



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

SISTEMA DE CONTROL Y ADQUISICIÓN DE SEÑALES CON HARDWARE Y SOFTWARE LIBRE Y SU INCIDENCIA EN LA INDUSTRIA

PAMELA MONSERRATH ESPEJO VELASCO

Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo, presentado ante el Instituto de Posgrado y Educación Continua de la ESPOCH, como requisito parcial para la obtención del grado de:

MAGISTER EN SISTEMAS DE CONTROL Y AUTOMATIZACIÓN INDUSTRIAL

Riobamba - Ecuador

Abril - 2017



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

CERTIFICACIÓN:

EL TRIBUNAL DE TRABAJO DE TITULACIÓN CERTIFICA QUE:

El Trabajo de Titulación modalidad **Proyectos de Investigación y Desarrollo**, titulado: “SISTEMA DE CONTROL Y ADQUISICIÓN DE SEÑALES CON HARDWARE Y SOFTWARE LIBRE Y SU INCIDENCIA EN LA INDUSTRIA”, de responsabilidad de la Ing. Pamela Monserrath Espejo Velasco ha sido prolijamente revisado y se autoriza su presentación.

Tribunal:

ING. JUAN VARGAS GUAMBO; M.Sc.

PRESIDENTE

FIRMA

ING. DARÍO MENDOZA CHIPANTASI; M.Sc.

DIRECTOR

FIRMA

ING. ANDRÉS GORDÓN GARCÉS; M.Sc.

MIEMBRO

FIRMA

ING. PATRICIO ENCALADA RUIZ; M.Sc.

MIEMBRO

FIRMA

Riobamba, Abril – 2017

DERECHOS INTELECTUALES

Yo, Pamela Monserrath Espejo Velasco, declaro que soy responsable de las ideas, doctrinas y resultados expuestos en el **Trabajo de Titulación modalidad Proyectos de Investigación y Desarrollo**, y que el patrimonio intelectual generado por la misma pertenece exclusivamente a la Escuela Superior Politécnica de Chimborazo.

PAMELA MONSERRATH ESPEJO VELASCO

N° Cédula: 180320392-4

DEDICATORIA

Dedico este esfuerzo a Dios, mis padres Xavier y Tannia, a mis hermanos Obrian y Steven, a mi sobrino Josué, a mis abuelitos Jorge, Ruth, Edgar y Laura, a mis primos Vero, Maura, David, Andrés, Casandra, Erika, Mathiu, Doménica, Salomé, Damián, Edgar, Daniela, Oscarito, Lucianito y a todos mis tíos propios y políticos. Todas estas personas fueron mi motivación para seguir adelante y nunca darme por vencida.

Pamela Espejo

AGRADECIMIENTO

Agradezco en primer lugar a Dios ya que sin Él no hubiese llegado a donde estoy ahora, a mis padres por el apoyo incondicional en esta etapa de mi vida, a mi hermano Obrian Espejo quien si su ayuda no hubiese logrado terminar esta meta, a mi compañero de maestría Fernando Saá, Verónica Gallo, Wendy Romero y Hernán Ramos por todos sus consejos para no quedarme en este proceso y seguir adelante, a mis compañeros de trabajo Verónica Gutiérrez, Juan Ballesteros, Lenin Rodríguez y especialmente Freddy Manotoa por estar siempre pendientes de todo este proceso, al Ing. Juan Pablo Pallo e Ing. Manzano, por la ayuda en la implementación de este proyecto, a todos mis amigos por su apoyo incondicional, a la ESCUELA POLITÉCNICA SUPERIOR DE CHIMBORAZO por abrirme las puertas de su prestigiosa institución, a todos mis docentes que me enseñaron durante este año cinco meses todo lo necesario para estar preparada un nivel más para la vida laboral y a mis Tutores Ing. Darío Mendoza M.Sc, Ing. Patricio Encalada M.Sc e Ing. Andrés Gordón M.Sc por guiarnos en este proceso de titulación.

Pamela Espejo

TABLA CONTENIDO

PORTADA	i
CERTIFICACIÓN:	ii
DERECHOS INTELECTUALES	iii
DEDICATORIA	iv
AGRADECIMIENTO	v
TABLA CONTENIDO	vi
ÍNDICE DE TABLAS	ix
ÍNDICE DE FIGURAS	x
RESUMEN	xii
SUMMARY	xiii
CAPÍTULO I	xiii
1. INTRODUCCIÓN	1
1.1. Planteamiento del problema	1
1.2. Justificación de la investigación	1
1.3. Objetivos	2
1.3.1. Objetivo general	2
1.3.2. Objetivos específicos	2
CAPÍTULO II	3
2. Fundamentación Teórica	3
2.1. Estudio del Arte	3
2.2. Fundamentación Teórica	4
2.3. Definición de la Automatización	4
2.4. Pirámide de la Automatización	4
2.5. La Automatización de la fabricación	5
2.6. Tipos de Sistema de Automatización	6
2.6.1. Automatización Rígida	6
2.6.2. Automatización Flexible	8
2.7. Adquisición de datos	9
2.8. Sensores	10
2.8.1. Sensores Digitales	10
2.8.2. Sensores Analógicos	10

2.9.	Actuadores	11
2.10.	Controladores Lógicos Programables	11
2.10.1.	<i>Elección de un PLC</i>	12
2.11.	Protocolos de Comunicación	13
2.12.	Fiabilidad Industrial	13
2.12.1.	<i>EMC (Compatibilidad Electromagnética)</i>	13
2.12.2.	<i>Rohs (Estándar industrial para PLC)</i>	13
2.13.	Hardware y Software Libre en la industria.....	14
2.14.	PLCOpen	15
2.15.	Interfaz Humano-Máquina	16
 CAPÍTULO III		17
3.	METODOLOGIA INVESTIGATIVA	17
3.1.	Metodología	17
3.2.	Modalidad de Investigación.....	17
3.3.	Recolección de Información	17
3.4.	Procesamiento y Análisis de datos	17
3.5.	Desarrollo del Proyecto.....	20
 CAPÍTULO IV		21
4.	PROPUESTA DE SOLUCIÓN	21
4.1.	Introducción.....	21
4.2.	Normas y Estándares Internacionales para la Automatización Industrial.....	21
4.2.1.	<i>Cumplimiento de Norma IEC-61131 en los Controladores Lógicos Programables</i> ...	21
4.3.	Requerimientos de funcionamiento del sistema	24
4.4.	Prototipo del sistema de control basado en tecnología propietaria	25
4.4.1.	<i>Materiales a utilizar</i>	25
4.4.2.	<i>Software</i>	33
4.4.3.	<i>Especificaciones técnicas de elementos usados</i>	33
4.4.4.	<i>Conexión</i>	35
4.4.5.	<i>Programación del PLC- TIA PORTAL</i>	36
4.4.6.	<i>Tablas de características de variables</i>	36
4.4.7.	<i>Configuración del variador para el Motor AEG TYP AM 63 NY4</i>	38
4.4.8.	<i>CONFIGURACIÓN OPC SERVER</i>	39
4.4.9.	<i>Programación LABVIEW</i>	41
4.4.10.	<i>Configuración Física y Lógica del sistema</i>	43
4.5.	Prototipo del sistema de control con hardware y software libre.....	43

4.6. Análisis de Hardware Libre	44
4.6.1. Especificaciones técnicas de elementos usados	44
4.6.2. Conexión	45
4.6.3. Programación del PLC-M-Duino	46
4.6.4. Ensamblaje del mecanismo neumático	47
4.6.5. Ensamblaje del mecanismo neumático	48
4.7. Pruebas y Resultados	57
4.7.2. Software Libre	59
4.8. Tabla de comparación de comunicaciones industriales entre el PLC M-Duino y el Siemens S7-1200 (Ethernet, I2C, Serial, Rs485)	61
4.9. Tabla de compatibilidad del PLC M-Duino y sensores	63
4.10. Comparación de Entradas Digitales	64
4.11. Comparación de Salidas Digitales	64
4.12. Comparación de Entradas Analógicas	65
CONCLUSIONES	66
RECOMENDACIONES	67
BIBLIOGRAFÍA	
ANEXOS	

ÍNDICE DE TABLAS

Tabla 1-2: Clasificación de sensores.....	10
Tabla 2-2: Clasificación de los actuadores	11
Tabla 1-3: Aspectos específicos.....	18
Tabla 2-3: Aspectos generales	19
Tabla 1-4: Características M-DUINO PLC.....	22
Tabla 2-4: Normas sobre características mecánicas, eléctricas y electromagnéticas	23
Tabla 3-4: Materiales tipo Hardware	25
Tabla 4-4: Características y Propiedades del Sensor	27
Tabla 5-4: Datos Técnicos Termocupla	31
Tabla 6-4: Compresor parámetros técnicos	31
Tabla 7-4: Datos Técnicos Electroválvulas.....	32
Tabla 8-4: Materiales tipo Software.....	33
Tabla 9-4: Especificaciones técnicas de entradas digitales.....	33
Tabla 10-4: Especificaciones técnicas de salidas digitales	34
Tabla 11-4: Especificaciones técnicas de entradas analógicas.....	34
Tabla 12-4: Especificaciones técnicas de salidas analógicas	34
Tabla 13-4: Variables del PLC.....	37
Tabla 14-4: Variables del PLC.....	37
Tabla 15-4: Bloque de programación.....	38
Tabla 16-4: Materiales tipo Hardware	43
Tabla 17-4: Materiales tipo Software.....	44
Tabla 18-4: Especificaciones Técnicas del PLC M-Duino	45
Tabla 19-4: Tabla de comparación de comunicaciones industriales entre el PLC M-Duino y el Siemens S7-1200.....	61
Tabla 20-4: Características de comunicaciones industriales para PLC M-Duino y el Siemens S7-1200.....	61
Tabla 21-4: Compatibilidad de sensores con el PLC M-DUINO	63
Tabla 22-4: Comparación de Entradas Digitales	64
Tabla 23-4: Comparación de Salidas Digitales.....	64
Tabla 24-4: Comparación de entradas analógicas.....	65

ÍNDICE DE FIGURAS

Figura 1-2: Pirámide de la Automatización	5
Figura 2-2: Máquina Transfer.....	7
Figura 3-2: Módulo Flexible.....	8
Figura 4-2: Variedad de Productos automatizados y no automatizados	9
Figura 5-2: Etapas de un sistema de adquisición de datos	9
Figura 6-2: Esquema básico de un sensor analógico	10
Figura 7-2: Autómata programable de la firma Allen Bradley	12
Figura 8-2: Estructura de un PLC	12
Figura 1-4: Software plcLib	23
Figura 2-4: Prensa Neumática.....	25
Figura 3-4: Sensor de proximidad Magneto resistivo.....	26
Figura 4-4: Esquema de conexiones	26
Figura 5-4: Sensor de proximidad capacitivo	27
Figura 6-4: Estructura interna	28
Figura 7-4: Conexión del sensor de posición capacitivo	29
Figura 8-4: Conexión del sensor de posición capacitivo	30
Figura 9-4: Termocupla tipo J	30
Figura 10-4: Conexión de la Termocupla tipo J	31
Figura 11-4: Compresor.....	31
Figura 12-4: Electroválvula biestable	32
Figura 13-4: Esquema de conexiones	32
Figura 14-4: Diagrama.....	35
Figura 15-4: Programación del PLC- TIA PORTAL	36
Figura 16-4: Creación de un nuevo canal	39
Figura 17-4: Siemens TCP/IP Ethernet	40
Figura 18-4: Agregar el dispositivo del PLC S71200.....	40
Figura 19-4: Agregar dirección en PLC 192.168.0.1.....	41
Figura 20-4: Asignación de variables	41
Figura 21-4: VI - Enlace de variables a controlar	42
Figura 22-4: Variables utilizadas en el OPC Server	42
Figura 23-4: Configuración de variables en OPC Server	43
Figura 24-4: PLC Arduino 24 Vcc M-Duino.....	45

Figura 25-4: Conexión	46
Figura 26-4: Algoritmo de programación del M-Duino	46
Figura 27-4: Ensamblaje del mecanismo neumático	47
Figura 28-4: Pantalla inicio de base de datos en phpMyAdmin	49
Figura 29-4: Base de datos en phpMyAdmin	50
Figura 30-4: Tabla de datos en phpMyAdmin Móviles.....	51
Figura 31-4: Creación de nuevo sitio en Dreamweaver.....	51
Figura 32-4: Parámetros de configuración de comunicación con el servidor FTP en Dreamweaver	52
Figura 33-4: Estado del servicio vsftp	53
Figura 34-4: Transferencia de archivos hacia el servidor	54
Figura 35-4: Archivo Html subido al servidor.....	54
Figura 36-4: Login de Usuarios	56
Figura 37-4: Página de bienvenida del sistema de monitoreo y presentación	56
Figura 38-4: Dashboard y estado de sensores.....	57
Figura 39-4: Esquema de programación LabVIEW	58
Figura 40-4: Interfaz LabVIEW	59
Figura 41-4: Control PID implementado en sistema propietario.....	59
Figura 42-4: Análisis de las variables de control en SCADA	60
Figura 43-4: Página de Control.....	60
Figura 44-4: Control PID implementado en M-DUINO.....	61

RESUMEN

El objetivo de la investigación fue la implementación de un sistema de control y adquisición de señales con hardware y software libre para determinar su aplicabilidad en la industria. El proyecto ejecutado consistió en realizar un análisis del comportamiento de una prensa hidráulica automatizada utilizando tecnología propietaria y libre, en donde se demuestra que el sistema de hardware y software permite a las empresas tener tecnología de punta en su planta de producción, adaptándose a sus necesidades y evolucionado conforme a los requerimientos futuros, debido a que el costo de implementación es bajo. El sistema automatizado consistió de sensores y actuadores analógicos y digitales industriales utilizando por un lado el sistema propietario controlado con un Controlador Lógico programable (PLC S7-1200) de Siemens y una interfaz de usuario basada en LabVIEW a través de un servidor OPC; y por otro lado el sistema libre controlado con un Controlador Lógico programable (PLC M-DUINO) basado en Arduino, monitoreado mediante un sistema Web Service alojado en un servidor LAMP, determinado que el sistema libre proporciona mejores prestaciones porque no requiere de librerías especiales para la compatibilidad de sensores y actuadores de distintas marcas y su costo de implementación es 10 veces menos que el propietario. Se recomienda usar correctamente la plataforma Arduino y el servidor LAMP.

Palabras Claves:

<TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <CONTROL AUTOMÁTICO>, <ADQUISICIÓN DE DATOS>, <INDUSTRIA>, <SISTEMA LIBRE>, <SISTEMA PROPIETARIO>, <HARDWARE LIBRE>, <SOFTWARE LIBRE>

SUMMARY

The objective of this research was the implementation of a system of control and signals acquisition with free hardware and software to determine its applicability in the industry. The executed project consisted of carrying out an analysis of the behavior of an automatized hydraulic press using proprietary and free technology; whit this, it is possible to demonstrate that the system of hardware and software allows companies to have leading-edge technology in their production plants, which adapts its necessities and evolving according to future requirements due to the implementation cost is low. The automatized system consisted of analogic and industrial digital sensors and actuators using on one side the proprietary system controlled by a Programmable Logical Controller (PLC S7-1200) of Siemens and user interface based on LabVIEW through an OPC server; on the other side free system controlled by a Programmable Logical Controlled (PLC M-DUINO) based on Arduino, monitored through a Web Service System hosted in a LAMP server. It is determined that the free system provides better performances due to it does not require special libraries for compatibility of sensors and actuators of diferente brands and its cost of implementation on ten times less than the proprietary. It is recommended to use correctly the Arduino Platform and the LAMP server.

Key Words:

<TECHNOLOGY AND ENGINEERING SCIENCES>, <AUTOMATIZED CONTROL>, <DATA ACQUISITION>, <INDUSTRY>, <FREE SYSTEM>, <PROPRIETARY SYSTEM>, <FREE HARDWARE>, <FREE SOFTWARE>

CAPÍTULO I

1. INTRODUCCIÓN

1.1. Planteamiento del problema

El Gobierno Ecuatoriano, pretende transformar la matriz productiva, incorporando: conocimiento, acción organizada de un sistema económico y sobre todo la transformación en las estructuras productivas que promuevan la sustitución de importaciones y la diversificación productiva; permitiendo revertir los procesos concentradores y excluyentes (PNBV, 2013).

En el 2012, la FLACSO publicó un estudio sobre la micro, pequeña y media empresa en la industria en el Ecuador, estableciendo que el nivel de automatización de los procesos productivos es bajo; al carecer de sistemas modernos de control las empresas obtienen bajos niveles de calidad en sus productos, tiempos de producción no aptos para un mercado competitivo y limitando a las empresas de este segmento la exportación de sus productos a otros países (Abril, 2012).

En las pequeñas y medias empresa en el país lo primordial es la rentabilidad, productividad y competitividad a corto y mediano plazo; siendo generadoras del 75% del empleo, según el Ministerio de la Productividad en el informe de agosto 2014. La automatización industrial consiste en realizar procesos repetitivos sin la intervención del operador humano, desarrollándose sistemas SCADA, permitiendo tener un funcionamiento controlado del proceso. Sin embargo, la mayoría de dispositivos de control y aplicaciones de desarrollo de sistemas SCADA son de tipo propietario, incrementado los costos de hardware y software de módulos para tareas específicas., además tienen esquemas y aplicaciones cerradas para interactuar con otros dispositivos de otras marcas introduciendo librerías externas de programación, en caso de que sea necesario realizar una adaptación al sistema (Abaffy, 2007).

1.2. Justificación de la investigación

Las empresas hoy en día buscan ser competitivas en el mercado, entregando sus productos a tiempo y mejorando la calidad en ellos, demandando tecnología de punta durante todo el proceso de producción, desde la transformación de materia prima hasta su distribución y el mantenimiento de maquinaria. Sin embargo, se ven limitadas por el elevado costo en la automatización de sus procesos de producción; debido a que las soluciones actuales involucran hardware y software propietario, repercutiendo de manera directa al precio del producto final. Los avances tecnológicos y a la colaboración de miles de personas, se han desarrollado soluciones de hardware y software libre para la industria, con altos niveles de fiabilidad y seguridad, permitiendo reducir los costos en la implementación de sistemas SCADA, logrando de esta manera que las pequeñas empresas accedan a esta tecnología para que se vuelvan competitivas, con la utilización de relativamente escaso presupuesto, pero alcanzando un elevado nivel de automatización y calidad en el producto.

En este contexto se plantea una guía de implementación de un SCADA, utilizando hardware y software libre, que le permita a las empresas tener tecnología de punta en su planta de producción, adaptándose a sus necesidades y evolucionado conforme a los requerimientos futuros.

1.3. Objetivos

1.3.1. *Objetivo general*

Implementar un sistema de control y adquisición de señales con hardware y software libre para determinar su aplicabilidad en la industria.

1.3.2. *Objetivos específicos*

- Analizar el cumplimiento de la normativa vigente de equipos destinados a la automatización industrial.
- Comparar el rendimiento del control de un sistema de prueba entre el sistema propietario y hardware libre.
- Análisis de la incidencia en la aplicación de soluciones de hardware y software libre y sistemas propietarios en la industria

CAPÍTULO II

2. Fundamentación Teórica

2.1. Estudio del Arte

Los productos que se elaboran en el Ecuador de forma automatizada son responsabilidad del buen funcionamiento de dispositivos de medición, control y reporte, rigiéndose a normas de programación y protección física para su trabajo eficiente en los medios en donde se desempeñan.

En la actualidad existen trabajos realizados con tecnología libre, en donde las licencias para el desarrollo de sistemas de control y supervisión son de precio nulo, permitiendo al usuario modificar tanto hardware como software de manera legal, citando los más relevantes:

En el trabajo realizado por Alberto Medrano en el 2015, denominado “Hardware libre y abierto, modelos de negocios para América Latina y el Caribe”, se expone la importancia de manejar códigos y diseños libres asociados a dispositivos electrónicos, para mejorar la rentabilidad en el control de plantas petroleras de Venezuela (Chourio & Medrano, 2015).

En el 2011, Barreño publicó un artículo denominado “Metodología Para la Transferencia Tecnológica en la Industria Electrónica Basada en Software Libre y Hardware Copyleft.”, se hace referencia al diseño de sistemas embebidos para lazos de control industriales basados en software libre, realizando un análisis de la escasa implementación en los países en vía de desarrollo con tecnología libre, esto se debe en gran parte a que las empresas de base tecnológica son muy pequeñas, con bajos niveles de producción y utilizan metodología de diseño desactualizadas. Uno de los principales argumentos para impulsar el desarrollo sustentable con hardware libre es la producción acelerada, innovadora y activa de bienes comunes (Camargo, 2012).

Lazalde en el 2014 y 2015 expone sus trabajos titulados: “Hardware libre Recomendaciones para el fomento de la innovación ciudadana” y “Ecosistemas de innovación y producción basados en hardware libre”, respectivamente en donde menciona que hardware libre es una tecnología idónea para impulsar espacios de innovación ciudadana con impacto social, esto es,

laboratorios como FabLabs y makerspaces que agrupen universidades, empresas, gobierno y ciudadanía en general (Alan, Torres, & Vela, 2015).

En el 2015, Besantes desarrollo un trabajo titulado: “Automatización de sistema de recirculación de agua caliente utilizando hardware libre en el Hospital IESS Latacunga”, demostrando que la tecnología de hardware y software libre puede ser aplicable a las pequeñas y medianas empresas con el fin de reducir costos de implementación y puesta en marcha de sistemas de automatización (Talve, 2015).

2.2. Fundamentación Teórica

2.3. Definición de la Automatización

El diccionario de mecánica y estructuras metálicas de Juan Fernández define: "La automatización es un sistema donde se transfieren tareas de producción, realizadas habitualmente por operadores humanos a un conjunto de elementos tecnológicos."

La Real Academia de Ciencias Exactas Físicas y Naturales define a la Automática como: “La ciencia que trata de los métodos y procedimientos cuya finalidad es la sustitución del operador humano por un operador artificial en la ejecución de una tarea, física o mental, previamente programada.”

En el glosario ALA define como: "La realización de una operación, una serie de operaciones o un proceso por auto control, automatización o por medios automáticos. La automatización implica la utilización de un equipo automático de procesos de datos que puede ser un ordenador u otros dispositivos que simplifiquen el trabajo”.

2.4. Pirámide de la Automatización

Los diferentes componentes que forman un Sistema de Control de Automatización y Control Industrial (IACS) y los que componen la pirámide de automatización, hacen referencia a la Figura 1-2, realizando la implementación a partir de los años 90 con la aparición de redes basadas en IP, en las redes de área local e Internet (Schwaiger, 2003) o la norma ISA- 95 (Scholten, 2007), tienen requisitos diferentes en cuanto a funcionamiento y rendimiento de cada nivel, tanto para los sistemas propiamente como para los protocolos de comunicaciones.



Figura 1-2: Pirámide de la Automatización

Fuente: Ierache J., 2012, pp. 5

La especificación de cada nivel mostrada en la Figura 1-2, se muestra a continuación:

- El primer nivel o "nivel de campo" incluye los dispositivos físicos presentes en la industria, como los actuadores y sensores.
- El segundo nivel o "nivel de control" incluye los dispositivos controladores como ordenadores, PLCs, PIDs, etc.
- El "nivel de supervisión" (tercer nivel) corresponde a los sistemas de supervisión, control y adquisición de datos (SCADA).
- En un nivel superior o "nivel de planificación" se encuentran los sistemas de ejecución de la producción (MES).
- La cúspide de la pirámide ("nivel de gestión") la componen los sistemas de gestión integral de la empresa (ERP) (SMC, 2016).

2.5. La Automatización de la fabricación

La Automatización es el estudio y aplicación de la Automática en el control de procesos industriales, tanto en lazo abierto como en lazo cerrado. Bajo este concepto pueden englobarse la automatización de procesos de fabricación continua como la automatización de procesos de fabricación de piezas.

En general, la Automatización implica la utilización y el conocimiento de tres disciplinas: la mecánica, la electrónica y la informática (hardware y software), incluyendo:

- Sistemas sensoriales/instrumentación
- Sistemas de actuación
- Máquinas con control numérico
- Robots Industriales
- Sistemas de almacenamiento y transporte
- Sistemas de regulación
- Equipos de control
- Sistemas de comunicaciones

En los sistemas de fabricación modernos muchas de las operaciones están automatizadas y/o computarizadas. Sin embargo, los sistemas de fabricación incluyen personas que los hacen funcionar y que deben ser considerados en los procesos de automatización. (Barrietos & Gambao, 2012).

El ciclo de proceso de fabricación incluye tareas diferentes a las de producción y el trabajo del obrero, comprendiendo la colaboración entre el obrero y el profesional especializado en una serie de etapas que van desde la especificación del producto por la empresa o por el cliente, pasando por el diseño del producto, su venta y distribución. La automatización industrial tradicionalmente se ha ocupado de aquellas etapas relativas al ciclo de fabricación y relacionadas con máquinas, líneas de ensamblado, robots y procesos que se podían reducir en una serie de operaciones repetitivas y claramente definidas y supervisadas por obreros. Para casi todos los procesos industriales el porcentaje de los tiempos de maquinado suele ser pequeño respecto al tiempo total de fabricación. Por el contrario, el diseño, la planificación y venta, son las funciones que consumen la mayor parte de costos del ciclo de producción (B. Wendy, 1989).

2.6. Tipos de Sistema de Automatización

2.6.1. *Automatización Rígida*

Se diseñan máquinas para producir un artículo normalizado, como por ejemplo un motor, un cilindro, un engrane, etc. y suele ser rentable en la producción de productos de gran demanda. Estas máquinas son especializadas y carecen de flexibilidad; típicamente la automatización consiste en la concatenación de opciones simples, el coste de la instalación suele ser elevado, pero se compensa gracias a la producción de un número elevado de piezas con una mayor

productividad. El caso típico de la automatización rígida la constituyen las denominadas “máquinas transfer”, diseñadas y construidas específicamente para una aplicación concreta y en general no válidas para la fabricación de otro producto diferente (Barrietos & Gambao, 2012).

En la automatización rígida se diseñan las máquinas como por ejemplo un monoblock, una válvula, un engrane o un husillo, como muestra en la Figura 2-2. El tamaño del producto y los parámetros de procesamiento (como velocidad y profundidad de corte) se pueden cambiar, aunque no se puede modificar en grado apreciable para procesar productos que tengan distintas formas y dimensiones.

Las máquinas que se usan en aplicaciones de automatización dura se suele construir con el principio modular o de bloque constructivo. denominándoles en general máquinas de transferencia y consisten en dos partes principales: unidades de producción de cabezal motorizado y mecanismos de transferencia.

Unidades de producción de cabezal motorizado. - Está conformado: por un chasis o armazón, motores, sistemas de engranajes y husillos, relacionados entre sí. Estos sistemas mecánicos permiten adaptabilidad y flexibilidad para procesos industriales formadas por uno o más motores en modelos de línea recta, circular o en U sobre una superficie plana.

Mecanismos de transferencia y líneas de transferencia. - Son mecanismos que permite transportar el producto de una estación a otra dentro de la línea de producción con retroalimentación para el control de la calidad.

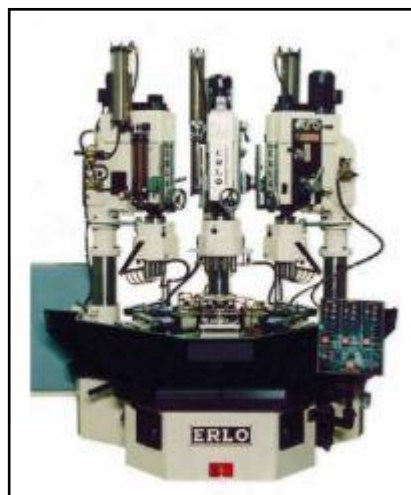


Figura 2-2: Máquina Transfer

Fuente: ERLO, 2012, pp. 8

2.6.2. *Automatización Flexible*

Consiste en la fabricación de diferentes productos con un cambio de las órdenes del sistema con alteraciones mínimas, en la Figura 3-2 se observa un módulo flexible de fabricación de productos de FESTO.

Una máquina que fabrica cierta variedad de productos en partidas pequeñas, se le conoce como estrategia de reprogramación de flujo flexible, suele ser programada para alternarse en la elaboración de dos productos.

Cuando una máquina ha sido destinada a un producto o una familia de productos en particular, se le conoce como un flujo de línea. La Cummins Engine Company, fabricante de motores diésel con sede en Colombus, Indiana, utiliza dicha flexibilidad para manejar frecuentes modificaciones de diseño (Lee & Larry, 2008).

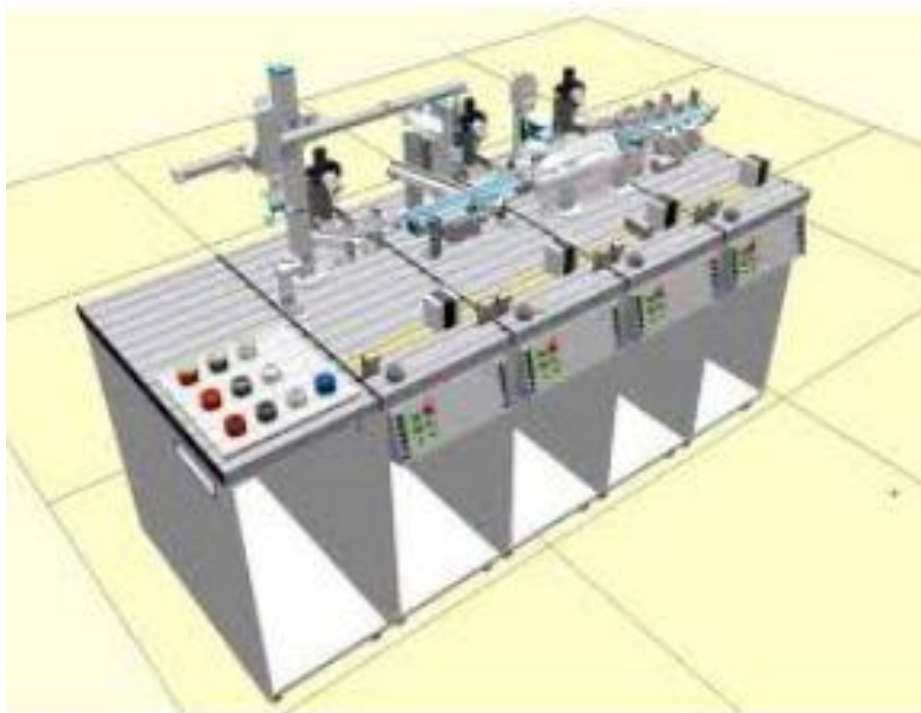


Figura 3-2: Módulo Flexible

Fuente: FESTO, 2012, pp. 5

En la Figura 4-2 se observa una relación entre productos realizados con los diferentes tipos de automatización, así como los productos realizados de forma manual.

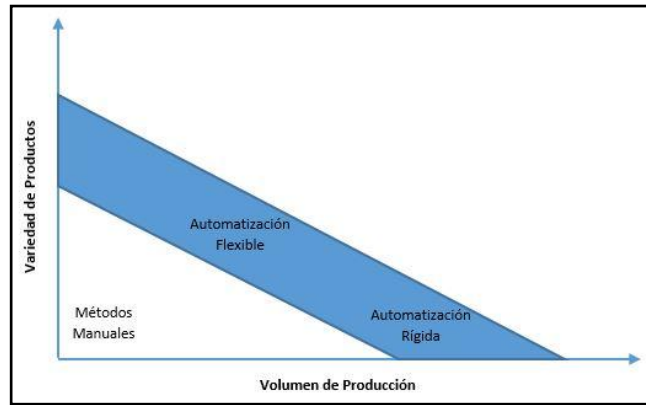


Figura 4-2: Variedad de Productos automatizados y no automatizados

Realizado por: Pamela Espejo. 2017

2.7. Adquisición de datos

Es la conversión de magnitudes físicas a magnitudes eléctricas acondicionadas para su procesamiento. La Figura 5-2 muestra un esquema que representa la arquitectura de un sistema básico de adquisición de datos.

Componentes de un sistema de adquisición de datos son:

- Ordenador Personal
- Sensores y actuadores
- Acondicionadores de señal
- Hardware de adquisición de datos
- Software

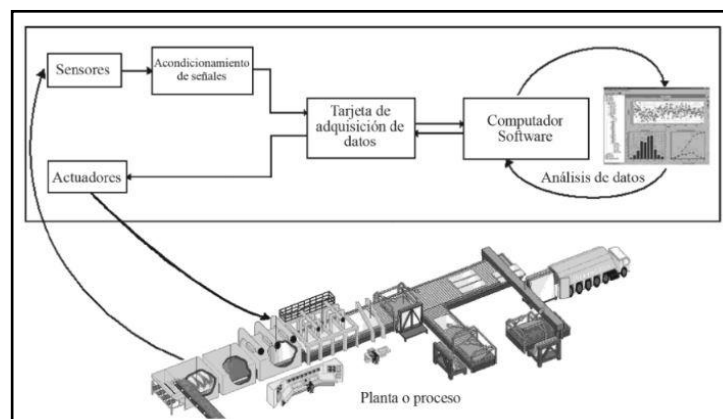


Figura 5-2: Etapas de un sistema de adquisición de datos

Realizado por: Pamela Espejo. 2017

En el sistema de adquisición de datos de la Figura 5-2 las flechas entre los bloques: Tarjeta de Adquisición de Datos y Computador/Software, indican que los datos se transmiten y reciben en las dos direcciones. Un sistema con esta estructura se conoce como “Sistema bidireccional de adquisición de datos” (Quintero, Oñate, & Harías, 2011).

2.8. Sensores

Son dispositivos que obtienen señales eléctricas en respuestas de magnitudes física (Pallas, 2003).

En la Tabla 1-2 se muestra la clasificación de los sensores según su modo de operación, señal de salida y aporte de energía.

Tabla 1-2: Clasificación de sensores

Criterio	Clases	Ejemplos
Aporte de energía	Moduladores	Termistor
	Generadores	Termopar
Señal de salida	Analógicos	Potenciómetro
	Digitales	Codificador de posición
Modo de operación	De deflexión	Aceleración de deflexión
	De comparación	Servoacelerómetro

Realizado por: Pamela Espejo. 2017

2.8.1. Sensores Digitales

Son aquellos en que las salidas tienen valores discretos e inmunes a las interferencias electromagnéticas, obteniendo mayor fiabilidad, fidelidad y exactitud en los valores obtenidos, ideales para aplicaciones de sistemas de control, telecomunicaciones entre otros.

2.8.2. Sensores Analógicos

Los sensores analógicos proporcionan una señal continua en el tiempo de forma continua. En la Figura 2-6 se presenta el esquema básico de un sensor analógico (Pallas, 2003).

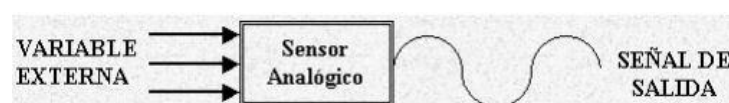


Figura 6-2: Esquema básico de un sensor analógico

Fuente: Pallas, 2003, pp. 20

Las propiedades de un sensor analógico son:

- Calibración
- Rango de funcionamiento
- Confiabilidad
- Velocidad de respuesta
- Exactitud precisión
- Sensibilidad
- Linealidad (Pallas, 2003).

2.9. Actuadores

Es un dispositivo mecánico que permite la interacción con otro dispositivo mecánico. Los actuadores pueden ser de tipo: eléctrico, neumático e hidráulico.

El actuador rotatorio dependiendo de su diseño como muestra en la Tabla 2.2 que consta de las siguientes partes móviles básicas:

Tabla 2-2: Clasificación de los actuadores

	Actuador Neumático	Actuador Eléctrico	Actuador Hidráulico
Fuerza Generadora de movimiento	Presión de aire	Energía Eléctrica	Presión Hidráulica
Elemento Motriz	Pistón	Motor Eléctrico	Pistón
Torque	Eje	Reductor	Eje
Conversión Mecánica	Piñón	No hay	Piñón

Realizado por: Pamela Espejo. 2017

2.10. Controladores Lógicos Programables

Se define como autómata programable (Figura 7-2) a un computador digital diseñado para programar y controlar procesos secuenciales en tiempo real y con capacidad de trabajar en ambientes industriales.



Figura 7-2: Autómata programable de la firma Allen Bradley

Fuente: Alley Bradley, 2012, pp. 25

La estructura de un controlador lógico programable (Figura 8-2) está constituido por un conjunto de tarjetas o circuitos impresos, sobre los cuales están ubicados componentes electrónicos (InfoPLC, 2009).

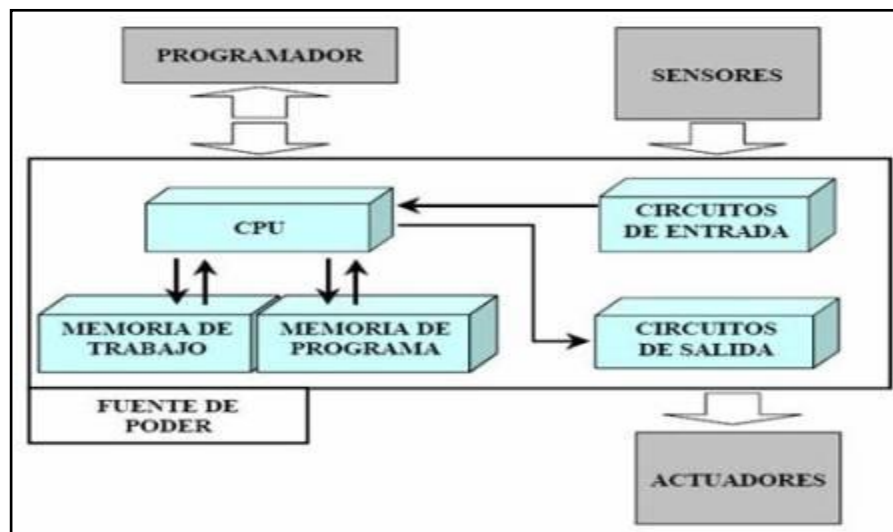


Figura 8-2: Estructura de un PLC

Fuente: Info PLC, 2009, pp. 15

2.10.1. Elección de un PLC

Según la publicación de (InfoPLC, 2013) se enumera algunos aspectos importantes para la elección de un autómata, entre ellas se tiene:

- Número de entradas y salidas
- Capacidad de memoria de programación

- Comunicaciones
- Escalabilidad
- Control de movimiento
- Seguridad
- Software
- Precio
- Servicio Técnico
- Voluntad del Cliente Final

2.11. Protocolos de Comunicación

Son un conjunto de reglas y normas que permiten la comunicación del CPU con otros dispositivos de una red industrial; los protocolos más usados en la actualidad son: RS-232C, RS-485, Profibus, etc., según el fabricante.

2.12. Fiabilidad Industrial

La fiabilidad viene definida de la probabilidad de un aparato o dispositivo trabaje correctamente por un tiempo determinado en condiciones de servicio sin que falle. Información útil para establecer las políticas de mantenimiento e inspección en una empresa, para determinar los tiempos de garantía de los productos (Ponsati, 2002).

2.12.1. EMC (*Compatibilidad Electromagnética*)

La EMC es un nuevo concepto a tener en cuenta en el diseño de los circuitos electrónicos, que estudia los mecanismos para eliminar, disminuir y prevenir los efectos de acoplamiento de un equipo eléctrico o electrónico en un entorno electromagnético (López, 2010).

2.12.2. Rohs (*Estándar industrial para PLC*)

El estándar internacional IEC 61131 es una colección completa de estándares referentes a controladores programables y sus periféricos asociados. Consiste de las siguientes partes:

Parte 1: Información General Establece las definiciones e identifica las principales características significativas a la selección y aplicación de los controladores programables y sus periféricos asociados.

Parte 2: Equipo requerimientos y pruebas Especifica los requisitos del equipo y pruebas relacionadas para los controladores programables (PLC) y sus periféricos asociados.

Parte 3: Lenguajes de Programación Define como un conjunto mínimo, los elementos básicos de programación. Reglas sintácticas y semánticas para los lenguajes de programación usados más comúnmente, incluyendo los lenguajes gráficos de Diagrama de Escalera y Diagrama de Bloques de Funciones y los lenguajes textuales de Lista de Instrucciones y Texto estructurado. Así como sus principales campos de aplicación, pruebas aplicables y los medios por los cuales los fabricantes pueden expandir o adaptar esos conjuntos básicos a sus propias implementaciones de controlador programable.

Parte 4: Guías de Usuario. Un reporte técnico que proporciona una vista general y guías de aplicación del estándar para los usuarios finales de los controladores programables.

Parte 5: Especificación del servicio de Mensajería. Define la comunicación de datos entre controladores programables y otros sistemas electrónicos usando el “Manufacturing Message Specification” (MMS, acorde al ISO/IEC 9506).

Parte 6: Programación en lógica difusa. Define los elementos básicos de programación de “lógica difusa” para su uso en Controladores programables.

Parte 7: Guías para aplicación e implementación de lenguajes de programación. Proporciona una guía para los desarrolladores de software para los lenguajes de programación definidos en la parte 3 (Mandado, Acevedo, Fernandez, & Armesto, 2009).

2.13. Hardware y Software Libre en la industria

El uso de Hardware y Software Libre permite la sostenibilidad y soberanía tecnológicas, reduciendo los costos respecto al diseño propietario y su eventual valor educativo al insertarse en procesos de aprendizaje y de trabajo necesariamente colaborativo (FLOK Society, 2015). El primer Open Hardware Summit, celebrado en Nueva York en 2010, partía de la idea de definir y concretar los principios que deberían darles forma a unas especificaciones genéricas sobre dispositivos libres (Alan, Torres, & Vela, 2015).

Arduino y Hardware/Software Libre

La plataforma Arduino ha revolucionado la electrónica por la facilidad de la creación de proyectos, esta placa consta de un microcontrolador Atmel AVR y puertos de entrada/salida que le permiten adaptarse en forma rápida en la automatización de procesos industriales. La programación de esta placa se lo realiza a través del lenguaje de programación Processing/Wiring y el cargador de arranque (boot loader).

2.14. PLCOpen

PLC Open es una organización mundial independiente de fabricantes que busca ofrecer un valor añadido a los usuarios de controles industriales a través del estándar de desarrollo software IEC 61131-3 (Karl-Heinz & Michael, 1995).

IEC 1131 es el primer paso en la estandarización de los autómatas programables y sus periféricos, incluyendo los lenguajes de programación que se deben utilizar. Esta norma se divide en cinco partes:

- Parte 1: Vista general.
- Parte 2: Hardware.
- Parte 3: Lenguaje de programación.
- Parte 4: Guías de usuario.
- Parte 5: Comunicación.

IEC 1131-3, pretende es la base real para estandarizar los lenguajes de programación en la automatización industrial, haciendo el trabajo independiente de cualquier compañía. Hay muchas maneras de describir el trabajo desarrollado en la tercera parte de esta norma, indicaremos algunas de ellas son:

IEC 1131-3 es el resultado del gran esfuerzo realizado por 7 multinacionales a los que se añaden muchos años de experiencia en el campo de la automatización industrial.

Incluye 200 páginas de texto aproximadamente, con más de 60 tablas. IEC-1131-3 son las especificaciones de la sintaxis y semántica de un lenguaje de programación, incluyendo el modelo de software y la estructura del lenguaje.

2.15. Interfaz Humano-Máquina

Permite la interacción del humano y la máquina a través de paneles de control programados en lenguajes de programación orientados a objetos, que permite al operador del sistema de control la supervisión y la interacción de los procesos. En la comunicación hombre-máquina se estudia entre otros aspectos:

- Cantidad de información
- Forma de actuación
- Puesto de mando
- Propiedades fisiológicas y psíquicas del hombre-operador.

CAPÍTULO III

3. METODOLOGIA INVESTIGATIVA

3.1. Metodología

3.2. Modalidad de Investigación

La presente es una investigación aplicada, la que se desarrolló utilizando:

- Investigación Bibliográfica, porque la explicación científica de los contenidos del tema de investigación se realizó consultando en libros de electrónica, revistas y publicaciones de internet referentes a los sistemas de control y adquisición de señales con hardware y software libre y su incidencia en la industria.
- Investigación de Campo, para lo cual se realizó un estudio sistemático basado en un prototipo que permitió la comparación entre PLC de la marca SIEMENS S-7200 con el M-DUINO y su incidencia en la industria.

3.3. Recolección de Información

Fuentes, técnicas e instrumentos. Para el presente proyecto se recopiló información de libros, internet, revistas científicas, trabajos de investigación, papers, patentes, guía del tutor para el desarrollo del proyecto, etc.

3.4. Procesamiento y Análisis de datos

Una vez que se ha obtenido la información necesaria acerca de la investigación, esta fue procesada de manera ordenada y sistemática, de acuerdo a la matriz de consistencia de la Tabla 1-3 y Tabla 2-3.

Tabla 1-3: Aspectos específicos

Variable	Conceptualización	Dimensiones	Indicadores
Variable independiente: Sistema de control y adquisición de Señales con Hardware y Software libre	Es un sistema formado por equipos electrónico cuya función es el de control y almacenamiento de una o varias variables en cualquier proceso.	Equipo	Hardware Libre
		electrónico	Equipos Autónomos no flexibles
		Control	Precisión
			Estabilidad
			Eficiencia
		Variables físicas	Temperatura
			Voltaje
			Corriente
		Variable dependiente: Incidencia en la Industrial	En la industria es el conjunto de proceso y actividades que tiene la finalidad de transformar materias primas en productos elaborados en un tiempo determinado.
Procesos	Salidas		
Materia Prima	Funciones de Transferencia		
	Control de calidad		
	Transporte		
Productos elaborados	Insumo		
	Producto final		
	Distribución		
	Consumo		

Realizado por: Pamela Espejo. 2017

Tabla 2-3: Aspectos generales

Problema	Objetivos	Hipótesis	Variables	Metodología	Población
<p>Problema general ¿De qué manera un sistema de control y adquisición de señales con hardware y software libre, permitirá optimizar procesos, eficiencia, recursos y costos en la industria?</p> <p>Problemas Específicos ¿Los equipos destinados al control industrial con hardware libre cumplen con las normas vigentes destinados a la automatización industrial? ¿Qué parámetros técnicos se compararán para el rendimiento de control del extrusor de polímero entre un sistema propietario y hardware libre? ¿Qué incidencias se tendrá en la aplicación de soluciones de hardware y software libre y sistemas propietarios en la industria?</p>	<p>Objetivo General Implementar un sistema de control y adquisición de señales con hardware y software libre para determinar su aplicabilidad en la industria.</p> <p>Objetivos Específicos Analizar el cumplimiento de la normativa vigente de equipos destinados a la automatización industrial. Comparar el rendimiento del control del extrusor de polímero entre el sistema propietario y hardware libre. Análisis de la incidencia en la aplicación de soluciones de hardware y software libre y sistemas propietarios en la industria</p>	<p>Hipótesis General El sistema de control y adquisición de señales con hardware y software libre permitirá la reducción de costos de implementación de procesos de control en la industria sin afectar al rendimiento y calidad en el proceso industrial para determinar su confiabilidad.</p>	<p>Variable 1: Sistema de control y adquisición de Señales con Hardware y Software libre Variable 2: Incidencia en la Industrial</p>	<p>Investigación Bibliográfica, porque la explicación científica de los contenidos del tema de investigación se realizará consultando en libros de electrónica, revistas y publicaciones de internet referentes a los sistemas de control y adquisición de señales con hardware y software libre y su incidencia en la industria. Investigación de Campo, para lo cual se realizará un estudio sistemático de los sistemas de control y adquisición de señales con hardware y software libre, así como también su incidencia en la industria.</p>	<p>Por la característica de la investigación no se requiere de población ni muestra.</p>

Realizado por: Pamela Espejo. 2017

3.5. Desarrollo del Proyecto

Para el desarrollo de la investigación se efectuó los siguientes pasos:

- Analizar los estándares y normas internacionales para la automatización industrial.
- Analizar las normativas vigentes en el Ecuador de los equipos destinados a la automatización industrial.
- Analizar el prototipo de un sistema de control basado en software propietario.
- Estudiar el prototipo de un sistema de control en hardware libre
- Seleccionar los elementos electrónicos que se utilizaran en el prototipo de un sistema de control
- Simular el algoritmo de control del prototipo de un sistema de control.
- Diseñar el sistema de control en hardware libre del prototipo
- Implementar el sistema de control en hardware libre del prototipo.
- Realizar ensayos de funcionamiento sistema de control en hardware libre, para corrección de errores.
- Realizar un cuadro técnico de comparación del nivel de incidencia en la aplicación de soluciones de hardware y software libre y sistemas propietarios en la industria.

CAPÍTULO IV

4. PROPUESTA DE SOLUCIÓN

4.1. Introducción

La presente investigación plantea la automatización de una prensa neumática, basado en software y hardware libre, que permita trabajar con tecnología de punta en la planta de producción de las PYMES; adaptándose a sus necesidades y evolucionado conforme a los requerimientos futuros, por este motivo se ha decidido analizar el comportamiento de la automatización del sistema utilizando tecnología propietaria probada en la industria, con un controlador S7-1200 de Siemens y una interfaz de usuario basada en LabVIEW, a través de un servidor OPC y compararla con tecnología libre trabajando con un controlador M-DUINO basado en Arduino, monitoreando el sistema con un servidor LAMP, que aloja una página web programada en PHP; generando la reducción de costos de implementación en procesos de control, sin afectar al rendimiento y calidad en el proceso industrial para determinar su confiabilidad.

4.2. Normas y Estándares Internacionales para la Automatización Industrial

4.2.1. *Cumplimiento de Norma IEC-61131 en los Controladores Lógicos Programables*

Para la automatización en la industria es necesario el cumplimiento de la Norma IEC-61131, enfocada en los autómatas programables y sus periféricos; por este motivo es necesario el análisis del cumplimiento de la norma en el PLC M-DUINO, tal como lo hace el PLC S7-1200 (Siemens, 2013):

- Parte 1: Visión General.

En este apartado, hace referencia a las características fundamentales que se debe tener en cuenta al momento de elegir un PLC (InfoPLC, 2013).

Tabla 1-4: Características M-DUINO PLC

Ítem	Descripción	Arduino Industrial M-DUINO PLC
1	Número de Entradas digitales	7 Entradas Digitales (5-12-24 Vdc), configurables.
2	Número de Salidas digitales	(8x) Salidas Digitales Aisladas (5-12-24Vcc) / (3 de las 8x) PWM Aisladas.
3	Número de entradas analógicas	(6x) Entradas Analógicas (0-10Vcc) / Entradas Digitales (5-12-24Vcc) configurables.
4	Número de Salidas analógicas	(3x) Salidas Analógicas (0-10Vcc) configurable por Switch.
5	Capacidad de Memoria	Flash Memory: 256kB (8kB utilizados para el bootloader) SRAM: 8kB, EEPROM: 4kB, Clock Speed: 16MHz
6	Comunicaciones	Ethernet, Usb 2.0, I2C, RS 232, RS 485
7	Escalabilidad	Puerto I2C permite conectar módulos adicionales, así como compatibilidad con PLC ArdBox. Esclavos tipo Modbus, Rs485 y Rs 232
8	Control de Movimiento	Compatibilidad con salidas analógicas estándares de 0-10 Vdc para control de VDF. Compatibilidad con Drives con comunicación Modbus. Compatibilidad con Drives con comunicación RS-485
9	Seguridad en programación e instalación	EN 61131-2:2003 EN 61000-3-2:2000 EN 61000-3-3:1995+A1:2001 EN 61131-2:2003 Seguridad Electromagnética Declaración CE Rohs
10	Software	IDE – Arduino y librerías
11	Precio	\$230
12	Servicio Técnico	Industrial Shield Arduino
13	Capacidad de Visualización (HMI)	Servidor Web. Humm Touch 10.1 (Linux) Humm Touch 10.1 Android

Realizado por: Pamela Espejo. 2017

- Parte 2: Hardware.

Se especifican las características eléctricas, mecánicas y compatibilidad electromagnética que debe cumplir los controladores.

En el Anexo A, hace referencia sobre el cumplimiento de las normas:

Tabla 2-4: Normas sobre características mecánicas, eléctricas y electromagnéticas

Norma	Descripción
Emisiones RFI: 61131-2:2003	Producto Estándar
Emisiones RFI: 61000-3-2: 2000	Límites para las emisiones de corriente armónica
Emisiones RFI: 61000-3-3:1995+A1:2001	Limitación de fluctuaciones en sistemas de baja tensión de suministro de voltaje.
Inmunidad: 61131-2:2003	Producto Estándar

Realizado por: Pamela Espejo. 2017

- Parte 3: Lenguaje de programación.

La programación del PLC M-DUINO, no cumple con el lenguaje texto estructurado, establecido por el estándar IEC 61131-3; sin embargo, se cuenta con herramientas externas (Figura 1-4) para cumplir de manera indirecta con el estándar. Cabe recalcar que el fabricante del dispositivo Industrial Shields, se encuentra desarrollando el lenguaje Escalera para el cumplimiento del estándar.

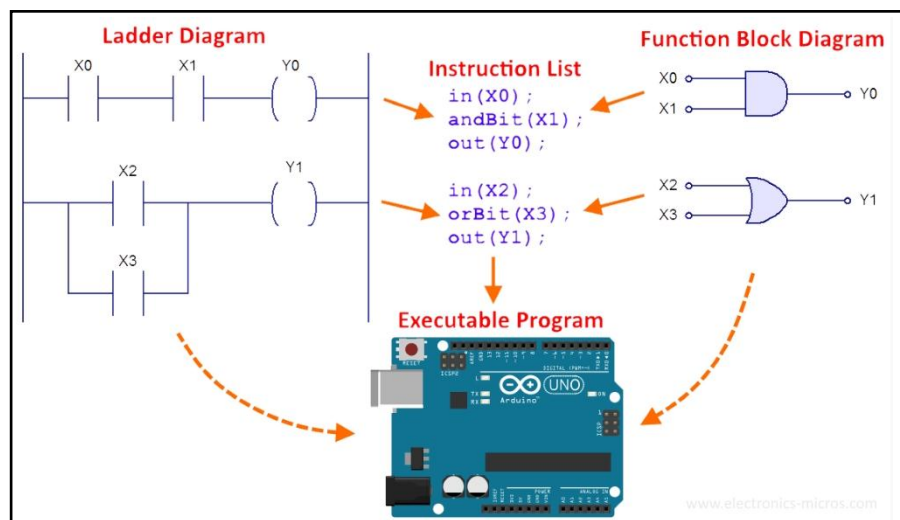


Figura 1-4: Software plcLib

Realizado por: Pamela Espejo. 2017

- Parte 4: Guías de Usuario y Parte 8: Guías para el diseño, implementación y uso de lenguaje de programación.

En la página web (Industrial Shields, 2016) del fabricante existe documentación de soporte para el usuario final como: CE M-Duino 21, M-Duino Ejemplo, M-Duino 21/42/58 7.0 guía de usuario, M-Duino 21/42/58 7.0 Layout, cumpliendo con el estándar IEC 61131-4/8.

- Parte 5: Comunicación.

El PLC M-Duino cumple con el estándar IEC 61131-5, al ofrecer comunicaciones entre diferentes dispositivos a través de los puertos:

(1x) Ethernet port.

(1x) Ethernet port.

(1x) USB port (type B).

(1x) I2C port.

(3x) TTL ports.

(1x) RS-232 port.

(1x) RS-485 port.

TCP/IP / Modbus RTU / Modbus TCP.

WebServer capacity

- Parte 6: Control Fuzzy o difuso.

Se cuenta con librerías para control difuso para el IDE de Arduino, dando cumplimiento con el estándar IEC 61131-7

4.3. Requerimientos de funcionamiento del sistema

El control de la prensa neumática (ver Figura 2-4) cuenta con botoneras para el encendido (ON) y apagado (OFF) del sistema, además de un paro de emergencia. Al presionar el botón ON se calienta una resistencia calefactora de la prensa a 30°C (monitoreado por un control PID de lazo cerrado y un sensor de temperatura) y se enciende el motor M1 conectado a un variador de frecuencia (VDF) a una velocidad preestablecida que lleva un producto por la banda transportadora hacia la prensa, la cual se encuentra retraída; el pistón en retroceso es monitoreado por el sensor SB1 y el avance por el sensor SB2. El sensor SB3 detecta el producto a pensar y baja la prensa, durante 4 segundos; transcurrido este tiempo el pistón se eleva y se activa el motor de la banda transportadora para dar paso al siguiente producto.

Existe un regulador para controlar de forma manual la presión que entra al pistón, siendo adquirida por un sensor SB4 de presión absoluta, la presión normal es de 2 bares y cuando pasa de 5 bares se enciende la alarma en el HMI.

El sensor SB3 realiza un conteo de todos los productos que se prensen, además de controlar atascos en la banda transportadora.

Existen 2 luces piloto que indican cuando el sistema se encuentra encendido y cuando se encuentra apagado, presentando alertas por atascos, presión y temperatura alta y baja.

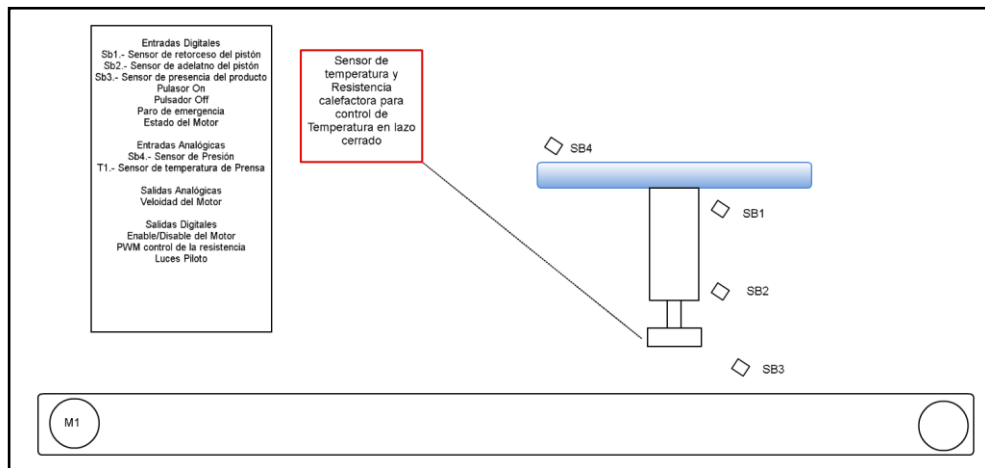


Figura 2-4: Prensa Neumática

Realizado por: Pamela Espejo. 2017

4.4. Prototipo del sistema de control basado en tecnología propietaria

4.4.1. Materiales a utilizar

Hardware

Tabla 3-4: Materiales tipo Hardware

Cantidad	Descripción
1	Sensores de proximidad SMT8.
3	Sensor de temperatura
1	Sensor de presión 539757. hila
1	Sensor de posición inductivo M18, montaje saliente 548645
1	Motor trifásico AEG TYP AM 63 NY4
1	Electroválvula biestable de 5/2 vías con LED 539778
1	Banda transportadora 16.5x40 cm.

1	Pistón 152888
1	PLC-S71200.
10	Cables con jacks tipo banana
1	Variador de Frecuencia Micromaster 440
1	Válvula de cierre con unidad de filtro y regulador 540691
1	Bloque de distribución (152896)
5	Canaletas
1	Multímetro.
4	Desarmadores
6	Mangueras neumáticas.

Realizado por: Pamela Espejo 2017

Sensores

A continuación, se presenta el sensor de proximidad SMT-8-SL-PS-LED-24-B 562019 para la detección del producto a procesarse (Figura 3-4).

Sensor de proximidad SMT-8-SL-PS-LED-24-B 562019 (Festo AG, 2009)



Figura 3-4: Sensor de proximidad Magneto resistivo

Fuente: Festo Didactic, 2009, pp. 28

Estructura Interna

El esquema de conexiones interna del sensor esta descrito por la Figura 4-4

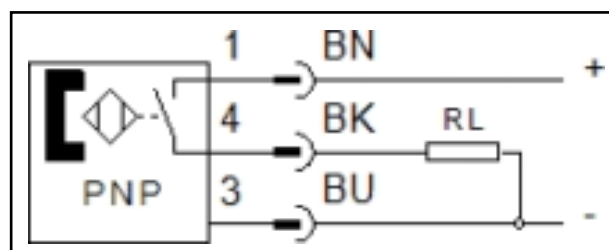


Figura 4-4: Esquema de conexiones

Fuente: Festo Didactic, 2009, pp. 31

Tabla 4-4: Características y Propiedades del Sensor

Característica	Propiedades
Construcción	Para ranura en T
Homologación	RCM Mark c UL us - Listed (OL)
Indicación sobre el material	Exento de cobre y PTFE Conforme con RoHS
Principio de medición	Magnetorresistivo
Temperatura ambiente	-20°C a 70 °C
Salida	PNP
Función del elemento de conmutación	contacto de trabajo
Corriente máxima de salida	100 mA
Rendimiento DC máximo de conmutación	3 W
Anticortocircuitaje	Si
Resistencia a sobrecargas	presente
Tensión de servicio calculada DC	24 V
Margen de tensión de funcionamiento DC	10 a 30 V
Polos inconfundibles	Para todas las conexiones eléctricas
Conexión eléctrica	Conector M8x1 3 contactos
Tipo de fijación	Fijado con tornillos Encajable longitudinalmente en la ranura
Peso del producto	4,2 g
Indicación del estado	LED amarillo
Tipo de protección	IP65 IP68

Fuente: Festo Didactic, 2009, pp. 32

Sensor de posición capacitivo M12 548651 (Festo Didactic, 2009)



Figura 5-4: Sensor de proximidad capacitivo

Fuente: Festo Didactic, 2009, pp. 33

Estructura Interna

En la Figura 6-4 se presenta la estructura que la compone.

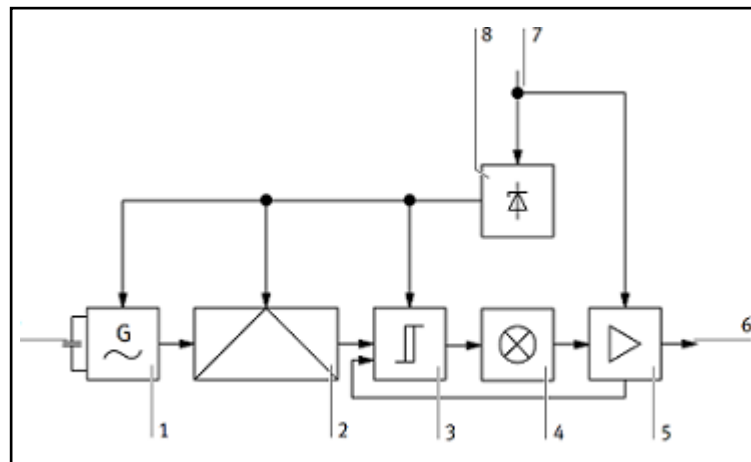


Figura 6-4: Estructura interna

Fuente: Festo Didactic, 2009, pp. 33

1: oscilador; 2: demodulador; 3: nivel de iniciador; 4: indicador del estado de conmutación; 5: salida con circuito de protección; 6: salida; 7: tensión externa; 8: fuente interna de tensión constante; 9: condensador con zona activa (Festo Didactic, 2009).

Funcionamiento

Un sensor de posición capacitivo evalúa el cambio de la capacidad de un condensador en un circuito de resonancia RC. Si se acerca una pieza al sensor de posición, aumenta la capacidad del condensador. De esta manera se produce un cambio cuantificable del comportamiento de la resonancia en el circuito RC. El cambio de la capacidad depende fundamentalmente de la distancia, de las dimensiones y de la constante dieléctrica de la pieza. El sensor de posición tiene una salida PNP, lo que significa que, en estado activado, la línea de transmisión de señales cambia a positivo. El conector está normalmente abierto. La conexión de la carga se realiza entre la salida de señales del sensor de posición y la masa. Los diodos luminosos (LED) indican el estado de conmutación. El sensor de posición capacitivo no se puede montar a ras.

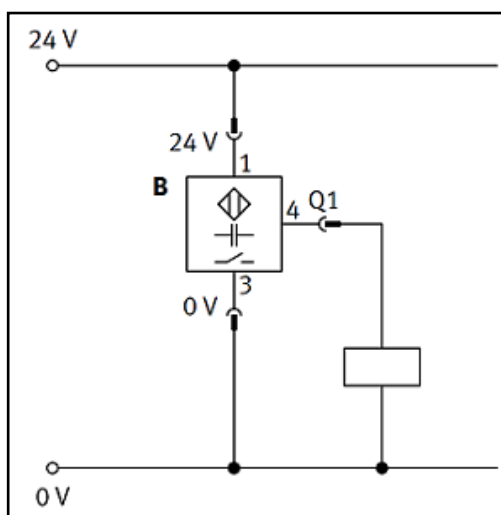


Figura 7-4: Conexión del sensor de posición capacitivo

Fuente: Festo Didactic, 2009, pp. 36

Pistón

Datos técnicos

El pistón utilizado es un DSNU 20 – 100 PPV – A, pistón que posee un diámetro de 20mm con una rosca tipo M8 con amortiguación neumática e indiferente a la posición de montaje, la estructura de este pistón se basa en un barril cilíndrico, en un vástago de émbolo y el pistón el cual esta estandarizado bajo la norma ISO6432 y CETOP RP 52P (Festo AG, 2015). La construcción del pistón está dada por las siguientes características, la barra del pistón está hecha de acero de alta aleación resistente a la corrosión, el barril del cilindro del pistón está hecho de acero de alta aleación resistente a la corrosión, la temperatura de trabajo es de -20° a 80°C (Festo AG, 2015). Las características de trabajo del pistón están dadas en la presión de trabajo de 1 a 10 Bar, la energía de impacto de 0.2 Joules, la longitud de amortiguación es de 15mm, el modo de operación es de doble efecto y el medio de operación es mediante aire comprimido (Festo AG, 2015).

Conexión

La conexión esta descrita en la siguiente imagen:

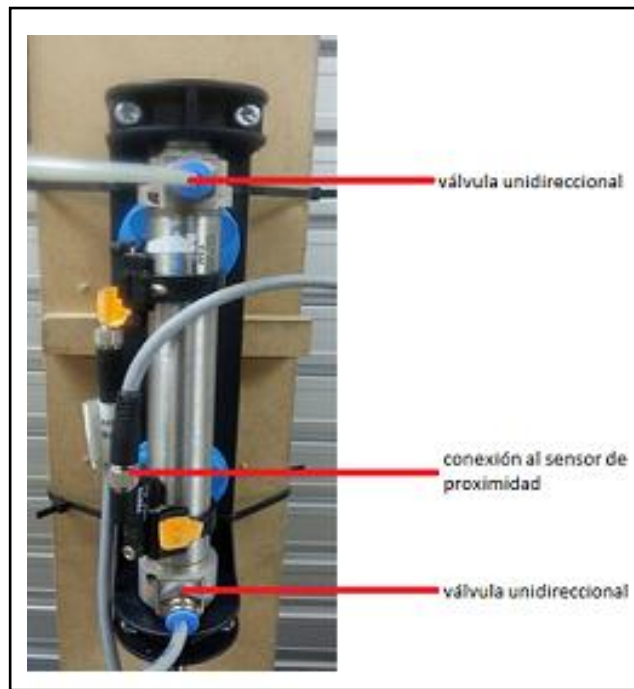


Figura 8-4: Conexión del sensor de posición capacitivo

Fuente: Festo Didactic, 2009, pp. 38

Las válvulas unidireccionales del pistón están conectados a las salidas de la electroválvula.

Termocupla

Termocupla tipo J56200



Figura 9-4: Termocupla tipo J

Fuente: Festo Didactic, 2009, pp. 40

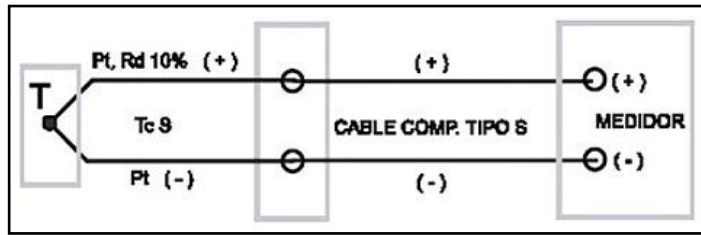


Figura 10-4: Conexión de la Termocupla tipo J

Fuente: Screw Thermocouple, 2016, pp. 18

Tabla 5-4: Datos Técnicos Termocupla

Característica	Propiedades
Ensamble de Termopar	"J" (Hierro-constantan)
Temperatura	0 - 760 °C
Límites de Error de 0 a 350	± 0.5 °C 0.4%
Hot Junction	Tierra

Fuente: Screw Thermocouple, 2016, pp. 18

Compresor

Compresor que trabaja a 110voltios 780Wattios 3.6 amperios con una frecuencia de 60Hz.



Figura 11-4: Compresor

Realizado por: Pamela Espejo. 2017

Datos técnicos

Tabla 6-4: Compresor parámetros técnicos

Característica	Propiedades
Ensamble de Termopar	"J" (Hierro-constantan)
Temperatura	0 - 760 °C
Límites de Error de 0 a 350	± 0.5 °C 0.4%
Hot Junction	Tierra

Fuente: Screw Thermocouple, 2016, pp. 30

Electroválvula biestable de 5/2 vías con LED



Figura 12-4: Electroválvula biestable

Fuente: Festo Didactic, 2009, pp. 42

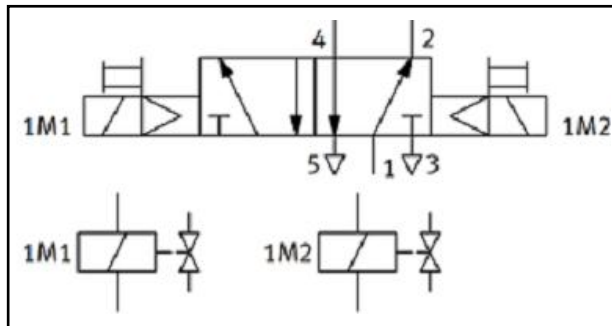


Figura 13-4: Esquema de conexiones

Fuente: Festo Didactic, 2009, pp. 42

Funcionamiento

La electroválvula biestable conmuta al aplicar tensión en una bobina y al retirar la señal mantiene su estado de conmutación hasta la siguiente señal. El diodo luminoso indica el estado de conmutación. La válvula dispone de dos accionamientos manuales auxiliares. La electroválvula se caracteriza por consumir muy poca potencia eléctrica y por calentarse poco. El diodo luminoso y el circuito protector tienen polos inconfundibles. Las conexiones de las bobinas están identificadas con los números 12 y 14.

Datos técnicos

Tabla 7-4: Datos Técnicos Electroválvulas

Datos Técnicos	
Parte neumática	
Fluido	Aire comprimido, con o sin lubricación
Construcción	Válvula de corredera servopilotada

Margen de presión	Desde 300 hasta 800 kPa (desde 3 hasta 8 bar)
Tiempo de respuesta con 600 kPa (6 bar)	15 ms
Caudal nominal normal	500 l/min
Conexión	3 QS-1/8-4-I para tubo flexible PUN 4 x 0,75
Parte eléctrica	
Tensión	24 V DC
Duración de conexión	100 %
Clase de protección	IP65
Conexión	Conector central tipo clavija M 8x1, cable con conector tipo zócalo

Fuente: Festo Didactic, 2009, pp. 43

4.4.2. *Software*

Tabla 8-4: Materiales tipo Software

Cantidad	Descripción
1	TIA Portalssdasd
1	LabVIEW
1	OPC Server
1	Autocad Electrical

Realizado por: Pamela Espejo. 2017

4.4.3. *Especificaciones técnicas de elementos usados*

Tabla 9-4: Especificaciones técnicas de entradas digitales

Entradas Digitales	Sensores Capacitivos	Sensores Inductivos (548645)	Sensor Fotoeléctrico	Compatibilidad con PLC (S71200)
Consumo de energía (mA)	300mA	200	50 mA max	SI
NPN	100		si	SI
PNP	100	Normalmente Abierto	si	SI
Alimentación	10-48 V DC	15 – 34 V DC	10- 30 VDC	SI
Voltaje de Salida			50mA	SI

Tipo de conexión	3 hilos	3 hilos	3 hilos	SI
	2, 3, 4 hilos.			
Normativas CE	CE	CE	CE	SI
Rohs				
Compatibilidad				
Electromagnética				

Realizado por: Pamela Espejo. 2017

Tabla 10-4: Especificaciones técnicas de salidas digitales

Salidas Digitales	Actuador	Compatibilidad con PLC(S71200)
Consumo de energía (mA)	2	no
Alimentación	24	no
Normativas CE Rohs		
Compatibilidad Electromagnética	CE	si

Realizado por: Pamela Espejo. 2017

Tabla 11-4: Especificaciones técnicas de entradas analógicas

Entradas Digitales	Sensor de Presión (548622)	Sensor de temperatura (Pt-1000)	Resolución de la entrada analógica del PLC	Compatibilidad con PLC (S71200)
Consumo de energía (mA)	150	1	1	si
Alimentación	15-30VDC	4-30VDC	0-10V	si
Voltaje de Salida	0-10V	0-30	35VDC	si
Tipo de conexión	4 hilos	4 hilos	2 hilos	si
	2, 3, 4 hilos.			
Normativas CE	CE	CE	CE	si
Rohs				
Compatibilidad				
Electromagnética				

Realizado por: Pamela Espejo. 2017

Tabla 12-4: Especificaciones técnicas de salidas analógicas

Entradas Digitales	Salida necesaria para conectar el	Resolución de la salida analógica	Compatibilidad con PLC (S71200)
--------------------	-----------------------------------	-----------------------------------	---------------------------------

	VDF	del PLC	
Consumo de energía (mA)	5.1A	5 V DC, con 1 mA at 2,5 mA	si
Alimentación	200-240v	24 V	si
Voltaje de Salida	240 V2	5 - 15 v DC	si
Tipo de conexión	2 hilos 2, 3, 4 hilos.	2 hilos	si
Normativas CE	CE	CE	CE
Rohs			
Compatibilidad Electromagnética			

Realizado por: Pamela Espejo. 2017

4.4.4. *Conexión*

En el presente diagrama (ver Figura 14-4) se representa como van las conexiones de aire y del variador de frecuencia. Ver planos de conexión en Anexo B.

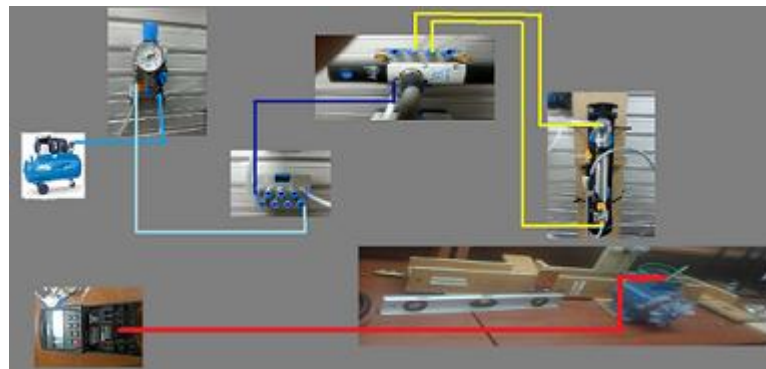


Figura 14-4: Diagrama

Realizado por: Pamela Espejo. 2017

- Conexión del aire que sale del compresor hacia la unidad de mantenimiento donde se va a regular la cantidad de aire o presión con la que se va a trabajar en los siguientes dispositivos.
- Aire regulado que sale desde la unidad de mantenimiento hacia la unidad de distribución, donde se va a tomar para el aire regulado para los diferentes procesos.
- Toma de aire que llega a la electroválvula la cual va a ser controlada para los procesos con el pistón.
- Aire que es controlado por la electroválvula y va directo a la activación del pistón.

■ Conexión de las líneas de alimentación del motor AC con el variador de frecuencia para el control de la velocidad de giro.

4.4.5. Programación del PLC- TIA PORTAL

La configuración lógica del funcionamiento del sistema (Anexo C) se estructura se presenta en la Figura 15-4:

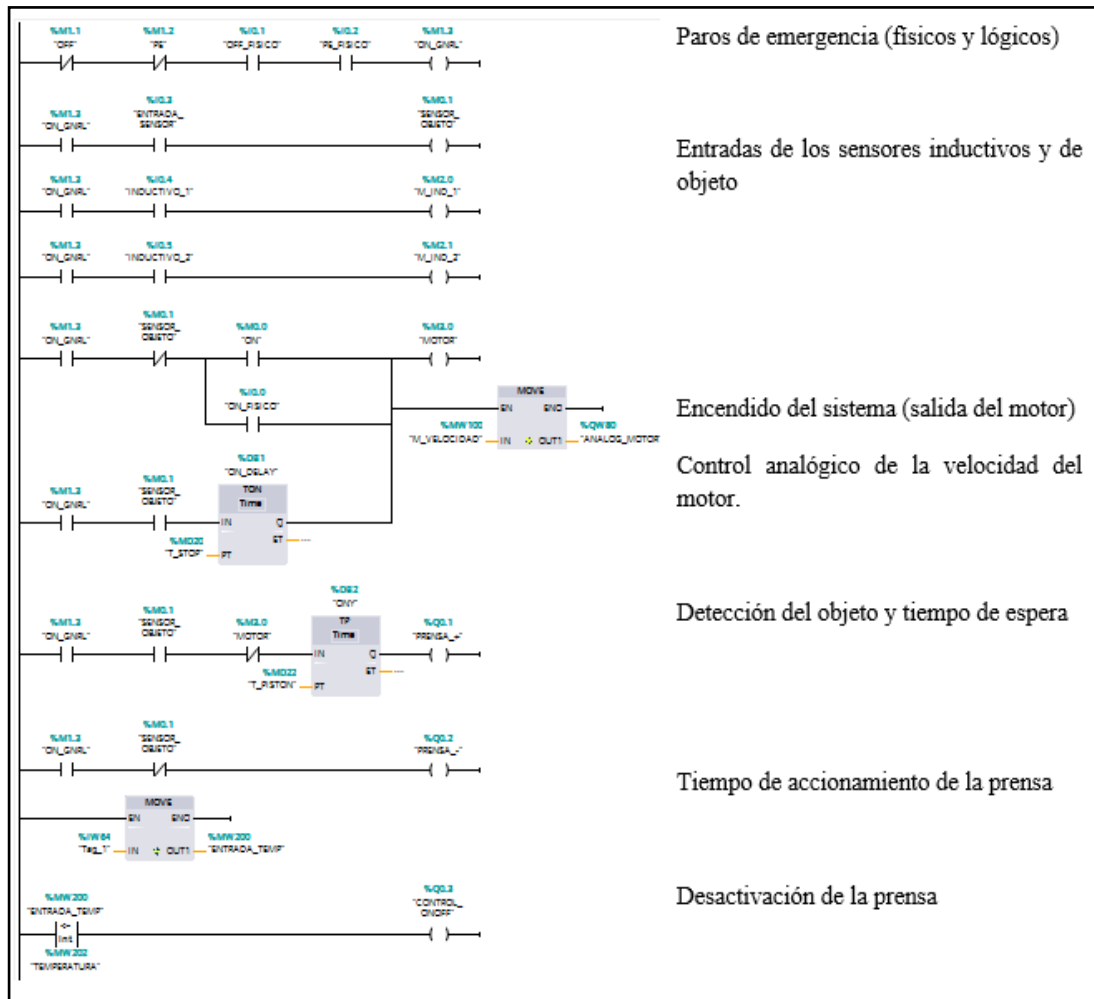


Figura 15-4: Programación del PLC- TIA PORTAL

Realizado por: Pamela Espejo. 2017

4.4.6. Tablas de características de variables

Tabla 13-4: Variables del PLC

Name	Path	Data Type	Logical Address	HMI Visible	HMI Accessible
ON	Tabla de variables estándar	Bool	%M0.0	True	True
MOTOR	Tabla de variables estándar	Bool	%M3.0	True	True
SENSOR_OBJETO	Tabla de variables estándar	Bool	%M0.1	True	True
OFF	Tabla de variables estándar	Bool	%M1.1	True	True
PE	Tabla de variables estándar	Bool	%M1.2	True	True
ON_GNRL	Tabla de variables estándar	Bool	%M1.3	True	True
PRENSA_+	Tabla de variables estándar	Bool	%Q0.1	True	True
PRENSA_-	Tabla de variables estándar	Bool	%Q0.0	True	True
M_IND_1	Tabla de variables estándar	Bool	%M2.0	True	True
ON_FISICO	Tabla de variables estándar	Bool	%I0.0	True	True
OFF_FISICO	Tabla de variables estándar	Bool	%I0.1	True	True
PE_FISICO	Tabla de variables estándar	Bool	%I0.2	True	True

Realizado por: Pamela Espejo. 2017

Tabla 14-4: Variables del PLC

Name	Path	Data Type	Logical Address	HMI Visible	HMI Accessible
-------------	-------------	------------------	------------------------	--------------------	-----------------------

ENTRADA_SENSOR	Tabla de variables estándar	Bool	%I0.3	True	True
INDUCTIVO_1	Tabla de variables estándar	Bool	%I0.4	True	True
INDUCTIVO_2	Tabla de variables estándar	Bool	%I0.5	True	True
M_IND_2	Tabla de variables estándar	Bool	%M2.1	True	True
T_STOP	Tabla de variables estándar	Bool	%MD20	True	True
T_PISTON	Tabla de variables estándar	Bool	%MD22	True	True
M_VELOCIDAD	Tabla de variables estándar	Bool	%MW100	True	True
ANALOG_MOTOR	Tabla de variables estándar	Bool	%QW80	True	True
Tag_1	Tabla de variables estándar	Bool	%IW64	True	True
ENTRADA_TEMP	Tabla de variables estándar	Bool	%MW200	True	True
TEMPERATURA	Tabla de variables estándar	Bool	%MW202	True	True
CONTROL_ON/OFF	Tabla de variables estándar	Bool	%Q0.3	True	True

Realizado por: Pamela Espejo. 2017

4.4.7. Configuración del variador para el Motor AEG TYP AM 63 NY4

Tabla 15-4: Bloque de programación

Código	Código	Descripción
P10	1/0	Reset / Normal
P100	2	Entrada configuración americana
P304	440	Configuración Estrella motor
P305	1.12A	Amperaje Motor
P307	0.11kw	HP
P310	60.00Hz	Frecuencia
P311	1550xmn ⁻¹	Rpms
P700	1	Entrada Analógica
P1000	2	Ingreso de variador Análogo

P1080	0.00	Voltaje inicial motor
P1082	60.00	Hz Frecuencia
P1120	10.00	Rango Voltaje
P1121	10.00	Tiempo respuesta 10V
P3900	1/0	Guarda cambios / Estado normal

Realizado por: Pamela Espejo. 2017

4.4.8. CONFIGURACIÓN OPC SERVER

Creación de un nuevo canal, que permita la comunicación entre LabVIEW y PLC:

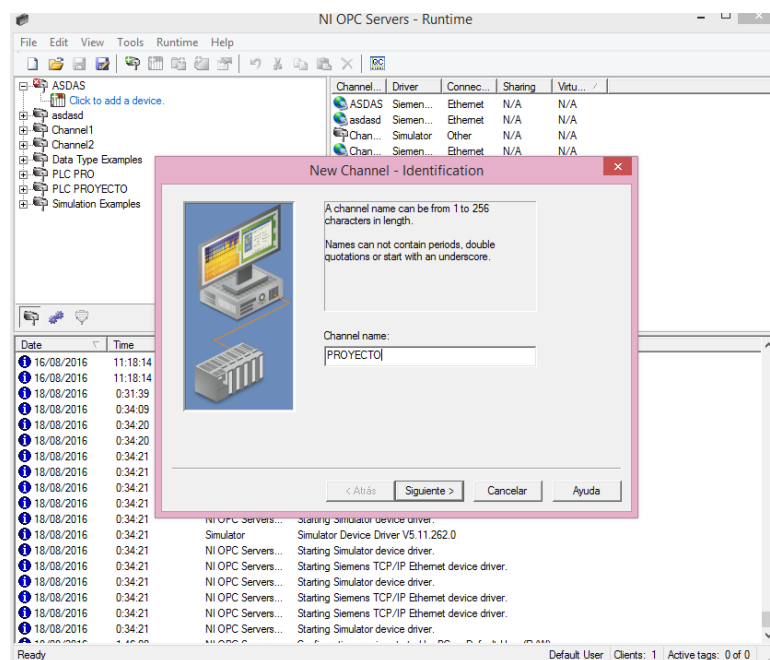


Figura 16-4: Creación de un nuevo canal

Realizado por: Pamela Espejo. 2017

Seleccionar la opción Siemens TCP/IP Ethernet:

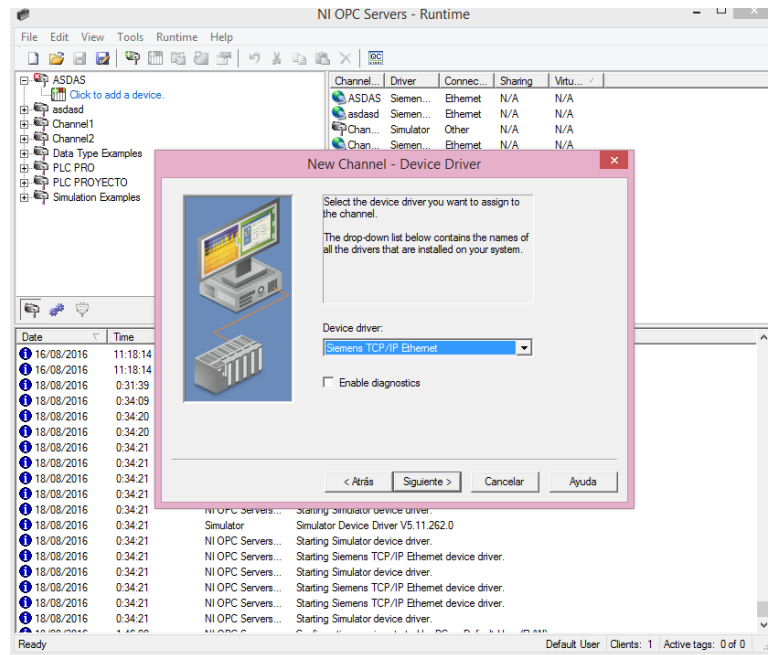


Figura 17-4: Siemens TCP/IP Ethernet

Realizado por: Pamela Espejo. 2017

Una vez agregado el canal, agregar el dispositivo del PLC S71200

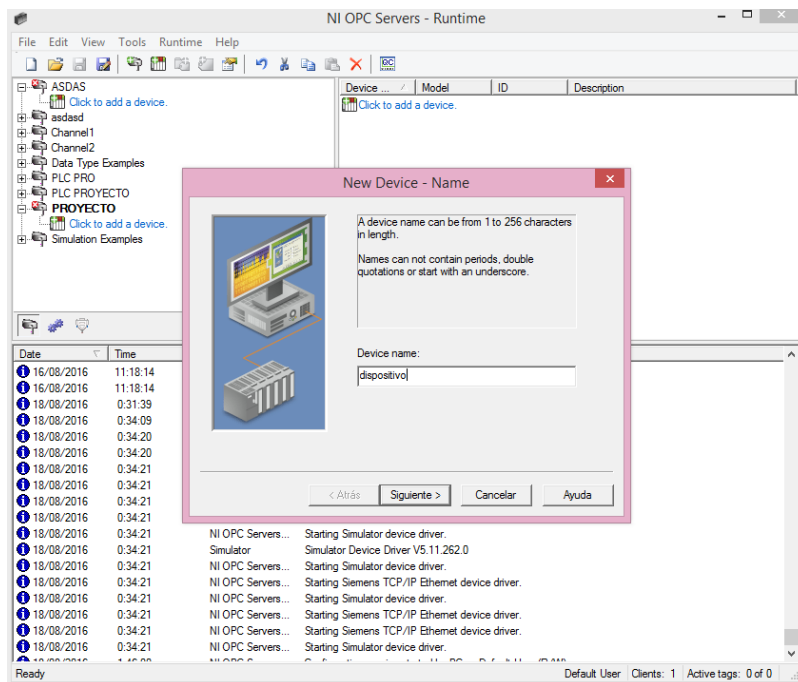


Figura 18-4: Agregar el dispositivo del PLC S71200

Realizado por: Pamela Espejo. 2017

Establecer la dirección del PLC 192.168.0.1, que se encontrar en la configuración de dispositivos de Software TiaPortal.

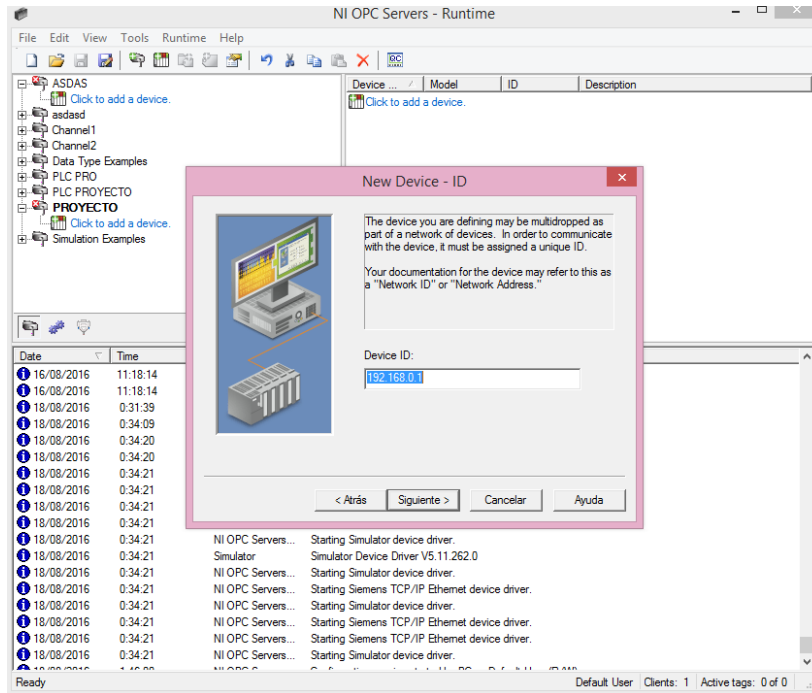


Figura 19-4: Agregar dirección en PLC 192.168.0.1

Realizado por: Pamela Espejo. 2017

Se agrega las variables que se utilizaran en el programa, dicho procedimiento se realizara cuantas veces se requieran en el programa:

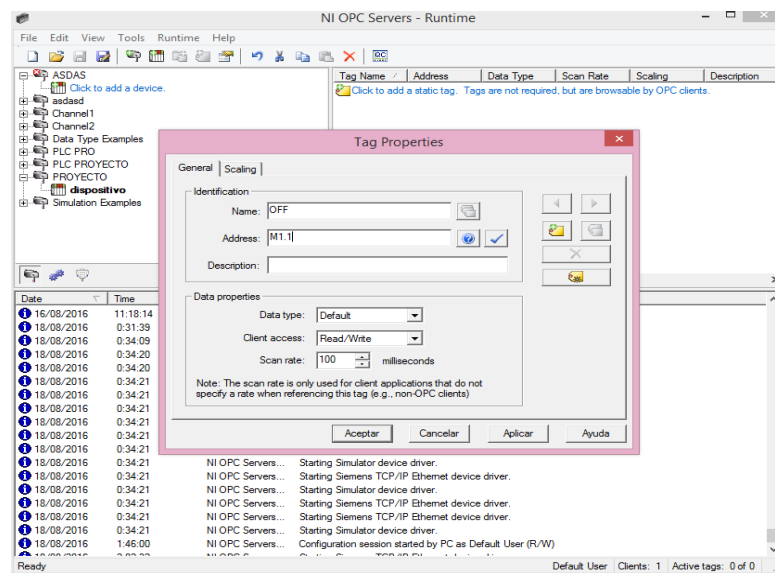


Figura 20-4: Asignación de variables

Realizado por: Pamela Espejo. 2017

4.4.9. Programación LABVIEW

En la interfaz de programación de LabVIEW, se desarrolló un VI (ver Figura 3.21) que permite el enlace con las variables del software OPC Server. Para el enlace con la interfaz de LabVIEW y el PLC, se utiliza las variables a controlar desde la interfaz del PLC a variables como: controles del motor (encendido y apagado), manipulación de temperatura, indicadores para sensores, motores, temperatura, etc. Ver programación completa en Anexo D.

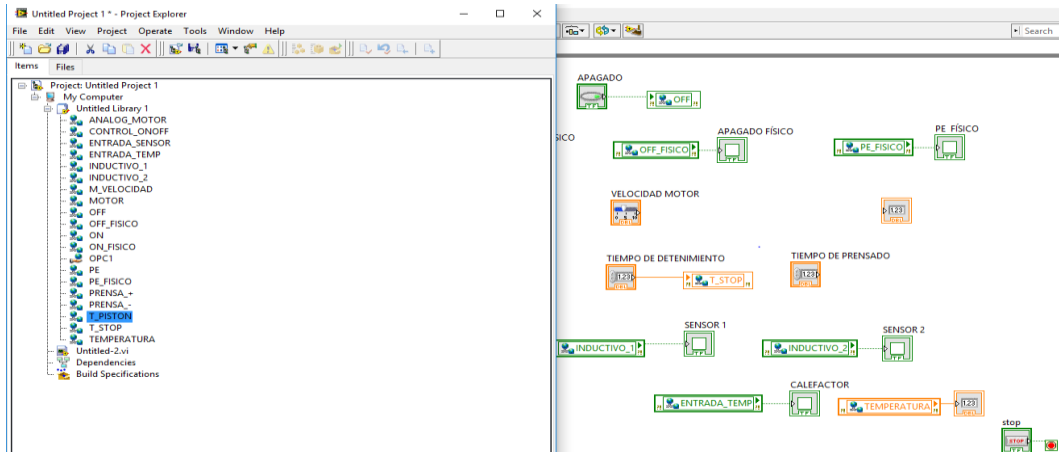


Figura 21-4: VI - Enlace de variables a controlar

Realizado por: Pamela Espejo. 2017

Las variables creadas en el OPC Server de la PC, se selecciona a variables del tipo lectura (booleana), como se ilustra en la Figura 21-4

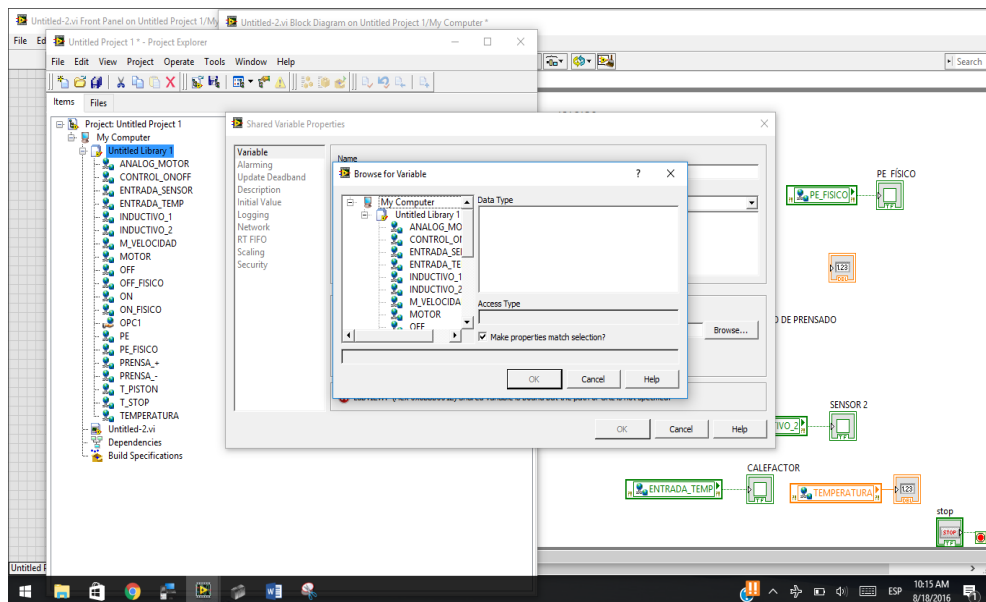


Figura 22-4: Variables utilizadas en el OPC Server

Realizado por: Pamela Espejo. 2017

En la Figura 23-4, se indica las variables creadas en el OPC server, con su descripción y función dada.

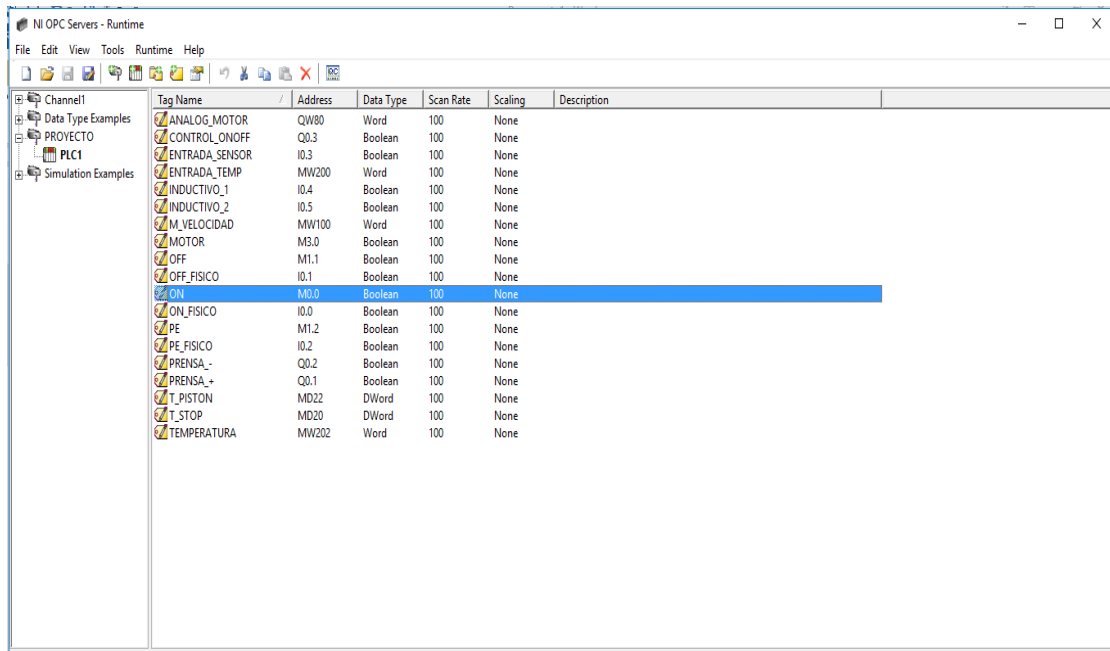


Figura 23-4: Configuración de variables en OPC Server

Realizado por: Pamela Espejo. 2017

4.4.10. Configuración Física y Lógica del sistema

La configuración física y lógica del sistema se detallada en los Anexos 2 respectivamente.

4.5. Prototipo del sistema de control con hardware y software libre

Materiales a utilizarse

Tabla 16-4: Materiales tipo Hardware

Cantidad	Descripción
1	Sensores de proximidad SMT8.
3	Sensor de temperatura
1	Sensor de presión 539757.
1	Sensor de posición inductivo M18, montaje saliente 548645
1	Motor trifásico AEG TYP AM 63 NY4
1	Electroválvula biestable de 5/2 vías con LED 539778
1	Banda transportadora 16.5x40 cm.
1	Pistón 152888
1	PLC-MDuino
10	Cables con jacks tipo banana

1	Variador de Frecuencia Micromaster 440
1	Válvula de cierre con unidad de filtro y regulador 540691
1	Bloque de distribución (152896)
5	Canaletas
1	Multímetro.
4	Desarmadores
6	Mangueras neumáticas.

Realizado por: Pamela Espejo. 2017

Software

Tabla 17-4: Materiales tipo Software

Cantidad	Descripción
1	Servidor LAMP
1	Lenguaje HTML
1	Lenguaje PHP

Fuente: Pamela Espejo. 2017

4.6. Análisis de Hardware Libre

4.6.1. Especificaciones técnicas de elementos usados

En el Tabla se hace referencia de las características técnicas de los dispositivos usados para la implementación de la prensa neumática basada en hardware libre, a continuación, se describe al PLC-M DUINO, que es un sistema profesional con interfaces de comunicación y tecnología Arduino para mejorar la flexibilidad y control, puede trabajar hasta 127 módulos a través de I2C que le permiten tener hasta 7100 entradas y salidas generando conexiones master/slave.

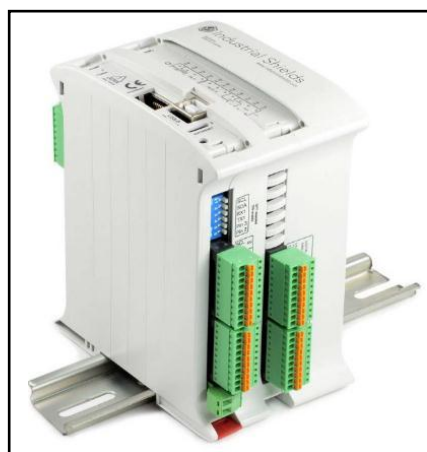


Figura 24-4: PLC Arduino 24 Vcc M-Duino

Realizado por: Pamela Espejo. 2017

Las especificaciones técnicas son detalladas en la Tabla 18-4

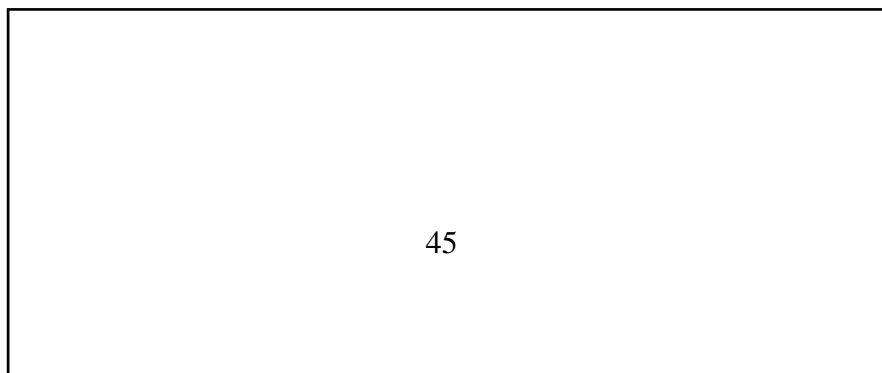
Tabla 18-4: Especificaciones Técnicas del PLC M-Duino

TECNOLOGÍA	PLC M-Duino
ETHERNET	1 PUERTO
USB	1 PUERTO
RS-232	4 PUERTOS*
RS-485	4 PUERTOS*
I2C	1 PUERTO
MODBUS	1 PUERTO
Líneas	SDA y SCL
Número de bits del Bus	8 bits
Velocidad Transmisión (modo estándar)	100 Kbps
Velocidad Transmisión (modo rápido)	400 Kbps
Tensión mínima entrada	1,5 V
Frecuencia de trabajo	100-400 KHz
ESPECIFICACIONES TÉCNICAS	
Entradas Digitales	6 Entradas Análogo/Digital 5 Entradas PNP Digital 2 Entradas Interrupción Digital
Entradas Analógicas	6 Entradas Análogo/Digital
Salidas Digitales	3 Salidas Análogas / Digital /PWM 5 salidas PNP Digital
Salidas Analógicas	3 Salidas Análogas / Digital /PWM

Realizado por: Pamela Espejo. 2017

4.6.2. *Conexión*

En la siguiente Figura 25-4 se indica las conexiones físicas entre los diferentes elementos que conforman el sistema.



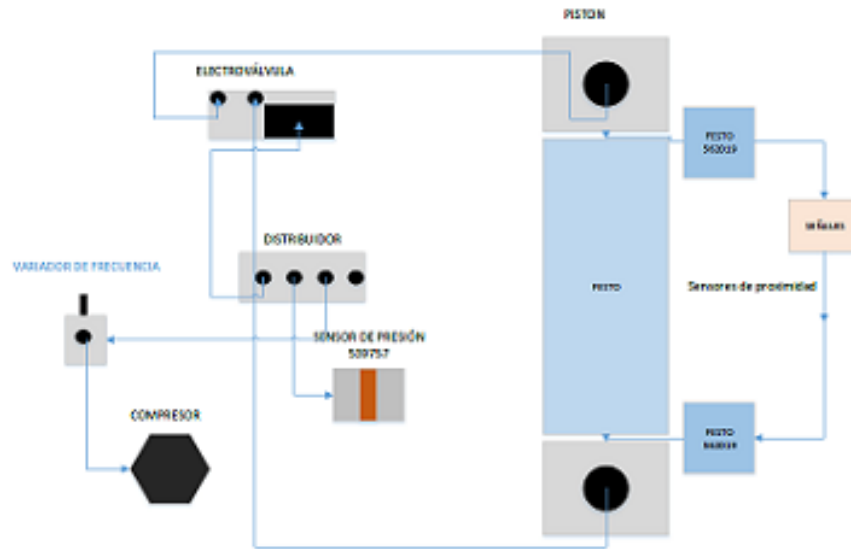


Figura 25-4: Conexión

Realizado por: Pamela Espejo. 2017

4.6.3. Programación del PLC-M-Duino

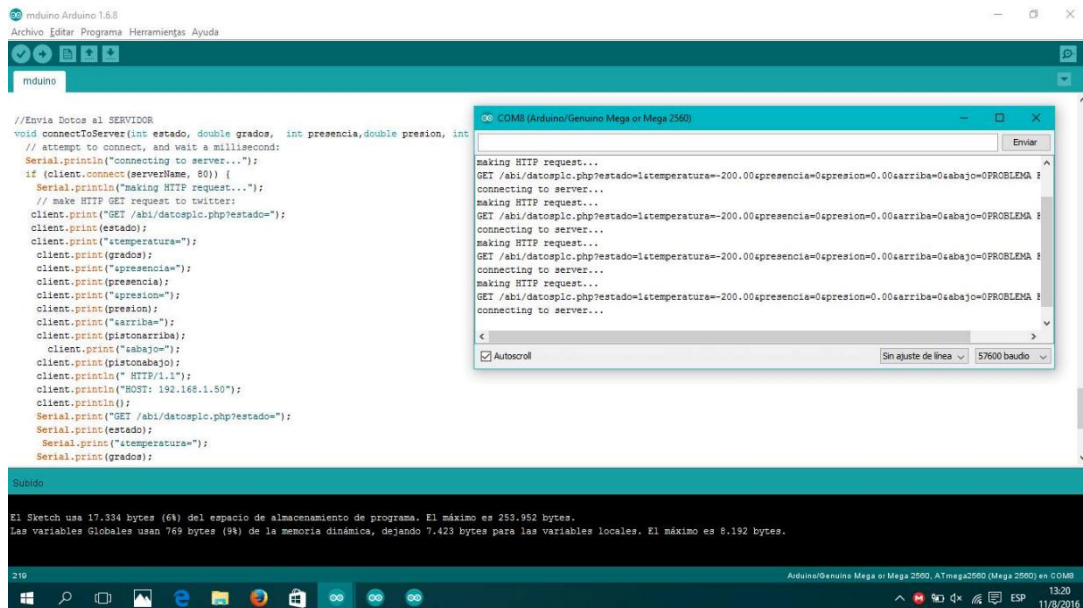


Figura 26-4: Algoritmo de programación del M-Duino

Realizado por: Pamela Espejo. 2017

M_DUINO PLC cumple la función de integrador de tanto la parte mecánica, física, actuadores, sensores y la comunicación entre ellos. El M-Duino recoge el valor digital que envía el sensor de proximidad el cuál varía de 0 a 24 [V], esta información es manipulada por el PLC el cual la transforma en salidas analógicas: alto detecta presencia y ordena que la banda transportadora se

detenga, baja el pistón por un lapso de 3 [s] se inicia un tiempo de retardo para que vuelva a correr la banda y vuelva a sensar. Ver código completo en Anexo E.

4.6.4. *Ensamblaje del mecanismo neumático*

En el proceso de construcción de la maqueta se muestra en la Figura 27-4, el proceso de unión de la banda transportadora con la base del pistón; este proceso consiste en dimensionar posiciones de cada uno de los sensores que se utilizara, con el objetivo de conseguir distancias exactas para cada uno de los elementos que se añadirán posteriormente.

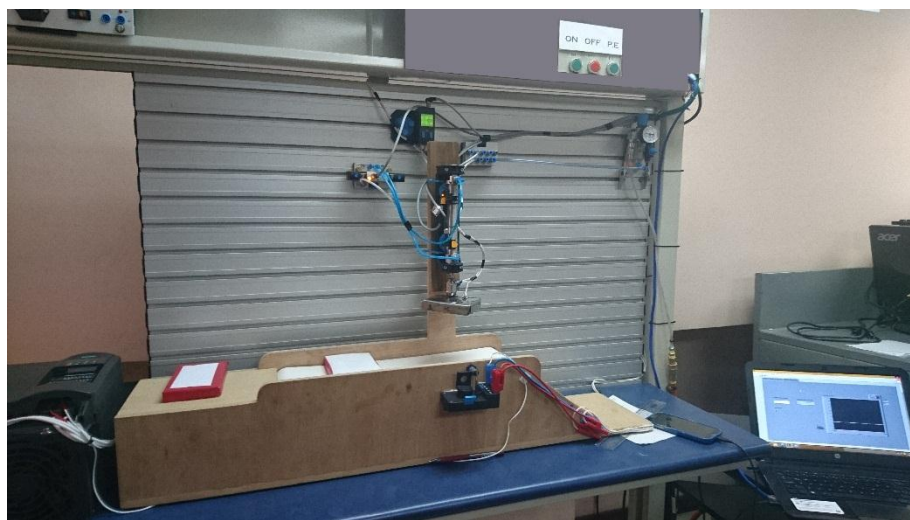


Figura 27-4: Ensamblaje del mecanismo neumático

Realizado por: Pamela Espejo. 2017

A continuación, se detalla los pasos a seguir para el ensamblaje del prototipo:

- Colocación de los diferentes actuadores para la simulación completa de las etapas de control.
- Proceso de programación y configuración de M_DUINO PLC
- Conexión de compresor neumático para la unidad de mantenimiento para la obtención de la presión necesaria para el funcionamiento óptimo del pistón de forma correcta, la presión regulada a la que se va a manejar para el funcionamiento es de 2 bares, esto se lo puede regular desde el actuador mismo; si el rango de la presión se sale de ese intervalo las diferentes alarmas se irán activando paulatinamente.
- Se realiza la conexión de la etapa anterior al bloque de distribución que es el encargado de llevar la compresión del aire a los diferentes sensores a los que va a controlar para una seguridad mayor, en esta misma etapa se coloca una manera de flujo de aire de distribución que va a estar conectada a una electroválvula que es la cual va a ser la encargada de regular

la salida del aire comprimido tanto al sensor de presencia como al de presión que se los va a detallar más adelante.

- Conexión del sensor de presencia y presión
 - Sensor de Presión: Este actuador va a ser el encargado de verificar la presión existente que ingresa al pistón ya que como lo habíamos detallado anteriormente la presión correcta que se va a manejar es de 2 bares. Consta de salidas de voltaje de 24V cada una cada una de ellas se las va a utilizar para los dos momentos del pistón cuando este arriba y el otro cuando este abajo así se lo ha programado.
 - Sensor de Presencia: Este actuador va a ser el encargado de verificar la presencia del embolo interno que maneja el pistón a la presión correcta de 2 BARES. Ya que este tipo de sensor maneja 0-10V la equivalencia de conversión a bares es la misma, es decir que 2 Bares = 2V. Con esto se asegura el óptimo funcionamiento del sensor ya que si se sale de los 2V permitidos se activará la alarma programada.
- Conexión de pistón y manejo de posición.
 - El pistón tiene 2 posiciones arriba y abajo cada una manejada por un voltaje de 24 V así se lo programo para mayor seguridad. y confiabilidad. Con esto se lo habilita y se procede a su prueba.

4.6.5. *Ensamblaje del mecanismo neumático*

Características de funcionamiento

Para lograr la adquisición y almacenamiento de datos de las variables disponibles en la prensa neumática primeramente se habilita el puerto ethernet o el puerto USB del Arduino del M-Duino mediante las librerías Ethernet Shield para obtener la comunicación del PLC con el ordenador para su configuración y compilación.

La presencia de un parpadeo constante de dos Leds verifica la habilitación del puerto.

Se hace uso de un servidor WEB, el servidor web instalado es APACHE (de software libre), en este servidor se encuentra instalado la base de datos PhpMyAdmin mediante MariaDB.

Creación de base de datos

Para crear la base de datos primero se debe instalar el APP server con todos sus componentes y contar con sistema operativo con todas las herramientas necesarias:

- Apache.

- MySQL.
- MariaDB.
- PhpMyAdmin.
- FTP.

Una vez realizada la instalación de todos los componentes para poder crear bases de datos se siguieron los siguientes pasos:

1. Escoger el nombre para la base de datos, como ejemplo se tiene: ejemplo.
2. Para ingresar a la base de datos de phpMyAdmin se ingresa en el navegador la dirección URL de la ubicación de base de datos: 192.168.1.50/phpMyAdmin/. Se muestra la siguiente interfaz:
3. El usuario actual es root y la contraseña jr.

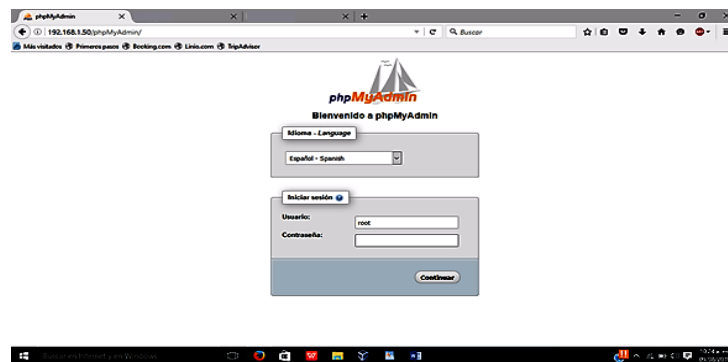


Figura 28-4: Pantalla inicio de base de datos en phpMyAdmin

Realizado por: Pamela Espejo. 2017

4. Se ubica en la sección Base de datos
5. Crear nueva base de datos, donde se coloca un nombre para la nueva base de datos MySQL, ejemplo.
6. Crear

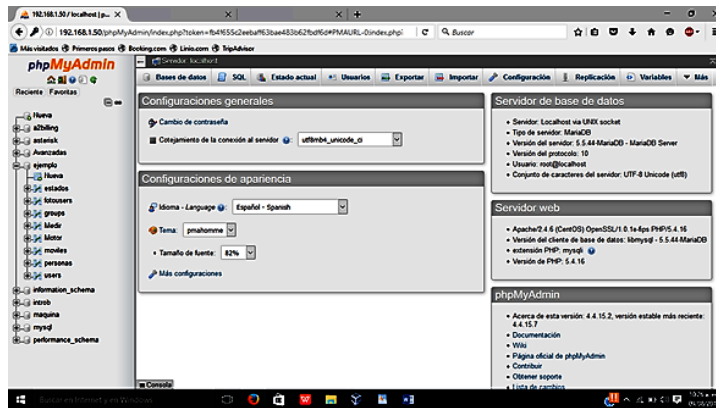


Figura 29-4: Base de datos en phpMyAdmin

Realizado por: Pamela Espejo. 2017

7. La creación de la tabla de datos.
8. La interfaz gráfica permite fácilmente crear un esquema de filas y columnas, la base de datos, las filas y columnas se actualizan según lo establecido en la programación del Arduino y el periodo de muestra de datos de la prensa, los datos manejados y almacenados son:

- ID.
- Estado (on/off).
- Temperatura.
- Presencia.
- Presión.
- Arriba (pistón).
- Abajo (pistón).
- Atasco (pistón).
- Fecha y hora.

9. En la tabla se muestran los valores en tiempo de real de todas las variables obtenidas por los sensores, procesadas por el PLC Arduino y almacenadas en la base de datos.

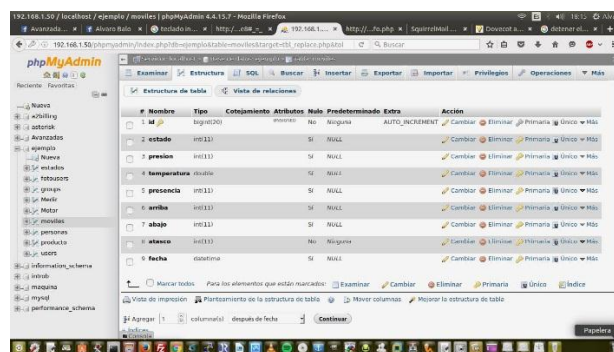


Figura 30-4: Tabla de datos en phpMyAdmin Móviles

Realizado por: Pamela Espejo. 2017

La base de datos genera diferentes tablas las cuales serán utilizadas en la aplicación Web para la visualización de la información de una manera más efectiva para el usuario.

Manejo de la información adquirida

La programación de la adquisición desde el puerto Ethernet del Arduino se la realizó con lenguaje php. Se usó del programa Dreamweaver para subir los archivos programados al servidor WEB mediante FTP.

Pasos para la creación de un sitio en Dreamweaver:

1. Al abrir Dreamweaver, un nuevo sitio desde el menú Sitio.
2. Administrar nuevos sitios, y colocamos y el nombre del sitio y su ubicación en el disco local.
3. En el menú servidores configuramos la comunicación con nuestro servidor mediante FTP.

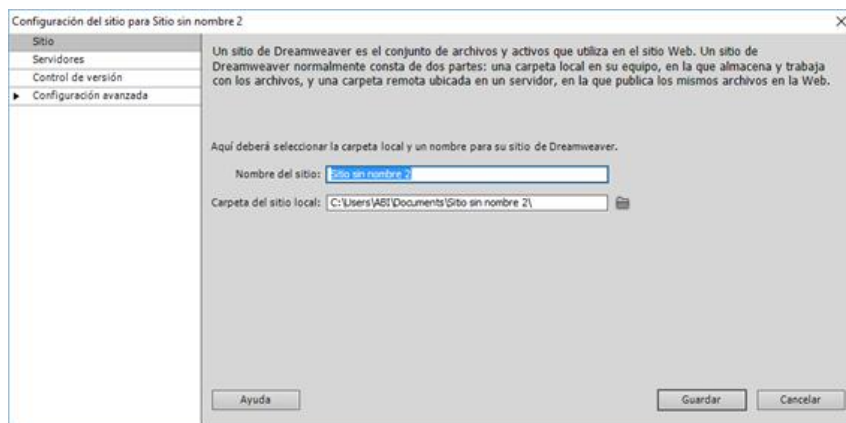


Figura 31-4: Creación de nuevo sitio en Dreamweaver

Realizado por: Pamela Espejo. 2017

4. Se realizan las siguientes configuraciones:

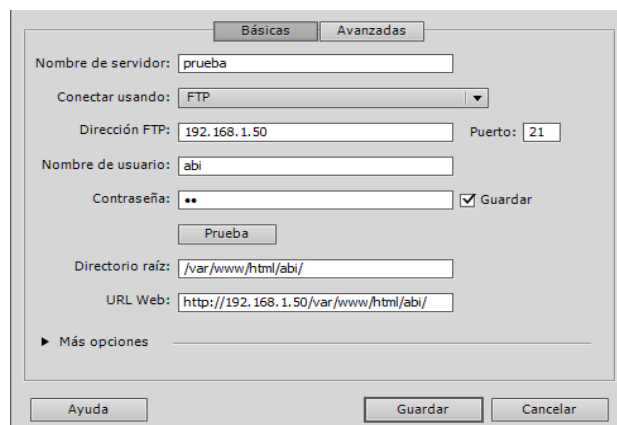


Figura 32-4: Parámetros de configuración de comunicación con el servidor FTP en Dreamweaver

Realizado por: Pamela Espejo. 2017

Monitoreo y Presentación de información del sistema

Características de funcionamiento

En un servidor FTP se guardan los archivos de la página Web creada, y se los transfiere por medio de filezilla, además el servidor web implementado usa la información de la base de datos My-SQL previamente creada para la presentación de la información adquirida. Para el acceso a esta página Web solamente usuarios registrados con anterioridad pueden ver el historial de los datos adquiridos, por medio de una clave de acceso remoto al sistema.

Este acceso es específicamente para monitoreo y presentación de la información de las variables manejadas en la prensa neumática además del estado de alarmas e indicadores. Todos estos parámetros visualizados en diagramas amplitud vs tiempo.

El monitoreo y presentación de información del sistema incluye los siguientes parámetros:

- Estado ON/OFF de la prensa
- Temperatura de la prensa (°C)
- Sensor de presión (Bar)
- Conteo de productos
- Estado del Pistón, arriba o abajo.
- Atascos del Pistón.
- Fecha y hora

El sistema de monitoreo es amigable al usuario con una interfaz gráfica fácil de manejar y con herramientas que facilitan el acceso a la información de datos en tiempo real e históricos del proceso.

Previo a el desarrollo del servidor FTP y la Aplicación Web, la máquina que funcionará como servidora debe cumplir con los requerimientos base de estructura funcional de Centos basado en Linux/GNU, para este tipo de aplicaciones para desarrollo Web (como Apache, Mariadb, LAMP, Joomla, etc).

Configuración del servidor FTP

Para configurar el servidor FTP se requieren seguir los siguientes pasos:

- Ingresar al modo root
- Instalar los paquetes para el servicio FTP con el comando: `yum install vsftpd`
- Ingresar al archivo del servidor FTP a través del comando: `vi /etc/vsftpd/vsftpd.conf`
- Una vez que se despliegue el archivo, descomentar los siguientes parámetros:
 - `anonymous_enable=NO`
 - `local_enable=YES`
 - `write_enable=YES`
 - `chroot_local_user=YES`
- Ingresar las siguientes líneas dentro del archivo:
 - `allow_writeable_chroot=YES`
 - `pasv_enable=Yes`
 - `pasv_min_port=40000`
 - `pasv_max_port=40100`
- Guardar los cambios e iniciar el servicio vsftpd mediante el comando: `systemctl restart vsftpd.service`
- Configurar para que el servicio se inicie automáticamente con el comando: `systemctl enable vsftpd.service`
- Una vez culminada la configuración, verificar que el servicio este activado ingresando el comando: `service vsftpd status`
- A continuación, se desplegará la información del estado del servicio en donde aparecerá que se encuentra activo. Figura 33-4

```

cdxf@localhost/home/cdxf
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
├─2286 /usr/sbin/httpd -DFOREGROUND
├─2287 /usr/sbin/httpd -DFOREGROUND
├─2288 /usr/sbin/httpd -DFOREGROUND
├─2289 /usr/sbin/httpd -DFOREGROUND
└─2290 /usr/sbin/httpd -DFOREGROUND

may 11 17:32:30 localhost.localdomain httpd[974]: AH00558: httpd: Could not r...
may 11 17:32:32 localhost.localdomain systemd[1]: Started The Apache HTTP Ser...
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost cdxf]# service vsftpd status
Redirecting to /bin/systemctl status vsftpd.service
vsftpd.service - Vsftpd ftp daemon
Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled)
Active: active (running) since lun 2015-05-11 17:32:16 ECT; 1h 35min ago
Process: 963 ExecStart=/usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf (code=exited,
status=0/SUCCESS)
Main PID: 969 (vsftpd)
CGroup: /system.slice/vsftpd.service
└─969 /usr/sbin/vsftpd /etc/vsftpd/vsftpd.conf

may 11 17:32:16 localhost.localdomain systemd[1]: Starting Vsftpd ftp daemon...
may 11 17:32:16 localhost.localdomain systemd[1]: Started Vsftpd ftp daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost cdxf]#

```

Figura 33-4: Estado del servicio vsftpd

Realizado por: Pamela Espejo. 2017

- Crear un usuario, el cual tendrá los permisos necesarios para poder acceder al servidor y manejar los archivos, para esto se deben ingresar los siguientes comandos:
 - `useradd -m user -s /sbin/nologin`
 - `passwd user`

- mkdir /home/user/www
- mkdir /var/www/html/user /home/user/www
- mount –bind /var/www/html/user /home/user/www
- chmod 777 /var/www/html/user
- Finalmente, para verificar la conexión entre el cliente y el servidor se emplea el software FileZilla el cual permite comprobar que la conexión se realice correctamente y a su vez facilita la transferencia de archivos con el servidor.
- A continuación, se procedió a crear un archivo html denominado “Inicio”, el cual será transferido hacia el servidor para poder acceder a él desde el navegador (ver Figura 34-4).

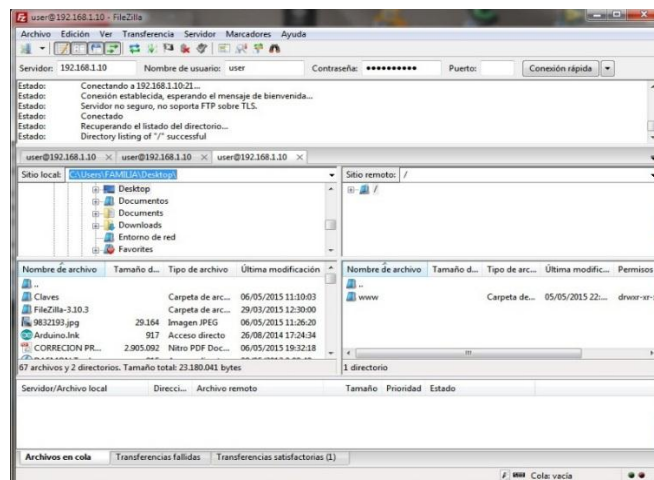


Figura 34-4: Transferencia de archivos hacia el servidor

Realizado por: Pamela Espejo. 2017

- Finalmente, para verificar que el archivo se haya cargado correctamente, se ingresó en el navegador la dirección IP del servidor seguido de la ubicación del archivo html: 192.168.1.10/user/Inicio.html (ver Figura 35-4).



Figura 35-4: Archivo Html subido al servidor

Realizado por: Pamela Espejo. 2017

- Como se puede evidenciar la conexión con el servidor y la transferencia de archivos a través del servicio ftp se realizó correctamente, culminando de esta forma la presente práctica.

Componentes de la Aplicación Web

La aplicación está basada en 3 componentes:

- PHP (Backend Language)
- Javascript (AngularJS) (Frontend Framework)
- MySQL (Base de Datos)

Código PHP de la página, disponible en Anexo F y el código HTML se presenta en el Anexo G.

La aplicación usa lenguaje PHP para la capa de acceso de datos (back-end), con este lenguaje se generaron los siguientes archivos que son la base del funcionamiento del sistema de monitoreo y presentación de información:

- AppController. - Es el corazón de la aplicación Web y recoge la información de la base de datos. (Ver código en Anexo H)
- DatosController. - Es el código que guarda las características del servicio Web y trabaja junto el AppController para el manejo de datos. (Ver código en Anexo I)
- LoginController. - Código desarrollado para el registro de usuarios y contraseñas (el usuario actual es root y la contraseña root), estos usuarios que se incluyen en el código con su respectiva contraseña son almacenados y registrados en la base de datos. (Ver código en Anexo J)
- UserController. - Permite la vinculación de los usuarios, incluyendo el registro de usuarios desde el Frontend Framework y tiene iteración directa con el LoginController. (Ver código en Anexo K)

Estos 4 archivos construidos con la plantilla de programación AngularJS y lenguaje PHP son transferidos al servidor FTP con ayuda de la herramienta Filezilla.

AngularJS fue utilizado como plantilla de programación para PHP, por medio de Interfaces Javascript generando uno de los componentes del sistema como es el Frontend Framework. Permitiendo la organización del código y re-uso de código. Así como la programación orientada a componentes interactivos como menús desplegables, dashboards, configuraciones, idioma, etc.

Para el acceso a la presentación de información primero se accede a la URL del servidor Web, la dirección IP es asignada por la red de área Local por DHCP o se puede establecer una IP estática.

Un ejemplo para el acceso de un PC remoto dentro de la misma red es ingresar en el navegador la dirección URL: 192.168.1.205/inicio#/login.

Como resultado se muestra en la siguiente Figura 36-4.

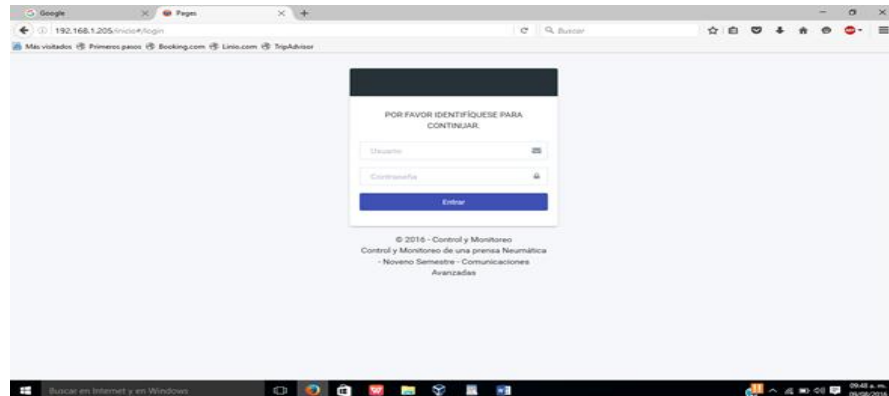


Figura 36-4: Login de Usuarios

Realizado por: Pamela Espejo. 2017

El usuario actual es root y la contraseña root, a continuación, la página de bienvenida del sistema de monitoreo y presentación:

Esta interfaz está compuesta por varios bloques como son: Navegación principal, logo de bienvenida, grupos y usuarios, menú de configuraciones y acceso directo.

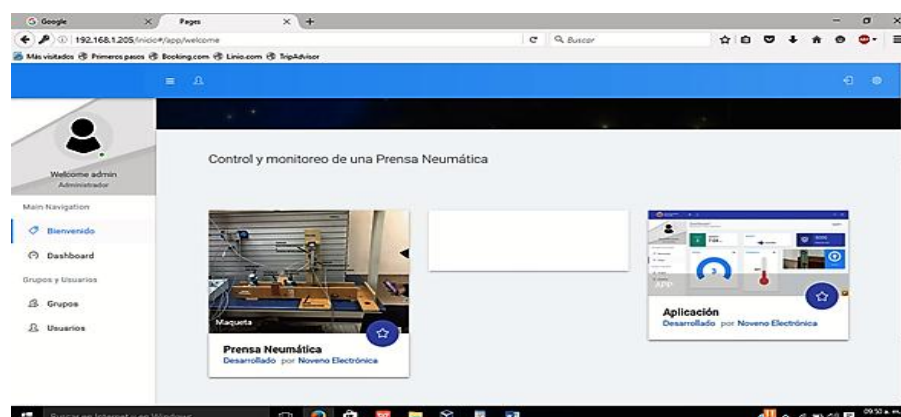


Figura 37-4: Página de bienvenida del sistema de monitoreo y presentación 2

Realizado por: Pamela Espejo. 2017

En el componente de menú principal Dashboard se muestra: Estado ON/OFF de la prensa, Temperatura de la prensa (°C), Sensor de presión (Bar), Conteo de productos, Estado del Pistón, arriba o abajo, Fecha y hora.

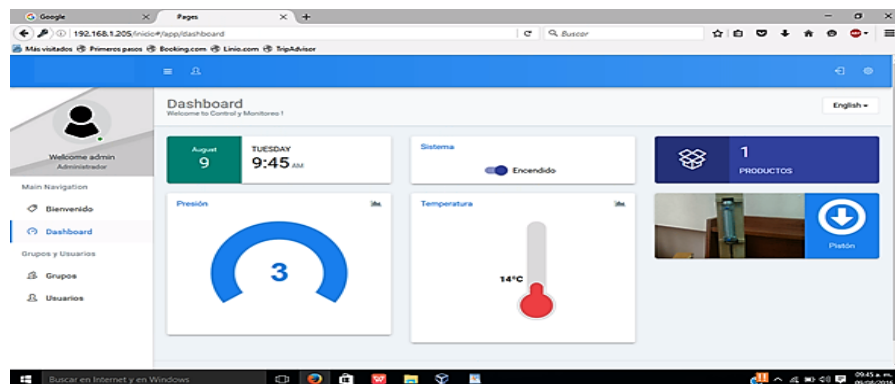


Figura 38-4: Dashboard y estado de sensores

Realizado por: Pamela Espejo. 2017

Menú Grupos: se tiene 2 tipos de grupos:

- Administrador. - Usuarios que tienen acceso a la administración de base de datos y al monitoreo de las variables.
- Operador. - Usuarios que pueden únicamente acceder al monitoreo de las variables.
- Menú Usuarios: Contiene la cantidad de usuarios registrados en el sistema y además se tiene la posibilidad de crear nuevos usuarios, el usuario registrado para administración del sistema es “admin”.
- Para crear nuevos usuarios clic en el botón “Nuevo Usuario” y aparecerá la siguiente pantalla en el cual se ingresan los parámetros como: Nombre, contraseña, grupo e identificador de usuario.

4.7. Pruebas y Resultados

4.7.1. Sistema Propietario

El control de temperatura del sistema para el proceso del prensado se realizó en perfectas condiciones teniendo un control de temperatura preciso y garantizado con la temperatura deseada de 50 grados centígrados, teniendo a nuestra salida una variación al ambiente en que se maneje el sistema teniendo una temperatura actual de 56 grados centígrados, estos resultados se los podrá ver en la interfaz de LabVIEW en donde presencia una relación entre las temperaturas deseadas y requeridas.

El control del sistema se lo realiza en una interfaz hecha en LabVIEW cada pulsador tiene su función específica contando con un pulsador de INICIO y un ENCENDIDO FISICO en donde podremos ver por un led encendido, y así mismo. Un apagado que se podrá realizar mediante la interfaz en LabVIEW APAGADO, y un led que se encenderá al apagar desde la botonera física APAGADO FISICO, los leds de los sensores se encenderán, en el momento óptimo donde los sensores detectan algo o de acuerdo a su función, SENSOR 1, SENSOR2, se encenderá un led en el momento en que el pistón de la prensadora esté actuando PISTON, y otro sensor en donde detecta la presencia de un elemento DETECCIÓN ELEMENTO. Se podrá observar los pulsos que se envían en cada variable en una interfaz de secuencias de estados.

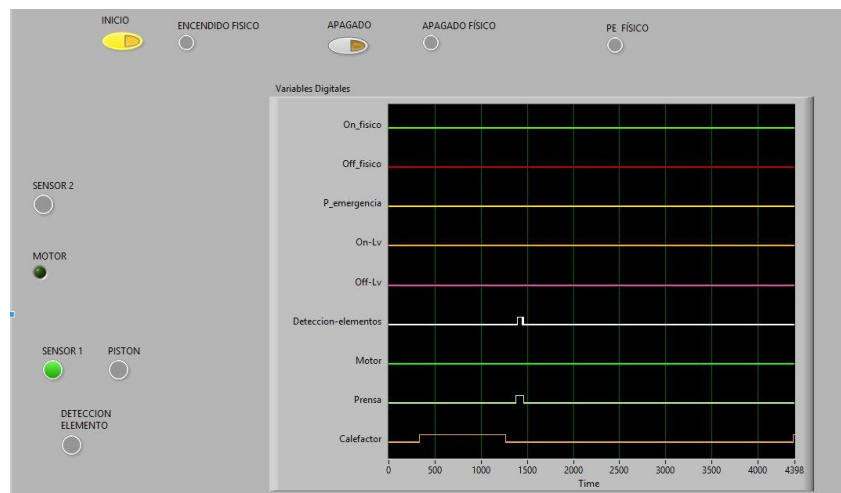


Figura 39-4: Esquema de programación LabVIEW

Realizado por: Pamela Espejo. 2017

La animación del sistema de como se está realizando el proceso se podrá observar en otra interfaz de LabVIEW, el tiempo de prensado en milisegundos, que es una variable que se puede cambiar según el requerimiento del usuario, TIEMPO DE PRESENSADO, se puede controlar la velocidad de la banda transportadora por un control en donde 2700 equivale a 10 voltios, la velocidad óptima para el sistema se instaló en 700 equivalente a 3 voltios, la animación de los diferentes gráficos en LabVIEW nos indica en forma clara del funcionamiento del sistema automatizado.

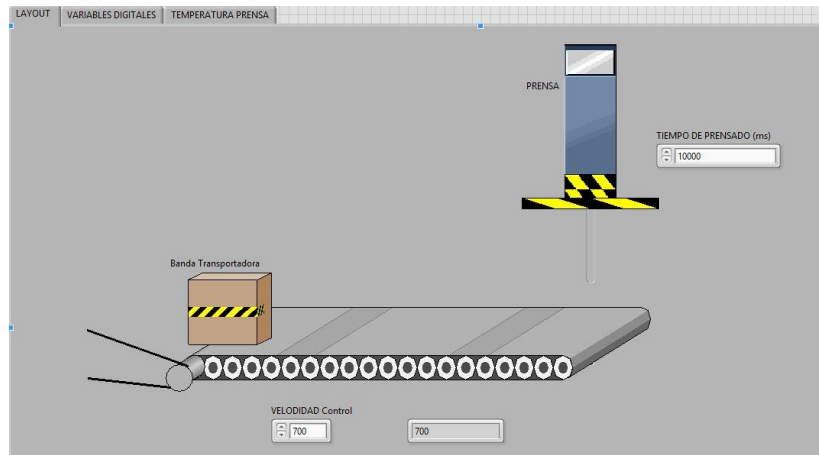


Figura 40-4: Interfaz LabVIEW

Realizado por: Pamela Espejo. 2017

Como se observa en la Figura 41-4 el control proporcional, integral y derivativo (PID) implementado para mantener la temperatura controlada a un valor de 30°C ha resultado un éxito, el tiempo de reacción y estabilización del sistema de control a partir de la puesta en marcha del sistema de producción es óptimo para el proceso a desarrollarse.

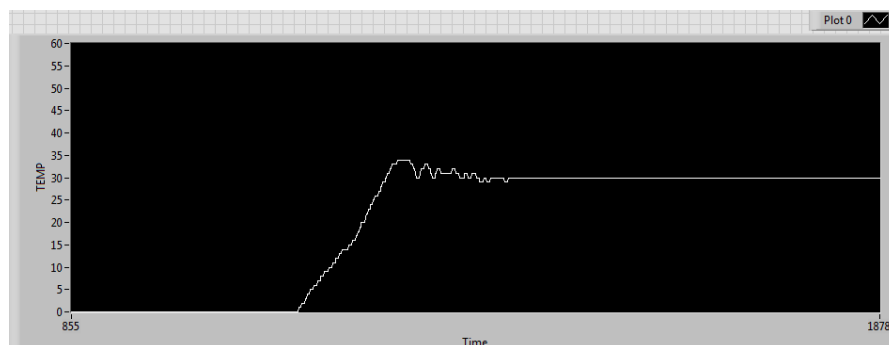


Figura 41-4: Control PID implementado en sistema propietario

Realizado por: Pamela Espejo. 2017

4.7.2. *Software Libre*

A su vez se ha implementado el mismo sistema de control y se ha sometido a las mismas pruebas bajo las mismas características al proceso industrial utilizando el servidor y el PLC M-DUINO generando los siguientes resultados:

Para el análisis de la reacción de las señales de control de los actuadores, de igual manera se presenta en la Fig. 42-4; las señales de control frente a la repuesta de los actuadores es instantánea debido al uso de interrupciones habilitadas en el controlador.

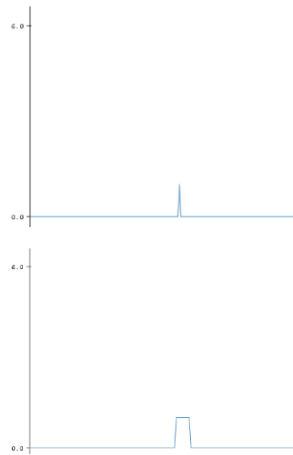


Figura 42-4: Análisis de las variables de control en SCADA
Realizado por: Pamela Espejo. 2017

La interfaz de visualización presentada en la Figura 4-43 cumple la función equivalente a la interfaz presentada en LabVIEW, siendo la diferencia primordial frente al sistema anteriormente presentado que no es necesario el software propietario en el terminal del usuario sino un dispositivo con navegador web; además de ser un sistema multipantalla, multiplataforma y con seguridades necesarias para que no todos los usuarios que se encuentren en la red puedan acceder al sistema.

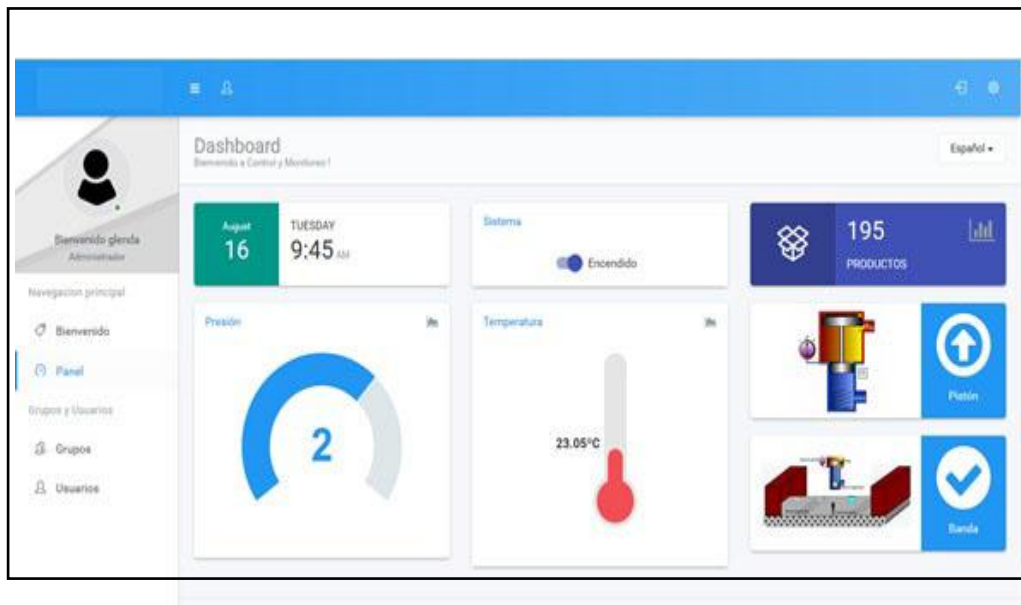


Figura 43-4: Página de Control
Realizado por: Pamela Espejo. 2017

La Figura 4-44 presenta la reacción del control PID implementado en el PLC M-DUINO, en la cual se puede observar una respuesta similar a la del sistema implementado en hardware propietario y cumple su tarea con la misma fiabilidad; demostrando que puede ser un sustituto efectivo para etapas de control en la industria.

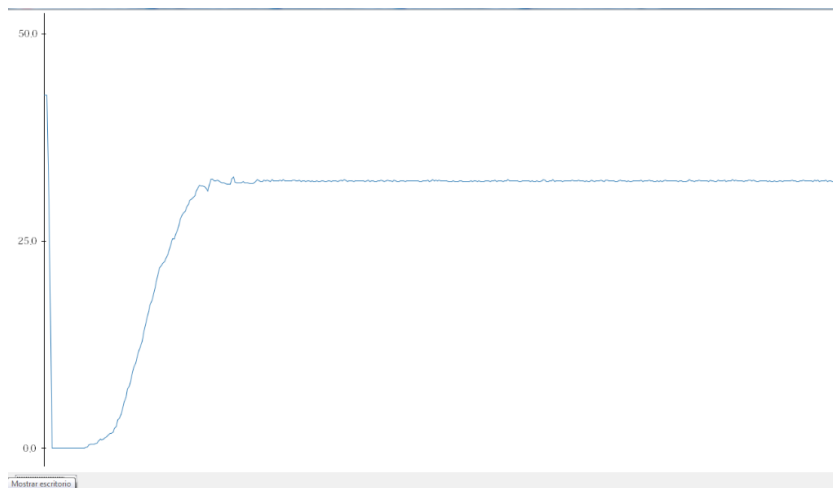


Figura 44-4: Control PID implementado en M-DUINO

Realizado por: Pamela Espejo. 2017

4.8. Tabla de comparación de comunicaciones industriales entre el PLC M-Duino y el Siemens S7-1200 (Ethernet, I2C, Serial, Rs485)

Tabla 19-4: Tabla de comparación de comunicaciones industriales entre el PLC M-Duino y el Siemens S7-1200.

TECNOLOGÍA	PLC M-Duino	Siemens S7-1200
ETHERNET	1 PUERTO	1 PUERTO**
USB	1 PUERTO	NO
RS-232	4 PUERTOS*	1 PUERTO**
RS-485	4 PUERTOS*	1 PUERTO**
I2C	1 PUERTO	NO
MODBUS	1 PUERTO	1 PUERTO

Realizado por: Pamela Espejo. 2017

Tabla 20-4: Características de comunicaciones industriales para PLC M-Duino y el Siemens S7-1200.

PARÁMETRO	TECNOLOGÍAS
ETHERNET	
Velocidad	10/100 Mbps
Longitud de Cable	Apantallado 1000 m máximo
Ancho de banda	100 MHz
Topologías	Bus, estrella, anillo y árbol
Conectores	RJ-45
Cables	UTP y FTP

<i>RS-485</i>	
Velocidad	10 Mbps
Configuración de línea	Unipolar
Modo de operación	Simplex o Full-Dúplex
Valores lógicos típicos	+/- 5 V a +/-15 V
Impedancia de entrada mínima del receptor	3 a 7 Kohms
Tensión de salida	2 V
Longitud de cable	Apantallado 1000 m máximo
Rango de tensión	-7 a +12 V
Sensibilidad en recepción	+/- 0,2 V mínimo, (200 mV M-duino).
<i>RS-232</i>	
Velocidad	20 Kbps
Configuración de línea	Diferencial
Modo de operación	Simplex o Half-Dúplex
Valores lógicos típicos	+/- 1,5 V a +/-6 V
Impedancia de entrada mínima del receptor	12 Kohms
Tensión de salida del transmisor	+/- 5 V
Tensión de entrada del receptor	+/- 30 V DC máx
Longitud de cable	Apantallado 10 m máximo
Sensibilidad en recepción	+/- 0,2 V mínimo, (200 mV M-duino).
<i>I2C M-Duino</i>	
Líneas	SDA y SCL
Número de bits del Bus	8 bits
Velocidad Transmisión (modo estándar)	100 Kbps
Velocidad Transmisión (modo rápido)	400 Kbps
Tensión mínima entrada	1,5 V
Frecuencia de trabajo	100-400 KHz

Realizado por: Pamela Espejo. 2017

4.9. Tabla de compatibilidad del PLC M-Duino y sensores

Tabla 21-4: Compatibilidad de sensores con el PLC M-DUINO

	DEFINICION	FUNCION	CARACTERISTICAS	I/O DIGITALES
Sensor de Humedad y Temperatura	Un elemento capacitivo mide la humedad relativa y la temperatura se mide por un coeficiente negativo de temperatura (NTC).	Conecte el pin + a 5v y - pin a GND, pin S es la salida de los datos de temperatura y humedad	Voltaje de operación 5 V CD Humedad relativa y medición de la temperatura Compatible con PLC M-DUINO	Salida digital pre-calibrada
DTH11 Sensor de Inclinación	Éste sensor de inclinación es un interruptor equivalente a un botón	Cuando esta nivelado el interruptor queda abierto, y cuando se inclina, se cierra el interruptor emitirá la señal a S	Voltaje de operación 5 V CD Interface de 3 pines Compatible con PLC M-DUINO	Se utiliza como una entrada digital
Sensor Magnético	Esta es una interfaz electrónica que se puede utilizar como interruptor magnético.	El interruptor magnético es un instrumento que puedes utilizar para un circuito de encendido y apagado basado en	Voltaje de operación 5 V CD Interface de 3 pines Compatible con PLC M-DUINO	Utiliza como una salida digital
Sensor de Gas MQ-2	Este pequeño modulo cuenta con un sensor de gas MQ-2.	Se puede utilizar para la detección de fugas de gas (hogar y la industria). Es capaz de detectar LPG, isobutano,	Voltaje de operación 5 V CD Compatible con PLC M-DUINO Sencilla interface de 3 pines	Utiliza una salida analógica
Sensor Infrarrojo PIR	Este módulo es simple y sencillo cuenta con un sensor de movimiento (PIR).	Se puede utilizar para un sistema de alarma contra robo, automatizar un proceso industrial, etc.	Voltaje de operación 5 V CD Interface de 3 pins Compatible con ARDUINO Angulo de detección: 120° Distancia efectiva: 5m	Utiliza una salida analógica
Sensor de Luz	El módulo de sensor de luz utiliza un fotoresistor para detectar la intensidad de la luz.	La resistencia del sensor disminuye cuando la intensidad de la luz aumenta. Cuenta con un LM358 chip que se utiliza como un comparador	Voltaje de operación 5 V CD Interface de 3 pins Compatible con ARDUINO Salida digital o salida analógica Bajo consumo de energía	Puede utilizarse una salida analógica o una salida digital con la ventaja que tiene incorporado un LM35

Realizado por: Pamela Espejo. 2017

4.10. Comparación de Entradas Digitales

Tabla 22-4: Comparación de Entradas Digitales

Entradas Digitales	Sensores Capacitivos (M12 548651)	Sensores Inductivos	Sensor Fotoeléctrico (B562019)	Compatibilidad con PLC (w5100.h)
Consumo de Energía (mA)	40 mA	40 mA	100 mA	5mA
NPN	normalmente abierto	normalmente abierto	normalmente abierto	normalmente abierto
PNP	normalmente cerrado	normalmente abierto	normalmente abierto	normalmente abierto
Alimentación	12-24V	12-24V	24V	12-24V
Voltaje de Salida	5-12-24V	5-12-24V	5-12-24V	5-12-24V
Tipo de conexión	3 hilos	2 hilos	2 hilos	1 hilos
2,3,4 hilos				
Normativa CE	UE-EMC	UE-EMC	UE-EMC	UE-EMC
Robs				
Compatibilidad Electromagnética				

Realizado por: Pamela Espejo. 2017

4.11. Comparación de Salidas Digitales

Tabla 23-4: Comparación de Salidas Digitales

Salidas Digitales	Relay	Luces Piloto	Electroválvula	Compatibilidad con PLC
Consumo de energía	15 [mA _{dc}]	10-20 [mA]	1- 1.5 [W]	4 [mA]
Alimentación	4 -15 [V _{dc}]	12- 24 Ac/Dc	24 [V _{dc}]	24 [V _{dc}]
Normativa CE	EU-EMC	EU-EMC	EU-EMC	EU-EMC
Robs				
Compatibilidad Electromagnética				

Realizado por: Pamela Espejo. 2017

4.12. Comparación de Entradas Analógicas

Tabla 24-4: Comparación de entradas analógicas

Entradas analógicas	Sensor de Presión 548622	Sensor de temperatura J	Resolución de la entrada analógica del PLC	Compatibilidad con PLC siemens s7-1200
Consumo de energía (mA)	35 mA	2 - 90 uA	1.2 A	4 mA
Alimentación	15 - 30 VCC	20 - 30 VCC	12 - 24 VCC	24 VCC
Voltaje o Corriente de Salida	100 mA	4 - 20 mA	0 - 10 VCC	24 VCC
Tipo de conexión 2, 3, 4 hilos	4	2	1	1
Normativa CE Rohns Compatibilidad Electromagnética	EU-EMC	EU-EMC	EU-EMC	EU-EMC

Realizado por: Pamela Espejo. 2017

CONCLUSIONES

Los sensores Festo son los más utilizados en el control industrial con la ventaja de su confiabilidad y funcionamiento y versatilidad y a su manejo de voltajes fijos y variables en sus salidas para el control de procesos de producción y en otras aplicaciones en otros ámbitos.

PLC M-DUINO ofrece la ventaja de trabajar en software y hardware libre sin necesidad de pagar costos adicionales después de su compra en comparación de los PLC tradicionales.

El servicio FTP implementado permite realizar la transferencia de archivos entre el cliente y el servidor de una manera eficaz ya que se creó un usuario específico para la administración de la conexión con el servidor; trabaja por defecto en los puertos 20 y 21. El puerto 20 es el utilizado para el flujo de datos entre el cliente y el servidor y el puerto 21 para el flujo de control.

La creación del entorno visual y la adquisición de los datos de las variables: estado ON/OFF, temperatura, presión, presencia y velocidad se facilita el almacenamiento de información en entorno dinámico; los valores de las señales enviadas se guardan en tiempo real, proporcionando un control estable de las mismas.

La aplicación Web creada muestra el Estado ON/OFF de la prensa, Temperatura de la prensa (°C), Sensor de presión (Bar), Conteo de productos, Estado del Pistón, arriba o abajo, Fecha y hora. Para el acceso a esta información se debe estar registrado en el sistema, sea como administrador u operador.

La interfaz gráfica de la aplicación Web es amigable con el usuario que desea acceder al sistema y cuenta con una presentación idónea que permite una iteración cómoda entre el usuario y el proceso que se está controlando y monitoreando.

La base de datos creada luego de la adquisición es compatible y de fácil manejo desde la aplicación Web ya que usa phpMyAdmin para la gestión de tablas de datos.

Asterisk cuenta con una gama de módulos, con distintas funcionalidades que permiten realizar conexiones hacia distintas bases de datos con una llamada por medio de un IVR.

RECOMENDACIONES

Verificar el funcionamiento de cada uno de las partes del proceso para detectar las posibles falencias y correcciones a tiempo.

Revisar los respectivos manuales de los dispositivos para evitar malas conexiones y daños de los equipos.

Para poder transferir archivos mediante FileZilla se requiere montar previamente la dirección del archivo denominado www en donde se va a guardar los datos desde el cliente, ya que caso contrario se produce un error en la transferencia.

Para iniciar con la creación de la base de datos se debe previamente instalar el APP server con todos sus componentes es decir requiere el uso de servidores Apache, php, MySQL y phpMyAdmin.

Se recomienda inicialmente extraer los datos directamente del Arduino y guardarlos en una base de datos local de la aplicación para realizar pruebas. Luego se procede a unir la página Web con la base de datos creada para la adquisición y almacenamiento de datos.

Utilizar las versiones probadas de asterisk porque las nuevas versiones vienen con nuevas políticas de seguridad que impedirían las comunicaciones.

Las alertas monitorizadas de los sensores deben ser controladas por un usuario el cual sea capaz de analizar dichas alertas y comprender cada una de las señales para tomar medias dependiendo el nivel de la alerta.

BIBLIOGRAFÍA

- ABAFFY, C.** (2007). Desarrollo de Scada en una Plataforma de Software Libre. *CITEG*.
- ABRIL, C.** (2012). Los procesos de producción y la productividad en la industria de calzado ecuatoriana: Caso empresa Mabelyz. *I Congreso Iberoamericano de Investigación sobre MIPyME*.
- ALAN, L., TORRES, J., & VELA, D.** (2015). Hardware libre Recomendaciones para el fomento de la innovación ciudadana. Infraestructuras técnicas abiertas. pp. 1-2.
- B. WENDY.** (1989). *Aplicaciones de la inteligencia aritificial en la actividad empresarial la ciencia y en la industria*. Madrid: DIAS DE SANTO S.A.
- BARRIETOS, A., & GAMBAO, E.** (2012). *Automatizacion. En Sistemas de Produccion Automatizados*. España: DEXTRA. pp. 200
- CAMARGO, C.** (2012). Metodología para la transferencia tecnológica en la industria electrónica basada en software libre y hardware copyleft. *Congreso Argentino de Sistemas Embebidos CASE 2012*.
- CHOURIO, L., & MEDRANO, J.** (2015). Hardware libre y abierto, modelos de negocios para América Latina y el Caribe. *ALTEC*. pp. 1-5.
- FESTO AG.** (2009). *Sensores de proximidad SMT/SME-8, para ranura en T*. Obtenido de <https://w5.siemens.com/spain/web/es/industry/automatizacion/simatic/Documents/S71200-MANUAL%20DEL%20SISTEMA.PDF>
- FESTOAG.**(2015).Obtenido de https://www.festo.com/cat/esmx_mx/data/doc_es/PDF/ES/DSNU-ISO_ES.PDF
- FESTO DIDACTIC.** (2009). *Festo Didactic*. Obtenido de http://www.festo-didactic.com/ov3/media/customers/1100/548651_es_naehierungsschalter_kapazitiv_m12.pdf

- FLOK SOCIETY.** (2015). Hardware libre Recomendaciones para el fomento de la innovación ciudadana. *Línea 4: Infraestructuras técnicas abiertas.*
- INDUSTRIAL SHIELDS.** (2016). *Downloads.* Obtenido de Industrial Shields Web site: <http://www.industrialshields.com/es/downloads/>
- INFOPLC.** (2009). *Automatas Programables.* Obtenido de http://www.infopl.net/files/descargas/rockwell/infoPLC_net_Ejemplo_Programacion_Control_Logix.pdf
- INFOPLC.** (2013). *10 aspectos para elegir un autómata PLC.* Obtenido de InfoPLC: <http://www.infopl.net/blogs-automatizacion/item/101501-10-aspectos-para-elegir-automata-plc>
- KARL-HEINZ, J., & MICHAEL, T.** (1995). *IEC 61131-3: Programming Industrial Automation Systems.* Berlin: Springer Verlag.
- LEE, J., & LARRY, P.** (2008). *Administración de operaciones: estrategia y análisis.* Mexico: PEARSON.
- LÓPEZ, J.** (2010). *Compatibilidad electromagnética y seguridad funcional en sistemas electrónicos.* España: MARCOMBO S.A.
- MANDADO, E., ACEVEDO, J., FERNANDEZ, C., & ARMESTO, J.** (2009). *Autómatas programables y sistemas de automatización.* España: MARCOMBO, S.A.
- PALLAS, R.** (2003). *Sensores y acondicionadores de señal.* España: Marcombo.
- PLAN NACIONAL DEL BUEN VIVIR.** (2013). *Plan Nacional del Buen Vivir.* Quito: Senplades.
- PONSATI, G.** (2002). *Gestión de la Calidad.* España: UPC. España : UPC.
- QUINTERO, C., OÑATE, J., & HARIAS, H.** (2011). *Instrumentación electrónica aplicada. Prácticas de la laboratorio.* Colombia: Universidad del Norte.

SCHOLTEN, B. (2007). *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. ISA, 2007.

SIEMENS. (2013). S7 Controlador programable S7-1200 .

SMC. (2016). *SMC International Training*. Obtenido de <http://www.smctraining.com/webpage/indexpage/311/>

TALVE, W. (2015). *Automatización de sistema de recirculación de agua caliente utilizando hardware libre en el hospital IESS Latacunga*. Ingeniero. Ambato: Universidad Técnica de Ambato.

ANEXOS

Electromagnetic Compatibility (EMC)

General

The aim of electromagnetic compatibility considerations is to avoid or minimize the influence of electromagnetic phenomena on a device, equipment or system and on living or inert matter. To achieve this, methods of measurement and test, as well as limits and levels of electromagnetic emission and of electromagnetic immunity are defined in a large number of EMC publications.

Note: The European EMC Directive does not cover the interaction between technical apparatus and the biological environment.

Types of EMC Publications

EMC publications and standards, developed by the IEC and other standardization bodies, can be divided into four general categories:

Generic EMC Standards, Basic EMC Publications, Product family EMC Standards and Product EMC Standards. These terms are defined in the corresponding chapters below.

Product and Product Family EMC Standards take precedence over Generic EMC Standards and Product EMC Standards take precedence over Product Family EMC Standards.

The distinction between these four types of EMC publications is not yet consequently applied within the IEC, CISPR and Cenelec (EN).

Generic EMC Standards

Generic EMC Standards specify requirements for products and systems operating in residential or industrial environments. They apply to products for which no dedicated Product Family EMC Standards or Product EMC Standards exist.

Generic EMC Standards can be considered to be general Product Standards in that they specify a limited number of essential emission and immunity tests, maximum emission levels as well as minimum immunity test levels with assigned performance criteria but refer to the Basic EMC Standards for detailed measurement and test methods.

Table 1: Generic EMC Emission Standards (only parts relevant to power supplies)

Port	IEC 61000-8-3/EN 50081-1: Electromagnetic compatibility – Generic standards – Emission standard for residential, commercial and light-industrial environments		IEC 61000-8-4/EN 50081-2: Electromagnetic compatibility – Generic standards – Emission standard for industrial environments	
	Referenced basic standard	Frequency range and required limit	Referenced basic standard	Frequency range and required limit
Case (radiated)	CISPR 22/ EN 55022	30...1000 MHz, class B	CISPR 11/ EN 55011	30...1000 MHz, Group 1, class A
AC mains (conducted)		0.15...30 MHz, class B		0.15...30 MHz, Group 1, class A
	IEC 61000-3-2/ EN 60555-2	0...2 kHz		
	IEC 61000-3-3/ EN 60555-3	0...2 kHz		

Table 2: Generic EMC Immunity Standards (only parts relevant to power supplies)

Referenced Basic standard	Port	Requirements of Generic Standard EN 50082-1 (IEC 61000-6-1): Electromagnetic compatibility – Generic standards – Immunity standard for residential, commercial and light-industrial environments	Requirements of Generic Standard IEC 61000-6-2/EN 50082-2: Electromagnetic compatibility – Generic standards – Immunity standard for industrial environments	Performance criterion
IEC/EN 61000-4-2: Electrostatic discharge	Enclosure (case)	±4 kV contact discharge ±8 kV air discharge	±4 kV contact discharge ±8 kV air discharge	B
IEC/EN 61000-4-3/ ENV 50204: Electromagnetic field	Enclosure (case)	3 V/m, 80...1000 MHz, AM 80%, 1 kHz, 900 MHz, PM 200 Hz ¹	10 V/m, 80...1000 MHz, AM 80%, 1 kHz, 900 MHz, PM 200 Hz ¹	A
IEC/EN 61000-4-4: Fast transients (burst)	Signal and control	±0.5 kV, 5 kHz, capacitive clamp (only if cables are longer than 3 m)	±1 kV, 5 kHz, capacitive clamp (only if cables are longer than 3 m), ±2 kV, 5 kHz, capacitive clamp (only for process control and according to EN)	B
	DC Input and output	±0.5 kV, 5 kHz, direct Injection (only if cables are longer than 10 m)	±2 kV, 5 kHz, direct Injection	
	AC Input and output	±1 kV, 5 kHz		
IEC/EN 61000-4-5: Surges	DC Input (and output)	±0.5 kV line to earth, ±0.5 kV line to line	±0.5 kV line to earth, ±0.5 kV line to line	B
	AC Input	±2 kV line to earth, ±1 kV line to line	±4 kV line to earth, ±2 kV line to line	
IEC/EN 61000-4-6: Common mode conducted disturbances	Signal and control lines	3 V, 0.15...80 MHz, AM 80%, 1 kHz (only if cables are longer than 3 m)	10 V, 0.15...80 MHz, AM 80%, 1 kHz (according to IEC only if cables are longer than 3 m)	A
	DC Input and output			
	AC Input and output	3 V, 0.15...80 MHz, AM 80%, 1 kHz	10 V, 0.15...80 MHz, AM 80%, 1 kHz	
IEC/EN 61000-4-11: Voltage dips	AC Input	30%, 10 ms	30%, 0.5 periods	B
		60%, 100 ms	60%, 5 and 50 periods	C
IEC/EN 61000-4-11: Voltage Interruptions		>95%, 5000 ms	>95%, 250 periods	

¹ 900 MHz is referenced in EN standards only.

Basic EMC Publications

Basic EMC Publications specify the general and fundamental conditions and rules as well as the measurement and testing techniques for the verification of EMC and serve as reference documents for the product committees of the standardization bodies. *Basic EMC Publications* relate to general information, to the disturbing phenomena and to measurement or testing techniques.

Basic EMC Publications are general standards or technical reports and are not dedicated to specific product families or products. They should not include prescribed limits and related performances. These are covered by the *Generic, Product Family or Product Standards*.

Basic Emission Standards

Electromagnetic emission is the phenomenon by which electromagnetic energy emanates from a source.

Radio Frequency Emission

Radio frequency emission, also called radio frequency interference (RFI) is an electromagnetic disturbance which is generated by electrical apparatus and can be received by other equipment.

In power supplies, the RFI is generated by the switching devices. The steeper the voltage steps and the higher the switching frequency, the greater the high frequency content of generated disturbances. These disturbances can couple to other components in the power supply, producing noise sources radiating over a broad frequency spectrum.

International basic emission standards are prepared by the CISPR (Comité international spécial des perturbations radioélectriques – International special committee on radio interference) and adopted by national or multinational authorities.

It is usual to measure disturbances at low frequencies (150 kHz up to approx. 30 MHz) as voltages between the supply lines and earth (conducted disturbances) and at higher frequencies (above approx. 30 MHz) as field strength or power (radiated disturbances).

The graphs below provide an overview of the most common international standards for both conducted and radiated emissions.

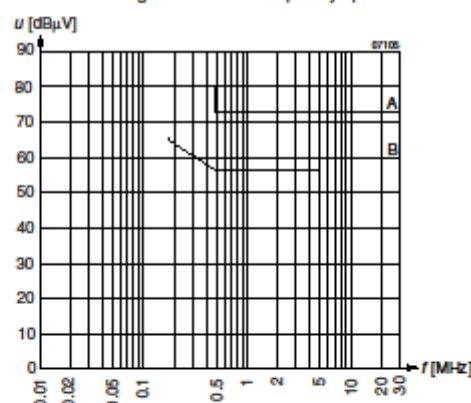


Fig. 1
Disturbance voltage limits (quasi-peak) according to CISPR 11/EN 55011 and CISPR 22/EN 55022

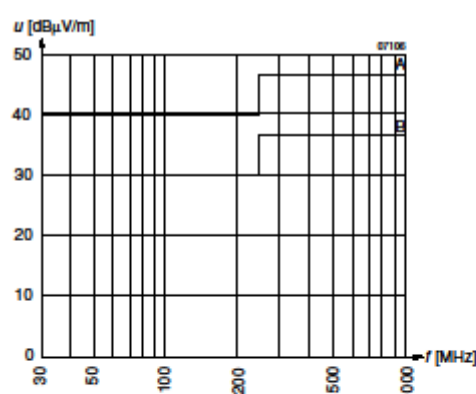


Fig. 2
Disturbance field strength limits (quasi-peak) according to CISPR 11/EN 55011 and CISPR 22/EN 55022, normalized to a distance of 10 m

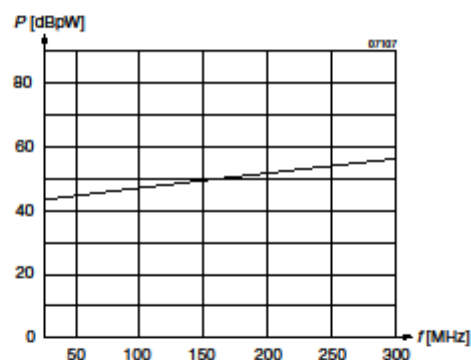


Fig. 3
Disturbance power limits (quasi-peak) according to CISPR 14/EN 55014

CISPR 11/EN 55011: Industrial, scientific and medical (ISM) radio-frequency equipment – Electromagnetic disturbance characteristics – Limits and methods of measurement

CISPR 14/EN 55014 (= VDE 0875, part 14): Limits and methods of measurement of radio disturbance characteristics of electric motor-operated and thermal appliances for household and similar purposes, electric tools and similar electric apparatus

This standard is not directly applicable to power supplies, but describes methods of measuring radiated electromagnetic power using an absorbing (MDS) clamp. The clamp allows the simple measurement over a cable of a defined length.

CISPR 22/EN 55022 (= VDE 0878, part 22; corresponds to FCC, part 15): Limits and methods of measurement of radio disturbance characteristics of information technology (IT) equipment

MIL-STD-461C: Electromagnetic emission and susceptibility requirements for the control of electromagnetic interference

Harmonic Current Emission

Electrical and electronic equipment inject harmonic currents into their AC supply system. In order to ensure electromagnetic compatibility, maximum harmonic disturbance levels have been defined in the corresponding standards.

cos φ

For undistorted, sinusoidal waveforms cos φ is the displacement factor of the load current compared to the mains voltage (fig. cos φ).

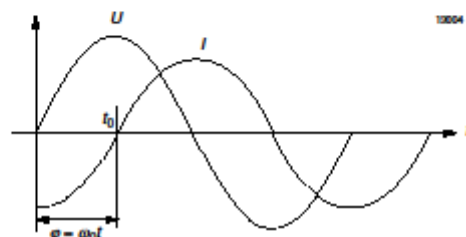


Fig. 4
cos φ

For this special case φ is also equal to the phase difference between true power and the apparent power.

Apparent power: $S = U_{rms} \cdot I_{rms}$
 True power: $P = U_{rms} \cdot I_{rms} \cdot \cos \varphi = S \cdot \cos \varphi$
 Reactive power: $Q = U_{rms} \cdot I_{rms} \cdot \sin \varphi$

Apparent power is the vector sum of the true and reactive power (fig. AC-power components).

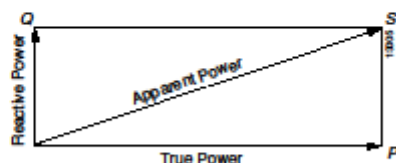


Fig. 5
AC-power components

Power Factor

For single phase systems the power factor is defined as the ratio of the true power delivered to the load to the apparent power. Power factor is the generalized case of cos φ, since the waveforms do not have to be sinusoidal. For the special case where the waveforms are sinusoidal, the power factor is equal to cos φ.

$$PF = \frac{\text{True power}}{\text{Apparent power}} = \frac{P}{S}$$

In practice the mains voltage can be assumed to be sinusoidal. So the power factor only depends on the distortion of the load current: The more the waveform is distorted, the lower the power factor. A perfect power factor would be equal to 1, whereas the worst case power factor is 0.

$$PF = \frac{I_0 \cdot \cos \varphi_0}{I_{rms}}$$

- I_0 : Fundamental of load current
- $\cos \varphi_0$: Displacement factor of fundamental of load current
- I_{rms} : rms-value of load current

Harmonic Distortion

For nonlinear loads (for example rectifier) the load current will not be sinusoidal anymore (fig. Harmonic distortion).

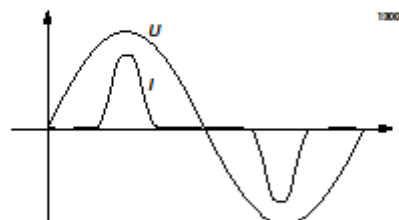


Fig. 6
Harmonic distortion

Since true power only takes place at the fundamental frequency (frequency of mains voltage), the analysis of the current into fundamental and harmonic currents has to be calculated (Fourier analysis):

$$I(t) = I_0 \cdot \sin(\omega_0 t + \varphi_0) + I_1 \cdot \sin(\omega_1 t + \varphi_1) + \dots + I_n \cdot \sin(\omega_n t + \varphi_n)$$

- I_0 : amplitude of fundamental
- I_i : amplitude of i^{th} harmonic
- $\omega_0 = 2 \pi \cdot f_0$ $f_0 = 50/60 \text{ Hz}$ frequency of fundamental
- $\omega_i = i \cdot \omega_0$ $i = 1 \dots n$ frequency of i^{th} harmonic
- φ_0 : phase shift of fundamental
- φ_i : phase shift of i^{th} harmonic

Based on this decomposition the Distortion Factor can be calculated:

$$K = \frac{\text{Contents of harmonics}}{\text{rms-value}}$$

$$K = \sqrt{\frac{\sum_{i=1}^n I_i^2}{\sum_{i=0}^n I_i^2}}$$

IEC/EN 61000-3-2: Limits for harmonic current emissions (equipment input current ≤ 16 A per phase)

This standard deals with the limitation of harmonic currents injected into the public supply systems with nominal voltages of 220 V (line to neutral) or more. It is applicable to

equipment having an input current up to and including 16 A per phase. The standard defines 4 classes of equipment with different harmonic current limits:

Class	Description	Limits		Unit
		Harmonic order	Maximum permissible harmonic current	
A	Balanced three-phase equipment and all other equipment, except those included in one of the following classes.	2	1.06	A
		3	2.30	
		4	0.43	
		5	1.14	
		6	0.30	
		7...40	0.77...0.046	
B	Portable tools	Limits of class A, multiplied by 1.5		
C	Lighting equipment	2	2	% of input current at the fundamental frequency
		5	10	
		7	7	
		9	5	
		11...39 (odd only)	3	
		3	$30 \cdot \lambda$	%; λ = circuit power factor
D	Equipment having an input current with a "special wave shape" (as defined in the standard) and an active input power above 75 W up to 600 W.	3	3.4	mAW
		5	1.9	
		7	1.0	
		9	0.5	
		11...39 (odd only)	0.35...0.1	

Power-One power supplies which don't supply portable tools or lightning equipment are deemed to be class A if they provide an active power factor correction circuit or

have an active input power above 600 W, otherwise they are class D. The lower active power limit for class D equipment will be reduced from 75 W to 50 W.

Basic Immunity (Susceptibility) Standards

Electromagnetic immunity is the ability of a device, equipment or system to perform without degradation in the presence of electromagnetic disturbances. Electromagnetic susceptibility is the inability to perform without degradation, i.e. susceptibility is the lack of immunity.

The aim of basic immunity standards is to provide test procedures and test levels to verify electromagnetic immunity. Both conducted and radiated phenomena are considered. The results of tests are classified in terms of the loss of function or degradation of performance of the equipment under test as follows:

Performance Criteria

A Normal performance within limits specified by the manufacturer.

Power-One specifies normal performance as the output voltage of a power supply not deviating by more than 3% from the value prior to the test.

B Temporary loss of function or degradation of performance which ceases after the disturbance ceases.

Power-One specifies degradation of performance as a voltage deviation of more than 3% from the value before the test.

C Temporary loss of function or degradation of performance, the correction of which requires operator intervention or system reset.

In case of Power-One power supplies, operator intervention can mean turning the unit off and on by removal and re-application of the input supply or by means of the inhibit input.

D Loss of function or degradation of performance which is not recoverable, due to damage of hardware or software or loss of data.

IEC/EN 61000-4-2: Electrostatic discharge (ESD) immunity test

Static electricity discharges may be present in environments with low relative humidity, using low-conductivity (artificial fibre) carpets, vinyl garments, etc.

This standard specifies requirements and tests for the immunity of electrical or electronic equipment which are subject to electrostatic discharge. The tests are based on a human body model with a capacitance of 150 pF and a discharge resistance of 330 Ω .

The discharge shall either be applied in direct contact or over a short air distance to parts of the device under test which can be touched by persons during normal use.

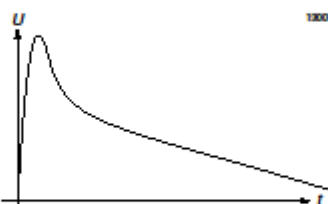


Fig. 7
Electrostatic discharge pulse

Contact discharge			Air discharge		
Test level	Test voltage kV	Peak current A	Test level	Test voltage kV	Peak current A
1	2	7.5	1	2	7.5
2	4	15	2	4	15
3	6	22.5	3	8	30
4	8	30	4	15	56
x	special		x	special	

IEC/EN 61000-4-3: Radiated, radio-frequency, electromagnetic field immunity test

Most electronic equipment is affected by electromagnetic radiation which is generated by hand-held radio transceivers, fixed-station radio and television transmitters and various industrial electromagnetic sources.

The radiated immunity test is applicable to all products, where radio frequency fields are present.

The test is performed by applying an electromagnetic far field of defined strength while varying the frequency in the range 80...1000 MHz. The table gives the field strength of the unmodulated signal. For testing, this signal is 80% amplitude modulated with a 1 kHz sinewave.

Test level	Test field strength V/m
1	1
2	3
3	10
x	special

ENV 50204: Radiated electromagnetic field from digital radio telephones – Immunity test

The radiated immunity test is applicable to all products, where radio frequency fields from digital radio telephones are present.

The test is performed by applying an electromagnetic far field of defined strength with a fixed frequency. Two different spot frequencies are defined: 900 \pm 5 MHz and 1890 \pm 10 MHz. The table gives the field strength of the unmodulated signal. For testing, this signal is pulse modulated with a duty cycle of 50% at a repetition frequency of 200 Hz.

This test is subject to be implemented in IEC/EN 61000-4-3.

Test level	Test field strength V/m
1	1
2	3
3	10
4	30
x	special

IEC/EN 61000-4-4: Electrical fast transient/burst immunity test

The fast transients test is applicable to products which are connected to AC or DC power systems or have cables in close proximity to such sources. It is intended to demonstrate immunity to transient disturbances such as those originating from switching transients (interruption of inductive loads, relay contact bounce, etc.).

The waveform of a single transient is characterized by a rise time of 5 ns, a time to half-value of 50 ns and a maximum energy of 4 mJ at 2 kV into a 50 Ω load. The impedance of the transient source is 50 Ω. The repetition frequency of the transients is a function of the test level. Burst duration is 15 ms and burst period 300 ms. Coupling is performed via a coupling/decoupling network with coupling ca-

pacitors of 33 nF for power supply ports and via a capacitive coupling clamp with a capacitance of 50...200 pF for signal, data and control ports.

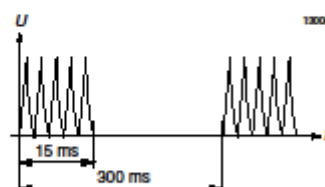


Fig. 8
Electrical fast transients/bursts

Test level	On AC and DC power supply ports		On signal, data and control ports	
	Open-circuit peak voltage kV	Repetition frequency kHz	Open-circuit peak voltage kV	Repetition frequency kHz
1	0.5	5	0.25	5
2	1	5	0.5	5
3	2	5	1	5
4	4	2.5	2	5
x	special	special	special	special

IEC/EN 61000-4-5: Surge immunity test

Surges are mainly generated by switching transients or by lightning strokes injecting high currents producing voltages or inducing high voltages/currents via electromagnetic fields. Switching transients can be generated by power system switching, load changes, short circuits or arcing faults to the earthing system of the installation.

The surge immunity test is applicable to products, which are connected to mains supplies or other networks leaving the building. It is intended to demonstrate immunity to surge voltages caused by switching and lightning effects. It is not intended to test the capability of the insulation to withstand high-voltage stress.

The standard specifies two different open-circuit surge waveforms: One with a rise time of 1.2 μs and a time to half-value of 50 μs, and the other with a rise time of 10 μs and a time to half-value of 700 μs. The source impedance for the 1.2/50 μs surge, is 2 Ω for line to line coupling and 12 Ω for line to earth coupling. The 10/700 μs surge impedance is 42 Ω. Coupling for both waveforms is performed via a coupling/decoupling network with coupling capacitors of 0.1, 0.5, 9 or 18 μF, or with arrestors, depending on the kind of lines to be tested.

Test level	Open circuit test voltage kV
1	0.5
2	1
3	2
4	4
x	special

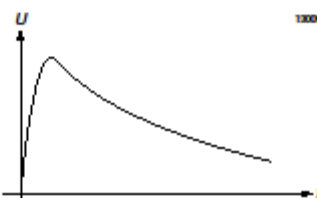


Fig. 9
Surge

IEC/EN 61000-4-6: Immunity to conducted disturbances, induced by radio-frequency fields

The source of conducted disturbances is basically an electromagnetic field, emanating from RF transmitters, that may act on the whole length of cables connected to installed equipment. The in-going and out-going leads behave as passive antenna networks if they are several wavelengths long.

The conducted immunity test is applicable to products operating in environments where radio frequency fields are present and which are connected to mains supplies or other networks (signal or control lines). This test is a supplement to IEC/EN 61000-4-3 which defines test methods for immunity to radiated electromagnetic energy.

The test is performed by applying a voltage of a defined value to the port to be tested while varying the frequency in the range 150 kHz...80 MHz. The signal is 80% amplitude modulated with a 1 kHz sinewave. The table gives the field strength of the unmodulated signal. The impedance of the test generator is 50 Ω. The waveform is coupled to each of the n lines of the port to be tested (common mode) via a coupling/decoupling device with an impedance of $n \cdot 100 \Omega$ per line or via an injection clamp.

Test level	Open circuit test voltage (rms)	
	V	dB (μV)
1	1	120
2	3	130
3	10	140
x	special	special

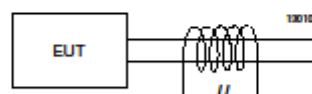


Fig. 10
Testing of immunity to conducted disturbances with injection clamp

IEC/EN 61000-4-8: Power frequency magnetic field immunity test

The power frequency magnetic field is mainly generated by power frequency currents in conductors.

In general this test is limited to products which are susceptible to magnetic fields (e.g. hall effect devices, CRTs and special products to be installed in high magnetic field environments). It has no relevance to power supplies.

The test is performed by applying a 50 or 60 Hz magnetic field of a defined strength to the equipment to be tested.

Continuous		Short duration (1...3 s)	
Test level	Magnetic field strength A/m (rms)	Test level	Magnetic field strength A/m (rms)
1	1	1	not defined
2	3	2	not defined
3	10	3	not defined
4	30	4	300
5	100	5	1000
x	special	x	special

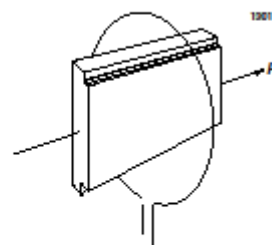


Fig. 11
Testing with power frequency magnetic field

IEC/EN 61000-4-9: Pulse magnetic field immunity test

Pulse magnetic fields are generated by lightning strokes on buildings and metal structures including aerial masts, earth conductors and earth networks and by initial fault transients in low voltage, medium voltage and high voltage systems.

This test is mainly applicable to products to be installed in electrical power plants (e.g. telecontrol centres in close proximity to switchgear). It is not relevant for distribution network equipment.

The test is performed by applying $6.4/16 \mu\text{s}$ magnetic field pulses of defined strength to the equipment to be tested.

Test level	Pulse magnetic field strength A/m (peak)
1	not defined
2	not defined
3	100
4	300
5	1000
x	special

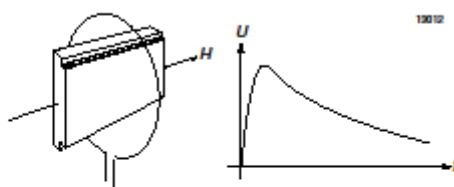


Fig. 12
Testing with pulse magnetic field

IEC/EN 61000-4-10: Damped oscillatory magnetic field immunity test

The damped oscillatory magnetic field is generated by the switching of high voltage bus-bars by isolators.

This test is mainly applicable to products to be installed in high voltage substations.

The test is performed by applying damped oscillatory magnetic field (sinusoid waves) of defined strength to the equipment to be tested.

Waveform specification	"Low frequency"	"High frequency"
Oscillation frequency	0.1 MHz	1 MHz
Decay rate	50% of the peak value after 3 to 6 cycles	
Repetition rate	≥ 40 Hz	≥ 400 Hz
Test duration	2 s	2 s

Test level	Damped oscillatory magnetic field strength A/m (peak)
1	not defined
2	not defined
3	10
4	30
5	100
x	special

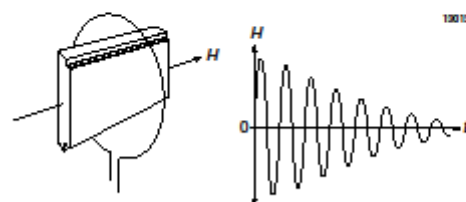


Fig. 13
Testing with damped oscillatory magnetic field

IEC/EN 61000-4-11: Voltage dips, short interruptions and voltage variations immunity tests

Voltage dips and short interruptions are caused by faults in the network or in installations or by a sudden large change of load. Voltage variations are caused by the continuously varying loads connected to the network.

These tests are applicable to equipment connected to AC mains with rated input currents not exceeding 16 A per phase.

The test for immunity to voltage dips and short interruptions is performed by an abrupt change of the supply voltage of the equipment under test at any phase angle on the mains voltage at the lower and at the upper voltage of the nominal voltage range.

For power supplies, immunity to these phenomenon can normally be derived from the specifications of input voltage range and hold-up time.

Preferred test levels and duration for voltage dips and short interruptions

Test level in % of rated voltage for the equipment under test	Corresponding voltage change in % of rated voltage for the equipment under test	Duration in number of periods. Any duration may apply to any test level.
0	100	0.5, 1, 5, 10, 25, 50, x (special)
40	60	
70	30	

The test for immunity to voltage variations is optional. It specifies a defined transition between the lower and upper voltage of the nominal voltage range and the changed voltage.

Timing of short-term supply voltage variations

Test level in % of rated voltage for the equipment under test	Time for decreasing the voltage in s	Time at reduced voltage in s	Time for increasing the voltage in s
40	2 ±20%	1 ±20%	2 ±20%
0	2 ±20%	1 ±20%	2 ±20%
40	x (special)	x (special)	x (special)
0	x (special)	x (special)	x (special)

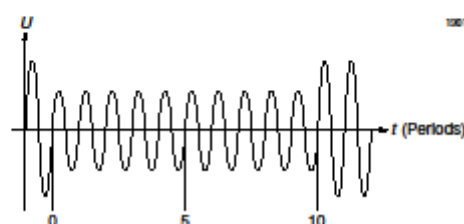


Fig. 14
Voltage dips

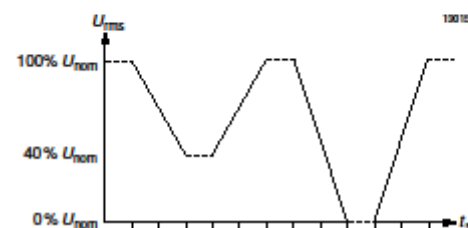


Fig. 15
Voltage variations

IEC/EN 61000-4-12: Oscillatory waves immunity test

This standard specifies tests to simulate two phenomena:

The ring wave (non-repetitive) appears at the terminals of equipment as a consequence of switching in power and control lines, as well as a consequence of lightning. The ring wave test is applicable to equipment connected to AC mains in certain countries (e.g. the mains network in the USA). This test has existed for a long time under the designation ANSI/IEEE C.62.41-1991.

The damped oscillatory wave (repetitive) appears at the terminals of equipment as a consequence of switching with restriking of the arc. The damped oscillatory wave test is applicable to equipment used in high voltage substations

(static relays). This test, but with a frequency of 1 MHz only, has existed for a long time under IEC 255-4, appendix E5: High frequency disturbance test, also known as ANSI/IEEE C.37.90a-1989, and has been transferred to IEC 60255-22-1: Electrical disturbance tests for measuring relays and protection equipment – 1 MHz burst disturbance tests. By inclusion in IEC 61000-4-12 the frequency range was extended.

Coupling is performed via a coupling/decoupling network with coupling capacitors of 0.5, 3 or 10 μF , depending on the output impedance of the test generator.

Wave specification	Ring wave	Damped oscillatory wave
Open circuit peak voltage	250 to 4000 V	250 to 2500 V
Oscillation frequency	100 kHz	100 kHz and 1 MHz
Decay rate	60% of the preceding peak value	50% of the peak value between the third and sixth periods
Repetition rate	1 to 60 per minute	≥ 40 Hz for 100 kHz and ≥ 400 Hz for 1 MHz
Output Impedance	12 Ω , 30 Ω and 200 Ω	200 Ω
Coupling mode	Common mode and differential mode	Common mode and differential mode
Test duration per port and coupling mode	≥ 5 positive and ≥ 5 negative transients	≥ 2 s

Test level	Ring wave voltage kV (peak)		Damped oscillatory wave voltage kV (peak)	
	Common mode	Differential mode	Common mode	Differential mode
1	0.5	0.25	0.5	0.25
2	1	0.5	1	0.5
3	2	1	2	1
4	4	2	-	-
x	special	special	special	special

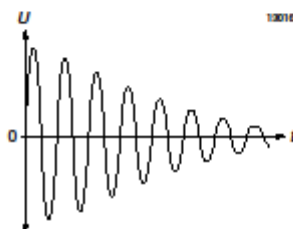


Fig. 16
Damped oscillatory wave

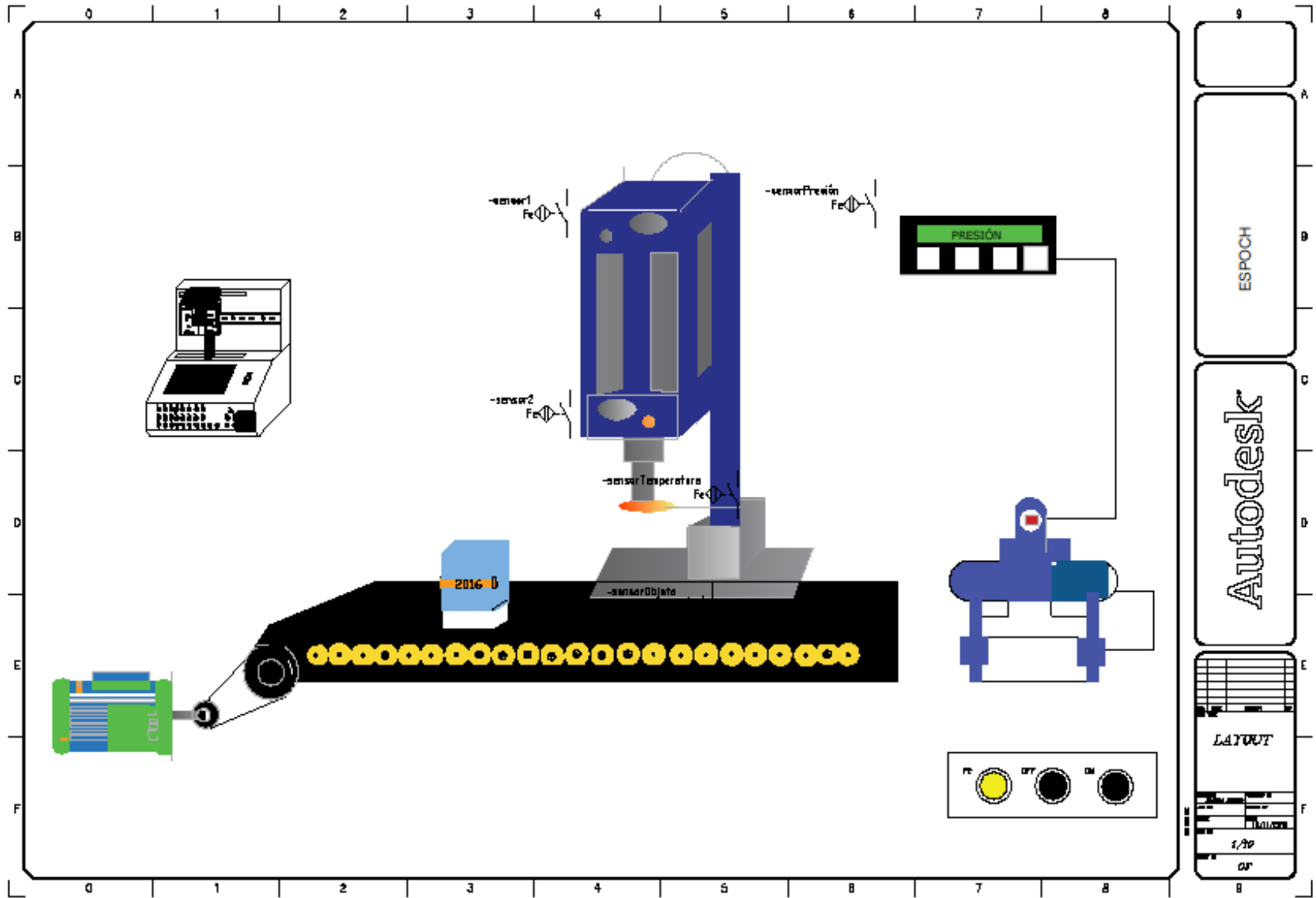
Product Family EMC Standards

A product family is a group of similar products for which the same standards can be applied. *Product Family EMC Standards* define specific electromagnetic (EM) requirements and test procedures dedicated to particular product families. They also give precise performance criteria taking into account the purpose of the equipment where possible. They apply the *Basic Standards* and coordinate with the *Generic Standards* to the extent practicable. Applicable *Product Family EMC Standards* take precedence over *Generic Standards*.

Product EMC Standards

Product EMC Standards relate to a particular type of product for which product specific requirements should be considered. Applicable *Product EMC Standards* take precedence over *Generic Standards* and *Product Family EMC Standards*.

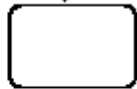
Anexo B. Planos de conexión

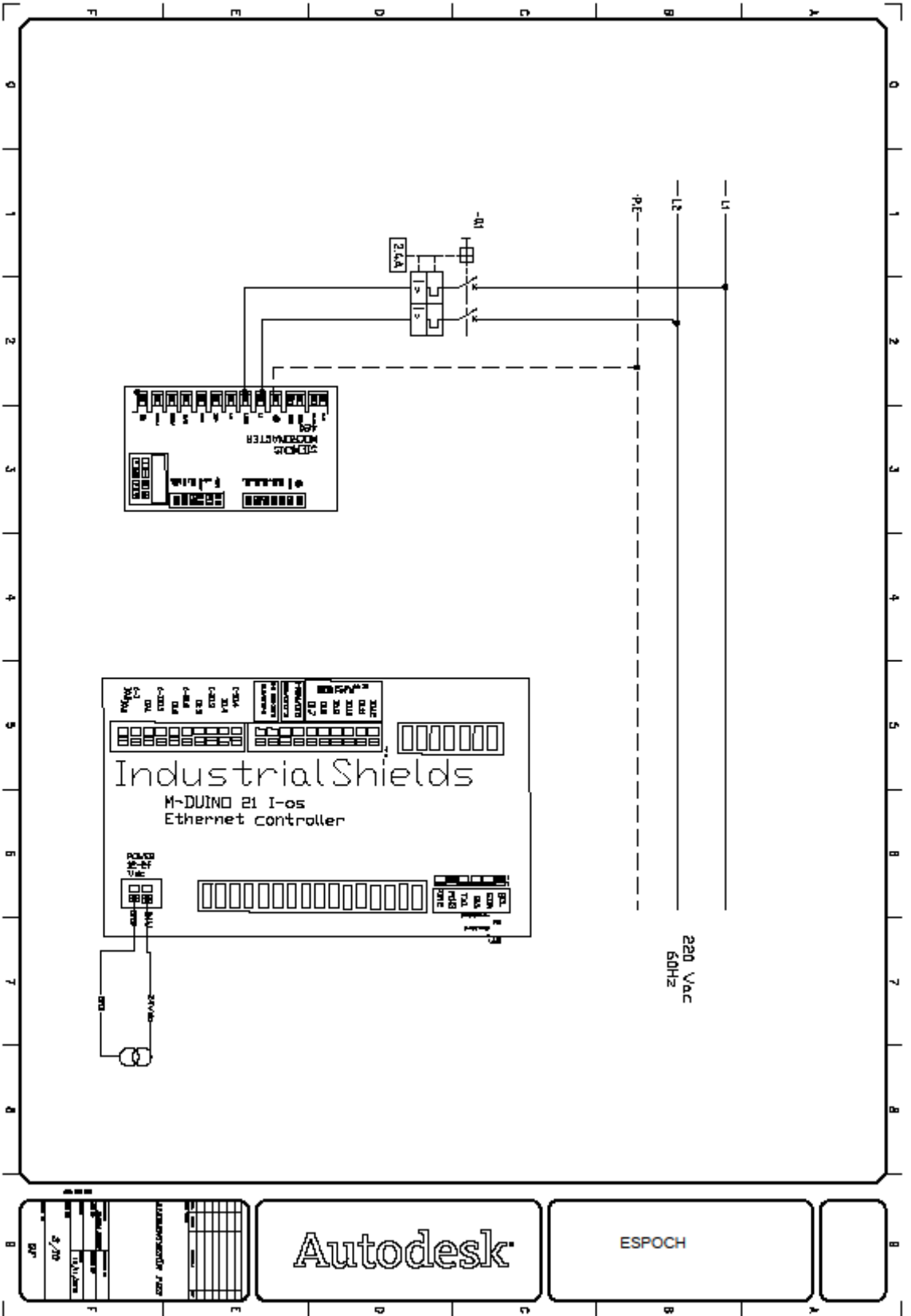


A										
B										
C										
D										
E										
F										
	0	1	2	3	4	5	6	7	8	9

ÍNDICE.

- 1. LAYOUT.
- 2. ÍNDICE.
- 3. ALIMENTACIÓN (TRIFÁSICA vac, 220 vac, 24 vdc).
- 4. ENTRADAS DIGITALES A PLC.
- 5. SALIDAS DIGITALES DEL PLC.
- 6. ENTRADAS ANALÓGICAS AL PLC.
- 7. SALIDAS ANALÓGICAS DEL PLC.
- 8. POTENCIA (MOTOR Y VDF).
- 9. PLANO NEUMÁTICO.



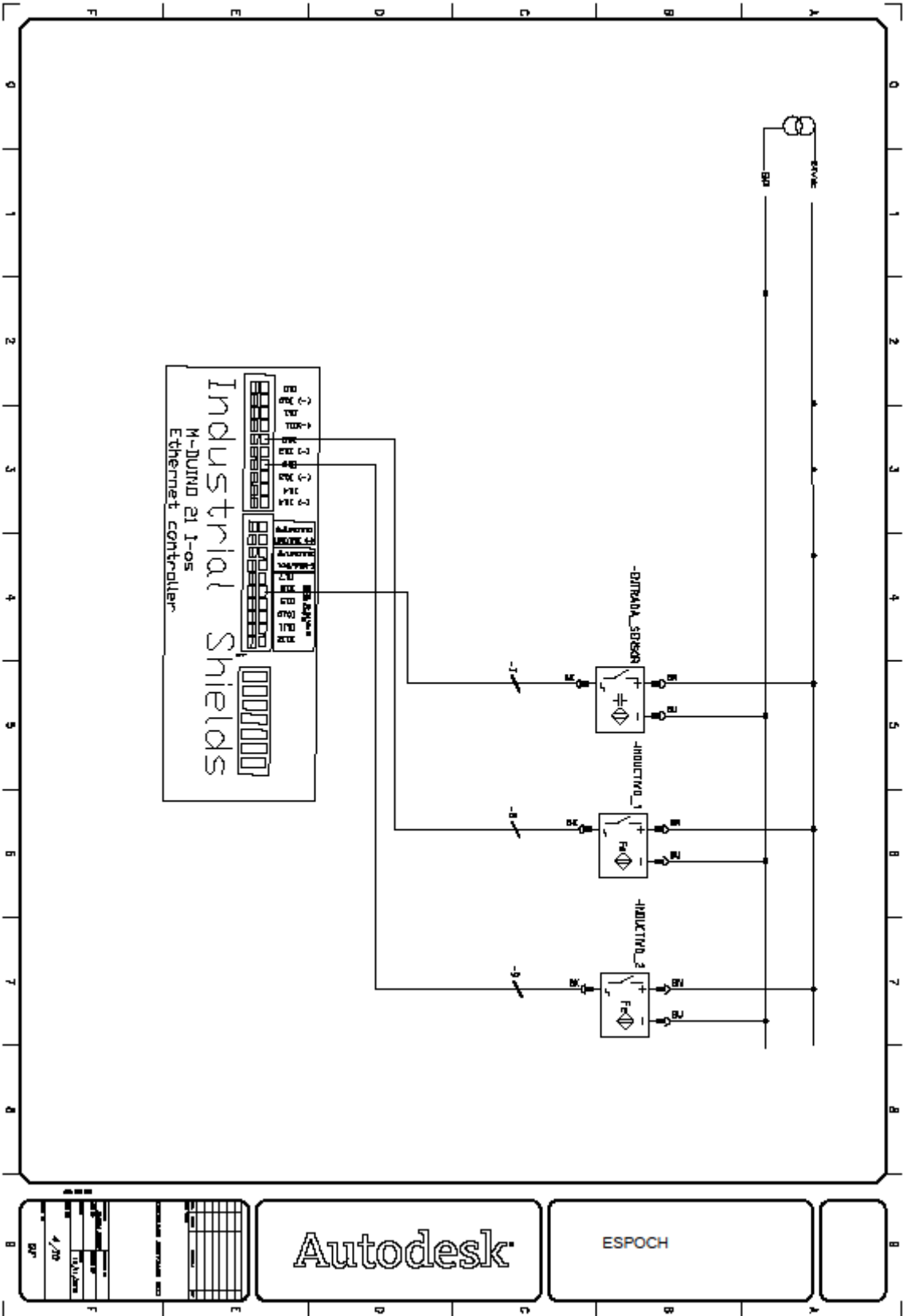


Autodesk	ESPOCH
----------	--------

Autodesk

ESPOCH

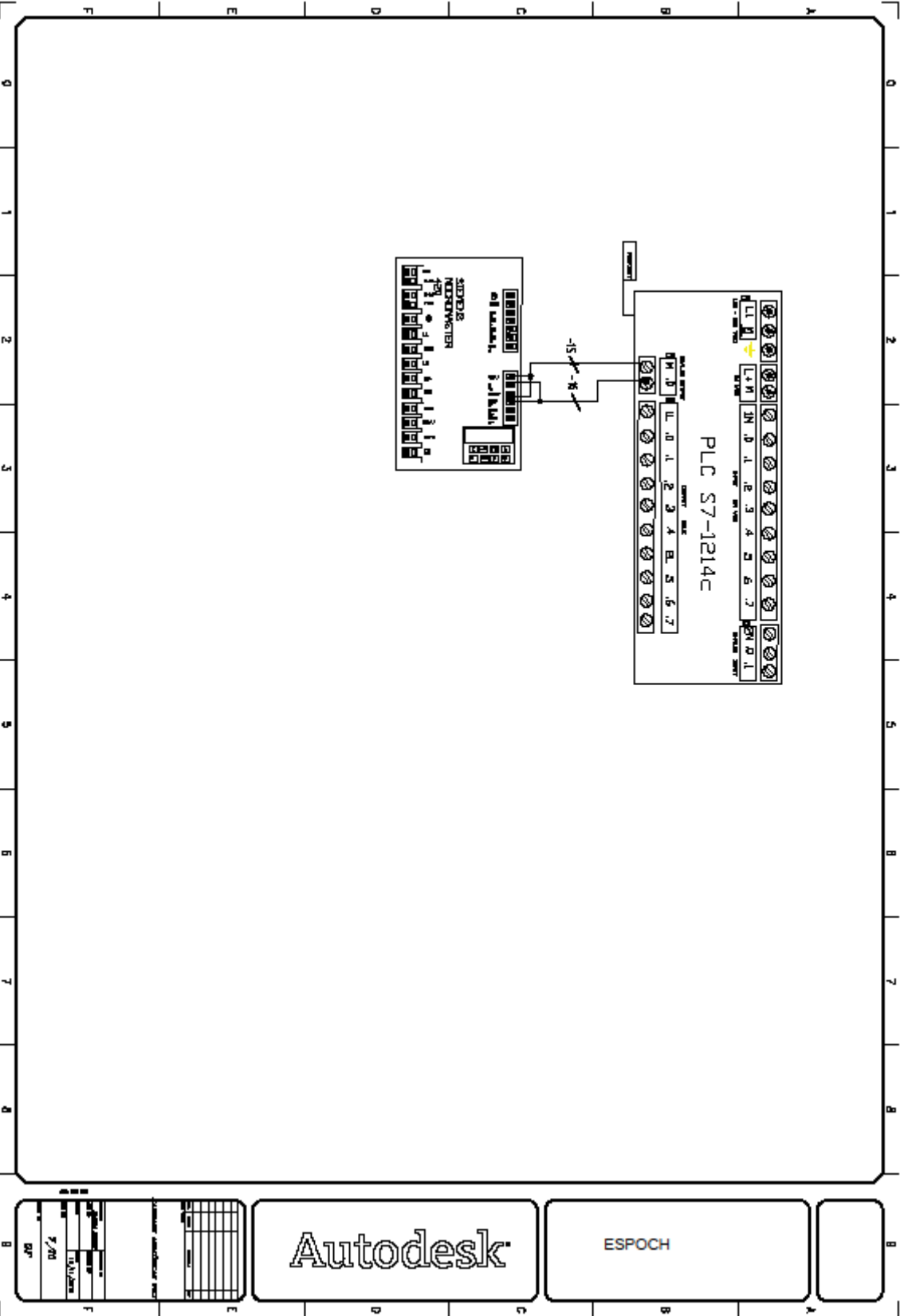




Autodesk

ESPOCH

NO	REV	DATE	BY	CHK
1	1	4/20	ESPOCH	
PROJECT: INDUSTRIAL CONTROL SYSTEM DRAWING: SCHEMATIC SHEET: 1 OF 1				



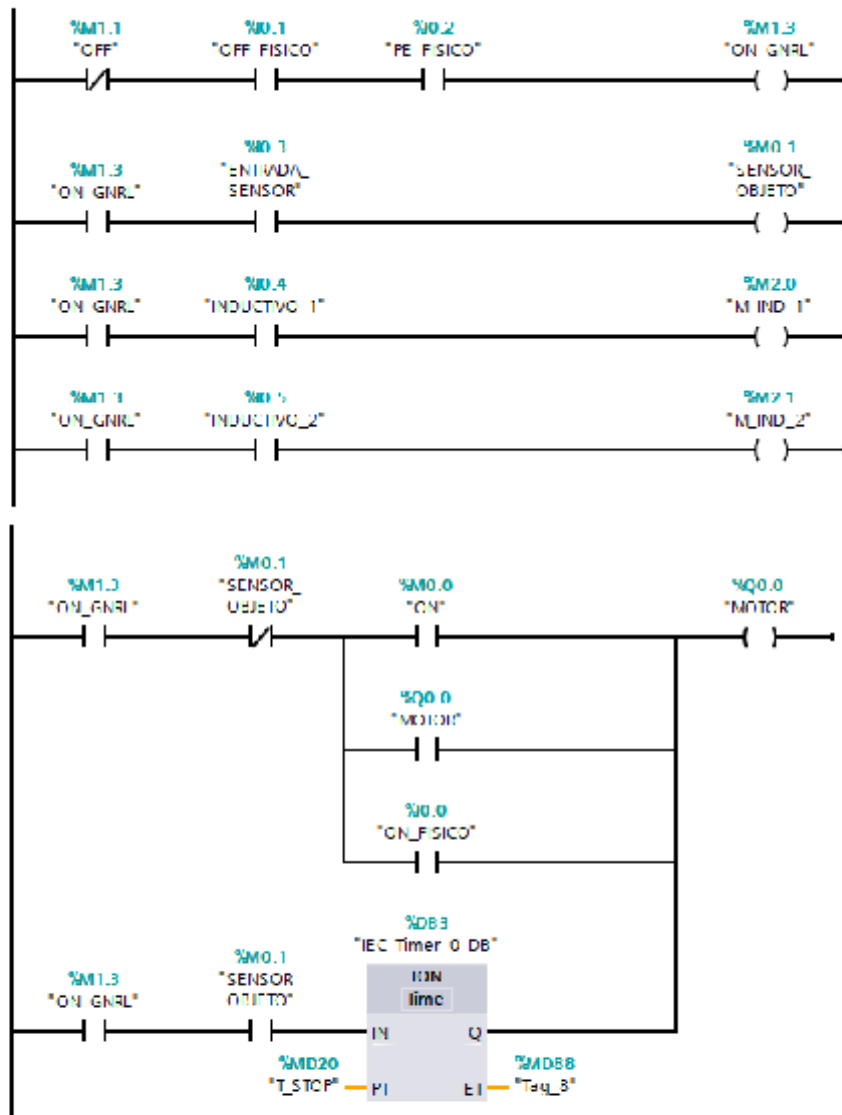
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

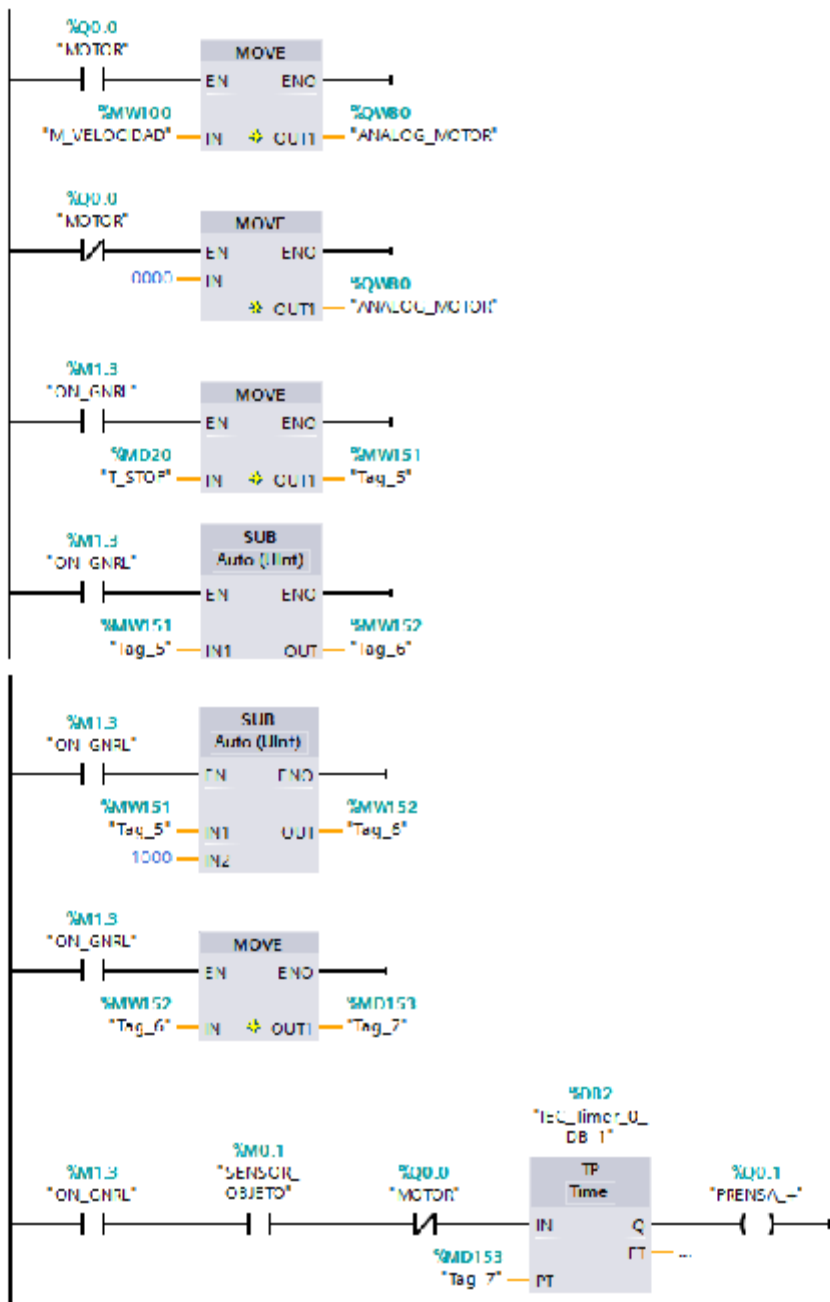
Autodesk

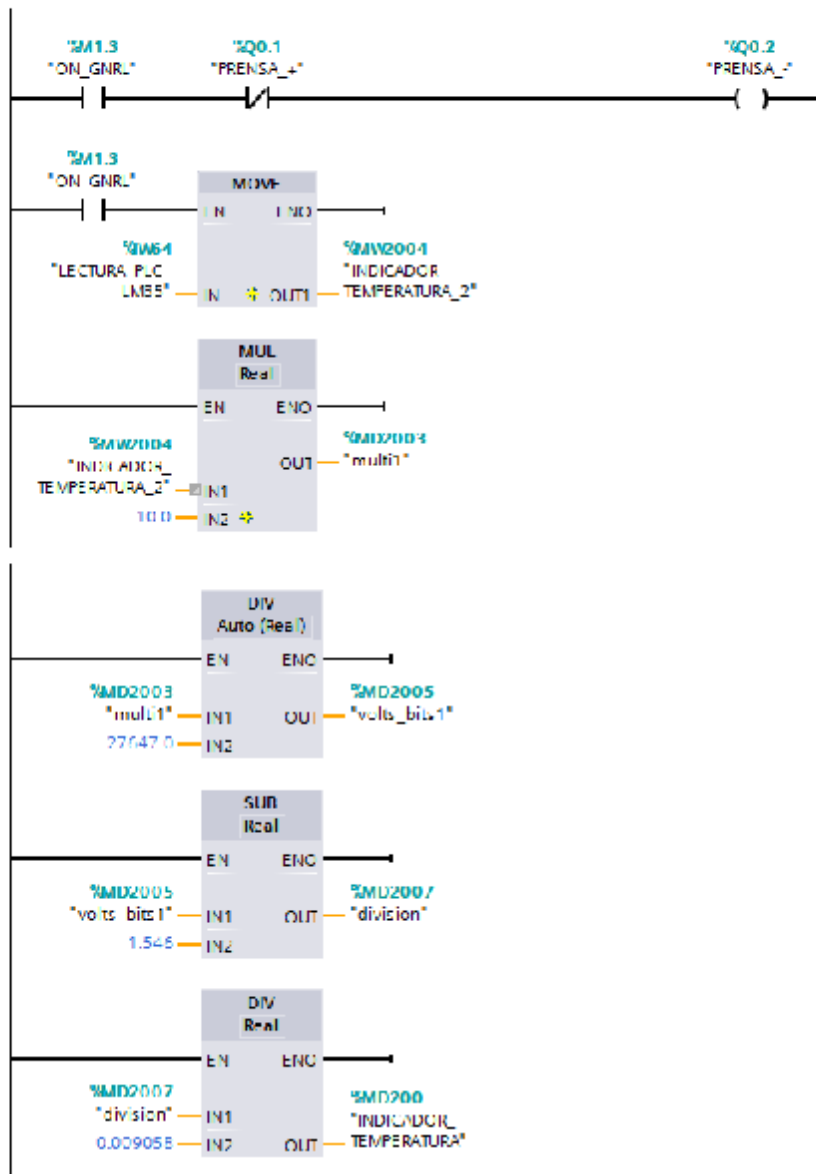
ESPOCH

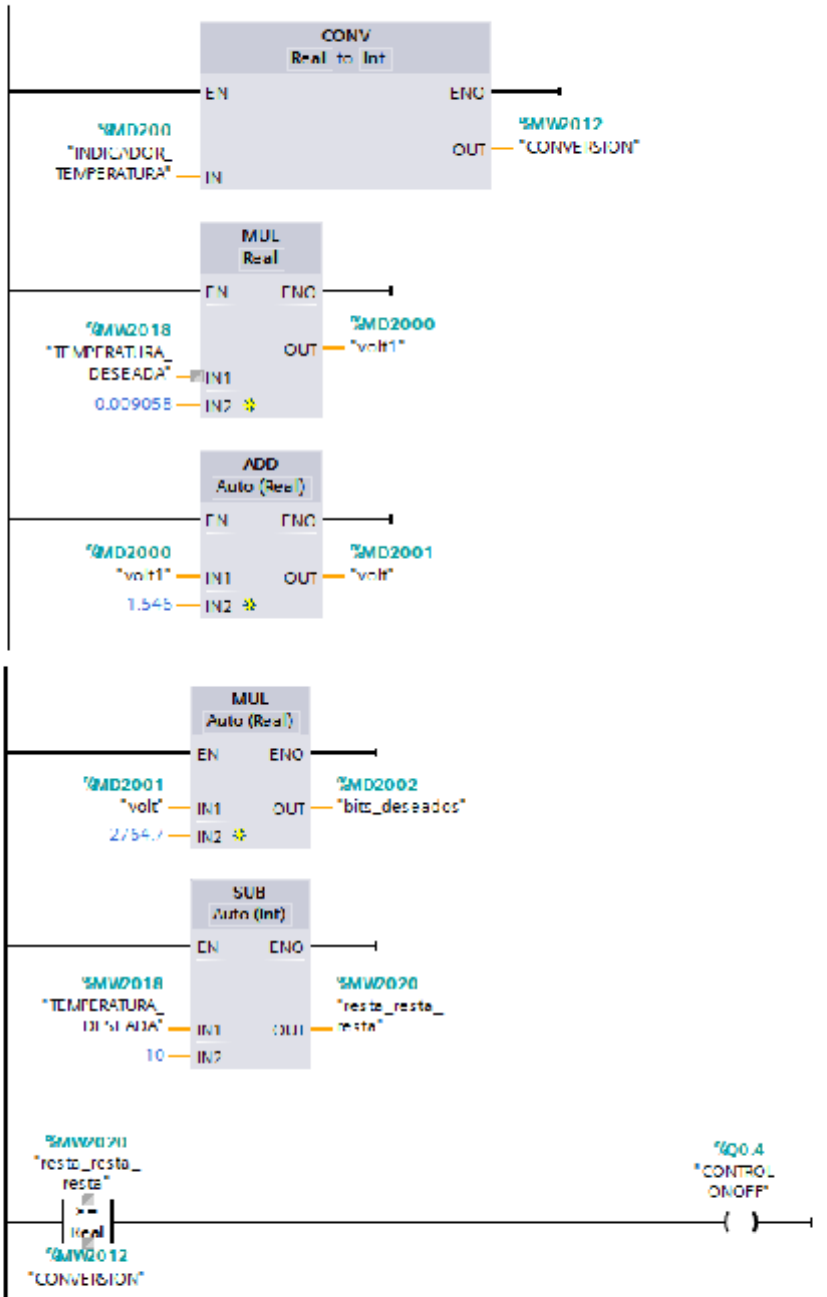


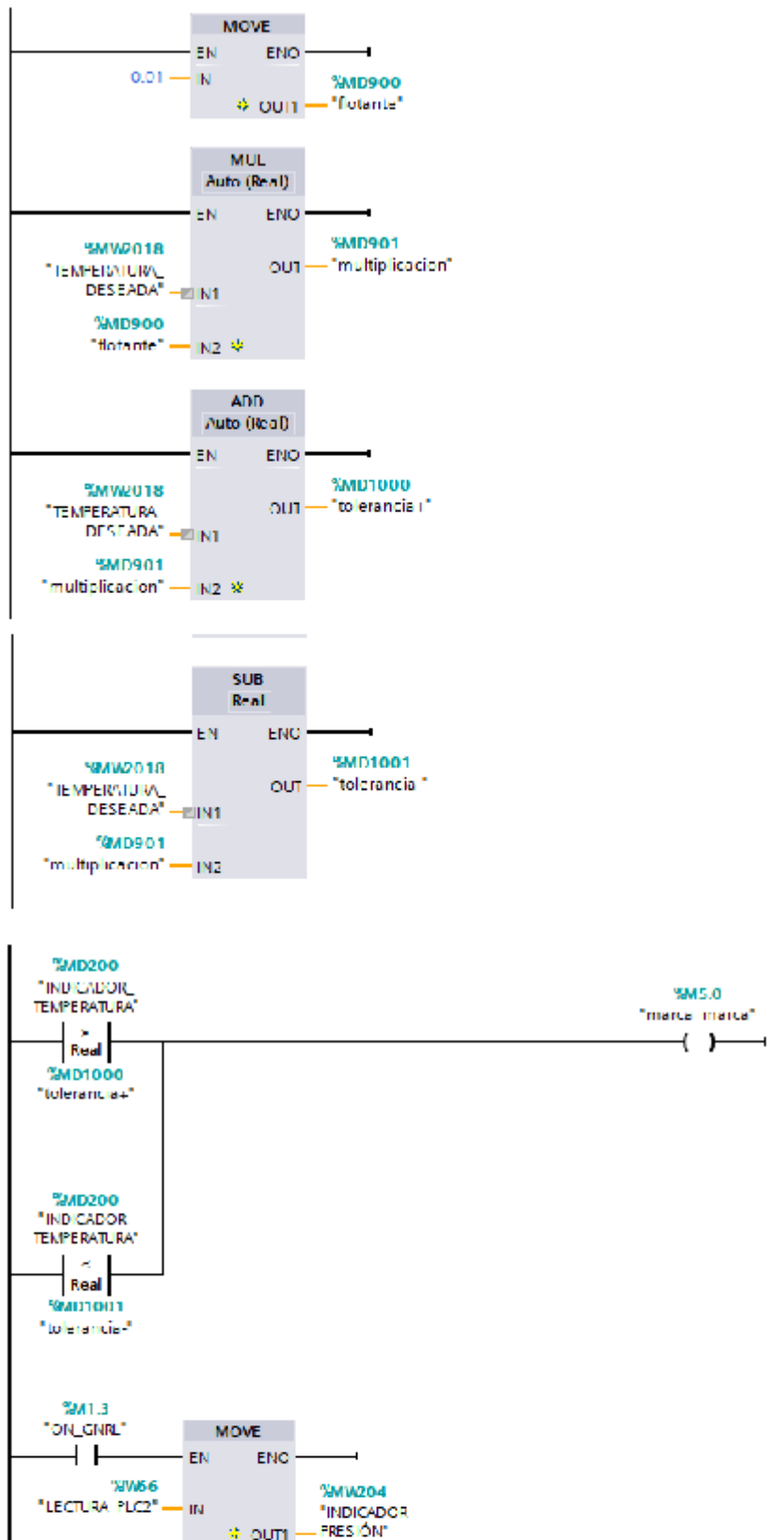
Anexo C. Código de programación para PLC S7-1200





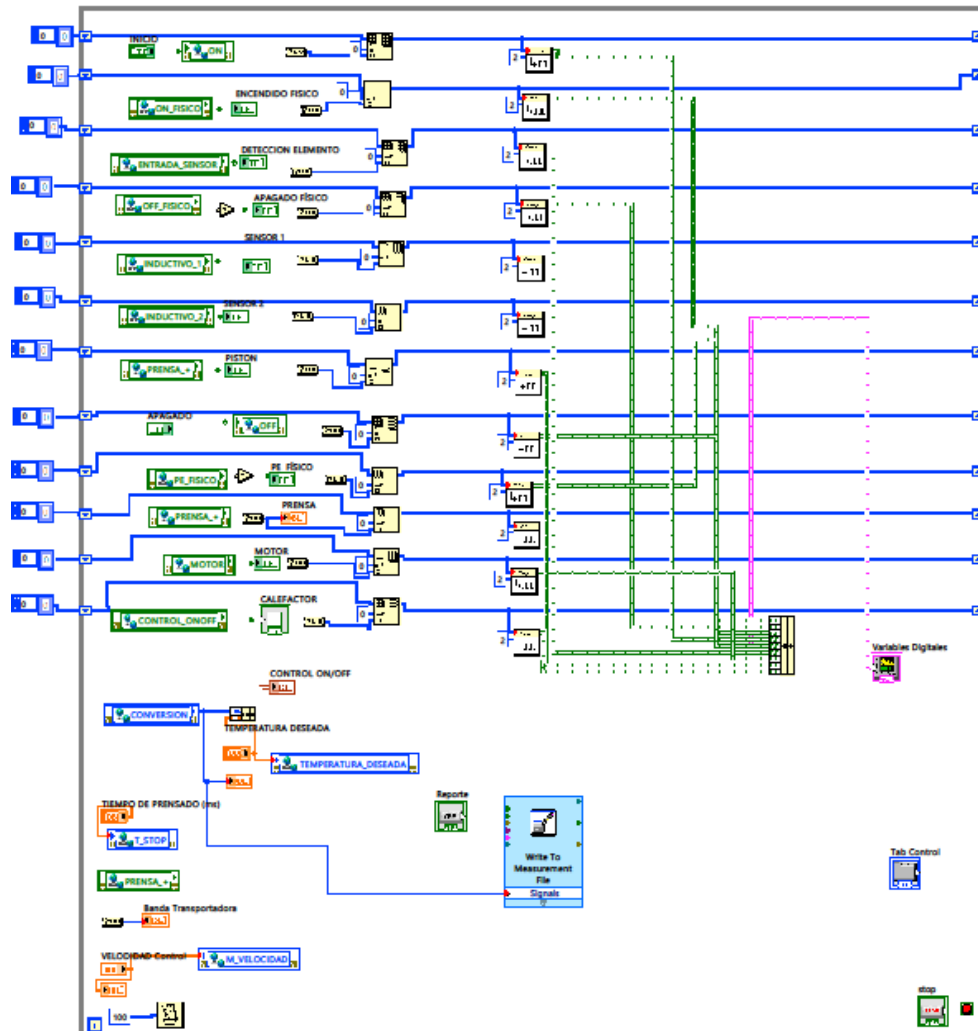






Anexo D. Código de programación interfaz LabVIEW

Untitled Library 1.lvlib:Untitled 1.vi
C:\Users\WINDOWS 8.1\Documents\UTA\Septimo\PLC\PROYECTO FINAL\proyecto plc\labview\
Untitled 1.vi
Last modified on 24/08/2016 at 18:44
Printed on 27/08/2016 at 11:08



Anexo E. Programación PLC M-DUINO

código arduino .ino

```
#include <SPI.h> // Librería SPL
#include <Ethernet.h> // Librería Ethernet
//PINES PLC
//Configuración Digital IN (24v]
int I06 = 3; //Interrupción número1
int I05 = 2; //interrupción número 0
int I16 = 19; // interrupción número 4
int I15 = 18; // interrupción número 5
int I26 = 21; // interrupción número 2
int I25 = 20; //I interrupción número 3
int I08 = A1; // Puerto Análogo PIN A1
int I07 = A0; // Puerto Análogo PIN A0
//Configuración Digital IN (24Vdc)
int I04 = 26; // selector Análogo/PWM
int I03 = 25; // Entrada digital
int I02 = 24; // Entrada digital
int I01 = 23; // Entrada digital
int I00 = 22; // Entrada digital
int I14 = 31; // Entrada digital
int I13 = 30; // Entrada digital
int I12 = 29; // Entrada digital
int I11 = 28; // Entrada digital
int I10 = 27; // Entrada digital
int I23 = 35; // Entrada digital
int I22 = 34; // Entrada digital
int I21 = 33; // Entrada digital
int I20 = 32; // Entrada digital
//Configuración Analogo/PWM/Digital IN (0-10Vdc or 24Vdc) int Q05 =
t; // selección análogo PWM Salida Digital
int Q06 = 5; // selección análogo PWM Salida Digital
int Q07 = 6; // selección análogo PWM Salida Digital
int Q15 = 8; // selección análogo PWM Salida Digital
int Q16 = 10; // selección análogo PWM Salida Digital
int Q17 = 11; // selección análogo PWM Salida Digital
int Q25 = 12; // selección análogo PWM Salida Digital
int Q26 = 13; // selección análogo PWM Salida Digital
//Configuración Digital OUT (24Vdc)
int Q00 = 36; // salida digital
int Q01 = 37; // salida digital
int Q02 = 38; // salida digital
int Q03 = 39; // salida digital
int Q04 = 40; // salida digital
int Q10 = 41; // salida digital
int Q11 = 42; // salida digital
```



```

int Q12 = 43; // salida digital
int Q13 = 44; // salida digital
int Q14 = 45; // salida digital
int Q20 = 46; // salida digital
int Q21 = 47; // salida digital
int Q22 = 48; // salida digital
int Q23 = 49; // salida digital
volatile int state = LOW; // Estado de la interrupción
byte mac[] = {
  0x08,0x6E,0x8E,0xEC,0x24,0xAD }; // Ingreso de la MAC Ethernet-
Client client; // Inicializar la libreria del cliente Ethernet IPAddress ip(192,168,1,20);
// Direccion IP del cliente
char serverName[] = "192.168.1.50"; // Direccion IP del servidor const
unsigned long requestInterval = 60000;
unsigned long lastAttemptTime = 0; boolean readingTweet = false;
boolean requested;
void setup() {
  //Configuración Análoga pwm
  pinMode(Q05, OUTPUT); // se asigna al pin la variable para el control
Q05
  pinMode(Q06, OUTPUT); // se asigna al pin la variable para el control
Q06
  pinMode(Q07, OUTPUT); // se asigna al pin la variable para el control
Q07
  pinMode(Q15, OUTPUT); // se asigna al pin la variable para el control
Q15
  pinMode(Q16, OUTPUT); // se asigna al pin la variable para el control
Q16
  pinMode(Q26, OUTPUT); // se asigna al pin la variable para el control
Q26
  pinMode(Q25, OUTPUT); // se asigna al pin la variable para el control
Q25
  //Configuración Digital
  pinMode(I00, INPUT); // se configura al pin como entrada la variable
I00
  pinMode(I01, INPUT); // se configura al pin como entrada la variable
I01
  pinMode(I02, INPUT); // se configura al pin como entrada la variable
I02
  pinMode(I03, INPUT); // se configura al pin como entrada la variable
I03
  pinMode(I04, INPUT); // se configura al pin como entrada la variable
I04
  pinMode(I10, INPUT); // se configura al pin como entrada la variable
I10
  pinMode(I11, INPUT); // se configura al pin como entrada la variable
I11

```

```

    pinMode(I12, INPUT); // se configura al pin como entrada la variable
I12
    pinMode(I13, INPUT); // se configura al pin como entrada la variable
I13
    pinMode(I14, INPUT); // se configura al pin como entrada la variable
I14
    pinMode(I20, INPUT); // se configura al pin como entrada la variable
I20
    pinMode(I21, INPUT); // se configura al pin como entrada la variable
I21
    pinMode(I22, INPUT); // se configura al pin como entrada la variable
I22
    pinMode(I23, INPUT); // se configura al pin como entrada la variable
I23
    pinMode(I06, INPUT); // se configura al pin como entrada la variable
I06
    pinMode(I07, INPUT); // se configura al pin como entrada la variable
I07
    pinMode(I08, INPUT); // se configura al pin como entrada la variable
I00
    pinMode(I15, INPUT); // se configura al pin como entrada la variable I15
    pinMode(I16, INPUT); // se configura al pin como entrada la variable
I16
    pinMode(I25, INPUT); // se configura al pin como entrada la variable
I25
    pinMode(I26, INPUT); // se configura al pin como entrada la variable
I26
    //Configuration Digital OUT (24Vdc)
    pinMode(Q00, OUTPUT); //Pin para subir pistón
    pinMode(Q01, OUTPUT); //Pin para bajar pistón
    Serial.begin(9600); // Control Baudics
    Ethernet.begin(mac,ip); // Declaración Mac
    Serial.print("My IP address: "); // Configuración Serial
    for (byte thisByte = 0; thisByte < 4; thisByte++) { // Declaración
de Condición
        Serial.print(Ethernet.localIP()[thisByte], DEC); // Imprimir la va-
riable
        Serial.print("."); // Impresión de punto
    }
    Serial.println();
    Serial.println(Ethernet.localIP()); // Impresión de la dirección IP
}
void loop() {
    //*****SENSORES*****
    double temp =45; // Asignación del valor a la variable temp
    int presencia = digitalRead(I04); // lectura de digital asignada al valor
presencia

```

```

    int presion = analogRead(I08); // lectura del puerto análogo declarada
a la variable presión
    Serial.println(presion); // imprimir el valor presión
    if(presencia == 1){ //Sensor de presencia activado
        delay(1000); // declaración de tiempo
        digitalWrite(Q02, HIGH); // Pistón Abajo 10
        digitalWrite(Q01, LOW);
        analogWrite(Q07, 0); //detener banda
        delay(5000); // Espera por 3 segundos
        Serial.println(presion);
        digitalWrite(Q02, LOW); // Pistón Arriba 01
        digitalWrite(Q01, HIGH);
        analogWrite(Q07, 0); //detener banda
        // delay(1000); // Espera por dos segundos
    }
    else{ //Sensor de presencia desactivado
        digitalWrite(Q02, LOW); // Pistón Arriba 01
        digitalWrite(Q01, HIGH);
        analogWrite(Q07, 40); //mover banda Serial.println(presion);
        //delay(1000); }
        digitalWrite(Q06,150);
        //*****ENVIO DATOS *****
        //dpresion= em=encia;
        //double tempC = presion;
        int arriba= 34; // asignar el valor 34 a la variable arriba
        int abajo = 12; // asignar el valor 12 a la variable abajo connectToSer-
ver(temp, presencia,presion, arriba, abajo);
    }
    void connectToServer(double temp, int presencia,double presion,
int arriba, int abajo) { // Configuración para conectar
        Serial.println("connecting to server..."); // Impresión Serial
        if (client.connect(serverName, 80)) { // Condición para el cliente
            Serial.println("making HTTP request..."); // Comparte el valor del
pedido :
            client.print("GET /abi/datospic.php?temperatura=");
            client.print(temp);
            client.print("&presencia=");
            client.print(presencia);
            client.print("&presion=");
            client.print(presion);
            client.print("&arriba=");
            client.print(arriba);
            client.print("&abajo=");
            client.print(abajo);
            client.println(" HTTP/1.1"); // Mensaje HTTP
            client.println("HOST: 192.168.1.50"); // Ip del cliente

```

```

    client.println(); Serial.print("GET /abi/datosplc.php?temperatura="); // impresión del mensaje temperatura
    Serial.print(temp); // impresión del valor temp Serial.print("&presencia="); // impresión de la palabra presencia
    Serial.print(presencia); // impresión del valor presencia Serial.print("&presion="); // impresión de la palabra presión
    Serial.print(presion); // impresión del valor presión Serial.print("&arriba="); // impresión de la palabra arriba
    Serial.print(arriba); // impresión del valor arriba Serial.print("&abajo="); // impresión de la palabra abajo
    Serial.print(abajo); // impresión del valor abajo
  }
  // note the time of this connect attempt:
  lastAttemptTime = millis(); client.stop();
  client.flush();
  //delay(100); // espera 5 segundos antes de volver a sensar la temperatura
}

```

Anexo F. Código PHP del Sistema SCADA

Código PHP

```
<?php //inicio de la programación
include_once ($ _SERVER['DOCUMENT_ROOT'] . '/Aracely/config.inc.php'); //incluir el documento config.inc.php
include_once (DIR_INC . "class.mysql.inc.php"); //incluir el documento clases mysql.inc.php
$a = new class_mysql(); //creamos la variable a
insert_personal($ _POST['nombre'],$ _POST['apellido'],$ _POST['direccion'],$ _POST['telefono'],$ _POST['cargo']); // incluimos los datos obtenidos nombre, apellido ,dirección, teléfono mediante formularios
$a->insert_arduino($ _GET['Id'],$ _GET['Fecha'],$ _GET['Hora'],$ _GET['Estado ON/OFF'],$ _GET['Temperatura'],$ _GET['Sensor_de_presion'],$ _GET['Sensor_presencia_del_producto'],$ _GET['SA_Velocidad_del_motor'],$ _GET['Sensor_piston'],$ _GET['SD_Enable/disable_motor'],$ _GET['PWM_motor']);
// incluimos los datos obtenidos Fecha, Hora, estado ron/off, temperatura, velocidad de motor, sensor pistón, enable, pwm, enviados en el url
$str_sql = "SELECT * FROM datos"; //creación de str_sql $rs_usuarios
$rs_usuarios = $a->query($str_sql); //creación de rs_usuarios
echo $rs_usuarios; //nos muestra los caracteres de rs_usuarios
while ($usr_data = mysql_fetch_assoc($rs_usuarios))
{ //asignación de condición while -mientras
    $IdL = $usr_data['Id']; //asignamos a la variable IDL el valor de IP de pdh
    $FechaL = $usr_data['Fecha']; //asignamos a la variable Fecha el valor de Fecha de pdh
    $HoraL = $usr_data['Hora']; //asignamos a la variable HoraL el valor de Hora de pdh
    $TemperaturaL = $usr_data['Estado ON/OFF']; //asignamos a la variable TemperaturaL el valor de Estado ON/OFF de pdh
    $HumedadL = $usr_data['Temperatura de prensa']; //asignamos a la variable humedadL el valor de Temperatura de prensa de pdh
    $PresionL = $usr_data['Sensor_de_presion']; //asignamos a la variable PresionL el valor de Sensor_de_prensa de pdh
    $ProductoL = $usr_data['Sensor_presencia_del_producto']; //asignamos a la variable ProductoL el valor de Sensor de prensa de producto de pdh
    $VelmotorL = $usr_data['SA_Velocidad_del_motor']; //asignamos a la variable VelmotorL el valor de Velocidad del motor de pdh
    $SenspistonL = $usr_data['Sensor_piston']; //asignamos a la variable SenspistonL el valor de Sensor de piston de pdh
    $SDL = $usr_data['SD_Enable/disable_motor'];
    $PWML = $usr_data['PWM_motor']; //asignamos a la variable PWML el valor de PWM motor de pdh
}
date_default_timezone_set('America/Bogota');
echo $fecha['d-m-y']."<br>"; // Ingreso de día mes y año
```

```
?>  
<?php // Inicio de PHP  
refresh:100; // Actualizar cada 100 mil segundos  
?>  
<html lang='es'>
```

Anexo G. Código HTML del Sistema SCADA

HTML

```
<head> // Cabecera de la página HTML
<meta charset='UTF-8'> <meta http-equiv="refresh" content="5">

<meta http-equiv="content-type|default-style|refresh">
<title> Arduino Shell Ethernet // Título de la página
</title>
<link
href='EstiloLetra.css'
rel='stylesheet'
type='text/css' />
<link rel='stylesheet'
href='Estilo1.css'>
</head>
<body> // Cuerpo de la página
<div class='page-wrap'>
<header class='header'>
<h1> // Mensaje Comunicaciones Avanzadas<br> Sensores + ARDUINO
</header>
<section class='content-wrap'>
<div class='device'>
<div class='device-name'>
<h2> FISEI </h2> // Incursión de Subtitulo FISEI
</div>
<div class='forms'>
<form class='transition button on'>
<?php
echo "<input type='button' value='ON/OFF='; // Impresión de
mensaje ON/OFF
echo $temperaturaL; // impresión de temperaturaL
echo"/>";
?>
</form>
</div>
<div class='forms'>
<form class='transition button on'>
<?php
echo "<input type='button' value='Sen_presion='; // Ingreso de
aracteres
echo $PresionL; // Impresión de la variable PresionL
echo"/>"
?>
</form>
</div>
<div class='forms'>
```

```

    <form class='transition button on'>
    <?php
    echo "<input type='button' value='SenProduct='"; //Imprimimos
mensaje de señalización botón
    echo $ProductoL; // Imprimimos la variable ProductoL
    echo"/>"; ?> // finalización del código
    </form> // Forma
    </div>
    <div class='forms'>
    <form class='transition button on'>
    <?php // Inicio de código
    echo "<input type='button' value='Velmotor='"; // Imprimimos los
valores
    echo $VelmotorL; // Imprimimos los valores de VelmotorL
    echo"/>";
    ?>
    </form>
    </div>
    <div class='forms'>
    <form class='transition button on'>
    <?php
    echo "<input type='button' value='Temp_prensa='"; // Imprimi-
mos el valor de Temp_Prensa
    echo $humedadL; // Imprimimos el valor de humedadL
    echo"/>";
    ?>
    </form>
    </div>
    <div class='forms'>
    <form class='transition button off'> //nombre de los botones <input
type='button' value='Salir' /> //
    </form>
    </div>
    </div>
    </section>
    </div>
    <h2>UTA <?php //Ingreso del mensaje UTA
    ?>

```


Anexo H. Código APP CONTROLLER

App Controller

```
<?php
namespace App\Controller; //sociaciones cake
php3 use Cake\Controller\Controller;
use Cake\Event\Event; use Cake\Core\Configure;
use Cake\Network\Exception\ForbiddenException; //
/**
 * @link http://book.cakephp.org/3.0/en/controllers.html#the-app-controller
 */
class AppController extends Controller
{
/**
 * User Id
 */
public $user_id;
/**
 * @return void
 */
public function initialize() // Método para agregar código de inicialización, como componentes de carga .
{
parent::initialize();
$this->loadComponent('RequestHandler');
$this->loadComponent('Flash');
$this->loadComponent('Auth', [
'loginAction'=>['controller'=>'Pages',
'action'=>'unauthorized', '_ext'=>'json'], 'authorize'=>['Controller'],
'authError'=>"Error"
]);
}
public function beforeFilter(Event $event) //Método petición de id del usuario
{
$this->Auth->allow();
$this->user_id = $this->Auth->user('id');
if($this->user_id)
{
//Validar token del usuario que ha iniciado
if(!$this->checkUserToken()
) {
}
}
$this->Auth->allow();
}
}
```

```

    }
    public function checkUserToken() //Metodo de comprobaci3n del token del usuario
    {
        $request_token = $this->getRequestToken();
        if (!$request_token) { return false; }
        if ($request_token != $this->userToken()
        ){
            return false;
        }
        return true;
    }
    public function getRequestToken() //Método para obtener token de solicitud {
        $headers = $this->getHeaders();
        if (!isset($headers['Authorization']))
            return false;
        $token = explode(" ",
            $headers['Authorization']); return $token[1];
    }
    private function getHeaders() //Método para obtener las solicitudes de cabecera {
        $headers = getAllheaders();
        return $headers;
    }
    public function userToken() //Método para obtener token de usuario {
        return
            $this->Auth->user('token');
    }
    public function isAuthorized($user) //Método de predeterminado si la
    autorización es correcta
    {
        return false;
    }
    /**
     * @param \Cake\Event\Event $event The beforeRender event.
     * @return void
     */
    public function beforeRender(Event $event) //Método de llamada
    cake.php {
        if (array_key_exists('_serialize', $this->viewVars) && in_array(
            $this->response->type(), ['application/json', 'application/xml']
        ) {
            $this->set('_serialize', true);
        }
    }
}

```

Anexo I. Código DATOS CONTROLLER

```
Datos Controller
<?php
namespace App\Controller;
use App\Controller\AppController;
use Cake\I18n\Time;
class DatosController extends AppController
{
/**
 * @return \Cake\Network\Response|null
 */
public function index()
{
    $datos = $this->paginate($this->Datos);
    $this->set(compact('datos'));
    $this->set('_serialize', ['datos']);
}
public function view($id = null) //Método de visualización de datos
{
    $dato = $this->Datos->get($id, [ 'contain' => [] ]);
    $this->set('dato', $dato);
    $this->set('_serialize', ['dato']);
}
public function add() //Método de guardado de datos
{
    $dato = $this->Datos->newEntity();
    if ($this->request->is('post')) {
        $dato = $this->Datos->patchEntity($dato, $this->request->data);

        if ($this->Datos->save($dato))
        {
            $this->Flash->success(__('The dato has been saved.'));
            return $this->redirect(['action' => 'index']);
        }
        else {
            $this->Flash->error(__('The dato could not be saved. Please,
try again.'));
        }
    }
    $this->set(compact('dato'));
    $this->set('_serialize', ['dato']);
}
public function edit($id = null) //Método de edición de datos
{
    $dato = $this->Datos->get($id, [ 'contain' => [] ]);
    if ($this->request->is(['patch', 'post', 'put']))
    {
```

```

    $dato = $this->Datos->patchEntity($dato, $this->request->data);

    if ($this->Datos->save($dato))
    {
        $this->Flash->success(__("The dato has been saved.));
        return
        $this->redirect(['action' => 'index']);
    } else {
        $this->Flash->error(__("The dato could not be saved. Please,
try again.));
    } }
    $this->set(compact('dato')); $this->set('_serialize', ['dato']);
    }
    public function delete($id = null) //Métodode eliminación de el-
ementos
    {
        $this->request->allowMethod(['post', 'delete']);
        $dato = $this->Datos->get($id); if ($this->Datos->delete($dato))

        {
            $this->Flash->success(__("The dato has been deleted.));
        } else {
            $this->Flash->error(__("The dato could not be deleted. Please,
try again.));
        }
        return $this->redirect(['action' => 'index']);
    }
    public function datosplc($id = null) // Método adquisición de
datos PLC
    {
        $dato = $this->Datos->newEntity();
        if ($this->request->is('get')) {
            $this->request->data['estadocistema']=1;
            $this->request->data['temperatura']=$this->request->query['tem-
peratura'];
            $this->request->data['produccion']=$this->request->query['pres-
encia'];
            $this->request->data['presion']=$this->request->query['presion'];

            if($this->request->query['arriba']){
                $this->request->data['piston']=1; //Solicitud estado del pistón
            } if($this->request->query['abajo']){
                $this->request->data['piston']=0; //Solicitud estado del pistón
            } $this->request->data['fecha']= Time::now(); //Solicitud de hora
actual
            $dato = $this->Datos->patchEntity($dato, $this->request->data);

```

```

        if ($this->Datos->save($dato)) {
            $this->Flash->success(__('The dato has been saved.)); //Men-
saje de confirmación
            return $this->redirect(['action' => 'index']);
        } else {
            $this->Flash->error(__('The dato could not be saved. Please,
try again.));
        } }
        $this->autoRender = FALSE;
        //echo $this->request->query['datos'];
        $file = fopen("/var/www/ress.txt", "w");
        fwrite($file, $this->request->query['temperatura'], ' ',$this->request-
>query['presencia'], ' ',$this->request->query['presion'], ' ',$this->request-
>query['arriba'], ' ',$this->request->query['abajo'].PHP_EOL);
        fclose($file);
    }
    public function datospresionjson($num = null){ //Método de datos
de presión
        $this->autoRender = FALSE;
        $time = $this->Datos->find()->func()->date_format([
            'fecha' => 'identifler',
            "'%T'" => 'literal'
        ]);
        $results = $this->Datos->find()
        ->
        select(['presion','hora'=>$time])
        ->where(function ($exp, $q) {
            $now = $q->func()->now();
            $now2 = $q->func()->date_sub(['now()'=>'literal','interval 1 day'=>
'literal']);
            return $exp->between('fecha',$now2,$now);
        })
        ->order(['id' => 'DESC'])
        ->limit(5)
        ->all();
        echo json_encode($results);
    }
    public function datostemperaturajson($num = null) { //Método de
datos de temperatura
        $this->autoRender = FALSE;
        $time = $this->Datos->find()->func()->date_format(['fecha' =>
'identifler',
            "'%T'" => 'literal' ]);
        $results = $this->Datos->find() ->select(['temperatura','hora'=>$time])
        ->where(function ($exp, $q) {

```

```
$now = $q->func()->now();
$now2 = $q->func()->date_sub(['now()'=>'literal','interval 1 day'=>
'literal']);
return
$exp->between('fecha',$now2,$now);
})
->order(['id' => 'DESC']) ->limit(5) ->all();
echo json_encode($results);
}}
```

Anexo J. Código LOGIN CONTROLLER

Login Controller

```
<?php
namespace App\Controller;
use App\Controller\AppController;
//sociaciones cake php3
use Cake\Core\Configure;
use Cake\Network\Exception\InternalErrorException;
use Cake\Network\Exception\UnauthorizedException;
use Cake\Event\Event;
use Cake\Utility\Text;
use Cake\Utility\Security;
use Cake\Auth\DefaultPasswordHasher;
use Cake\ORM\TableRegistry;
use Cake\Datasource\ConnectionManager;
use Cake\Network\Response;
// //Extension de parametros de Appcontroller.php
class LoginController extends AppController
{
public function initialize()
{
$this->loadModel('User');
parent::initialize();
}
public function beforeFilter(Event $event)
{
parent::beforeFilter($event);
$this->Auth->allow(['index', 'logout']);
}
/** * Index Login method API URL /api/login method: POST
* @return json response
*/
public function index()
{
$this->autoRender = FALSE;
$session=null;
$a = array('estado' => FALSE, 'datos'=>null, 'titulo'=>null,'msg'=>null,'tipomsg'=>'b:
gloss'); $results=null;
try {
if(!isset($this->request->data['username']))
{
//Solicitud de ingreso de usuario throw new UnauthorizedException("Please
enter your username");//Mensaje de Solicitud de ingreso de usuario
}
if(!isset($this->request->data['password']))
{

```

```

//Solicitud de ingreso de contraseña
throw new UnauthorizedException("Please enter your password"); //Men-
saje de Solicitud de ingreso de contraseña
}
$username = $this->request->data['username'];
//Petición de usuario
$password = $this->request->data['password']; //Petición de con-
traseña
$user = $this->Auth->identify(); //Comprobar las credenciales de usuario
if(!$user)
{
throw new UnauthorizedException("Invalid login"); // Advertencia
codigo invalido
}
$this->Auth->setUser($user); //Autenticación de datos del usuario

$this->Personas=TableRegistry::get('Personas');
$this->Users=TableRegistry::get('Users');
$this->Fotousers=TableRegistry::get('Fotousers');
if($user['group_id']!=1)
{
$results = $this->Users->find() ->where(['username' => $user-
name]) ->first();
}
$token = Security::hash($user['id'],$user['username'], 'sha1', true);
// Generación y autenticación del token
$this->request->session()->write('Auth.User.token', $token); //
Añadir token de usuario en la sesión de autenticación
$this->response->header('Authorization', 'Bearer '.$token); //Re-
torno Autenticación del token
}
catch (UnauthorizedException $e)
{
throw new UnauthorizedException($e->getMessage(),401);
}
$session =['user'=>$this->Auth->user()];
$session['user']['token']=$this->request->session()->read('Auth.User.token');
//Lectura autorización del token
if(!empty($results))
{
$session['user']['datosUsuario']=$results;
if($user['group_id']==2)
{
$session['user']['datosUsuario']['grupo']='Administrador'; //Inicio de
sesión como administrador
} if($user['group_id']==3)

```



```

    {
        $session['user']['datosUsuario']['grupo']='Operador'; //Inicio de sesión
        como operador
    }
    if(!empty($results2)){ $session['user']['foto']=$results2; //Inicio se-
    sion de con foto usuario
    }
    $a['datos']=$session;
    $this->set('user', $session['user']); $this->set('_serialize', ['user']);

}
/** * Logout user * API URL /api/login method: DELETE
 * @return json response
 */
public function logout() //Metodo de salida de usuario { $this->Auth-
>logout();
    $this->set('message', 'You were logged out'); // Mensaje de salida de
sesión
    $this->set('_serialize', ['message']);
}
}

```

Anexo K. Código USER CONTROLLER

```
User Controller
<?php
namespace App\Controller;
use App\Controller\AppController;
/**
 * Users Controller
 *
 * @property \App\Model\Table\UsersTable $Users
 */
class UsersController extends AppController
{
/**
 *Index method
 *
 * @return void
 */
public function initialize()
{
parent::initialize();
}
public function index()
{
$this->paginate = [
'contain' => ['Groups']
];
$this->set('users', $this->paginate($this->Users));
$this->set('_serialize', ['users']);
}
public function indexJson()
{
$this->paginate = [
'contain' => ['Groups']
];
$results = $this->Users->find()
->
where(['group_id <>' => 1]);
$this->set('users', $this->paginate($results));
$this->set('_serialize', ['users']);
}
/**
 * View method
 *
 * @param string|null $id User id.
 * @return void
 * @throws \Cake\Network\Exception\NotFoundException When record not
found.
```

```

*/
public function view($id = null) //Método de visualización de usuarios
{
    $user = $this->Users->get($id, [
        'contain' => ['Groups', 'Fotousers']
    ]);
    $this->set('user', $user);
    $this->set('_serialize', ['user']);
}
public function viewJson($id = null) //Método Json de visualización
de usuarios
{
    $user = $this->Users->get($id, [
        'contain' => ['Groups', 'Fotousers']
    ]);
    $this->set('user', $user); $this->set('_serialize', ['user']);
}
/**
 * @return void Redirects on successful add, renders view otherwise.
 */ public function add() // Método para agregar nuevos usuarios
{
    $user = $this->Users->newEntity();
    if ($this->request->is('post'))
    {
        $user = $this->Users->patchEntity($user, $this->request->data);

        if ($this->Users->save($user))
        {
            $this->Flash->success(__('The user has been saved.));
            return $this->redirect(['action' => 'index']);
        } else {
            $this->Flash->error(__('The user could not be saved. Please,
try again.));
        }
    }
    $groups = $this->Users->Groups->find('list', ['limit' => 200]);
    $this->set(compact('user', 'groups'));
    $this->set('_serialize', ['user']);
}
public function addJson() // Método Json para agregar nuevos usuarios
{
    $a = array('estado' => FALSE, 'datos' => null, 'titulo' => null, 'msg' => null, 'tipomsg' => 'b.
gloss'); $user = $this->Users->newEntity(); if ($this->request->is('put'))
    {
        $this->autoRender = FALSE; $a['datos']=$this->request->data;
    }
}

```

```

    $user = $this->Users->patchEntity($user, $this->request->data);

    if(!$user->errors()) {
    if ($this->Users->save($user)) {
    $a['titulo']='Correcto!'; $a['msg']=['ok'=>' Usuario guardado cor-
    rectamente'];
    $a['estado']=true; $a['tipomsg']='success';
    } else {
    $a['titulo']='Error!'; $a['msg']=['error'=>' No se pudo guardar el
    Usuario'];
    $a['tipomsg']='danger';
    }
    }else {
    $errores = $user->errors(); foreach ($errores as $indice => $sub-
    array) { $a['msg'] = $subarray;
    }
    $a['tipomsg']='warning'; $a['titulo']='Aviso.'; } echo json_encode($a);

    }
}
/**
 * Edit method
 *
 * @param string|null $id User id.
 * @return void Redirects on successful edit, renders view otherwise.
 * @throws \Cake\Network\Exception\NotFoundException When record not
    found.
 */
public function edit($id = null) //Método de edición de usuarios
{
    $user = $this->Users->get($id, [ 'contain' => [] ]);
    if ($this->request->is(['patch', 'post', 'put']))
    {
    $user = $this->Users->patchEntity($user, $this->request->data);

    if ($this->Users->save($user))
    {
    $this->Flash->success(__('The user has been saved.));
    return
    $this->redirect(['action' => 'index']);
    } else {
    $this->Flash->error(__('The user could not be saved. Please,
    try again.));
    } }
    $groups = $this->Users->Groups->find('list', ['limit' => 200]);
    $this->set(compact('user', 'groups'));

```

```

    $this->set('_serialize', ['user']); }
    public function editJson($id = null) //Método Json de edición de
usuarios {
    $a = array('estado' => FALSE, 'datos'=>null, 'titulo'=>null, 'msg'=>null, 'tipomsg'=>'b.
gloss'); $this->autoRender = FALSE;
    $user = $this->Users->get($id, [ 'contain' => [] ]);
    if ($this->request->is('patch')) {
        $new=[ 'username'=> $this->request->data['username'], 'group_id'=>
$this->request->data['group_id'], 'identificador'=> $this->request-
>data['identificador'], // 'password'=> $this->request->data['password'],
'estado'=>$this->request->data['estado'] ];
        $user = $this->Users->patchEntity($user, $new);
        if(!$user->errors()) { if ($this->Users->save($user)) {
            $a['titulo']='Correcto!'; $a['msg']=['ok'=>] ' Usuario actualizado cor-
rectamente';
            $a['estado']=true; $a['tipomsg']='success';
        } else {
            $a['titulo']='Error!'; $a['msg']=['error'=>] ' No se pudo actualizar
el Usuario'; $a['tipomsg']='danger';
        }
    } else {
        $errores = $user->errors(); foreach ($errores as $indice => $sub-
array) {
            $a['msg'] = $subarray; } $a['tipomsg']='warning'; $a['titulo']='Aviso.';
        }
    }
    echo json_encode($a);
} }
/**
 * @param string|null $id User id.
 * @return \Cake\Network\Response|null Redirects to index.
 * @throws \Cake\Network\Exception\NotFoundException When record not
found.
 */
public function delete($id = null) //Método de eliminación de usuarios
{
    $this->request->allowMethod(['post', 'delete']);
    $user = $this->Users->get($id); if ($this->Users->delete($user))
{
    $this->Flash->success(__("The user has been deleted.));
    } else {
    $this->Flash->error(__("The user could not be deleted. Please,
try again.));
    }
    return
    $this->redirect(['action' => 'index']); }

```

```

public function deleteJson($id = null) //Método Json de eliminación
de usuarios
{
    $a = array('estado' => FALSE, 'datos' => null, 'titulo' => null, 'msg' => null, 'tipomsg' => 'black-
gloss'); $this->autoRender = FALSE;
    $this->request->allowMethod(['post', 'delete']);
    $user = $this->Users->get($id); if ($this->Users->delete($user))
    {
        $a['titulo']='Correcto!'; $a['msg']={['ok'=>' El usuario ha sido elimi-
nado.'];
        $a['estado']=true; $a['tipomsg']='success'; } else { $a['titulo']='Er-
ror!';
        $a['msg']={['error'=>' El usuario no se pudo eliminar. Por favor, inténtelo
de nuevo']; $a['tipomsg']='danger'; }
        echo json_encode($a);
    }
}

```