debido al uso del lenguaje Basic, permite que el PLCBM pueda ser programado por personas que ya tengan conocimiento del mismo y no del LAD sin la necesidad de aprender la simbología y el razonamiento que éste último utiliza.

El software permite la ejecución línea por línea en modo intérprete, cambiando a la siguiente línea hasta que el usuario así lo indica presionando un botón, para una depuración más sencilla del código, pudiendo ver el estado de las entradas y salidas en la pantalla en todo momento.

El software muestra el uso de programación en Visual Basic, el cuál es utilizado mayormente en el área de Ingeniería en Sistemas Computacionales y carreras afines, por lo que muestra el uso de la programación para crear intérprete y compilador.

Así mismo se emplea una interfaz, tema que se ve en diferentes materias de programación en áreas de electrónica, mecatrónica y electromecánica, por lo que el programa además de ser utilizado para la programación del PLCBM sirve como un ejemplo de aplicación de estas herramientas que se enseñan en el aula.

Referencias

Basile, F., Chiacchio, P., & Gerbasio, D. (2011). Progress in PLC programming for

distributed automation systems control. *2011 9th IEEE International Conference*

on *Industrial Informatics*, 621–627.
<http://doi.org/10.1109/INDIN.2011.6034950>

Rida, M. E., Liu, F., & Jadi, Y. (n.d.).
Design Mini-PLC based on
ATxmega256A3U-AU Microcontroller,
3–6.

Xiao, L., & Li, M. (2015). PLC Programs
' Checking Method and Strategy Based
on Module State Transfer *, (August),
702–706.