

Design of a SCADA automation and integration system between a collaborative Robot (Cobot UR) and the HT-PLC platform

Inzunza Villagomez Héctor Iván, Highest Degree³, Pérez Arce Beatrice, Highest Degree³, Hernández Ruiz Sergio Iván, Highest Degree², López Corella José Alejandro, Highest Degree², Herrera Velarde Maribel, Highest Degree², Meranza Castellón Manuel Omar, Highest Degree¹, Vázquez Cuevas Ignacio Javier Highest Degree²

Tecnológico Nacional de México / I.T. de Nogales, Metal Mecánica Ingeniería Mecatrónica
Av. Tecnológico 911, Nogales, Sonora, México
+52(631) 311 1881. Ext 1129.
sergiohernandez@itnogales.edu.mx

Abstract- The present work describe the integration and development of an automation composed of a collaborative robot, the HT-PLC platform [1] and a SCADA system (Supervisory Control And Data Acquisition).

The general objective of this work is to introduce the HT PLC platform in industrial processes to determine if it is a viable alternative as a PLC. For this, the platform has been subjected to controlling a simple automated pick & place process in collaboration with a COBOT UR.

The HT PLC platform is composed of an IDE (Integrated Development Environment) developed in c#, which allows programming a PLC prototype based on the Arduino development board using ladder language. Similarly, tools for monitoring, control and data acquisition have been developed, for the platform and for mobile devices, which can be used remotely within a LAN (Local Area Network), this in order to comply with trends within today's industry, such as the internet of things and Industry 4.0

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2021.1.1.49 ISBN: 978-958-52071-8-9 ISSN: 2414-6390
--

Diseño de un sistema de automatización e integración SCADA entre un Robot colaborativo (Cobot UR) y la plataforma HT-PLC.

Design of a SCADA automation and integration system between a collaborative Robot (Cobot UR) and the HT-PLC platform

Inzunza Villagomez Héctor Iván, Highest Degree³, Pérez Arce Beatrice, Highest Degree³, Hernández Ruiz Sergio Iván, Highest Degree², López Corella José Alejandro, Highest Degree², Herrera Velarde Maribel, Highest Degree², Meranza Castellón Manuel Omar, Highest Degree¹, Vázquez Cuevas Ignacio Javier Highest Degree²

Tecnológico Nacional de México / I.T. de Nogales, Metal Mecánica Ingeniería Mecatrónica
Av. Tecnológico 911, Nogales, Sonora, México
+52(631) 311 1881. Ext 1129.
sergiohernandez@itnogales.edu.mx

Resumen- En el presente trabajo se describe la integración y el desarrollo de una automatización compuesta por un robot colaborativo, la plataforma HT-PLC [1] y un sistema SCADA (Supervisory Control And Data Acquisition). El objetivo general de este trabajo es introducir a la plataforma HT PLC en procesos industriales para determinar si es una alternativa viable como PLC. Para esto se ha sometido la plataforma a controlar un proceso automatizado simple de pick & place en colaboración con un COBOT UR. La plataforma HT PLC está compuesta por una IDE (Integrated Development Environment) desarrollada en c#, la cual permite programar un prototipo de PLC basado en la placa de desarrollo Arduino usando lenguaje escalera. De igual forma, se han desarrollado herramientas de monitoreo, control y adquisición de datos, tanto para la plataforma como para dispositivos móviles, las cuales pueden ser utilizadas de forma remota dentro de una LAN (Local Area Network), esto con el fin de cumplir con las tendencias dentro de la industria actual, como el internet de las cosas (IoT) y la industria 4.0.

Palabras Clave: Arduino, C#, SCADA, PLC, COBOT UR.

Abstract- The present work describes the integration and development of an automation composed of a collaborative robot, the HT-PLC platform [1] and a SCADA system (Supervisory Control And Data Acquisition).

The general objective of this work is to introduce the HT PLC platform in industrial processes to determine if it is a viable alternative as a PLC. For this, the platform has been subjected to controlling a simple automated pick & place process in collaboration with a COBOT UR.

The HT PLC platform is composed of an IDE (Integrated Development Environment) developed in c#, which allows programming a PLC prototype based on the Arduino development board using ladder language. Similarly, tools for monitoring, control and data acquisition have been developed, for the platform and for mobile devices, which can be used remotely within a LAN (Local Area Network), this in order to comply with trends within today's industry, such as the internet of things and Industry 4.0

I. INTRODUCCION

El HT-PLC es una plataforma que se encuentra en desarrollo desde el año 2016, y ofrece una IDE desarrollada en C# que permite programar en lenguaje en escalera un prototipo de PLC basado en la placa de desarrollo Arduino.

Esta plataforma fue publicada por primera vez en el ELECTRO 2018 (Congreso Internacional de Ingeniería Electrónica) [1]. En el mes de octubre del mismo año se publicó en la IEEE ICA-ACCA (International Conference of Automation/ Congress of the Chilean Association of

Automatic Control) una nueva versión del IDE y del prototipo de PLC[2]. La última publicación fue en el año 2019, nuevamente en el congreso ELECTRO del mismo año [3].

Actualmente la plataforma permite al usuario crear diagramas en escalera con herramientas básicas, tales como contactos abiertos, cerrados, contadores, temporizadores, entre otros. Los diagramas creados en este software pueden ser interpretados por un prototipo de PLC basado en la placa de desarrollo Arduino, esto le permite operar como cualquier otro PLC.

El objetivo inicial de la plataforma fue tener una alternativa didáctica de otros PLC, teniendo como ventaja un menor costo de adquisición, no obstante, actualmente el objetivo es introducir la plataforma en procesos y entornos industriales para que en el futuro también permita ser una alternativa como PLC para las industrias.

Este trabajo detalla el primer proceso automatizado en el cual se implementa la plataforma HT-PLC con un robot colaborativo (COBOT UR), y se muestran las nuevas herramientas desarrolladas para la plataforma que permiten el monitoreo, control y adquisición de datos de forma remota dentro de una red local. También se ha integrado una nueva aplicación para dispositivos móviles, desarrollada en lenguaje de programación Python que ofrece estas últimas herramientas.

El proceso automatizado consiste en una tarea de *pick & place*, donde el brazo colaborativo se encarga del traslado de unas cajas, con la opción de poder cambiar su carga útil con el fin de adaptarse al peso de las mismas, el cual es determinado por el prototipo de PLC. El software del HT-PLC trabaja como servidor y se permite controlar, monitorear y adquirir datos del proceso. De igual forma, mediante una aplicación se puede conectar un dispositivo móvil para realizar el monitoreo y visualizar los datos adquiridos.

II. MARCO DE REFERENCIA

A. HT-PLC.

HT-PLC es una plataforma de que se compone de un IDE (*Integrated Development Environment*) y un prototipo de PLC.

El IDE HT-PLC es un software desarrollado en el entorno de desarrollo de Visual Studio, en el lenguaje de programación C#, y sus funciones principales son el monitoreo, control, adquisición de datos y el diseño de diagramas en escalera para la programación de la placa Arduino.

El prototipo de PLC se basa en la placa de desarrollo Arduino y cuenta con una interfaz de usuario que permite navegar entre las opciones que tiene este PLC, tales como opciones de ejecución y conexión. De igual forma este prototipo cuenta con componentes físicos que le permiten operar en un entorno industrial.

B. Arduino.

Arduino es una plataforma de desarrollo de código abierto, es decir, las características, componentes, funcionamiento y especificaciones son de acceso público, de manera que cualquier persona puede destinar o modificar estos elementos de la forma que se requiera.

En el IDE, los usuarios pueden programar con una computadora el microcontrolador que se encuentra en una placa de circuitos impresos, para controlar múltiples sensores y/o actuadores. Este IDE está basado en el entorno de *Processing* y en la estructura del lenguaje de programación *Wiring* [4,5].

C. Controlador lógico programable y diagrama en escalera.

Los PLC son autómatas diseñados específicamente para operar de manera segura en entornos y condiciones industriales, como zonas con vibraciones, temperaturas extremas y condiciones de humedad o polvo. La función más común que cumplen estos sistemas es el control automático de procesos electromecánicos en distintas industrias, y se caracterizan por su versatilidad para adaptarse a pequeños o grandes cambios en un proceso ya establecido.

El diagrama en escalera es un lenguaje de programación gráfico comúnmente utilizado para la programación de los PLC. Este lenguaje se basa en esquemas eléctricos de control clásicos [6,7,8].

D. Visual Studio.

Microsoft Visual Studio es una aplicación informática desarrollada por Microsoft que ofrece servicios integrales para el desarrollo de páginas web y software que trabajen con el sistema operativo Windows. Este IDE incluye compiladores, herramientas de autocompletado de código, herramientas de depuración, y más funciones para facilitar el proceso de desarrollo de software.

Este entorno soporta el uso de diversos lenguajes de programación, por ejemplo: C, Visual Basic, C++, F#, JavaScript, TypeScript, Python y R, que ya están incorporados en el programa [9], no obstante, existen extensiones que expanden las opciones del usuario al momento de querer hacer uso de otros lenguajes de programación [10].

E. Python.

Python es un lenguaje de programación potente, de alto nivel y multiparadigma [11]. Su popularidad ha incrementado en los últimos años por su –relativamente– baja curva de aprendizaje y menor tiempo de desarrollo de aplicaciones en relación a otros lenguajes [12], en vista de que su aprovechamiento puede garantizar soluciones más cortas y con buena legibilidad de código.

Este lenguaje de programación es caracterizado por su tipado dinámico, sintaxis clara y sencilla, el soporte de módulos y de paquetes [11]. Además de su licencia de código abierto, la cual permite la distribución de programas publicados y realizados en Python para propósitos comerciales [13].

F. Arquitectura cliente-servidor.

La arquitectura de software cliente-servidor es definida como la división de carga de trabajo entre los elementos de un sistema que cumplen el papel de cliente o servidor [14].

El cliente es el componente que da principio a la comunicación en el sistema, enviando una petición al servidor para luego esperar una respuesta.

El servidor, por su parte, es el componente cuyo trabajo es esperar una conexión de un cliente y –si es aceptada y mantenida– proveer acceso a los servicios que ofrece y entregar lo que sea requerido de él [15].

Los servicios pueden ser variados dependiendo de la capacidad del sistema, puesto que se pueden transferir desde mensajes hasta archivos con mayores requerimientos de almacenamiento (imágenes, videos y audios).

G. Robots colaborativos UR CB3.

Los robots colaborativos, también conocidos como “cobots”, nacieron de la necesidad de crear una herramienta que ayudara al ser humano a completar tareas repetitivas y tediosas, llevando a cabo dichas actividades en un entorno compartido en donde la seguridad del factor humano no se vea comprometida por la presencia de brazos robóticos sin barreras físicas a su alrededor [16].

Además de distinguirse por su capacidad de cooperación y riesgo mínimo, son fáciles de programar e instalar por su tamaño y peso [16].

Los robots colaborativos de Universal Robots, específicamente la familia CB3, son brazos robóticos con 6 ejes de libertad diseñados para automatizar diversos procesos, dada su adaptabilidad, así como operaciones de paletizado, *pick & place*, atornillado, entre otros.

TABLA I
TABLA DE COBOTS UR CB3[17].

Modelo	Peso	Carga útil	Alcance
UR3	11 kg	3 kg	500 mm
UR5	18,4 kg	5 kg	850 mm
UR10	28,9 kg	10 kg	1300 mm

Las diferencias técnicas más importantes entre los tres modelos quedan en su carga útil, peso y alcance [17], como se muestra en la tabla I. La precisión en la repetición de movimientos, límite de velocidad de operación, diseño mecánico y software de funcionamiento/ programación son iguales.

Los cobots UR presentan una característica especial que los han hecho destacar y ganar una posición en el mercado entre la competencia. Este factor ha sido la implementación de una pantalla táctil de 12 pulgadas con el software de Polyscope, desarrollado por Universal Robots, el cual actúa como la interfaz gráfica de usuario (GUI, por sus siglas en inglés) para la programación y operación del cobot y caja de control [18]. Lo que hace especial a este interfaz es que se acceden a todas las funciones disponibles del sistema robótico sin

dificultad y es muy intuitivo de usar, incluyendo un apartado para la representación 3D de los movimientos del robot.

III. SUPERVISIÓN, CONTROL Y ADQUISICIÓN DE DATOS CON HT-PLC

A. Descripción y funcionamiento.

El IDE HT-PLC es un software que ofrece herramientas para programar, monitorear, controlar y adquirir datos de una placa de desarrollo Arduino. La programación se realiza en lenguaje en escalera y el monitoreo, control y adquisición de datos se puede llevar a cabo de forma remota mediante conexión wifi o ethernet.

En la imagen 1 se puede visualizar la pantalla de inicio, donde se tiene acceso a todas las funciones que ofrece el entorno de desarrollo.



Figura 1. Menú de inicio.

En sección de la caja de información se muestran notificaciones y mensajes que permiten al usuario conocer cuando se ha llevado distintos eventos y el estado de estos. De igual forma pueden facilitar u orientar al usuario en el uso de este entorno, ya que también se muestra información sobre las distintas herramientas y consejos de su uso.

En la cinta de opciones localizada en la parte superior se puede acceder a accesos directos, herramientas, ayuda y a opciones de configuración de conexión para la programación del PLC y para la comunicación por wifi o ethernet.

En la zona de menú principal y menú rápido, se encanutaras las funciones que permiten crear diagramas en escalera y las funciones que permiten configurar, controlar y monitorear el PLC.

B. Configuración de conexión.

Para llevar a cabo el monitoreo del PLC desde este IDE es necesario configurar la conexión a internet del programa. La opción de “Conexión a internet” dentro de la cinta de opciones, como se muestra en la imagen 2, permite al usuario configurar el puerto de conexión, conectarse, abrir el

puerto seleccionado y seleccionar si solo se requiere realizar un monitoreo o ejecutar el sistema SCADA.

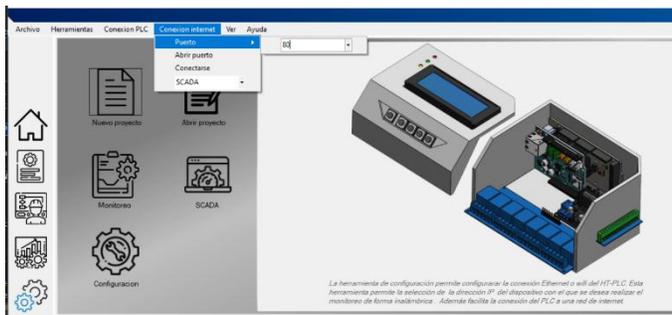


Figura 2. Configuración de conexión.

Para la configuración WIFI del PLC es necesario configurar el módulo de wifi integrado dentro del prototipo, para que este sea capaz de conectarse a una red. Para realizar esta tarea previamente el PLC debe estar conectado a la computadora, posteriormente se debe abrir la herramienta de configuración e ingresar los datos de la red en la que se desea conectar y la dirección IP y el puerto del dispositivo donde se realizará el monitoreo, tal como se ilustra en la figura 3. A un costado del cuadro de texto donde se ingresa la dirección IP, se encuentra un Check Box que al seleccionarlo permite escribir automáticamente la dirección IP de la computadora en la cual se está realizando la configuración.

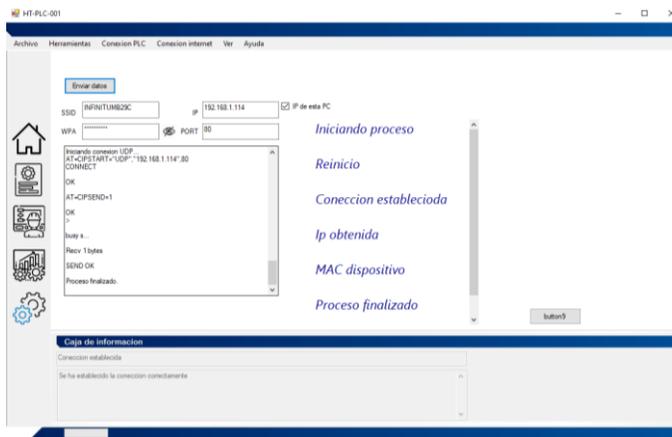


Figura 3. Configuración para conexión a red.

Una vez ingresados todos los datos requeridos se presiona el botón “enviar datos” para dar inicio a la configuración del módulo, mientras se lleva a cabo se pueden visualizar mensajes del estado del proceso de conexión y al finalizar en la caja de información se notifica si la conexión se ha establecido correctamente o si se ha producido un error.

C. SCADA.

En la imagen 4 se muestra esta herramienta, que al abrirla se puede visualizar distintas zonas donde el usuario puede monitorear y controlar el PLC de forma remota, y si lo desea

puede iniciar un proceso de adquisición de datos, el cual puede ser configurado para adaptarse a las necesidades del usuario.

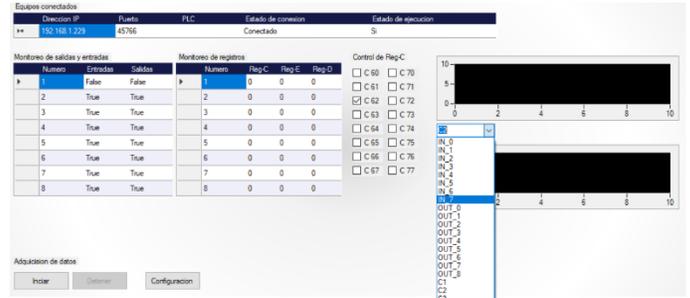


Figura 4. Sistema SCADA del software.

En la parte superior de esta ventana se encuentra la sección de conexión, la cual es una tabla que muestra información de todos los equipos conectados, donde se puede visualizar la dirección IP, el puerto y en el caso de que se conecten PLC, también se puede visualizar si este se encuentra en estado de ejecución o no.

Para iniciar el monitoreo y control del PLC de forma remota es necesario ejecutar un programa previamente cargado desde el prototipo y si este ya se encuentra configurado, el monitoreo inicia de forma automática, mostrando en pantalla la información del PLC y el estado de las entradas y salidas.

D. Monitoreo.

En la sección de monitoreo el usuario puede visualizar el estado actual de las entradas y salidas del PLC. Esta sección se compone de una tabla de 3 columnas y 8 filas. La primera columna indica el número de entrada y salida. La segunda columna indica el estado de las entradas. Y la tercera columna indica el estado de las salidas.

Esta misma estructura se repite para el monitoreo del resto de registros del PLC. Estos le permiten al usuario conocer el estado de las bobinas internas, de los temporizadores y contadores del prototipo.

En el costado derecho de esta imagen se encuentran dos herramientas de gráficas, en las cuales el usuario puede seleccionar que elementos desea graficar.

E. Control.

La sección de control permite controlar de forma remota los valores de los registros C 60 a C 77, esto según el programa que se esté ejecutando, permite al usuario controlar de forma remota el flujo del programa, el control de salidas específicas o controlar cualquier otra función que el usuario requiera.

F. Adquisición de datos.

Para llevar a cabo la adquisición de datos es necesario realizar tener un perfil de configuración. Este perfil se puede crear y guardar para posteriormente reutilizarlo. Dentro del perfil de configuración de adquisición de datos se guarda la dirección donde se creará el archivo con los datos recopilados, el

tiempo que se realiza para la adquisición y cuantos datos o ciclos se tomará, entre otras opciones.

Al presionar el botón de “Configuración” dentro de la sección de adquisición, se abre una ventana dividida en tres secciones a las que se puede acceder mediante la pestaña de la parte superior, como se muestra en la figura 5.

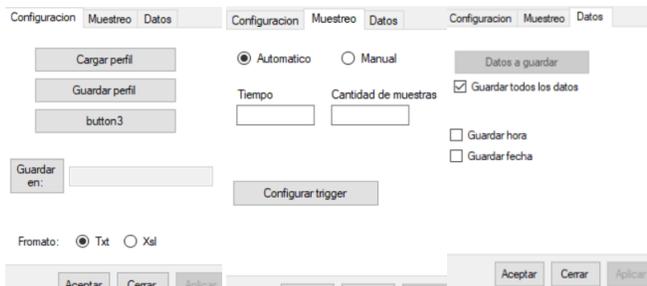


Figura 5. Configuración de adquisición de datos.

Dentro de las opciones de configuración, el usuario puede elegir guardar los datos en un archivo de texto o una hoja de Excel, en donde se va a guardar, que registros se guardaran durante cuánto tiempo y el la frecuencia en la que se tomará cada adquisición, si la adquisición se tiene de forma manual o automática, y por último, el usuario asociar datos de registro con opciones de disparo dentro de la configuración de *Trigger*, para dar inicio o detener la adquisición según las necesidades de la tarea a realizar.

IV. SUPERVISIÓN Y ADQUISICIÓN DE DATOS EN DISPOSITIVOS MOVILES

A. Descripción y funcionamiento.

La interfaz gráfica de usuario fue desarrollada haciendo uso de un marco de trabajo para Python denominado Kivy, dedicado a la creación de aplicaciones que son compatibles en diferentes sistemas operativos, tales como: Windows, Linux, Android, iOS, entre otros [20].

La aplicación móvil se compone de tres ventanas en las que la interacción del usuario es limitada, puesto que el usuario únicamente posee las opciones de elegir qué proceso desea observar en el momento: el monitoreo o la adquisición de datos del proceso. A continuación, se describirán las pantallas con mayor detalle.

B. Ventana principal.

La ventana principal aparece cada vez que la aplicación es abierta y se retorna de una opción posterior. Se constituye de cuatro elementos sencillos; primero, el título de la aplicación “HT-PLC”; segundo, un botón en la esquina superior derecha para cerrar correctamente la aplicación; tercero, el botón de “Monitoreo de robot UR”, al cual le corresponde llevar al usuario a la ventana dedicada al monitoreo en tiempo real del proceso del robot UR3; y, por último, el botón de “Adquisición de datos”, cuyo papel es accionar el traslado a una ventana que le corresponde mostrar los últimos registros del sistema automatizado.

C. Monitoreo.

La pantalla asignada a la demostración de la supervisión del proceso inicia la comunicación con el servidor y procede a solicitar el número de piezas que han pasado hasta el momento, dividiendo estos elementos en tres grupos de acuerdo al peso que poseen, que son de 1kg, 2kg y 3kg. De igual manera, se requiere de la carga actual con la que se encuentra trabajando el robot UR3. Posteriormente, al recibir todos los datos, la comunicación con el servidor es finalizada y la información es procesada para luego ser revelada en pantalla. Mientras se continúe observando la ventana, se seguirán pidiendo los datos continuamente hasta que sea presionado el botón “Regresar” y terminar con la petición de servicios al servidor. La aplicación que permite este monitoreo se muestra en la imagen 6.

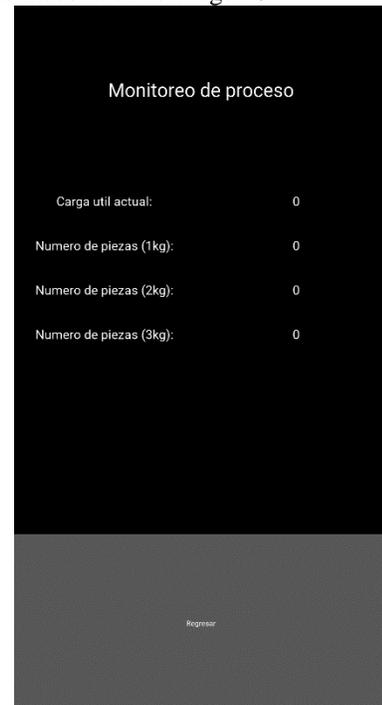


Figura 6. Ventana de monitoreo de proceso en el robot UR3.

D. Adquisición de datos.

La ventana que proporciona los datos adquiridos al usuario (Figura 7) también comienza una conexión con el servidor, al cual le pide los últimos cinco datos guardados en un archivo separado por comas (mayormente conocido como documentos CSV, por sus iniciales en inglés de *Comma Separated Values*), encargado de almacenar la información del proceso, correspondiente al conteo de piezas con respecto a sus pesos, la carga útil de operación del robot UR3 y asimismo la fecha y hora en la que los datos fueron recolectados. Una vez reunida la información y transformada a un formato para su lectura, es mostrada en pantalla. Al igual que en la supervisión del proceso, solo se dejará de hacer las peticiones al servidor una vez que el botón “Regresar” sea tocado por el usuario.

Carga útil actual (1kg)	Número de piezas (2kg)	Número de piezas (3kg)	Fecha	Hora
0	0	0	00/00/00	00:00:00
0	0	0	00/00/00	00:00:00
0	0	0	00/00/00	00:00:00
0	0	0	00/00/00	00:00:00
0	0	0	00/00/00	00:00:00

Figura 7. Ventana de adquisición de datos del proceso.

V. SISTEMA AUTOMATIZADO

A. Descripción y funcionamiento.

El sistema de automatización consiste en la participación de cuatro recursos físicos y digitales en una red LAN, con el fin de lograr una interacción exitosa entre ellos y que reaccionen de manera inmediata a los cambios hechos por el usuario. El sistema se basa en una arquitectura cliente-servidor, dividiendo la carga de trabajo y atribuyendo al servidor la tarea de entregar los servicios requeridos por los clientes.

El proceso a controlar e inspeccionar se centra en una tarea sencilla de *pick & place* que realiza un robot colaborativo UR3, responsable de tomar una pieza con una masa determinada y dejarla en otro sitio.

B. Servidor.

En este caso se tienen dos opciones para usar como servidor. El primero es el IDE de la plataforma HT PLC que se mostró en la sección 2 y la segunda opción es un programa sin interfaz gráfica que sirve como alternativa en el caso de que el usuario no desee utilizar las herramientas del IDE.

Esta alternativa de servidor es una computadora a la que se le encomienda la ejecución de un programa, hecho en Python, que identifica la dirección IP de los clientes (HT-PLC, UR3 y la aplicación de teléfono celular) y administra sus solicitudes, concediendo acceso y retornando la información que piden. Igualmente, efectúa el almacenamiento del proceso efectuado por el cobot UR3 en un archivo CSV, dependiendo del periodo de tiempo con el que sea configurado. El documento de adquisición de datos incluye la carga útil con la que trabajaba el cobot UR3, los

elementos que pasaron por el proceso –separados por su masa– y la fecha y hora de obtención.

Al establecer el funcionamiento del servidor, puede observarse que actúa como el núcleo del sistema de automatización, en vista de que los clientes precisan de sus servicios para la extracción de datos de otros componentes. En virtud de esto, es el único factor del sistema que siempre debe estar en ejecución.

C. Clientes.

D. HT-PLC.

En este proceso el PLC opera como un cliente del servidor y tiene como función enviar las señales que se encuentra en sus periféricos, tanto de entrada como de salida. En este caso el PLC envía al servidor cuando se ha detectado una caja y cuál es su peso.

E. Robot colaborativo UR3.

Como fue mencionado con anterioridad, el cobot UR3 es el brazo robótico que lleva a cabo el traslado de piezas con una masa de 1kg, 2kg o 3kg de un lugar a otro. Para conocer con que carga útil se trabajara en el momento, la caja de control del robot se comunica constantemente con el servidor y verifica si se ha hecho algún cambio. Si no se recibe un valor diferente de la computadora, el robot colaborativo sigue operando con la misma carga útil. En caso contrario, ejecuta sus movimientos tomando en cuenta la masa correspondiente a las piezas nuevas de trabajo.

Cada vez que el brazo robótico envía una solicitud a la computadora remota, le remite el número de partes que han pasado por el proceso –clasificadas por su masa– y alimenta a los otros componentes del sistema de automatización, específicamente al servidor y a la aplicación móvil.

F. Aplicación móvil.

La aplicación para celulares, cuyo funcionamiento ya ha sido expuesto en un apartado anterior, es dependiente de la comunicación con el servidor, el cual le comparte la información de los otros componentes del sistema y de sí mismo para presentar de forma gráfica al usuario el monitoreo en tiempo real y la adquisición de datos.

VI. RESULTADOS.

El proceso se ha llevado a cabo mediante una simulación en el entorno de desarrollo virtual URsim que ofrece universal robots mediante una máquina virtual. El PLC fue adaptado para tomar las mediciones de una celda de carga y así medir el peso de las piezas que posteriormente trasladaría el robot. Una vez realizada la medición el PLC mandaba la información al IDE, el cual guardaba almacenaba el valor y enviaba el comando correspondiente al cobot UR3 para que tomara la caja e hiciera el cambio de carga útil para adaptarse al peso actual de la caja. Una vez trasladada la caja de posición, el brazo robot mandaba una señal al servidor para indicar que la operación había finalizado y

estaba en posición para un nuevo ciclo. El sistema del proceso es ilustrado con un diagrama en la imagen 8.

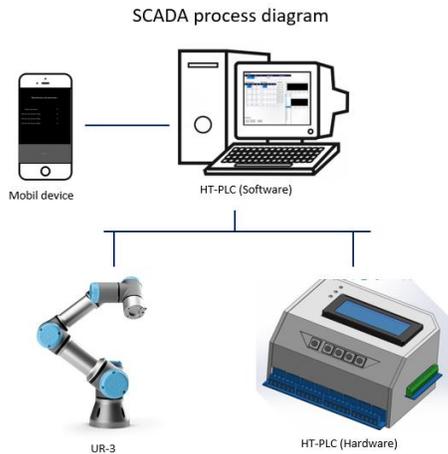


Figura 8. Datos registrados por la plataforma HT-PLC

La entrada física A1 del PLC se conectó un sensor para detectar las piezas y, dentro de la configuración de adquisición, esta entrada se configuro como *trigger* para el registro de datos. Los registros C1 y C2 se utilizaron para representar el peso medido y el registro de contador se utilizo para el conteo de las piezas.

En la Figura 9 se muestran los datos registrados del proceso, donde se probaron 15 piezas y en la imagen 10 se ilustra una gráfica del control estadístico del proceso, donde se consideran que las piezas con 3kg están fuera de los límites.

IN (A1)	C1	C2	CNT
1	0	1	1
1	1	0	2
1	0	1	3
1	1	1	4
1	0	1	5
1	1	0	6
1	1	0	7
1	0	1	8
1	1	0	9
1	1	0	10
1	1	1	11
1	0	1	12
1	0	1	13
1	1	0	14
1	0	1	15

Figura 9. Datos registrados por la plataforma HT-PLC

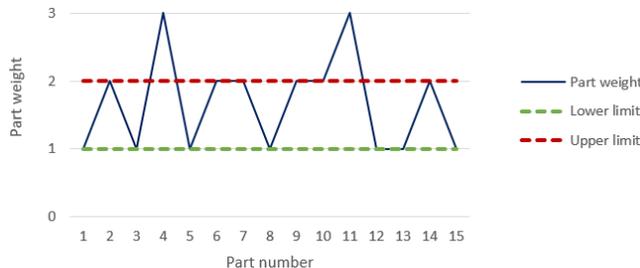


Figura 10. Control estadístico de proceso realizado.

En la tabla II se hace una comparación con dos controladores lógicos programables que se encuentran actualmente en el mercado y ofrecen características similares al prototipo de PLC que propone la plataforma HT-PLC.

TABLA II
COMPARACION DE COSTOS DEL HTPLC CON EQUIPOS [3].

Modelo	HT-PLC-001	LOGO! 0BA7	PLC Micro810
Entradas	8	12	8
Salidas	8	8	4
Herramientas básicas	5	8	
Herramientas especiales	2	30	
Costos de Hardware	450	2700	4700

CONCLUSIONES.

El control, monitoreo y la adquisición de datos del proceso simulado se llevó a cabo de forma exitosa. No obstante, es necesario someter a la plataforma en procesos de automatización reales dentro entornos industriales para concluir si esta será capaz de ser una alternativa viable para la industria en un futuro.

Sin embargo, los resultados obtenidos muestran que la plataforma, con las modificaciones adecuadas, es capaz de trabajar en conjunto con equipo que actualmente se utiliza en la industria. Esto indica que es necesario aumentar la versatilidad de las herramientas de conexión que ofrece la plataforma, para tener más facilidades al intentar comunicarse con otros equipos.

Siguiendo el objetivo general de este proyecto, en un futuro se pretende someter esta plataforma a procesos automatizados dentro de entorno industriales reales para identificar qué aspectos y herramientas, implementar, mejorar o cambiar, con el fin de ser una alternativa a los PLC que se encuentran actualmente en el mercado.

REFERENCIAS

- [1] Inzunza Villagomez Héctor Iván, diseño e implementación de un entorno de desarrollo de diagramas en escalera para Arduino, Congreso Internacional en Ingeniería Electrónica Electro 2018, Vol. 40, pp. 26-31, ISSN 1405-2172.
- [2] Inzunza Hector, Pérez Beatrice, Hernández Sergio, López José, Design and implementation of a development environment on ladder diagram (HT-PLC) for Arduino with Ethernet connection, ICA-ACCA 2018, October 17-19, Greater Concepción Chile, 978-1-5386-5586-3, pp. 1-6.

- [3] Pérez Arce Beatrice, sistema de monitoreo por wifi para el ht-plc utilizando el protocolo udp, Congreso Internacional en Ingeniería Electrónica Electro 2019, Vol. 41, pp. 26-31, ISSN 1405-2172.
- [4] <https://www.arduino.cc/>
- [5] Hernández R. Sergio, Sistema Distribuido con Arduino, wifi y LabView, utilizando el protocolo UDP, congreso Internacional en Ingeniería Electrónica Electro 2017, Vol. 39, pp. 22-26, ISSN 1405-2172.
- [6] <https://ab.rockwellautomation.com/>
- [7] https://books.google.com.mx/books/about/Programmable_Logic_Controllers.html?id=aVNIMwEACAAJ&redir_esc=y.
- [8] <https://www.idec.com/caen/products/Catalogs/Software/AO/overview.html>
- [9] <https://docs.microsoft.com/es-es/visualstudio/?view=vs-2019#pivot=languages>
- [10] <https://www.visualstudio.com/?rr=https%3A%2F%2Fwww.google.com%2F>
- [11] <https://www.python.org/doc/essays/blurb/>
- [12] <https://www.python.org/doc/essays/comparisons/>
- [13] <https://www.python.org/about/>
- [14] Ali, Sawsan & Alauldeen, Rana & Ruaa, Ali & Khamees,. (2020). What is Client-Server System: Architecture, Issues and Challenge of Client-Server System (Review). 10.5281/zenodo.3673071. pp 1
https://www.researchgate.net/publication/339340046_What_is_Client-Server_System_Architecture_Issues_and_Challenge_of_Client-Server_System_Review
- [15] Sulyman, Shakirat. (2014). Client-Server Model. IOSR Journal of Computer Engineering. 16. 57-71. pp 57, 58.
https://www.researchgate.net/publication/271295146_Client-Server_Model
- [16] Galin, Rinat & Meshcheryakov, Roman & Kamesheva, Saniya & Samoshina, Anna. (2020). Cobots and the benefits of their implementation in intelligent manufacturing. IOP Conference Series: Materials Science and Engineering. Pp 1.
- [17] <https://www.zacobria.com/universal-robots-zacobria-industrial-robot-ur5-ur10-specifications.html>
- [18] <https://www.universal-robots.com/cb3/>
- [19] https://www.universal-robots.com/media/50591/ur5_es.pdf
- [20] <https://kivy.org/doc/stable/gettingstarted/intro.html>