

## **Desarrollo de un Sistema Emulador de PLC con Telecontrol desde una Aplicación Móvil para Android y con Software de Teleprogramación vía Bluetooth que incluye Tecnología DDE para realizar Aplicaciones de Control Discreto por Computadora en Tiempo Real desde FESTO FluidSIM**

Jorge García

J. García

Universidad Tecnológica de Puebla. División de Mecatrónica Antiguo Camino a la Resurrección No. 1002-A Zona Industrial Oriente Puebla, Puebla. C.P. 72495

jorge.garcia@utpuebla.edu.mx

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

## Abstract

This paper deals about the development of a single-chip programmable automaton that emulates a PLC (Programmable Logic Controller) and which can be programmed by using KOP (Kontaktplan – ladder diagram) language in a wireless fashion from a personal computer via a Bluetooth link. The CPU (Central Processing Unit) of the automaton can be either of Microchip PIC18FX550 or Atmel ATMEGA 328 families because of the BIOS (Basic Input Output System) for each microcontroller was implemented. The programming language of the automaton supports instructions as described in specification IEC 61131 and introduces new ones that access to communication resources and media available in smart phones. This is because the automaton can be connected to a smart phone by using the same Bluetooth link mentioned above through a mobile application for Android. So, the user program may contains “coils” or “contacts” which are related with the transmission to or reception of messages from the smart phone by using SMS (Short Message Service). Finally, the software used for programming the automaton also contains a DDE (Data Dynamics Exchange) server which can be used in a concurrent fashion with FESTO FluidSIM, in order to make data acquisition and discrete control actions from a personal computer in real time.

## 9 Introducción

Los esquemas de contactos (KOP) conforman uno de los lenguajes para PLC (Controlador Lógico Programable) [1] que más se utilizan en la industria de la automatización para realizar acciones de control. Aún cuando existe el estándar IEC 61131 que se estableció en el año 1992 y que define las características que deben tener los PLC y sus lenguajes de programación [2], es común que cada empresa que se dedica a la fabricación de estos autómatas, establezca características bien específicas que se aplican únicamente a sus propios PLC.

Además, desde hace ya algunos años, se han estado utilizando los diagramas de escalera como lenguaje de programación también para microcontroladores. Los programas para PC LDMicro [3] y PICPLC [4] son algunos ejemplos que se pueden encontrar en la red y que permiten generar código máquina para varios tipos de microcontroladores a partir de diagramas de contactos. Esto ha permitido que el usuario que se inicia en la programación y la aplicación de los microcontroladores pero que tiene conocimientos de cómo programar los PLC, pueda interactuar de forma fácil con ellos, logrando incluso a reemplazar el uso del PLC por el de un microcontrolador en aplicaciones donde es posible hacerlo. Sin embargo, para este caso, se requiere que el usuario del programa tenga un programador de microcontroladores aparte para que el archivo generado por la aplicación (generalmente tipo HEX) pueda cargarse en el chip.

Por otro lado, con el alto impacto y penetración que han tenido las aplicaciones móviles de diversa índole en los usuarios de teléfonos inteligentes, han aparecido ya aplicaciones móviles que permiten editar y cargar hacia y desde el autómata programable, estrategias de control representados con diagramas de escalera. Para el caso de aplicaciones móviles para Android que se ofertan en Google Play Store [5], algunos ejemplos de este tipo de aplicaciones son como los que ofrecen empresas que se dedican a la manufactura de PLC como Koyo Electronics [6] y Siemens [7].

Estas aplicaciones permiten realizar una acción de teleprogramación o telemando de dichos autómatas programables. Sin embargo, para el caso de sistemas embebidos como Arduino y para aquellos que están basados en otros microcontroladores, si bien las aplicaciones móviles permiten la interacción remota del usuario con el autómata, generalmente ésta es sólo de telemando.

Además la aplicación no permite que el programa del usuario grabado en el autómata pueda interactuar con el teléfono, aprovechando así otros recursos de comunicación que posee como lo es por ejemplo el SMS (Short Message Service – Servicio de Mensajes Cortos).

En el caso de Arduino, esto es posible únicamente agregando un hardware adicional (shield – cubierta) [8], con el consecuente incremento en costos y tiempo de desarrollo de la aplicación por parte del programador para integrar esta tecnología en sus estrategias de control.

Así pues, en este trabajo se presenta el desarrollo de un sistema emulador de PLC basado en un autómata programable de un solo chip cuyo lenguaje de programación sigue el estándar IEC 61131 y que puede ser programado en forma inalámbrica desde una computadora personal utilizando un enlace Bluetooth.

Primero se presentan las características principales de la aplicación móvil para Android con la cual el usuario puede correr o detener la ejecución del programa previamente cargado en el autómata e interactuar con él, enviando y recibiendo datos hacia o desde los registros internos del autómata. De hecho a través de esta aplicación, el programa del usuario en el autómata puede utilizar el SMS para enviar o recibir mensajes de texto hacia o desde otros teléfonos inteligentes.

Después se presentan las características más sobresalientes del compilador del software de programación y cómo se realiza la transferencia del programa al autómata. Posteriormente se mostrará la utilización del servidor DDE (Dynamics Data Exchange – Intercambio Dinámico de Datos) [9] integrado en el software de programación del autómata el cual permite que se puedan realizar acciones de control discreto en tiempo real desde el software FESTO FluidSIM.

Finalmente se describirán algunas características del autómata programable y se mencionarán los alcances en la utilización del mismo como controlador embebido de bajo costo con acceso a los medios de comunicación de los que dispone un teléfono inteligente; pudiéndolo aplicar entonces como recurso didáctico para la enseñanza de la programación de autómatas o de PLC.

### Aplicación móvil para telecontrol del autómata

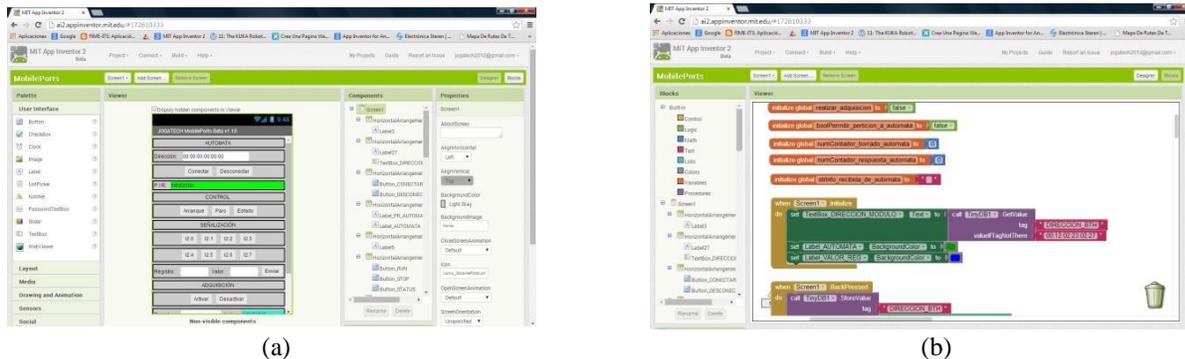
En la figura 1 se muestra la función que tiene la aplicación móvil en el sistema emulador.

**Figura 9** Utilización de la aplicación móvil como elemento de telecontrol del automata



Por conveniencia se ha designado a esta aplicación con el nombre de Mobileports. Esta aplicación se desarrolló utilizando el programa App Inventor el cual es un software de Google basado en la nube [10]. Debido a su naturaleza gráfica, App Inventor es utilizado para la realización de aplicaciones móviles de forma simple e intuitiva. En la figura 2 se muestra la apariencia del entorno de programación de la versión 2 de App Inventor.

**Figura 9.1** Apariencia de la ventana principal del programa App Inventor 2: (a) área de diseño y (b) área de bloques



Una aplicación con App Inventor requiere prácticamente dos aspectos a considerar [11]:

La estructuración de su GUI (Graphics User Interface – Interfaz Gráfica de Usuario) a través del uso de pantallas y otros componentes en el área del diseñador para introducción o despliegue de datos, uso de recursos de comunicación o medios entre otros.

La construcción del código de la aplicación utilizando el lenguaje de App Inventor en el editor de bloques.

El lenguaje de App Inventor está conformado de bloques de código representados por gráficos que semejan ser fichas de un rompecabezas. Para que el programador pueda crear una aplicación con App Inventor, tiene que trabajar en el editor de bloques e ir seleccionando y ensamblando las fichas que conformarán el código de su aplicación. Estos bloques pueden representar desde operaciones aritméticas simples, hasta más complejas como el acceso a recursos de comunicación y medios del teléfono inteligente [12].

Una vez que el programador ha implementado su aplicación, App Inventor le permite crear la versión ejecutable del mismo a través de un archivo con extensión APK el cual podrá ser descargado en su teléfono móvil o en su computadora para su posterior instalación.

En la figura 3 se muestra una captura de la pantalla principal que exhibe MobilePorts una vez que esta aplicación se ha instalado y está funcionando en el teléfono inteligente.

**Figura 9.2** Apariencia de la pantalla principal de MobilePorts



Básicamente, la aplicación MobilePorts tiene 4 áreas a través de las cuales el usuario puede interactuar de forma remota con el autómata:

Conectividad

Control

Señalización

Adquisición

En el área de conectividad, el usuario deberá introducir en el cuadro de texto “Dirección” la dirección del módulo Bluetooth a través del cual se realizará la comunicación entre la aplicación móvil y el autómata. Una vez hecho esto, se podrá conectar al autómata pulsando la tecla “Conectar”.

En el área de control, el usuario puede controlar de forma remota el arranque y paro del programa que se esté ejecutando en el autómata. También puede conocer el estado en la ejecución del mismo pulsando el botón “Estado”.

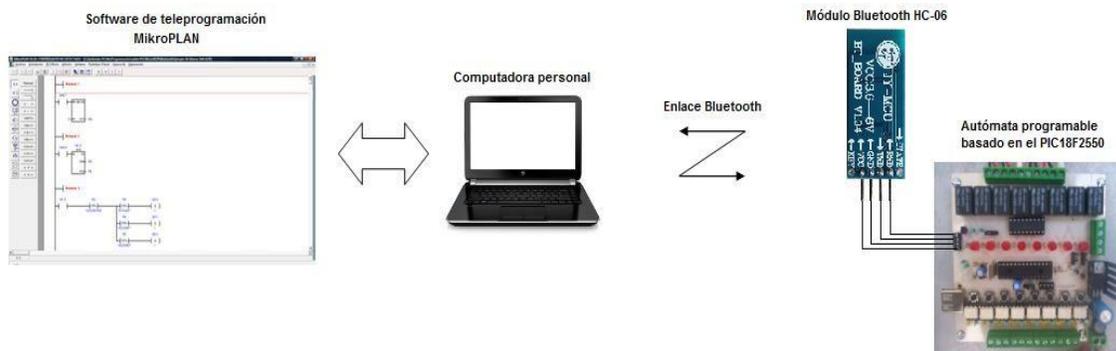
En el área de señalización, el usuario puede enviar valores numéricos específicos a registros internos del autómata. También puede activar remotamente los contactos virtuales I2.0 a I2.7 del autómata y a los que puede accederse desde el programa del usuario que está cargado en su memoria.

En el área de adquisición, el usuario puede obtener en forma continua el valor numérico de cualquiera de los registros internos del autómata. Además, a través de los indicadores tipo LED que se muestran en la pantalla, el usuario puede visualizar el estado lógico de los bits que corresponden al byte menos significativo del valor del registro que se muestra.

## Software de teleprogramación

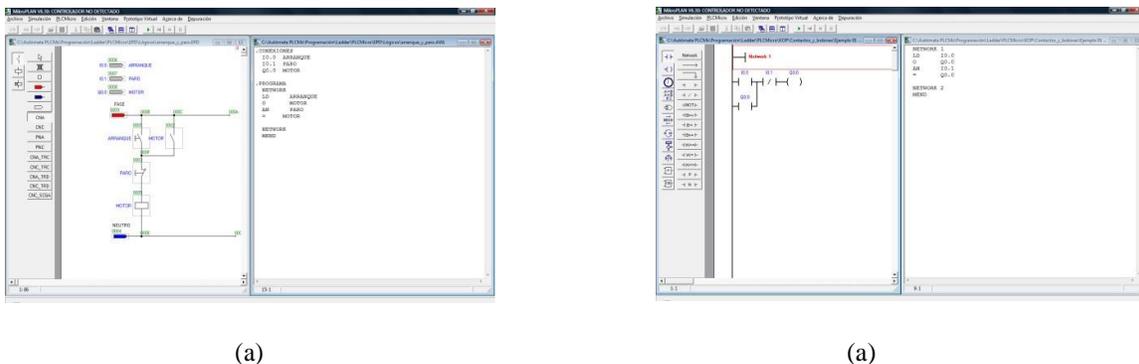
En la figura 4 se muestra la función que tiene el software de teleprogramación en el sistema emulador.

**Figura 9.3** Uso del software de teleprogramación en el sistema emulador



Por conveniencia se ha denominado a este software con el nombre MikroPLAN. Este programa es un desarrollo original que incorpora un editor de diagramas KOP, diagramas en DIN y código AWL (Anweisungsliste – Lista de instrucciones) compatible con la norma IEC 61131; un módulo compilador y un módulo grabador de programas. MikroPLAN fue desarrollado utilizando el lenguaje C++ Builder. El compilador de MikroPLAN genera en primera instancia el código AWL equivalente del tipo de esquema que se haya capturado. Esto se muestra en la figura 5.

**Figura 9.4** Generación de código AWL a partir de la captura esquemática: (a) Generación a partir de diagramas DIN y (b) Generación a partir de diagramas de escalera



Posteriormente, MikroPLAN detecta el tipo de microcontrolador destino y genera el código máquina adecuado. Para el caso de los microcontroladores PIC, el compilador genera el código máquina nativo para esos dispositivos correspondiente al programa que emula la lógica del esquema. Se genera tanto el código máquina de las rutinas que emulan el código AWL así como el código en el programa principal que se requiere para invocar a cada una ellas.

Para el caso del microcontrolador de Atmel, MikroPLAN genera códigos numéricos que representan a las rutinas y parámetros que emulan las instrucciones AWL que se obtuvieron después de la primera fase de compilación y que ya están implementadas en el BIOS del microcontrolador.

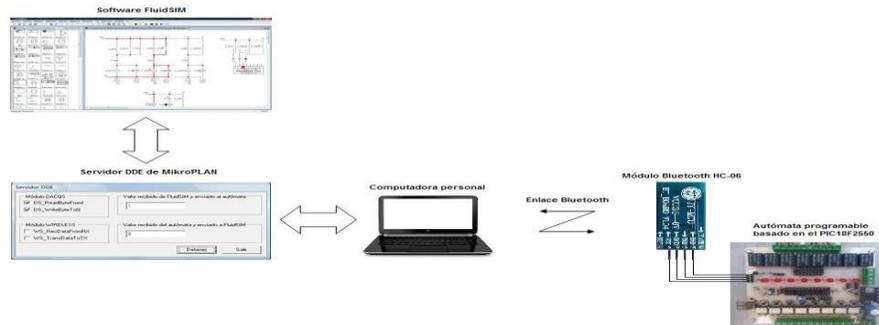
Después de generar los códigos numéricos, MikroPLAN realiza su transferencia al microcontrolador dependiendo del tipo de comunicación seleccionada: USB para comunicación alámbrica o bien comunicación serial para su transferencia por Bluetooth.

En cualquier caso, la transferencia del programa hacia el BIOS del autómatas se realiza por bloques de 64 bytes agregando códigos para detección de errores. Esto se lleva a cabo independientemente del tipo de microcontrolador destino.

### Servidor DDE

Para que FESTO FluidSIM pueda enviar datos al autómatas durante la simulación del esquema en DIN o ANSI, se requiere que en forma concurrente esté funcionando el módulo servidor DDE disponible en el software MikroPLAN. Este programa actúa como una interfaz de comunicación entre FESTO FluidSIM y el autómatas. Esto se representa en la figura 6.

**Figura 9.5** Uso del servidor DDE de MikroPLAN con FESTO FluidSIM



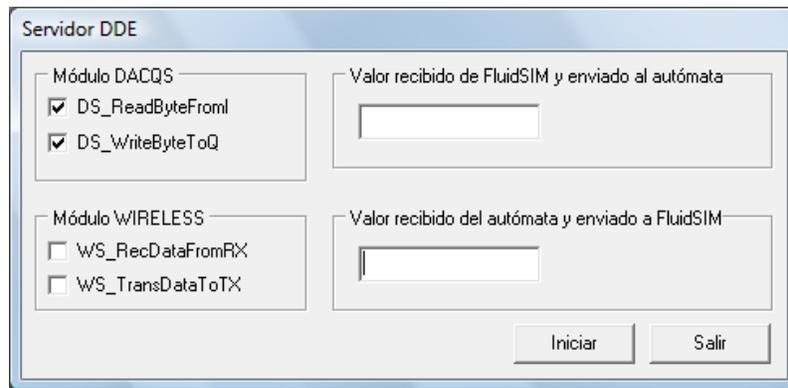
FESTO FluidSIM incorpora un módulo software a través del cual se puede comunicar con otras aplicaciones utilizando la tecnología DDE [13]. En la figura 7 se muestra el cuadro de diálogo que se utiliza en FESTO FluidSIM para que el usuario introduzca el nombre del servidor y el tópicos al que se conectará FluidSIM durante la simulación.

**Figura 9.6** Cuadro de diálogo para configurar la conexión DDE en FESTO FluidSIM



Es por esta razón que el software MikroPLAN debe de ejecutarse en forma concurrente con FluidSIM. En MikroPLAN, el servidor se activa mediante el cuadro de diálogo que se muestra en la figura 8.

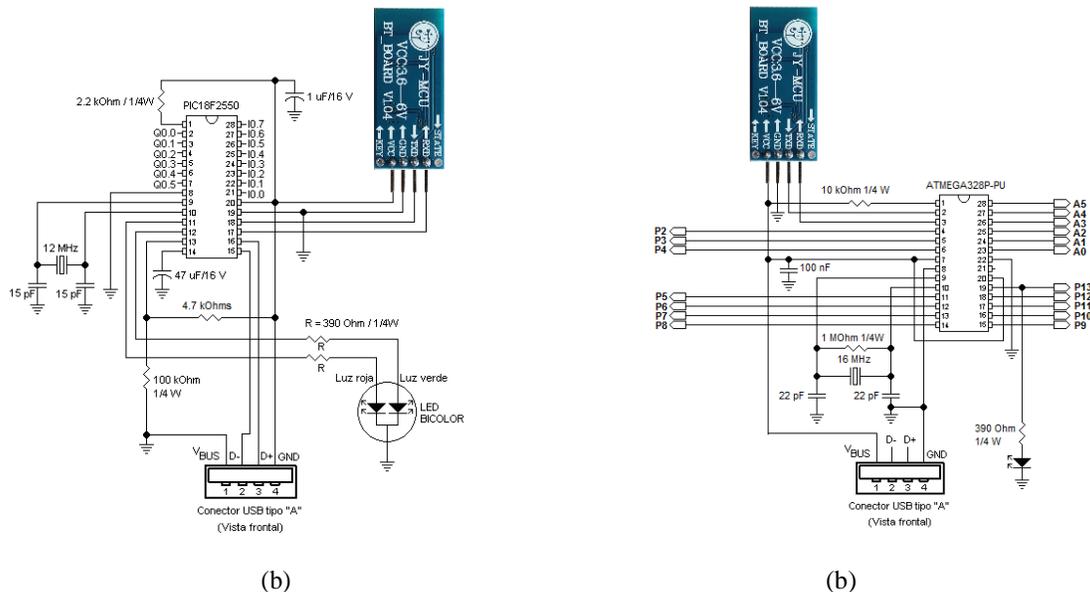
**Figura 9.7** Cuadro de diálogo para activar el servidor DDE de MikroPLAN



El servidor DDE de MikroPLAN soporta también la incorporación de otras rutinas de comunicación inalámbrica como el acceso a un puerto serie auxiliar en el autómeta, por lo cual es posible activarlas también en el mismo cuadro de diálogo.

Autómeta programable. En la figura 9 se muestran los diagramas electrónicos de dos versiones del autómeta programable que se desarrollaron: uno utilizando el microcontrolador PIC 18F2550 y el otro utilizando un microcontrolador ATMEGA328P-PU.

**Figura 9.8** Versiones del autómeta programable: (a) con el microcontrolador PIC18F2550 y (b) con el microcontrolador ATMEGA328P-PU



Como puede apreciarse en la figura de arriba, la arquitectura de los sistemas no es crítica. Sin embargo, independientemente del microcontrolador utilizado, éste debe tener cargado un BIOS el cual le permitirá comunicarse ya sea con el software MikroPLAN o con la aplicación móvil.

Para el caso de los microcontroladores PIC, su BIOS se implementó utilizando lenguaje ensamblador para los PIC 18FX550. Para el caso del microcontrolador de Atmel, se utilizó la plataforma de Arduino para desarrollar el BIOS de ese microcontrolador. Como se puede observar en la figura 9, en ambos autómatas se utiliza el módulo de comunicación Bluetooth HC-06 Esclavo. Sin embargo, solamente para el caso de los microcontroladores PIC, el BIOS puede comunicarse también vía USB.

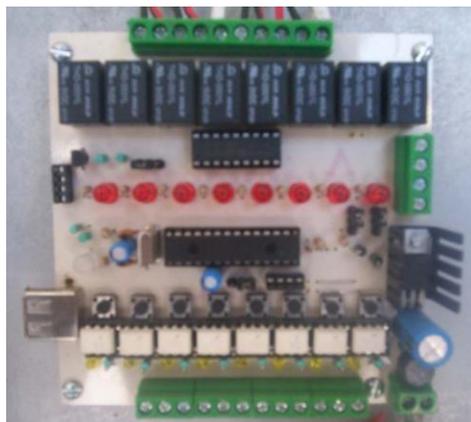
El módulo HC-06 permite la comunicación inalámbrica vía Bluetooth entre el microcontrolador en cuestión y una computadora, un teléfono inteligente o bien otro sistema embebido. En este último caso se requeriría un módulo Bluetooth maestro. La conexión del microcontrolador y el módulo Bluetooth tampoco es crítica ya que este dispositivo acepta comunicación serie con el microcontrolador a niveles de voltaje TTL (Transistor Transistor Logic – Lógica Transistor Transistor), por lo cual la conectividad entre ambos dispositivos es simple y sin requerir otro dispositivo electrónico de por medio.

La comunicación entre el BIOS del microcontrolador y el software MikroPLAN o con la aplicación móvil se realiza a base de peticiones y respuestas, por lo que la comunicación entre los dispositivos involucrados siempre es en forma bidireccional.

Si bien en algunos casos esto pudiera llegar a ralentizar el desempeño del sistema, en general esto permite que a ambos lados del canal de comunicación se puedan detectar errores como por ejemplo de tiempo fuera o de interpretación errónea de datos. En cuanto a la grabación del programa en el microcontrolador, para el caso de los PIC's, cuando el BIOS recibe de MikroPLAN el programa que se grabará en la memoria no volátil del microcontrolador, éste lo grabará en la memoria FLASH del mismo.

Para el caso del microcontrolador ATMEGA328P-PU, su BIOS grabará el programa en la EEPROM del dispositivo. En la figura 10 se muestra una fotografía de otra versión del autómata programable que incluye entradas optocopladas y salidas a relevador.

**Figura 9.9** Autómata programable basado en el microcontrolador PIC18F2550



## 9.1 Resultados y discusión

MikroPLAN es un software que se ha estado utilizando desde el año 2008 como DAQCS (Data Acquisition and Control System – Sistema de Control y Adquisición de Datos) didáctico para la enseñanza de la programación de controladores digitales en las asignaturas de Control Lógico Programable y Sistemas Hidráulicos y Neumáticos, así como en la asignatura de Sistemas de Control Automático, las cuales son parte de los planes de estudio de las Carreras de Técnico Superior Universitario en Mecatrónica Área Automatización y en Mecatrónica Área Instalaciones Eléctricas Eficientes. También se ha utilizado en la asignatura optativa de Domótica de la Carrera de Ingeniería en Tecnologías para la Automatización. Todas estas carreras pertenecientes a la División de Mecatrónica de la Universidad Tecnológica de Puebla. Si bien en principio se utilizó su versión de programación cableada con el autómeta, en el cuatrimestre enero a abril de este año de 2014 se utilizó su versión de programación inalámbrica.

Desde el año 2010, el BIOS del autómeta permite también la comunicación con otras aplicaciones de software como LabVIEW, C++, Visual C y Visual Basic. Por lo cual, desde ese mismo año, el autómeta también se ha utilizado en las asignaturas como Lógica de Programación, Lenguaje de Programación, Programación Visual, Instrumentación Virtual e Instrumentación Virtual Avanzada y que también forman parte de los planes de estudio de las carreras ya mencionadas.

## 9.2 Conclusiones

Se presentó el desarrollo de un sistema emulador de PLC compuesto de: Un software de teleprogramación de microcontroladores vía Bluetooth en lenguaje KOP, AWL y mediante diagramas en DIN Autómeta programable basado en un sólo microcontrolador. Aplicación móvil para Android con la cual se puede realizar el telecontrol del autómeta.

Se mencionó que los lenguajes KOP y AWL del autómeta programable son compatibles con la norma IEC 61131 y el software de teleprogramación contiene un servidor DDE que puede enlazarse al programa FESTO FluidSIM.; permitiendo de esta manera que éste último pueda utilizarse para realizar estrategias de control por computadora en tiempo real.

Se comentó también la utilización de este sistema como material didáctico en varias de las asignaturas que son parte de los planes de estudios en las carreras Técnico Superior Universitario e Ingeniería que se ofertan en la División de Mecatrónica de la Universidad Tecnológica de Puebla.

Este último aspecto es importante recalcarlo ya que, si por un lado, el lenguaje que soporta MikroPLAN es compatible con la norma IEC 61131, por el otro, debido a que el autómeta programable está basado en un solo chip, puede utilizarse como una opción de bajo costo para que el usuario del sistema pueda utilizarlo en aplicaciones de variada índole, incluso más allá del salón de clase.

Así pues, el módulo DDE que incorpora MikroPLAN para su comunicación con FluidSIM; la reciente incorporación a su lenguaje de programación de instrucciones de comunicación móvil y, a corto plazo, la integración de bloques de servo control, son factores decisivos que seguramente inclinarán la balanza a favor de utilizar este sistema en lugar de otros similares que actualmente se ofertan en el mercado.

## Agradecimientos

Deseo agradecer al Dr. Jorge Alfredo Guillén Muñoz, Rector de la UTP; a la MIA. María Oneida Rosado García, Secretaria Académica y en especial al Ing. Marcos Espinoza Martínez, Director de la División de Mecatrónica de la UTP por el apoyo brindado a un servidor al permitirme utilizar este material didáctico con los alumnos en las asignaturas que he impartido desde que se desarrolló este sistema.

## Referencias

[http://es.wikipedia.org/wiki/Controlador\\_1%C3%B3gico\\_programable](http://es.wikipedia.org/wiki/Controlador_1%C3%B3gico_programable)

[http://es.wikipedia.org/wiki/IEC\\_61131](http://es.wikipedia.org/wiki/IEC_61131)

<http://cq.cx/ladder.pl>

<http://www.mikroe.com/picplc16/>

<https://play.google.com/store/search?q=PLC%20tools>

<https://play.google.com/store/apps/details?id=kse.android.LadderTool>

<https://play.google.com/store/apps/details?id=cz.prunsys.s7plchmi>

<https://play.google.com/store/search?q=PLC%20Arduino&c=apps>

[http://es.wikipedia.org/wiki/Dynamic\\_Data\\_Exchange](http://es.wikipedia.org/wiki/Dynamic_Data_Exchange)

[http://es.wikipedia.org/wiki/App\\_Inventor](http://es.wikipedia.org/wiki/App_Inventor)

<http://explore.appinventor.mit.edu/get-started>

<http://ai2.appinventor.mit.edu/reference/components/>

<https://www.festo.com/net/startpage/>