

SISTEMA DE CONTROL ROBÓTICO CON APLICACIÓN *PICK AND PLACE* DE MATERIAL A GRANEL

C.P. Hernández-Marías, J.A. Rojas-Estrada, M.A. Ochoa-Villegas*, M.U. Zavala-Mejía
Tecnológico Nacional de México / Instituto Tecnológico de Nuevo León
División de Estudios de Posgrado e Investigación
Av. Eloy Cavazos 2001, Guadalupe, N.L. 67170, México
Tel. 81 8157 0500
patriciahdzmarías@gmail.com

*Autor de correspondencia: miguel.angel.ochoa@itnl.edu.mx

RESUMEN.

La necesidad de las plantas industriales por contar con procesos de producción más rápidos y eficientes conlleva al uso de los robots industriales. Una de las aplicaciones que predomina en la industria alimenticia es el *pick and place*, el cual consiste en mover un producto de un lugar a otro. Esta aplicación que se realiza, generalmente de manera manual, trae consigo problemas ergonómicos para el personal operativo, por lo que se opta automatizar esta tarea mediante el uso de robots manipuladores. La finalidad de este proyecto es el implementar el hardware necesario para desarrollar la programación en software que permitirá a un robot manipulador desarrollar una aplicación de *pick and place* mediante el envío de datos de posición y orientación desde un controlador lógico programable, lo cual elimina la necesidad de programar trayectorias complejas. Los resultados, implementados en una celda robótica industrial, demuestran la confiabilidad de este tipo de integración.

Palabras Clave: PLC, *pick and place*, robot manipulador.

ABSTRACT.

The need for factories to have faster and more efficient production processes lead to the use of industrial robots. One of the applications that predominates in the food industry is the *pick and place*, which consists of moving a product from one place to another. This application executed usually manually, brings ergonomic problems for the operative personnel, so it is decided to automate this task through the use of robot manipulators. The purpose of this project is to implement the necessary hardware to develop software programming that will allow a robot manipulator to develop a *pick and place* application by sending position and orientation data from a programmable logic controller, which eliminates the need to program complex trajectories. The results, implemented in an industrial robotic cell, demonstrate the reliability of this type of integration.

Keywords: PLC, *pick and place*, robot manipulator.

1. INTRODUCCIÓN.

Con el paso de los años se ha ido incrementando la complejidad en los procesos de producción. Las empresas están en busca de procesos de manufactura más rápidos, flexibles, precisos y altamente confiables; que aseguren un producto final conforme a los requerimientos de calidad. Estos nuevos requerimientos en la industria, representan la necesidad del uso de robots

industriales para la automatización de distintos procesos de manufactura [1].

Está comprobado que el uso de robots incrementa la productividad y efectividad en los procesos de producción. Según la Federación Internacional de Robótica (IFR), la venta de robots en todo el mundo continúa aumentando año tras año. Como un caso en particular, la IFR indica que México se ha convertido en un importante mercado emergente para los robots industriales, las ventas en el año 2016, ascienden a 5,900 unidades [2].

Dentro de la industria de alimentos, farmacéutica y electrónica; destaca la actividad denominada *pick and place*, la cual consiste en elegir un producto, tomarlo y desplazarlo hacia un lugar diferente. Las aplicaciones *pick and place* suelen ser pesadas y repetitivas, con demandas de alta rapidez y precisión, lo que provoca daños ergonómicos en las personas, así mismo, se ve afectada la productividad, debido a que el personal realiza un trabajo lento y con poca precisión [3].

Como alternativa principal para solucionar los problemas de baja productividad y ergonomía en las actividades de *pick and place* se implementan los robots industriales. La característica principal en robots que vayan a realizar aplicaciones de *pick and place* es la velocidad, por lo que muchos fabricantes ofrecen varios robots que cumplen con este requerimiento en específico. El modelo del robot dependerá de las características del producto a manipular como lo son el peso, volumen, forma, etc. El modelo típico para aplicaciones que demandan una muy alta velocidad y carga pequeña es el delta, mientras que para aplicaciones con alta velocidad y carga pesada es recomendable utilizar un robot articulado [4].

El presente trabajo busca dar a conocer la integración del hardware y software necesario para implementar, a escala industrial, una celda robotizada de manejo de materiales a granel, mediante la utilización de un robot articulado y un PLC como dispositivo maestro de la celda, con el fin de optimizar una aplicación *pick and place* al aumentar su productividad y versatilidad en el manejo de material gracias a la rapidez en el procesamiento e intercambio de datos en el PLC. El artículo está distribuido de la siguiente manera: la sección dos otorga una breve explicación sobre el hardware utilizado. La sección tres muestra el entorno de los principales programas a utilizar. La

sección cuatro muestra el desarrollo de la programación y además se incluyen los resultados de las pruebas realizadas. Finalmente, en la sección cinco, se muestran las conclusiones.

2. HARDWARE.

2.1. Robot manipulador.

El robot manipulador utilizado es un robot de la marca FANUC modelo R-2000iB/165F. Este robot es de tipo articulado con 6 grados de libertad. Dentro de sus características principales destaca su diseño compacto con una alta versatilidad para distintos tipos de aplicaciones como manejo de material, soldadura de puntos y ensamble [5]. En la figura 1 se muestra el robot utilizado.



Figura 1. Robot manipulador R-2000iB.

2.2. Controlador del robot manipulador.

El controlador para este tipo de robot es el denominado R-30iB de la marca FANUC. Este tipo de controlador cuenta con diferentes tipos de armarios acorde a los ambientes de trabajo. Cuenta con una integración flexible ya que posee 3 puertos Ethernet. Otra característica importante es el ahorro de energía gracias a una eficiencia energética optimizada [5]. Es sencillo de utilizar gracias al fácil manejo que ofrece el *Teach Pendant* que puede ser táctil, lo que le otorga la característica de ser intuitivo. En la figura 2 se muestra el controlador utilizado.



Figura 2. Controlador de robot R-30iB.

2.3. Controlador Lógico Programable (PLC).

El PLC es un aparato electrónico el cual cuenta con una memoria programable para el almacenamiento interno de instrucciones las cuales son programadas por el usuario, e implementa funciones específicas como lógicas, secuenciales, de temporización, conteo, aritméticas, entre otras [6]. Utiliza módulos de entradas/salidas digitales y analógicas para realizar el control de diversas maquinas o procesos industriales. La programación se realiza mediante una computadora o estación de trabajo, en un software específico desarrollado por el fabricante del PLC. El PLC que se utilizó es de la marca Allen Bradley modelo ControlLogix 5573, el cual se muestra en la figura 3.



Figura 3. PLC ControlLogix 5573.

2.4. Switch de red de área local (LAN).

Un *switch* o conmutador es un dispositivo de interconexión utilizado para conectar equipos dentro de una misma red, a la cual se le conoce comúnmente como red de área local (LAN) [7]. Las redes locales cableadas siguen el estándar Ethernet, en el cual se utiliza una topología en estrella, en este tipo de topología el *switch* es el elemento central de la red. El *switch* utilizado es de la marca Allen Bradley modelo Stratix 5700, este modelo cuenta con una amplia versatilidad para ambientes industriales. Se le pueden conectar diferentes tipos de dispositivos de comunicación industrial habilitados para Ethernet, como PLC, interfaces operador-maquina (HMI), variadores, sensores y E/S. La figura 4 muestra la arquitectura de red utilizada.

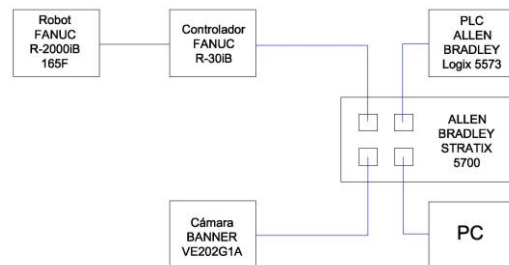


Figura 4. Arquitectura de red.

3. PROGRAMACIÓN EN SOFTWARE.

3.1. RSLinx.

El software RSLinx es propiedad de la marca Allen Bradley y se utiliza para crear la comunicación entre los dispositivos conectados a la red de área local mediante el protocolo Ethernet. Posee un entorno muy fácil de usar, el cual se muestra en la figura 5.

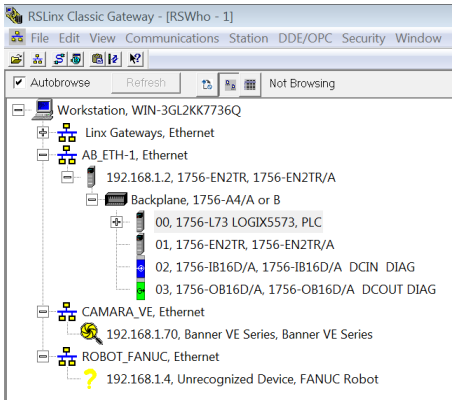


Figura 5. Ventana principal de RSLinx.

Los dispositivos como PLC y robot se dan de alta mediante la introducción de su dirección IP, el software RSLinx los identifica y reconoce, y de esta forma, poder utilizarlos dentro del software de programación.

3.2. Studio 5000.

El software Studio 5000 es propiedad de la marca Allen Bradley y se utiliza para desarrollar la programación en escalera necesaria para controlar el proceso. Para poder hacer uso de los módulos del PLC es necesario dar de alta cada uno de ellos dentro de este software. Se debe dar de alta un nuevo módulo y elegir dentro de un catálogo precargado el módulo deseado, así como también seleccionar ciertas propiedades del módulo. Para este caso en específico se dio de alta un módulo Ethernet. El módulo dado de alta se muestra en la figura 6.

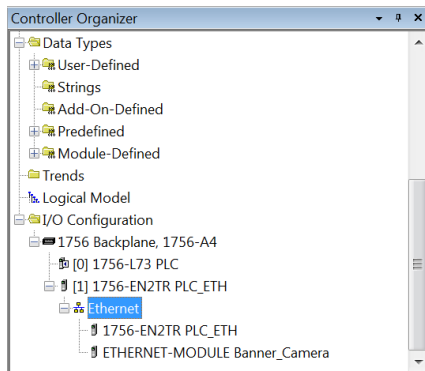


Figura 6. Creación de módulo Ethernet.

El siguiente módulo en dar de alta es el robot manipulador. Debido a que ya fue reconocido por el software RSLinx, el módulo del robot aparece por defecto en el catálogo precargado mostrado en la figura 7.

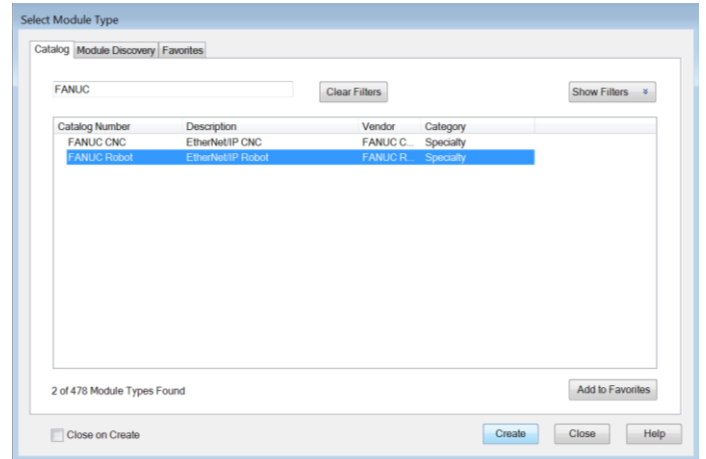


Figura 7. Catálogo precargado de dispositivos disponibles.

Una vez que el módulo del robot se ha dado de alta, es posible agregar las librerías denominadas *Add-On* para el mapeo de entradas y salidas digitales del robot. Estas librerías son importadas al software Studio 5000. En la figura 8 se muestran las librerías dentro de la carpeta *Data Types*.

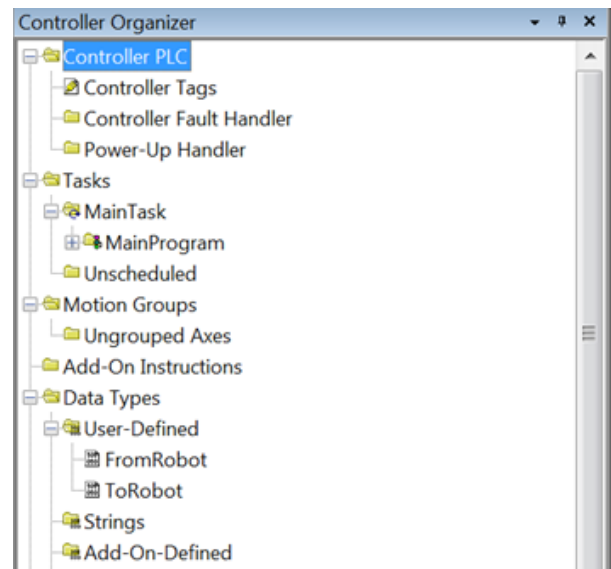


Figura 8. Importación de librerías Add-On.

Una vez que se han importado las librerías correspondientes, ya es posible mandar llamar en el diagrama escalera las entradas y salidas digitales del robot.

3.3. Software del controlador del robot.

La programación necesaria para el robot se realiza mediante el *teach pendant*. Es necesario crear un *tool frame* o marco de herramienta definido por el usuario. Por lo que se debe calibrar este marco con respecto al espacio de trabajo de la aplicación. Para poder crear el *tool frame* deseado fue necesario hacer uso del método de los tres puntos, para el cual se debe marcar un mismo punto con tres orientaciones diferentes y memorizar dichas posiciones en el *teach pendant* [8]. El resultado del marco de herramienta creado se muestra en la figura 9.

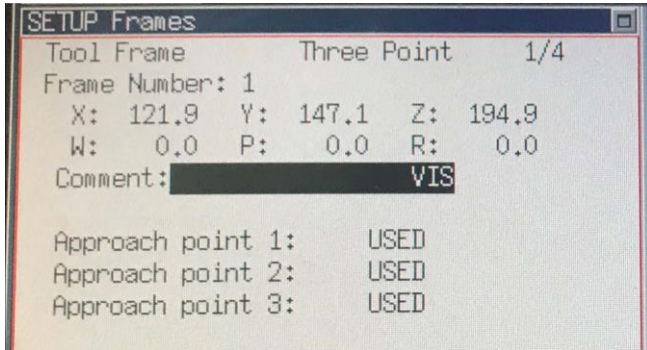


Figura 9. Ventana de marco de herramienta creado por el usuario.

Una vez definido el *tool frame* es necesario crear un *user frame* o sistema de referencia de usuario. Este marco se crea de la misma forma que el marco de herramienta, en base al espacio de trabajo deseado. El software maneja diferentes tipos de configuración, sin embargo, se utilizó el método de los tres puntos. El primer paso es memorizar la posición del origen de la referencia. El segundo paso es memorizar la dirección y sentido del eje X, y por último memorizar la dirección y sentido del eje Y [8]. El resultado del sistema de referencia creado se muestra en la figura 10.

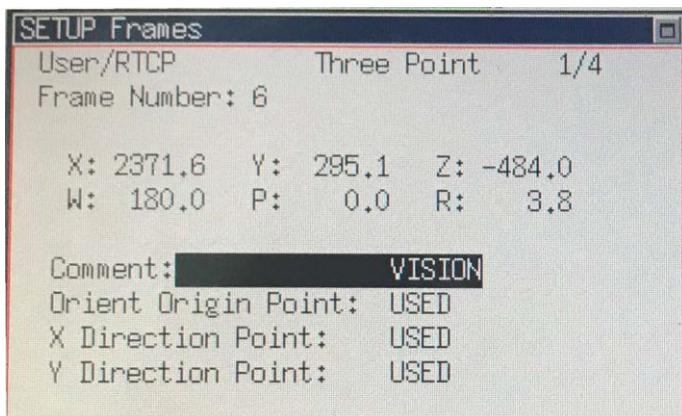


Figura 10. Ventana de sistema de referencia de usuario.

4. RESULTADOS.

4.1. Diagrama de escalera para registro de posiciones.

El diagrama de escalera desarrollado recibe los datos de las coordenadas X y Y así como el ángulo requerido, después estos datos los envía hacia registros del robot, los cuales son utilizados dentro de un programa en el controlador del robot, para que éste pueda dirigirse a la posición deseada.

Los datos de coordenadas y ángulo son adquiridos mediante un sistema de visión artificial, se procesan los datos y son enviados mediante el protocolo Ethernet hacia el PLC. La cámara del sistema de visión almacena los datos en formato decimal de 32-bits, por lo que en el primer peldaño del diagrama se realiza la conversión de decimal a formato real y poder así visualizar las medidas reales en milímetros y grados. En la figura 11 se muestran las instrucciones *copy* utilizadas para la conversión de los datos.

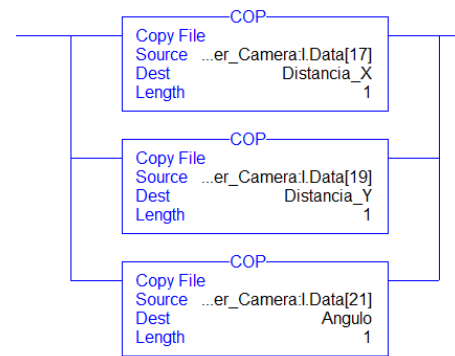


Figura 11. Diagrama de escalera para recepción de coordenadas.

En el segundo peldaño del diagrama mediante el uso de la instrucción *MOV*, se envían los datos de posición y ángulo hacia el controlador del robot. En la figura 12 se muestran las instrucciones *move* utilizadas para la posición y ángulo.

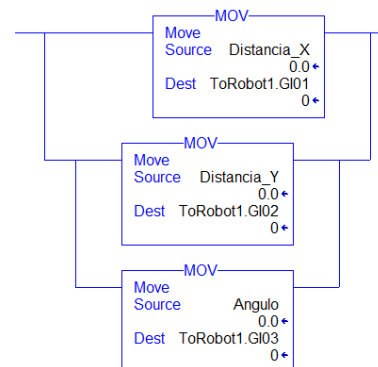


Figura 12. Diagrama de escalera para envío de coordenadas.

En el último peldaño se hizo uso de un par de instrucciones *copy* para que fuera posible el intercambio de información entre PLC y el controlador del robot, haciendo uso de las librerías *Add-On* que se dieron de alta previamente. En la figura 13 se muestran las instrucciones *copy* utilizadas para el intercambio de información.

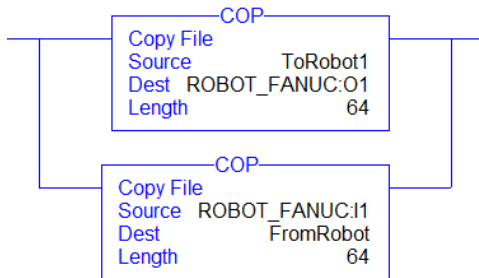


Figura 13. Diagrama de escalera para intercambio de información entre PLC y robot.

4.2. Programación de trayectoria en robot.

Los datos de posición y orientación enviados mediante el PLC deben ser almacenados en entradas y salidas de grupo, las cuales permiten acceder a los datos de varias señales de entradas o salidas digitales a la vez [8]. En la figura 14 se muestra la pantalla principal de entradas de grupo en el *teach pendant*.

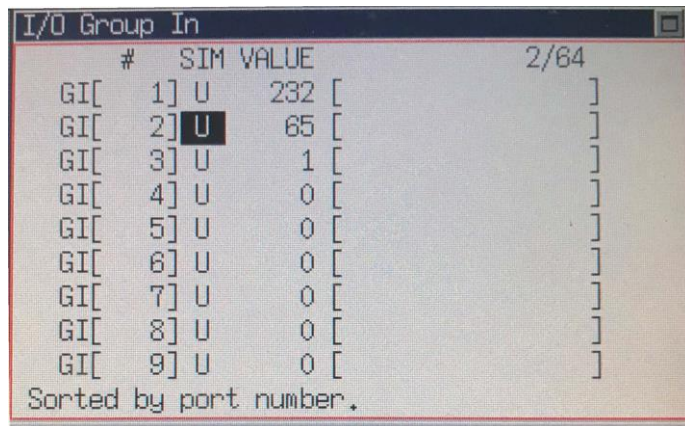


Figura 14. Ventana general de entradas de grupo.

Por lo tanto, la posición y orientación de la pieza se guardó en entradas de grupo y posteriormente la captura de estas entradas se realizó mediante un registro de posición, el cual contiene el valor decimal correspondiente al grupo de entradas digitales. Al ser guardado como registro de posición, se puede hacer uso directamente en el programa de la trayectoria deseada, lo cual reduce el tamaño del programa. En la figura 15 se muestra la pantalla de datos de un registro de posición.

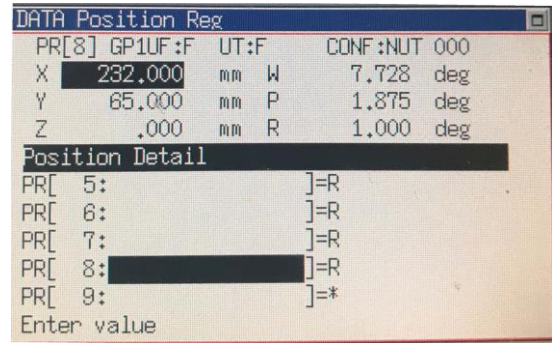


Figura 15. Ventana de datos de registro de posición.

Al inicio del programa de trayectoria se indica el número de marco de herramienta a utilizar, en este caso el marco de herramienta creado fue el número uno. De igual forma, se indica el sistema de referencia de usuario a utilizar, el sistema creado fue el número seis. A partir de esta línea, se comenzaron a programar los registros de posición, en los cuales se almacenaron los datos provenientes del PLC. En el registro de posición PR[8,1] se almacena la posición en el eje X, en el registro de posición PR[8,2] se almacena la posición en el eje Y, y en el registro de posición PR[8,6] se almacena la orientación, es decir, el ángulo de la pieza. Estos registros se estarán actualizando automáticamente cada que se detecte una pieza nueva. A continuación se enlista el fragmento de programa utilizado y en la figura 16, se muestra una pantalla del programa en el *teach pendant*.

```

1: UTOOL_NUM=1
2: UFRAME_NUM=6
3: PR[8,1]=GI[1]
4: PR[8,2]=GI[2]
.....
[End]
    
```

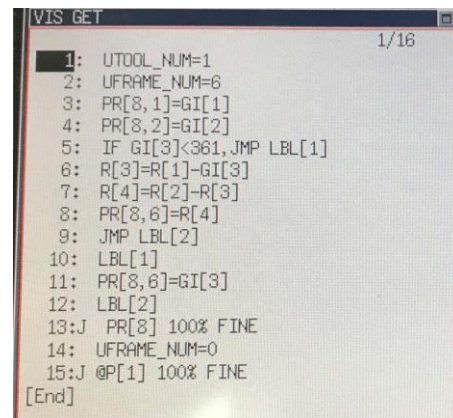


Figura 16. Ventana de programa de trayectoria del robot.

Se programó una condición con la instrucción IF, la cual evalúa si el ángulo de orientación es negativo o positivo. Si el ángulo resulta positivo, se utiliza una instrucción de salto de programa (JMP LBL) y se ejecuta la trayectoria almacenada. En cambio, si el ángulo resulta negativo, se realizan ciertas operaciones matemáticas para que en el programa se visualice el ángulo resultante en valores reales, ya que el controlador no reconoce valores negativos.

Por último, se programó que se regrese al sistema de referencia por default y que el robot regrese a su posición segura de *home*. En la figura 17 se muestra la realización de pruebas con piezas de forma rectangular; se varió la posición y ángulo de la pieza y el robot alcanza la posición y orientación deseadas. En el caso que la pieza fuera de otra forma, se tendrían que realizar los cambios correspondientes en el sistema de visión, para que la cámara pueda detectar la posición y orientación de la nueva forma y enviar los datos hacia el PLC.



Figura 17. Prueba con producto.

Las pruebas realizadas confirman que este proyecto es muy útil en aplicaciones *pick and place*, y en general en cualquier aplicación que involucre el manejo de material, ya que dotando a la celda robótica con un sistema de visión artificial y un dispositivo maestro como el PLC que se encargue del procesamiento e intercambio de datos, facilita el proceso de programación en el robot; ahorrando tiempo y personal técnico al realizar esta tarea, incluso haciendo que el robot ejecute el programa con mayor rapidez, debido a la poca cantidad de renglones programados. Además, le permite a la celda ser más versátil ya que podrá manipular diferentes modelos de piezas sin la necesidad de programar su trayectoria por separado, otorgándole una mejor capacidad de producción al proceso.

5. CONCLUSIONES.

A escala industrial, se implementó el hardware y el software necesario para considerar la integración de dispositivos como PLC, robot manipulador y diversas librerías propias de los mismos, para un sistema de *pick and place* mediante coordenadas enviadas por un sistema de visión artificial. Fue necesario crear una red de comunicación mediante el protocolo Ethernet IP para poder comunicar los diferentes dispositivos entre sí. El dispositivo que actuó como maestro fue el PLC, mientras que el robot y el sistema de visión actuaron como dispositivos esclavos. El PLC es el que recibe y envía los datos de posición y orientación, por lo que la programación fundamental se realizó dentro de este dispositivo. Los resultados demuestran que la recepción, conversión y envío de datos se realizan de manera satisfactoria, haciendo que el robot alcanzara el objetivo deseado, sin la necesidad de programar cada una de las distintas trayectorias, sino recibiendo los datos de manera automática mediante el PLC. Esta aplicación puede ser extensible a otro tipo de robot, por ejemplo con un robot delta, en caso de requerir mayor rapidez. La variante en este caso sería la librería *Add-On* para entradas y salidas del robot delta, se debería mapear todas las entradas y salidas del controlador hacia el PLC; y una vez hecho esto, ya se podrían utilizar libremente en el desarrollo del programa escalera para el envío y recepción de datos. La programación de trayectorias del robot también cambiaría ya que el software que maneja cada fabricante de robots es diferente, sin embargo, el entorno es muy similar.

6. AGRADECIMIENTOS.

Carmen Patricia Hernández Marías agradece el apoyo de la beca No. 888533 del CONACyT y parcialmente al proyecto de TecNM 6772.18-P convocatoria 2018-2.

7. REFERENCIAS.

- [1] Tornil, Sebastián y Gámiz, Juan (2014). "La robótica industrial en el ámbito de la automatización global: estado actual y tendencias". *Técnica Industrial*, 306, 26-39.
- [2] International Federation of Robotics. (2017). "*Executive Summary World Robotics 2017 Industrial Robots*". Frankfurt, Germany: World Robotics.
- [3] Delgado, R. "¿Cómo se lleva a cabo la aplicación del sistema pick and place en robótica?". *Revista digital INESEM*. [En línea]. Available: <https://revistadigital.inesem.es/gestion-integrada/pick-and-place/> [Último acceso: Mayo 2019]
- [4] <<Robots pick and place, innovación y necesidad>>. [En línea]. Available: <https://techpress.es/robots-pick-place-innovacion-y-necesidad/#> [Último acceso: Mayo 2019]
- [5] FANUC Europe Corporation (2017). Gama de Robots. [En línea]. Available: <https://www.fanuc.eu/es/es/robots> [Último acceso: Mayo 2019]
- [6] Vallejo, Horacio (2012). "Los controladores lógicos programables". *Saber Electrónica*. No. 166.
- [7] Tanenbaum, A. & Wetherall, D. (2012). *Redes de computadoras*. México. Editorial: Pearson Educación.
- [8] Fanuc Robotics (2003). Curso de programación TPE. *Fanuc Robotics Ibérica S.L. V3*.