



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO  
FACULTAD DE MECÁNICA  
ESCUELA DE INGENIERÍA DE MANTENIMIENTO**

**“IMPLEMENTACIÓN DE UN MÓDULO DE ENTRENAMIENTO  
PARA CONFIGURACIÓN DE COMUNICACIÓN MODBUS EN EL  
LABORATORIO DE CONTROL Y MANIPULACIÓN  
AUTOMÁTICA”.**

**CHOCO VÉLEZ, EDWIN JAVIER;  
ÁLVAREZ TOABANDA, JOSÉ LUIS**

**TRABAJO DE TITULACIÓN  
TIPO: PROPUESTAS TECNOLÓGICAS**

**Previa a la obtención del Título de:**

**INGENIERO DE MANTENIMIENTO**

**RIOBAMBA - ECUADOR  
2017**

**ESPOCH**

Facultad de Mecánica

---

**CERTIFICADO DE APROBACIÓN  
DEL TRABAJO DE TITULACIÓN**

---

2017-11-15

Yo recomiendo que el Trabajo de Titulación preparado por:

**CHOCO VÉLEZ EDWIN JAVIER**

Titulado:

**“IMPLEMENTACIÓN DE UN MÓDULO DE ENTRENAMIENTO PARA  
CONFIGURACIÓN DE COMUNICACIÓN MODBUS EN EL LABORATORIO  
DE CONTROL Y MANIPULACIÓN AUTOMÁTICA”**

Sea aceptado como parcial complementación de los requerimientos para el Título de:

**INGENIERO DE MANTENIMIENTO**

---

Ing. Carlos José Santillán Mariño  
**DECANO DE LA FAC. DE MECÁNICA**

Nosotros coincidimos con esta recomendación:

---

Ing. Pablo Ernesto Montalvo Jaramillo  
**DIRECTOR**

---

Dr. Marco Antonio Haro Medina  
**ASESOR**

**ESPOCH**

Facultad de Mecánica

---

**CERTIFICADO DE APROBACIÓN  
DEL TRABAJO DE TITULACIÓN**

---

2017-11-15

Yo recomiendo que el Trabajo de Titulación preparado por:

**ÁLVAREZ TOABANDA JOSÉ LUIS**

---

Titulado:

**“IMPLEMENTACIÓN DE UN MÓDULO DE ENTRENAMIENTO PARA  
CONFIGURACIÓN DE COMUNICACIÓN MODBUS EN EL LABORATORIO  
DE CONTROL Y MANIPULACIÓN AUTOMÁTICA”**

Sea aceptado como parcial complementación de los requerimientos para el Título de:

**INGENIERO DE MANTENIMIENTO**

---

Ing. Carlos José Santillán Mariño  
**DECANO DE LA FAC. DE MECÁNICA**

Nosotros coincidimos con esta recomendación:

---

Ing. Pablo Ernesto Montalvo Jaramillo  
**DIRECTOR**

---

Dr. Marco Antonio Haro Medina  
**ASESOR**

# ESPOCH

Facultad de Mecánica

## EXAMINACIÓN DEL TRABAJO DE TITULACIÓN

**NOMBRE DEL ESTUDIANTE:** CHOCO VÉLEZ EDWIN JAVIER

**TRABAJO DE TITULACIÓN:** “IMPLEMENTACIÓN DE UN MÓDULO DE ENTRENAMIENTO PARA CONFIGURACIÓN DE COMUNICACIÓN MODBUS EN EL LABORATORIO DE CONTROL Y MANIPULACIÓN AUTOMÁTICA”

**Fecha de Examinación:** 2017-11-15

**RESULTADO DE LA EXAMINACIÓN:**

COMITÉ DE EXAMINACIÓN	APRUEBA	NO PRUEBA	FIRMA
Ing. Ángel Guamán Mendoza <b>PRESIDENTE TRIB.DEFENSA</b>			
Ing. Pablo Montalvo Jaramillo <b>DIRECTOR</b>			
Dr. Marco Antonio Haro Medina <b>ASESOR</b>			

\* Más que un voto de no aprobación es razón suficiente para la falla total.

**RECOMENDACIONES:** \_\_\_\_\_  
\_\_\_\_\_

El Presidente del Tribunal certifica que las condiciones de la defensa se han cumplido.

\_\_\_\_\_  
Ing. Ángel Guamán Mendoza  
**PRESIDENTE DEL TRIBUNAL**

# ESPOCH

Facultad de Mecánica

## EXAMINACIÓN DEL TRABAJO DE TITULACIÓN

**NOMBRE DEL ESTUDIANTE:** ÁLVAREZ TOABANDA JOSÉ LUIS

**TRABAJO DE TITULACIÓN:** “IMPLEMENTACIÓN DE UN MÓDULO DE ENTRENAMIENTO PARA CONFIGURACIÓN DE COMUNICACIÓN MODBUS EN EL LABORATORIO DE CONTROL Y MANIPULACIÓN AUTOMÁTICA”

**Fecha de Examinación:** 2017-11-15

**RESULTADO DE LA EXAMINACIÓN:**

COMITÉ DE EXAMINACIÓN	APRUEBA	NO PRUEBA	FIRMA
Ing. Ángel Guamán Mendoza <b>PRESIDENTE TRIB.DEFENSA</b>			
Ing. Pablo Montalvo Jaramillo <b>DIRECTOR</b>			
Dr. Marco Antonio Haro Medina <b>ASESOR</b>			

\* Más que un voto de no aprobación es razón suficiente para la falla total.

**RECOMENDACIONES:** \_\_\_\_\_  
\_\_\_\_\_

El Presidente del Tribunal certifica que las condiciones de la defensa se han cumplido.

\_\_\_\_\_  
Ing. Ángel Guamán Mendoza  
**PRESIDENTE DEL TRIBUNAL**

## **DERECHOS DE AUTORIA**

El presente trabajo de titulación que presentamos es original y basado en el proceso de investigación y/o adaptación tecnológica establecido en la Facultad de Mecánica de la Escuela Superior Politécnica de Chimborazo. En tal virtud, los fundamentos teóricos - científicos y los resultados son de exclusiva responsabilidad de los autores. El patrimonio intelectual le pertenece a la Escuela Superior Politécnica de Chimborazo.

---

Choco Vélez Edwin Javier

---

Álvarez Toabanda José Luis

## **DEDICATORIA**

Dedico este trabajo a mis padres: Lauro Choco Fajardo y Noemí Vélez Segarra, por su apoyo incondicional a lo largo de mi vida y carrera universitaria, ya que son el pilar fundamental en mi existencia formándome con buenos hábitos y valores.

A mis hermanos: Franklin y Klever Choco Vélez, que siempre han estado junto a mí brindándome su apoyo y guía para culminar mi carrera universitaria.

**EDWIN JAVIER CHOCO VÉLEZ**

Este trabajo está dedicado a mis padres José Álvarez y Enriqueta Toabanda por sus cuidados, apoyo incondicional y guía durante el transcurso de mi vida. Gratitud a ellos por sus enseñanzas y cariño.

**JOSÉ LUIS ÁLVAREZ TOABANDA**

## **AGRADECIMIENTO**

A la Escuela Superior Politécnica de Chimborazo por darme la oportunidad de estudiar y ser un profesional. A mi tutor Ing. Pablo Montalvo y asesor Dr. Marco Haro, que con sus conocimientos, su experiencia, su paciencia y su motivación han logrado que pueda terminar mis estudios con éxito.

**EDWIN JAVIER CHOCO VÉLEZ**

A la planta docente de la Carrera de Ingeniería de Mantenimiento ya que cada uno ha contribuido en mi desarrollo profesional aportando conocimientos y experiencia, en especial a nuestro Tutor Ing. Pablo Montalvo por el apoyo y apertura para la realización del trabajo de titulación y a mi asesor Dr. Marco Haro por guiarme en el desarrollo del trabajo.

**JOSÉ LUIS ÁLVAREZ TOABANDA**



# CONTENIDO

Pág.

## CAPÍTULO I

1.	<b>INTRODUCCIÓN</b> .....	1
1.1.	Antecedentes .....	1
1.2.	Planteamiento del problema.....	1
1.3.	Justificación .....	2
1.4.	Objetivos .....	3
1.4.1	<i>Objetivo general</i> .....	3
1.4.2	<i>Objetivos específicos</i> .....	3

## CAPÍTULO II

2.	<b>MARCO TEÓRICO</b> .....	4
2.1.	Fundamentos de comunicación aplicada al proyecto.....	4
2.1.1	<i>Modelo OSI de comunicación.</i> .....	4
2.1.2	<i>Comunicación de campo.</i> .....	5
2.2.	Niveles de comunicación .....	5
2.3.	Protocolos de comunicación industriales .....	5
2.4.	Modbus.....	6
2.4.1	<i>Modos de transmisión Modbus.</i> .....	7
2.4.2	<i>Mapa de registro.</i> .....	8
2.4.3	<i>Modbus RTU.</i> .....	8
2.4.4	<i>Modbus TCP.</i> .....	10
2.5	PLC .....	11
2.5.1	<i>PLC modular TWDLMDA20DTK Twido.</i> .....	11
2.5.2	<i>Módulo de visualización TWDXCPODM.</i> .....	12
2.5.3	<i>Adaptador de interfaz</i> .....	13
2.5.4	<i>Convertidor bidireccional USB-RS485</i> .....	13
2.5.5	<i>Bus de comunicación Modbus</i> .....	13
2.6	EasyPort .....	14
2.6.1	<i>Interface SysLink para señales digitales.</i> .....	15
2.6.2	<i>Conector Sub D para señales analógicas</i> .....	16
2.6.3	<i>Interfases de datos para PC.</i> .....	16
2.6.4	<i>Comunicación con EasyPort USB.</i> .....	16
2.7	Software de programación. ....	18
2.7.1	<i>TIA Portal</i> .....	18
2.7.2	<i>TwidoSuite</i> .....	20
2.7.3	<i>LabVIEW</i> .....	20

### **CAPITULO III**

<b>3. DISEÑO Y CONSTRUCCIÓN.....</b>	<b>23</b>
3.1. Parámetros de diseño del módulo. ....	23
3.2. Diseño del módulo. ....	24
3.2.1 <i>Módulo con PLC Twido y PLC Siemens.</i> .....	24
3.2.2 <i>Módulo I/O</i> .....	24
3.3. Construcción del módulo. ....	24
3.3.1 <i>Carcasa</i> .....	24
3.3.2 <i>Montaje de equipos</i> .....	24
3.4. Configuración de la comunicación Modbus. ....	28
3.4.2 <i>Pasos para la configuración Modbus de distintos dispositivos</i> .....	30
3.4.3 <i>Comunicación de Easy Port con LabVIEW</i> .....	43
3.5. Pruebas y funcionamiento. ....	53
3.5.1 <i>Comunicación Modbus entre PLC Siemens y LabVIEW</i> .....	53
3.5.2 <i>Comunicación Modbus entre PLC Twido y LabVIEW</i> .....	59
3.5.3 <i>Comunicación Modbus entre el Módulo I/O Modbus y LabVIEW.</i> .....	64
3.5.4 <i>Comunicación Modbus entre PLCs y Módulo I/O Modbus con LabVIEW</i> .....	69
3.5.5 <i>Comunicación entre Easy Port y LabVIEW por OPC server</i> .....	72
3.5.6 <i>Comunicación entre EasyPort y LabVIEW por comunicación Serial</i> .....	75
3.6. Normas de seguridad para el uso del módulo. ....	80
3.7. Desarrollo de Guías de laboratorio. ....	81
3.7.1. <i>Guía de Laboratorio 1.</i> .....	81
3.7.2. <i>Guía de Laboratorio 2.</i> .....	81
3.7.3. <i>Guía de Laboratorio 3.</i> .....	82
3.7.4. <i>Guía de Laboratorio 4.</i> .....	82
3.7.5. <i>Guía de Laboratorio 5.</i> .....	82
3.8. Plan de mantenimiento del módulo.....	82
3.8.1 <i>Check List</i> .....	83

### **CAPITULO IV**

<b>4. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>85</b>
4.1. CONCLUSIONES. ....	85
4.2. RECOMENDACIONES .....	86

### **BIBLIOGRAFÍA**

### **ANEXOS**

## INDICE DE TABLAS

	<b>Pág.</b>
<b>Tabla 2-1</b> Breves funciones de las capas del modelo OSI.....	4
<b>Tabla 2-2</b> Niveles de comunicación.....	5
<b>Tabla 2-3</b> Mapa de registro.....	8
<b>Tabla 2-4</b> Correspondencia de registro MODBUS.....	8
<b>Tabla 2-5</b> Características del PLC Twido modular.....	11
<b>Tabla 2-6</b> Características Técnicas del EasyPort USB.....	15
<b>Tabla 2-7</b> Datos técnicos de la interface SysLink para señales digitales Puerto 1 y 2.15	15
<b>Tabla 2-8</b> Datos técnicos del conector Sub D tipo zócalo, para señales analógicas....	16
<b>Tabla 2-9</b> Datos técnicos de la interface RS232.....	16
<b>Tabla 2-10</b> Ejemplos de órdenes Modify y Display.....	18
<b>Tabla 2-11</b> Bloque Instrucción MB_CLIENT.....	19
<b>Tabla 2-12</b> Bloque de Instrucción MB_SERVER.....	19
<b>Tabla 2-13</b> Datos que soporta Modbus (resumen).....	22
<b>Tabla 3-1</b> Conexión entradas/salidas DB-25.....	27
<b>Tabla 3-2</b> Parámetros WRITE MULTIPLE COILS.....	41
<b>Tabla 3-3</b> Parámetros READ COILS.....	42
<b>Tabla 3-4</b> Parámetros READ INPUT REGISTERS.....	42
<b>Tabla 3-5</b> Direcciones OPC Puerto 1 del EasyPort.....	48
<b>Tabla 3-6</b> Direcciones OPC Puerto 2 del EasyPort.....	48
<b>Tabla 3-7</b> Direcciones OPC Puerto 3 EasyPort.....	48
<b>Tabla 3-8</b> Ejemplos códigos para EasyPort USB por puerto serial desde LabVIEW. 53	53
<b>Tabla 3-9</b> Parámetros Modbus PLC Siemens.....	54
<b>Tabla 3-10</b> Direccionamiento entradas y salidas físicas PLC Siemens en Modbus.....	54
<b>Tabla 3-11</b> (Continúa)Direccionamiento entradas y salidas PLC Siemens MB.....	55
<b>Tabla 3-12</b> Parámetros MODBUS.....	59
<b>Tabla 3-13</b> Direccionamiento Entradas del PLC Twido a Modbus API de LabVIEW. 60	60
<b>Tabla 3-14</b> Direccionamiento Salidas del PLC Twido a Modbus API de LabVIEW. ..	61
<b>Tabla 3-15</b> Bytes asignados entradas y salidas digitales del MÓDULO I/O Modbus... 65	65
<b>Tabla 3-16</b> Bytes asignados lectura de entradas analógicas del Módulo I/O Modbus. .	65
<b>Tabla 3-17</b> Códigos para las señales analógicas del EasyPort.....	79
<b>Tabla 3-18</b> Plan de mantenimiento.....	83
<b>Tabla 3-19</b> CHECK LIST.....	84

## INDICE DE FIGURAS

	<b>Pág.</b>
<b>Figura 2-1</b> Principales buses de campo. ....	6
<b>Figura 2-2</b> Transmisión de información entre esclavo y maestro.....	7
<b>Figura 2-3</b> Estructura de la comunicación MODBUS.....	7
<b>Figura 2-4</b> Estructura de Mensajes del Protocolo MODBUS.....	9
<b>Figura 2-5</b> Cableado comunicación MODBUS.....	9
<b>Figura 2-6</b> Estructura MODBUS TCP.....	10
<b>Figura 2-7</b> Partes del PLC Twido Modular. ....	11
<b>Figura 2-8</b> Esquema de conexiones del PLC Twido. ....	12
<b>Figura 2-9</b> Modulo de visualización TWDXCPODM.....	12
<b>Figura 2-10</b> Adaptador de interfaz TWDDNAC485T.....	13
<b>Figura 2-11</b> Convertidor bidireccional USB-RS485. ....	13
<b>Figura 2-12</b> Esquema ejemplo de comunicación Modbus.....	14
<b>Figura 2-13</b> Configuración con 4 módulos EasyPort USB.....	14
<b>Figura 2-14</b> Interface de Software para EasyPort USB .....	17
<b>Figura 2-15</b> Órdenes de escritura y lectura.....	17
<b>Figura 3-1</b> Hardware de la red MODBUS TCP/RTU. ....	23
<b>Figura 3-2</b> Diagrama conexión eléctrica de alimentación módulo PLC.....	25
<b>Figura 3-3</b> Conexión de equipos módulo.....	26
<b>Figura 3-4</b> Diagrama eléctrico PLC TWIDO. ....	26
<b>Figura 3-5</b> Diagrama eléctrico PLC Siemens. ....	27
<b>Figura 3-6</b> Vista lateral del módulo MODBUS.....	28
<b>Figura 3-7</b> Maestro de bajo nivel controlando temporización y una solicitud leída. ..	29
<b>Figura 3-8</b> Dispositivos Modbus TCP/IP y un servidor NI OPC Servers como host..	30
<b>Figura 3-9</b> Creación nuevo I/O server. ....	31
<b>Figura 3-10</b> Creación nuevo I/O server Modbus. ....	31
<b>Figura 3-11</b> Configuración parámetros Modbus I/O server. ....	32
<b>Figura 3-12</b> Parámetros de configuración avanzada Modbus.....	32
<b>Figura 3-13</b> Creación Bound Variables. ....	33
<b>Figura 3-14</b> Agregar rango de variables. ....	33
<b>Figura 3-15</b> Visualización variables agregadas. ....	34
<b>Figura 3-16</b> Editor múltiple de variables. ....	34
<b>Figura 3-17</b> Librería Modbus I/O server agregada al proyecto. ....	35
<b>Figura 3-18</b> Creación de un nuevo VI. ....	35
<b>Figura 3-19</b> VI con variables Modbus. ....	36

<b>Figura 3-20</b>	Indicadores booleanos conectados con variables Modbus.....	36
<b>Figura 3-21</b>	Control numérico memoria M101 del PLC. ....	37
<b>Figura 3-22</b>	VI completo para comunicación Modbus. ....	37
<b>Figura 3-23</b>	Librería NI MODBUS en VI Packet Manager .....	38
<b>Figura 3-24</b>	Librería Modbus NI en palletes .....	38
<b>Figura 3-25</b>	Elementos Esclavo API.....	39
<b>Figura 3-26</b>	Elementos Maestro API .....	39
<b>Figura 3-27</b>	Librería NI Modbus en LabVIEW. ....	40
<b>Figura 3-28</b>	Parámetros instancia de inicio Modbus New Serial Master. ....	40
<b>Figura 3-29</b>	Controles creados en la instancia Modbus con sus parámetros. ....	41
<b>Figura 3-30</b>	Conexión de elementos librería Modbus. ....	41
<b>Figura 3-31</b>	Creación de todos los indicadores y controles. ....	42
<b>Figura 3-32</b>	Panel Frontal LABVIEW.....	43
<b>Figura 3-33</b>	Creación de controles e indicadores en LabVIEW. ....	44
<b>Figura 3-34</b>	Enlace de datos. ....	44
<b>Figura 3-35</b>	Toma de datos. ....	45
<b>Figura 3-36</b>	Selección tipo de acceso. ....	45
<b>Figura 3-37</b>	Selección del acceso al Path. ....	46
<b>Figura 3-38</b>	Accesos posibles desde EzOPC. ....	46
<b>Figura 3-39</b>	Configuración de EzOPC para la supervisión y control desde LabVIEW. ....	47
<b>Figura 3-40</b>	Direccionamiento lectura de la entrada 00 del Puerto 1 del Easy Port. ....	48
<b>Figura 3-41</b>	Direccionamiento escritura salida digital 01 del Puerto 1 del Easy Port. ....	49
<b>Figura 3-42</b>	Direccionamiento salida analógica Out 0 del Puerto 3 del Easy Port. ....	49
<b>Figura 3-43</b>	Direccionamiento lectura de la entrada analógica In 1 del Easy Port. ....	49
<b>Figura 3-44</b>	Vista frontal y de diagrama de bloques final. ....	50
<b>Figura 3-45</b>	Elementos VISA Serial a ser utilizados. ....	50
<b>Figura 3-46</b>	Configuración del puerto de comunicación serial. ....	51
<b>Figura 3-47</b>	Inserción del elemento escribir. ....	51
<b>Figura 3-48</b>	Inserción elementos Read y Close VISA.....	52
<b>Figura 3-49</b>	Panel Frontal VI para escribir y leer el estado de EasyPort.....	52
<b>Figura 3-50</b>	Parámetros MODBUS_SERVER PLC SIEMENS.....	53
<b>Figura 3-51</b>	Creación de variables Modbus dentro de la librería. ....	55
<b>Figura 3-52</b>	Editor múltiple de variables. ....	56
<b>Figura 3-53</b>	Colocación de variables Modbus dentro del VI.....	56
<b>Figura 3-54</b>	Variables Modbus de un bit con sus indicadores booleanos.....	57
<b>Figura 3-55</b>	Configuración para la escritura de M101.0 y M101.1 en el PLC.....	57

<b>Figura 3-56</b>	Diagrama de Bloques para comunicación del PLC S7-1200 SIEMENS...	58
<b>Figura 3-57</b>	Panel frontal comunicación Modbus TCP PLC S7-1200 .....	58
<b>Figura 3-58</b>	Configuración elementos Modbus 485 en TwidoSuite.....	59
<b>Figura 3-59</b>	Configuración parámetros de la red Modbus.....	60
<b>Figura 3-60</b>	Elementos Master API de la librería NI Modbus.....	61
<b>Figura 3-61</b>	Configuración elemento new master serial con los parámetros de la red..	62
<b>Figura 3-62</b>	Creación de controles e indicadores en READ COILS. ....	62
<b>Figura 3-63</b>	Creación de controles e indicadores en WRITE MULTIPLE REGISTER	63
<b>Figura 3-64</b>	Creación de controles e indicadores en READ REGISTERS. ....	63
<b>Figura 3-65</b>	Panel frontal del VI en LabVIEW para el PLC Twido.....	64
<b>Figura 3-66</b>	Vista Frontal Módulo I/O Modbus con la asignación de bytes/bits.....	66
<b>Figura 3-67</b>	Diagrama de bloques Skated Sequence 0 lectura de bits señales digitales.	66
<b>Figura 3-68</b>	Diagrama de bloques Skated Sequence 1 escritura bits señales digitales..	67
<b>Figura 3-69</b>	Diagrama de bloques Skated Sequence 2 lectura de señales analógicas. ..	67
<b>Figura 3-70</b>	Vista Frontal con controles e indicadores para comunicación Modbus. ...	68
<b>Figura 3-71</b>	Puerto serial Modulo I/O Modbus. ....	68
<b>Figura 3-72</b>	Puertos de conexión/alimentación Módulo I/O Modbus. ....	69
<b>Figura 3-73</b>	Elemento Set Unit ID.....	69
<b>Figura 3-74</b>	Vista de diagrama de bloques VI con PLCs y Módulo I/O Modbus. ....	70
<b>Figura 3-75</b>	Vista frontal VI con PLC Siemens, Twido y Módulo I/O Modbus. ....	71
<b>Figura 3-76</b>	Vista frontal VI Easy Port a través de OPC.....	72
<b>Figura 3-77</b>	EasyPort en EzOPC. ....	73
<b>Figura 3-78</b>	Entradas/Salidas analógicas EasyPort en EzOPC.....	73
<b>Figura 3-79</b>	Vista de diagrama de bloques del VI Easy Port a través de OPC.....	74
<b>Figura 3-80</b>	Concatenar String para escritura de salidas digitales del EasyPort. ....	75
<b>Figura 3-81</b>	Combo box ID Device. ....	76
<b>Figura 3-82</b>	Clúster con elementos booleanos a escribirse.....	76
<b>Figura 3-83</b>	Método convertir Hexadecimal a booleanos.....	77
<b>Figura 3-84</b>	Vista Frontal VI comunicación EasyPort RS232 para señales digitales....	77
<b>Figura 3-85</b>	Diagrama de bloques comunicación EasyPort RS232 señales digitales....	78
<b>Figura 3-86</b>	Combo Box Activar y Leer IN Analog.....	79
<b>Figura 3-87</b>	Combo Box OUTPUT CHANNEL. ....	79
<b>Figura 3-88</b>	Vista Frontal VI comunicación EasyPort RS232 para señales analógicas.	80

## LISTA DE ABREVIACIONES

<b>SCADA</b>	Supervisory, control and data acquisition. (Supervisión, control y adquisición de datos)
<b>PLC</b>	Controlador Lógico Programable
<b>RTU</b>	Remote Terminal Unit
<b>TCP/IP</b>	Protocolo de Control de Transmisión/Protocolo de Internet
<b>KOP</b>	Esquema de Contactos
<b>E/S</b>	Entradas y Salidas
<b>I</b>	Señales de Entrada del Proceso
<b>Q</b>	Señales de Salida al Proceso
<b>M</b>	Memorias de programación marcas
<b>OSI</b>	Open Systems Interconnection
<b>API</b>	Application Programming Interface
<b>RS</b>	Recommended Standard
<b>OPC</b>	OLE for Process Control
<b>ASCII</b>	American Standard Code for Information Interchange
<b>DDE</b>	Dynamic Data Exchange
<b>DSC</b>	Distributed Control System
<b>NI</b>	National Instruments
<b>LABVIEW</b>	Laboratory Virtual Instrument Engineering Workbench
<b>VI</b>	Instrumentos Virtuales
<b>CRC</b>	Verificación por Redundancia Cíclica
<b>CPU</b>	Unidad Central de Procesamiento
<b>VISA</b>	Virtual Instrument Software Architecture

## **LISTA DE ANEXOS**

- A.** Guías de Laboratorio.
- B.** Diagrama de programa cargado en PLC Siemens.
- C.** Diagrama de programa cargado en PLC Twido.
- D.** Diagrama eléctrico Módulo I/O Modbus.
- E.** Código de colores cable SysLink Easy Port y cable de entradas analógicas.



## RESUMEN

Se realizó el diseño e implementación de un módulo de entrenamiento para la configuración MODBUS. Para el desarrollo de este trabajo, se investigó acerca de la comunicación a aplicar, topología y configuración para el envío y recepción de datos entre la PC y dispositivos electrónicos que utilizan el protocolo MODBUS. Creada la comunicación entre la PC y cada uno de los dispositivos electrónicos que constituyen el módulo se procede a programar un dispositivo a la vez. La red está diseñada para la transferencia de datos de control y monitoreo entre las estaciones maestro-esclavos, esta se implementó con estaciones de simulación de procesos ya existente en el laboratorio de Control y Manipulación Automática de la Escuela de Ingeniería de Mantenimiento, la estación que trabaja como maestro será una PC (Software LabVIEW), la que se encarga de enviar y recibir señales de escritura y lectura procedentes de las estaciones esclavos. Se realizaron las pruebas de funcionamiento de la transferencia de datos en la red de comunicación maestro-esclavo: Modbus TCP/IP entre LabVIEW - PLC S7-1200 (SIEMENS), Modbus RTU entre LabVIEW - PLC Twido (SCHNEIDER) y LabVIEW – Modulo I/O Modbus (Autores), así, como LabVIEW – EasyPort (FESTO) a través de serial RS232 y OPC, cabe recalcar que en todos los casos el software LabVIEW hace de maestro en la red. Previo a la manipulación del módulo se recomienda una correcta familiarización del uso de todos los elementos que se ha utilizado en el desarrollo de este trabajo.

**PALABRAS CLAVES:** <COMUNICACIONES INDUSTRIALES>, <REDES INDUSTRIALES>, <PROTOCOLOS DE COMUNICACIÓN>, <CONTROL INDUSTRIAL>, <LABVIEW>.

## ABSTRACT

The design and implementation of a training module for the MODBUS configuration was carried out. For the development of this project, the communication to be applied was investigated, topology and configuration for sending and receiving data between the PC and electronic devices using the MODBUS protocol. Once the communication between the PC and each of the electronic devices that constitute the module is created, the procedure would be to program one device at a time. The network is designed for the transfer of control and monitoring data between the master-slave stations, this is implemented with existing process simulation stations in the Control and Automatic Manipulation laboratory of the School of Engineering of Maintenance, the station that works as master will be a PC (LabVIEW Software), which is responsible for sending and receiving writing and reading signals from the slave stations. Data transfer operation tests were performed on the master-slave communication network: Modbus TCP/IP between LabVIEW – PLC S7-1200 (SIEMENS), Modbus RTU between LabVIEW – PLC Twido (SCHNEIDER) y LabVIEW – Modbus I/O Modbus (Authors), as well as LabVIEW - EasyPort (FESTO) to through the serial RS232 and OPC, it should be emphasized that in all cases LabVIEW software acts as master on the network. Before the manipulation of the module, it is recommended to correct familiarization of the use of all the elements that has been used in the development of this project.

**KEYWORDS:** <INDUSTRIAL COMMUNICATIONS>, <INDUSTRIAL NETWORKS>, <PROTOCOLS OF COMMUNICATION>, <INDUSTRIAL CONTROL>, <LABVIEW>

# CAPÍTULO I

## 1. INTRODUCCIÓN

### 1.1. Antecedentes

El laboratorio de Control y Manipulación Automática, de la Carrera de Ingeniería de Mantenimiento de la Facultad de Mecánica consta de dispositivos y equipos electrónicos que se comunican a través del protocolo industrial Modbus, el mismo que actualmente en las industrias es muy usado para el control y la manipulación de equipos y procesos debido a que su aplicación es de bajo coste, simple y sobre todo porque es un protocolo abierto, teniendo así una gama muy vasta de utilización de este protocolo.

Mediante un trabajo de titulación en 2009 se implementó en el laboratorio de Control y Manipulación Automática un banco de ensayos que ocupa este protocolo, pero se limita a la realización de prácticas con sensores y un PLC a través de LabVIEW, este módulo no puede ser utilizado para otros procesos existentes en el laboratorio o que podrían desarrollarse posteriormente.

Desde la creación del Laboratorio de Control y Manipulación Automática de la Carrera de Ingeniería de Mantenimiento no ha existido un módulo de entrenamiento para la configuración y comunicación de dispositivos Modbus y podemos citar su importancia referenciando los resultados que obtuvo un trabajo similar desarrollado en laboratorios de la UPS en 2015, donde demostró la eficiencia de la conexión y de manera práctica y física el protocolo Modbus aplicado a procesos industriales.

### 1.2. Planteamiento del problema

La inexistencia de un módulo para la configuración y aplicación práctica de Modbus, a pesar de que existen equipos que pueden comunicarse a través de este protocolo industrial impide que los estudiantes de la Carrera de Ingeniería de Mantenimiento que se perfilen hacia el campo de la automatización se entrenen, desarrollen prácticas de laboratorio y

ganen experiencia en la utilización de dicho protocolo de comunicación que envía y receipta datos de autómatas estándares presentes en la industria, permitiendo establecer comunicaciones entre dispositivos indistintamente de sus marcas.

Este módulo sería muy útil para el Ingeniero de Mantenimiento debido a que las industrias actualmente buscan automatizar sus procesos ya que de esta manera las plantas industriales obtienen una variedad de beneficios incluidos, entre ellos, una mayor calidad y mayor rapidez de trabajo.

### **1.3. Justificación**

Este proyecto tecnológico tiene como finalidad la implementación de un módulo de entrenamiento Modbus para el Laboratorio de Control y Manipulación Automática de la Carrera de Ingeniería de Mantenimiento, dicho módulo permitirá que los estudiantes desarrollen prácticas en la utilización de este protocolo de comunicación industrial, estableciendo la misma entre la variedad de dispositivos existentes en el laboratorio, creando así una guía de laboratorio para establecer una conexión maestro-esclavo entre PLCs Twido y Siemens, los cuales nos servirán posteriormente para realizar prácticas que permitan controlar y monitorizar procesos y visualizarlos en una PC a través del software LabVIEW, además podemos agregar que actualmente el control de procesos en las industrias tiene una muy alta demanda debido a la diversidad de equipos existentes en el campo laboral, especialmente la utilización del Protocolo de comunicación Modbus que es un protocolo abierto de bajo coste y sencillo de aplicarlo.

Esta red Modbus tiene fines académicos que acorde a la misión y visión de la CARRERA DE INGENIERÍA DE MANTENIMIENTO, complementará el desarrollo de aprendizaje de los estudiantes, mejorando su preparación y experiencia en el campo de la automatización, colaborando así con el compromiso de entregar profesionales de calidad que cumplan con los requerimientos de las industrias.

## **1.4. Objetivos**

### **1.4.1 *Objetivo general***

Implementar un módulo de entrenamiento para configuración de comunicación Modbus en el laboratorio de control y manipulación automática de la carrera de ingeniería de mantenimiento.

### **1.4.2 *Objetivos específicos***

Configurar los respectivos envíos y recepción de datos hacia los diferentes dispositivos que hablen Modbus con la ayuda de un PC.

Desarrollar una comunicación industrial maestro esclavo entre el software LabVIEW y el PLC Twido, PLC Siemens, EasyPort y el Módulo I/O Modbus mediante el protocolo Modbus.

Realizar una guía de laboratorio para la utilización de este módulo de entrenamiento Modbus en la ejecución de prácticas de programación, configuración y comunicación de equipos dentro de esta red.

## CAPÍTULO II

### 2. MARCO TEÓRICO.

#### 2.1. Fundamentos de comunicación aplicada al proyecto.

La comunicación industrial se ha hecho indispensable en las plantas de manufactura modernas, estos procesos automatizados se encuentran conformados por equipos de diferente fabricante y su funcionamiento se halla en distintos niveles de comunicación, sin embargo, a menudo se desea que estos equipos trabajen coordinadamente entre sí para un resultado satisfactorio, la forma de conseguir esta coordinación se ve encomendada en las redes industriales. (CASTRO SILVA, y otros, 2017).

**2.1.1 Modelo OSI de comunicación.** El modelo OSI por sus siglas en el idioma inglés Open Systems Interconnection es un modelo de referencia para la interconexión de sistemas abiertos, además es un protocolo de 7 capas, trabaja de tal forma que se ejecuten las funciones de cada capa. Las capas físicas y de aplicación son las únicas capas del modelo OSI con las que el usuario interactúa. (CASTRO SILVA, y otros, 2017)

**Tabla 2-1** Breves funciones de las capas del modelo OSI.

<b>Aplicación</b>	“Proporciona el acceso al entorno OSI y servicios de información distribuida para los usuarios”.
<b>Presentación</b>	“Proporciona a los procesos de aplicación, Independencia respecto a las diferencias en la representación de los datos (sintaxis)”.
<b>Sesión</b>	“Proporciona el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las acciones cooperadoras”.
<b>Transporte</b>	“Proporciona seguridad, transferencias y transparencia de datos entre los puntos finales, además de recuperación de errores y control de flujo origen-destino”.
<b>Red</b>	“Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y transmisión utilizada para conectar los sistemas, es responsable del establecimiento, mantenimiento y cierre de las conexiones”.
<b>Enlace de datos</b>	“Proporciona una transferencia de datos segura a través del enlace físico; envía tramas de datos llevando a cabo la sincronización, el control de errores y flujo”
<b>Físico</b>	“Se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico, está relacionado con las características mecánicas, eléctricas, funciones y procedimientos para acceder al medio físico”.

**Fuente:** (CASTRO SILVA, y otros, 2017)

**2.1.2 Comunicación de campo.** Los buses de comunicación permiten el envío/recepción de información en tiempo real, tienen la capacidad de interconectar los distintos tipos de controladores lógicos con los sensores y actuadores presentes en los procesos industriales automatizados. Además, se utilizan mensajes cortos para realizar el control e interconexión de los dispositivos electrónicos, se puede dividir en dos clases de buses, el bus de dispositivos que utiliza pocos bytes y el bus actuador-sensor que utiliza bits. (CASTRO SILVA, y otros, 2017)

## 2.2. Niveles de comunicación

En una red industrial de comunicación coexisten una gran variedad de dispositivos, los cuales se los puede jerarquizar. (HERNANDEZ TINOCO, 2016)

**Tabla 2-2** Niveles de comunicación.

<b>Niveles de comunicación</b>	<b>Descripción</b>	<b>Dispositivos/redes</b>
Nivel de gestión	Es el encargado del Monitoreo.	Aplicaciones en red o estaciones de trabajo.
Nivel de célula	Es el encargado de enlazar y dirigir.	<b>Dispositivos:</b> autómatas, control de calidad, ordenadores de diseño, programación. <b>Redes:</b> LAN.
Nivel de campo	Se encarga de la integración de pequeños automatismos dentro de sud-redes o islas.	<b>Redes:</b> buses de campo o las redes Ethernet
Nivel de sensores y actuadores	Comunicación con el mundo real.	<b>Dispositivos:</b> actuadores y sensores.

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (HERNANDEZ TINOCO, 2016)

## 2.3. Protocolos de comunicación industriales

Un protocolo de comunicación industrial es un conjunto de reglas que viene insertado dentro de los dispositivos electrónicos de control que permiten la transferencia e intercambio de datos entre los dispositivos que conforman una red. Los protocolos poseen importantes ventajas, tales como mayor precisión derivada de la integración de tecnología digital en las mediciones, mayor y mejor disponibilidad de información de los dispositivos de campo y diagnóstico remoto de componentes. (LAGOS, 2006)

Existen las categorías Buses de campo, Redes LAN y Redes LAN-WAN para el diseño de una estructura de redes de comunicación industrial.

Un bus de campo es un sistema que permite la integración de equipos para la medición y control de variables de proceso, por medio de la transmisión de información que simplifica considerablemente la instalación y operación de máquinas y equipos industriales utilizados en procesos de producción. (LAGOS, 2006)

<b>Buses de campo con mayor presencia en el area de control y automatización</b>	HART	
	Profibus	Profibus DP
		Profibus PA
		Profibus FMS
	Fieldbus Foundation	
	Modbus	ASCII
RTU		
TCP		
Devicenet		

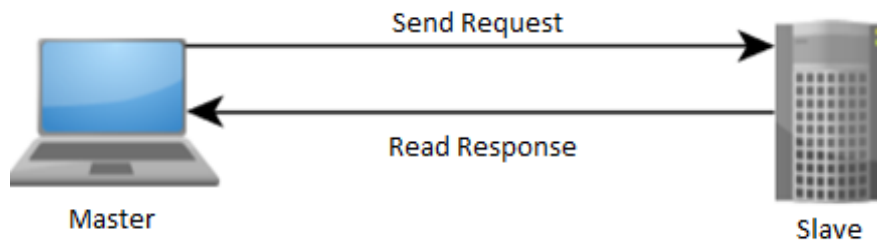
**Figura 2-1** Principales buses de campo.  
Fuente: (LAGOS, 2006)

## 2.4. Modbus

Modbus es un protocolo de comunicación serie creado por Modicon en el año de 1979. En sus inicios el protocolo estaba siendo usado únicamente en los PLCs de Modicon, pero en la actualidad es un protocolo de uso libre. Modbus es un protocolo para la comunicación entre dispositivos maestro-esclavo, el dispositivo (maestro) debe enviar solicitud y esperar una respuesta del otro dispositivo (esclavo), el maestro por lo general es un interfaz hombre-máquina (HMI) o sistema SCADA y el esclavo un sensor o un PLC. (HURTADO, 2015)

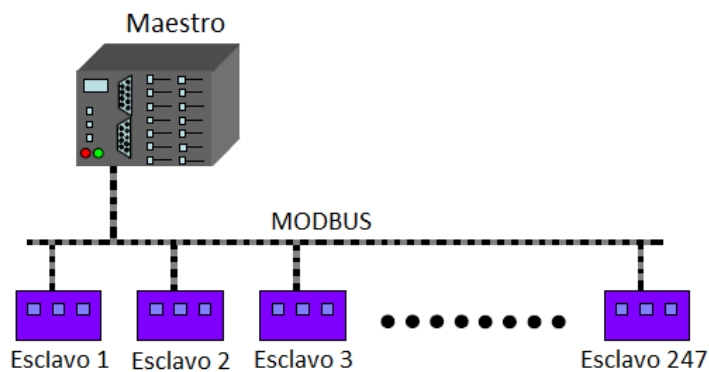
La conectividad de los equipos que utilizan este protocolo se realiza a través de líneas serie como pueden ser RS-232 o RS-485. Pero en la actualidad han aparecido variantes de conectividad como el Modbus TCP, que permite el encapsulamiento del Modbus serie en tramas Ethernet TCP/IP de forma sencilla, lo que ha producido que el protocolo Modbus sea uno de los protocolos más utilizados en las industrias, ya que permite conectar varios equipos de diferente marca en una red de comunicación sin necesidad de realizar acondicionamientos. (HURTADO, 2015)





**Figura 2-2** Transmisión de información entre esclavo y maestro.  
**Fuente:** (HURTADO, 2015)

“Los dispositivos Modbus usan interfaces serie compatibles con RS- 232C y RS-485, siendo el bus capaz de alcanzar distancias de transmisión de información de alrededor de 1 Km a velocidades de 19’2 Kbps”. (WEG, 2012)



**Figura 2-3** Estructura de la comunicación MODBUS.  
**Fuente:** (TERATRONIX, 2016)

El protocolo de comunicación Modbus realiza la transmisión de información entre distintos equipos electrónicos conectados a un mismo bus. En este bus existe únicamente un solo dispositivo maestro (Master) y varios equipos esclavos (Slaves). (HURTADO, 2015) (Ver Figura 2-3)

**2.4.1 Modos de transmisión Modbus.** Es posible realizar tres modos de transmisión: ASCII, RTU y TCP. “American Standard Code for Information Interchange o ASCII por sus siglas, es un sistema de codificación hexadecimal y cada carácter consta de 1 bit de inicio, 7 bits de codificación de los datos, 1 bit de paridad (opcional) y 1 o 2 bits de parada, o sea, un total de 9 a 11 bits por carácter”. (WEG, 2012)

“Remote Terminal Unit o RTU por sus siglas, es un sistema de codificación binario donde cada palabra transmitida posee 1 bit de inicio, 8 bits de codificación de los datos, 1 bit opcional de paridad y 1 o 2 bits de parada”. (WEG, 2012)

La versión Modbus/TCP es muy semejante al formato RTU, pero estableciendo la transmisión mediante paquetes TCP/IP. Y la comunicación se establece por redes Ethernet. (WEG, 2012)

**2.4.2 Mapa de registro.** El protocolo de comunicación Modbus tiene disponible un rango de memoria definido para el mapeo de cada tipo de dato, el rango de memoria se define según el tipo de dato y la función que se quiera realizar. (CASTRO SILVA, y otros, 2017)

**Tabla 2-3** Mapa de registro

Tipo de variable	Mapa de direcciones
Salidas digitales	0x0000 a 0x 9999
Entradas digitales	1x0000 a 1x 9999
VARIABLES ALMACENADAS	4x0000 a 4x 9999
Entradas analógicas	3x0000 a 3x 9999

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (CASTRO SILVA, y otros, 2017)

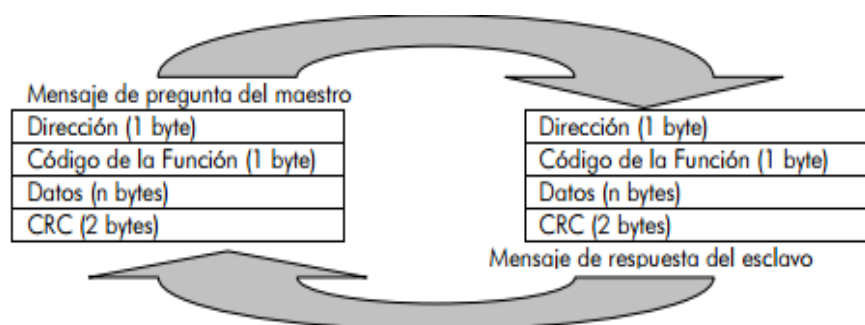
**Tabla 2-4** Correspondencia de registro MODBUS.

Función	Dirección MODBUS	Dirección en dispositivo	Dirección IEC61131	Tipo de registro	Tipo de acceso
16#01, 16#05, 16#0F	0 a 9.999	1 a 10.000	%M0, %M1...	Salidas o registros de aplicación digitales (bits)	Lectura y escritura
16#02	0 a 9.999	10.001 a 20.000	%I0, %I1...	Entradas digitales (bits)	Lectura
16#04	0 a 9.999	30.001 a 40.000	%IW0, %IW1...	Entradas analógicas (entero)	Lectura
16#03, 16#06, 16#10	0 a 9.999	40.001 a 50.000	%MW0, %MW1...	Registro general de la aplicación (entero)	Lectura y escritura

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (CANDELAS, 2011)

**2.4.3 Modbus RTU.** En la estructura RTU se permite únicamente un maestro y hasta un máximo de 247 esclavos. El inicio de toda la comunicación siempre la realiza el maestro, este envía una solicitud a uno de sus esclavos, y este responde la solicitud a su maestro, la estructura utilizada del mensaje del maestro y del esclavo es la misma: Dirección, Código de la Función, Datos y Suma de chequeo o Checksum. Solo el contenido de los datos posee tamaño variable. (WEG, 2012)



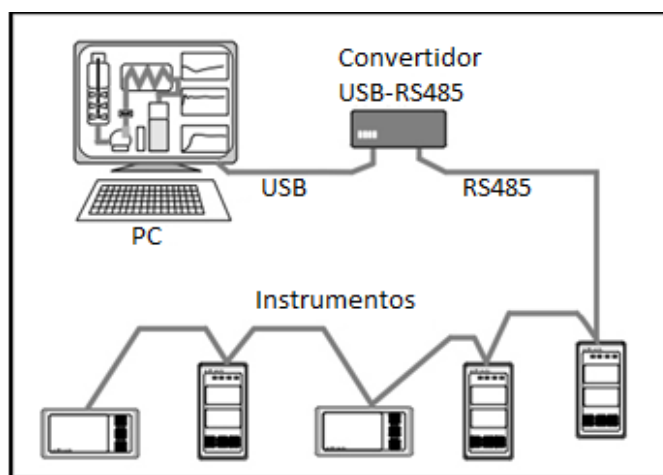
**Figura 2-4** Estructura de Mensajes del Protocolo MODBUS.  
Fuente: (WEG, 2012)

**2.4.3.1 Dirección.** “El maestro envía un byte con la dirección del esclavo. Al enviar la contestación el esclavo envía el telegrama con su propia dirección, esto con el fin de que el maestro sepa cuál es el esclavo que está enviando la respuesta”. (WEG, 2012)

**2.4.3.2 Código de la función.** “El código función está compuesto de un solo byte y es en este campo en donde el maestro especifica lo que solicita al esclavo (lectura, escritura, etc.)”. (WEG, 2012)

**2.4.3.3 Datos.** “Este es un campo con tamaño variable. El formato y el contenido de este campo dependen de la función utilizada. Este campo está descrito juntamente con la descripción de las funciones”. (WEG, 2012)

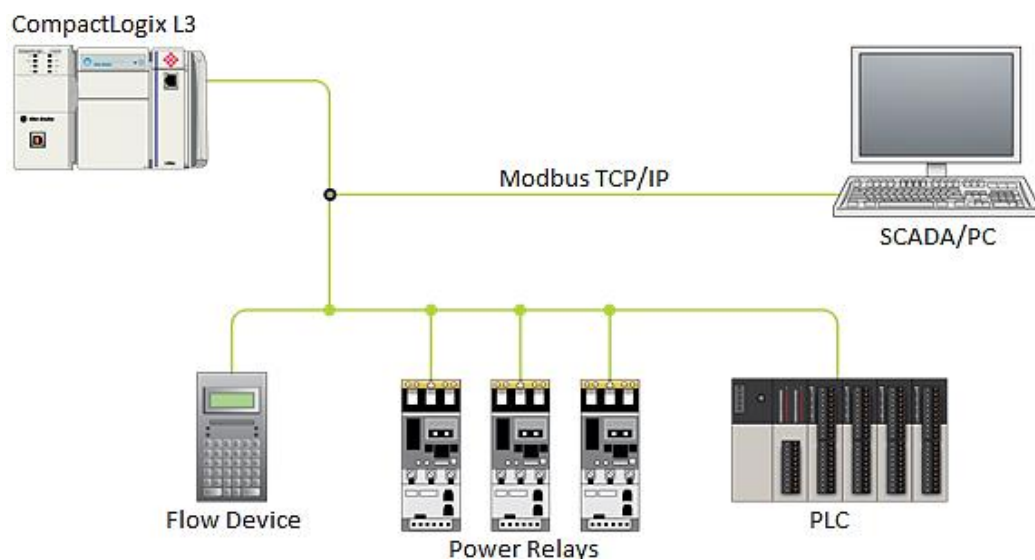
**2.4.3.4 CRC.** “Es un campo destinado al chequeo de errores de transmisión. Este campo está conformado por dos bytes, donde primero es transmitido el byte menos significativo (CRC-), y después el más significativo (CRC+)”. (WEG, 2012)



**Figura 2-5** Cableado comunicación MODBUS.  
Fuente: (ARIAN S.A., 2006)

**2.4.3.5 Cableado del bus RS-485.** El estándar RS-485 define un bus de transmisión de datos serie multipunto, este tipo de cableado es muy básico y se lo realiza con un par de hilos trenzados de cobre sobre el cual se transmite una señal diferencial para enviar los datos a largas distancias. Es este estándar de conexión un equipo puede enviar y recibir, pero no las dos a la misma vez y puede haber un equipo transmitiendo y varios equipos recibiendo la información. (ARIAN S.A., 2006)

**2.4.4 Modbus TCP.** Es un protocolo de comunicación simple y abierta, diseñado para la supervisión y control de procesos de producción, permite a los equipos de la industria comunicarse con el resto mediante una red de comunicación, estos equipos pueden ser PLCs, computadores, motores y otros dispositivos. En la trama MODBUS-TCP cada solicitud del maestro es tratada de forma independientemente por el esclavo, de esta forma se logra que las transferencias de datos no sufran rupturas debido al ruido en la red. En la versión Modbus-TCP todas las solicitudes y respuestas están diseñadas para verificar que el mensaje ha finalizado y se añade en la cabecera la longitud del mensaje, para conocer los límites del mismo. (OMROM Electronics SA, 2007)



**Figura 2-6** Estructura MODBUS TCP.  
Fuente: (PROSOFT, 2017)

**2.4.4.1 Ventajas del protocolo MODBUS-TCP.**

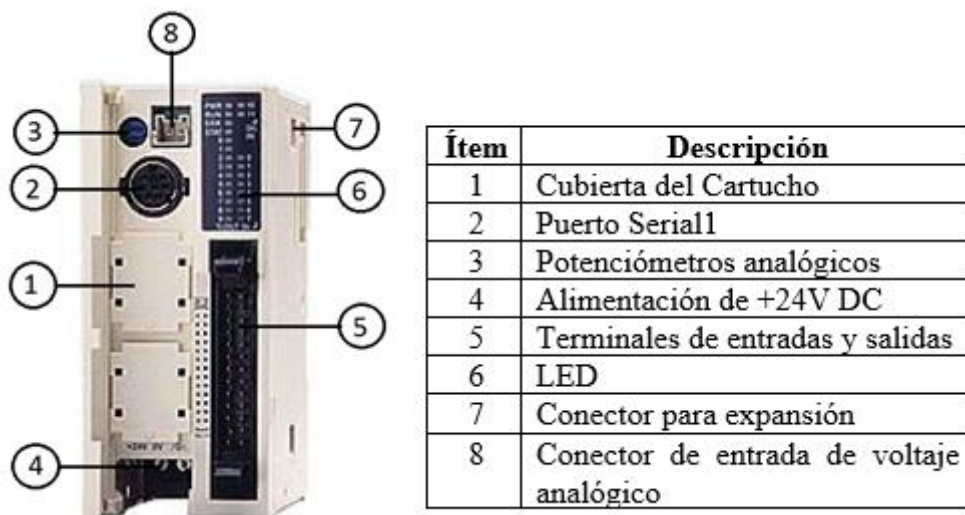
- Es un protocolo simple para administrar y expandir.
- Cualquier sistema con una batería de protocolos TCP/IP puede usar Modbus/TCP.
- Se puede usar para realizar la comunicación de una gran base de dispositivos MODBUS, usando productos de conversión. (OMROM Electronics SA, 2007)

## 2.5 PLC

Un controlador lógico programable o PLC por sus siglas en el idioma inglés, es un dispositivo electrónico que el usuario puede programar para ser utilizada en la automatización industrial, para el control de la maquinaria que realiza los procesos de fabricación en líneas de manufactura. Los PLC están diseñados con inmunidad al ruido eléctrico, resistencia a la vibración y al impacto. (SIEMENS, 2015)

**2.5.1 PLC modular TWDLMDA20DTK Twido.** Los PLCs Modulares no tienen una fuente de alimentación de AC integrada, sin embargo, el controlador puede ser alimentado con una tensión de red nominal de 24V-dc. (SCHNEIDER ELECTRIC, 2008).

### 2.5.1.1 Partes del PLC Twido Modular.



**Figura 2-7** Partes del PLC Twido Modular.  
Fuente: (SCHNEIDER ELECTRIC, 2008)

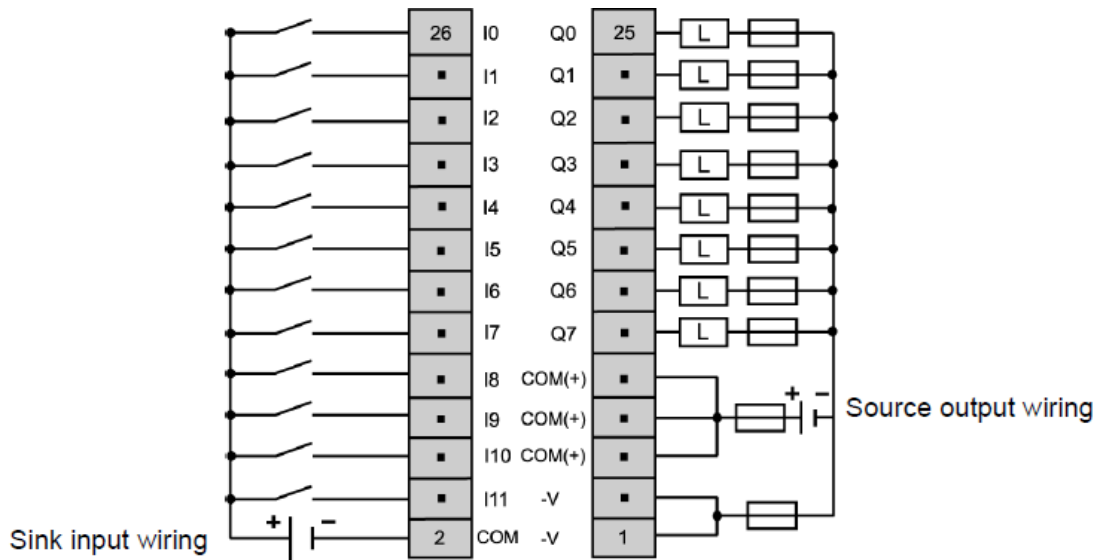
### 2.5.1.2 Características del PLC modular

**Tabla 2-5** Características del PLC Twido modular.

Especificación	Descripción	Especificación	Descripción
Serie del fabricante	TWD	Corriente de salida	0.3 A
Tensión de alimentación	24V DC	Tiempo de SCAN	1ms
Número de salidas	8 analógicos por transistor	Numero de puertos de comunicación	2
Puerto de comunicación	RS232 – RS485	Lenguaje de programación	Ladder Logic

**Realizado por:** Choco Edwin, Álvarez José. 2017  
**Fuente:** (SCHNEIDER ELECTRIC, 2008)

### 2.5.1.3 Esquema de conexiones del PLC TWDLMDA20DTK



**Figura 2-8** Esquema de conexiones del PLC Twido.  
**Fuente:** (SCHNEIDER ELECTRIC, 2008)

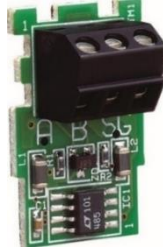
Los terminales COM (+) están conectados entre sí internamente, los terminales COM y COM (+) no están conectados entre sí internamente, los terminales -V están conectados entre sí internamente. Conecte un fusible adecuado para la carga. (SCHNEIDER Electric, 2008)

**2.5.2 Módulo de visualización TWDXCPODM.** Es un módulo de ampliación de controladores modulares Twido para la visualización y es en donde se encuentra el adaptador del segundo puerto serie. (SCHNEIDER ELECTRIC, 2008)



**Figura 2-9** Módulo de visualización TWDXCPODM.  
**Fuente:** (SCHNEIDER ELECTRIC, 2008)

**2.5.3 Adaptador de interfaz.** El Adaptador de interfaz TWDNAC485T es una tarjeta electrónica utilizada para la comunicación entre el PLC con la PC por medio del protocolo modbus. (SCHNEIDER ELECTRIC, 2008)



**Figura 2-10** Adaptador de interfaz TWDDNAC485T.  
Fuente: (SCHNEIDER ELECTRIC, 2008)

**2.5.4 Convertidor bidireccional USB-RS485.** Es un convertidor ampliamente usado en la automatización industrial, ya que permite la comunicación en ambos sentidos entre el puerto USB del computador y dispositivos que se comunican a través de RS 485. (SCHNEIDER ELECTRIC, 2008)



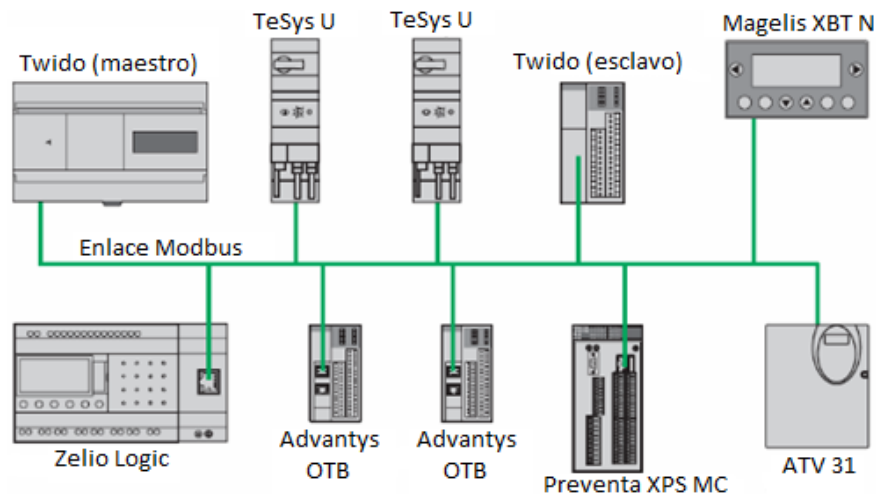
**Figura 2-11** Convertidor bidireccional USB-RS485.  
Fuente: (SCHNEIDER ELECTRIC, 2008)

**2.5.5 Bus de comunicación Modbus.** “El enlace serie Modbus permite responder a las arquitecturas maestro/esclavo (no obstante, es necesario comprobar que los servicios Modbus útiles para la aplicación se implanten en los equipos implicados)”. (SCHNEIDER ELECTRIC, 2008)

“El bus está constituido por una estación maestro y por estaciones esclavo. Solo la estación MAESTRO puede iniciar el intercambio (la comunicación directamente entre estaciones esclavo no es posible)”. (SCHNEIDER ELECTRIC, 2008)

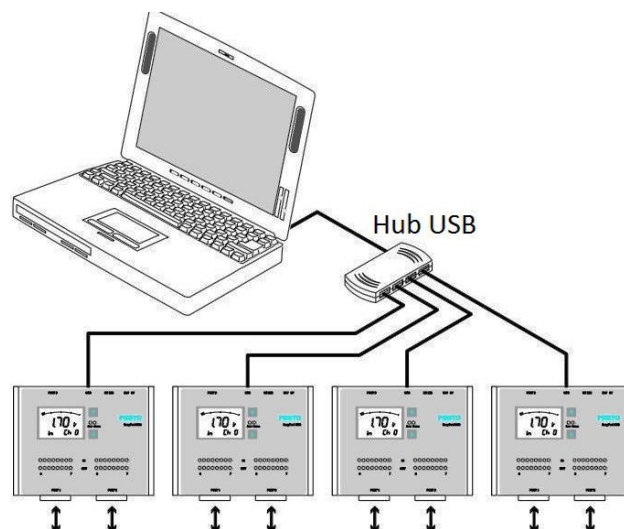
**Modo maestro de Modbus:** “El modo maestro de Modbus permite que el controlador pueda iniciar una transmisión de solicitudes Modbus, esperando una respuesta desde un esclavo Modbus”. (SCHNEIDER ELECTRIC, 2008)

**Modo esclavo de Modbus:** “El modo esclavo Modbus permite que el controlador pueda responder a las solicitudes de Modbus desde un maestro Modbus. Se trata del modo de comunicación predeterminado si no existe ninguna comunicación configurada”. (SCHNEIDER ELECTRIC, 2008)



**Figura 2-12** Esquema ejemplo de comunicación Modbus.  
**Fuente:** (SCHNEIDER ELECTRIC, 2008)

## 2.6 EasyPort



**Figura 2-13** Configuración con 4 módulos EasyPort USB.  
**Fuente:** (FESTO, 2008)

“La interface de procesos EasyPort USB se utiliza para la transmisión bidireccional de señales entre un proceso de control real (24 VDC) y un PC. Con el fin de excluir posibles interferencias en el PC, únicamente se utilizan interfaces galvánicamente separadas para realizar la transmisión de datos entre EasyPort USB y el PC”. (FESTO, 2008)



“En un distribuidor («Hub») de puertos USB conectado a un PC o en los puertos USB del propio PC, pueden conectarse como máximo cuatro módulos EasyPort USB. Deberá tenerse en cuenta que las direcciones de los módulos son diferentes”. (FESTO, 2008)

**Tabla 2-6** Características Técnicas del EasyPort USB

<b>Características EasyPort USB</b>	
Tensión de funcionamiento	24 VDC +/-10%
Consumo	3 VA
Cantidad de salidas	16 digitales de 24 VDC, 2 analógicas de 0...10 VDC ó -10...+10 VDC, resolución de 12 bit
Carga admisible	0,7 A por salida digital, 10 mA por salida analógica
Protección contra cortocircuitos	Sí
Cantidad de entradas	16 digitales de 24 VDC, 4 analógicas de 0...10 VDC ó -10...+10 VDC, resolución de 12 bit.
Umbral de conmutación de las entradas digitales	12 VDC
Histéresis de las entradas digitales	3 V
Interfaces de comunicación	RS232 y USB 2.0 con separación galvánica
Protocolo	ASCII, 115,2 kBaud, 8, N,1
Clase de protección	IP20
Temperatura ambiente admisible para funcionamiento/almacenamiento	0...55°C/0...70°C
Medidas (largo x ancho x alto x kg)	135 x 167 x 37 mm 0,65 kg.

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (FESTO, 2008)

**2.6.1 Interface SysLink para señales digitales.** “Las 8 entradas digitales y 8 salidas digitales del EasyPort USB están conectadas según IEEE 488 a un conector tipo zócalo de 24 contactos SysLink. Además, la entrada 0 en el puerto 1 o en el puerto 2 puede utilizarse como entrada de conteo rápido. Las entradas de conteo pueden funcionar con niveles de tensión desde 5 hasta 24 VDC”. (FESTO, 2008)

**Tabla 2-7** Datos técnicos de la interface SysLink para señales digitales Puerto 1 y 2.

Ítem	Contacto	Ítem	Contacto	Ítem	Contacto
OUTPUT 0	1	INPUT 0	13	0 VDC	11
OUTPUT 1	2	INPUT 1	14		12
OUTPUT 2	3	INPUT 2	15		23
OUTPUT 3	4	INPUT 3	16		24
OUTPUT 4	5	INPUT 4	17	24 VDC	9
OUTPUT 5	6	INPUT 5	18		10
OUTPUT 6	7	INPUT 6	19		21
OUTPUT 7	8	INPUT 7	20		22

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (FESTO, 2008)

**2.6.2 Conector Sub D para señales analógicas.** “Las 4 entradas y las 2 salidas analógicas del EasyPort USB están conectadas a un conector Sub D tipo zócalo de 15 contactos. La transformación analógica/digital se realiza con una resolución de 12 bit. La frecuencia de exploración es de 0,5 kHz”. (FESTO, 2008)

**Tabla 2-8** Datos técnicos del conector Sub D tipo zócalo, para señales analógicas.

<b>Puerto 3</b>	<b>Contacto</b>	<b>Puerto 3</b>	<b>Contacto</b>
OUTPUT 0	1	(Libre)	9
OUTPUT 1	2	(Libre)	10
0 V	3	+10VDC REF	11
(Libre)	4	(Libre)	12
(Libre)	5	(Libre)	13
0 V	6	INPUT 3	14
INPUT 1	7	INPUT 4	15
INPUT 2	8		

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (FESTO, 2008)

**2.6.3 Interfaces de datos para PC.** “El intercambio de datos con el PC se realiza a través del puerto serie RS232 o a través del puerto USB de EasyPort USB. La velocidad de la transmisión de datos es de 115,2 kBaud”. (FESTO, 2008)

**Tabla 2-9** Datos técnicos de la interface RS232.

<b>RS232</b>	<b>Contacto</b>	<b>RS232</b>	<b>Contacto</b>
(Libre)	1	(Libre)	6
Recibir datos	2	(Libre)	7
Enviar datos	3	(Libre)	8
(Libre)	4	(Libre)	9
Masa de señales GND	5		

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (FESTO, 2008)

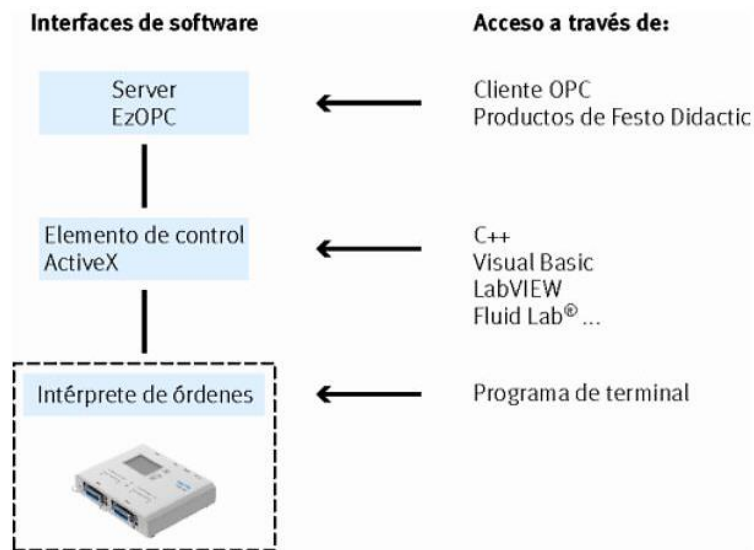
**2.6.4 Comunicación con EasyPort USB.** “La transmisión de datos desde y hacia una interface de procesos EasyPort USB se lleva a cabo a mediante órdenes de escritura y lectura individuales, atribuidas a determinadas direcciones”. (FESTO, 2008)

“Dependiendo cómo se incluye el EasyPort USB en la aplicación, existen diversos niveles de acceso para ejecutar esas órdenes de escritura/lectura”. (FESTO, 2008)

**Comunicación a través de EzOPC:** “OPC significa Open Connectivity Via Open Standards y se trata de una interface estándar utilizada en el sector de la automatización.

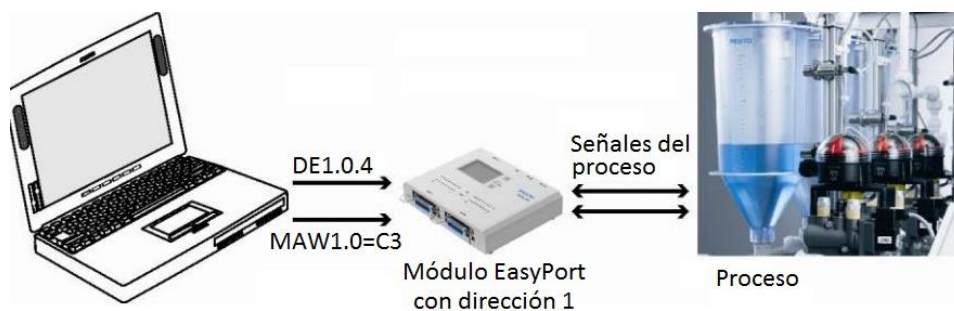
Esta interface garantiza un eficiente flujo de datos entre aplicaciones de Windows y aparatos de automatización”. (FESTO, 2008)

**Utilización del elemento de control ActiveX:** “Sirve de interface de programación para el EasyPort USB. Puede incluirse en el software como objeto COM. Los lenguajes de programación que soportan la interface COM son, entre otros, Visual Basic, C++ y, también, HTML. Pero también los sistemas de captación de datos de medición como LabVIEW y sistemas de visualización soportan la interface COM”. (FESTO, 2008)



**Figura 2-14** Interface de Software para EasyPort USB  
Fuente: (FESTO, 2008)

**Intérprete de órdenes:** “El intérprete para órdenes es el nivel de acceso inferior para la comunicación. El intérprete de órdenes conoce todas las órdenes que puede ejecutar el EasyPort USB. Mediante un programa de terminal como, por ejemplo, Hyper Terminal de Windows, es posible establecer una comunicación directa con el EasyPort USB y ejecutar las órdenes deseadas”. (FESTO, 2008)



**Figura 2-15** Órdenes de escritura y lectura.  
Fuente: (FESTO, 2008)

“Existen dos órdenes centrales para leer y escribir: Orden Display (D) para leer un elemento y Orden Modify (M) para modificar un elemento”. “Los elementos de un EasyPort USB pueden ser los siguientes: Entrada (E), Salida (A), Temporizador de operaciones (T) y Contador rápido (C)”. (FESTO, 2008)

**Tabla 2-10** Ejemplos de órdenes Modify y Display.

Dirección del módulo	Orden para el módulo	Respuesta del módulo	Observación
3	DAW3.0<CR>	AW3.0 = 1234<CR>	El valor de los datos es 1234 hexadecimal o 4660 decimal. Formato de palabra: xxxx
2	DEB2.0.1<CR>	EB2.0.1 = 12<CR>	El valor de los datos es 12 hexadecimal o 18 decimal. Formato de datos para bit: x
1	DE1.0.E<CR>	E1.0.E = 0<CR>	Lectura de bit de entrada 14. Formato de datos para bit: x
1	MAW1.0 = 201<CR>	AW1.0 = 201<CR>	El valor de los datos es 201 hexadecimal o 513 decimal.
3	MA3.0.A = 1<CR>	A3.0.A = 1<CR>	El bit de salida 10 se pone 1

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (FESTO, 2008)

## 2.7 Software de programación.

**2.7.1 TIA Portal.** Es un software innovador que permite de manera sencilla la programación de dispositivos pertenecientes a la marca SIEMENS, tales como PLCs y Pantallas HMI. Además, el software ofrece dos lenguajes de programación el KOP (bloques) y FUP (contactos). (CASTRO SILVA, y otros, 2017)

SIMATIC STEP 7 se ofrece en dos versiones que son compatibles con Windows 7 y 8.1:

- **STEP 7 Basic:** para programación de controladores SIMATIC S7-1200 y pantallas táctiles HMI.
- **STEP7 Professional:** Para la programación de controladores SIMATIC S7-1200, S7-300, S7-400, S7-1500 y WinAC. (SIEMENS, 2015)

**2.7.1.1 Librería Modbus TCP.** Un CPU que opere como servidor en la trama MODBUS-TCP podrá ser monitoreado por el maestro en la red, permitiendo la lectura de datos en la memoria del CPU. (CASTRO SILVA, y otros, 2017)

“**MB\_CLIENT** se comunica como cliente Modbus TCP a través del conector PROFINET del S7- 1200 CPU. No se requiere ningún módulo de hardware de comunicación adicional. Además, **MB\_CLIENTE** puede realizar una conexión cliente-servidor, enviar una solicitud de función Modbus, recibir una respuesta y controlar la desconexión desde un servidor Modbus TCP”. (CASTRO SILVA, y otros, 2017)

**Tabla 2-11** Bloque Instrucción **MB\_CLIENT**

LAD/FBD	SCL
<p style="text-align: center;"><b>%DB1</b> "MB_CLIENT_DB"</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;"><b>MB_CLIENT</b></p> </div>	<p><b>"MB_CLIENT_DB"</b>  REQ: _bool_in_,  DISCONNECT: _bool_in_,  CONNECT_ID: _uint_in_,  IP_OCTET_1: _byte_in_,  IP_OCTET_2: _byte_in_,  IP_OCTET_3: _byte_in_,  IP_OCTET_4: _byte_in_,  IP_PORT: _uint_in_,  MB_MODE: _usint_in_,  MB_DATA_ADDR: _udint_in_,  MB_DATA_LEN: _unit_in_,  DONE: _bool_out_,  BUSY: _bool_out_,  ERROR: _bool_out_,  STATUS: _word_out_,  MB_DATA_PTR: _variant_inout_.</p>

Fuente: (CASTRO SILVA, y otros, 2017)

“**MB\_SERVER** se comunica como servidor Modbus TCP a través del conector PROFINET de la CPU S7- 1200. Puede aceptar una petición para conectarse con el cliente Modbus TCP, recibir una petición de función Modbus y enviar un mensaje de respuesta”. (CASTRO SILVA, y otros, 2017)

**Tabla 2-12** Bloque de Instrucción **MB\_SERVER**

LAD/FBD	SCL
<p style="text-align: center;"><b>%DB2</b> "MB_SERVER_DB"</p> <div style="border: 1px solid black; padding: 5px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;"><b>MB_SERVER</b></p> </div>	<p><b>"MB_SERVER_DB"</b>  DISCONNECT: _bool_in_,  CONNECT_ID: _uint_in_,  IP_PORT: _uint_in_,  NDR: _bool_out_,  DR: _bool_out_,  ERROR: _bool_out_,  STATUS: _word_out_,  MB_HOLD_REG: variant_inout_.</p>

Fuente: (CASTRO SILVA, y otros, 2017)

**2.7.2 TwidoSuite.** Es un software utilizado para la programación de los dispositivos de Telemecanique, posee una navegación rápida y sencilla para el usuario gracias a su entorno de programación gráfica. Además, ofrece dos lenguajes de programación GRAFCET y FUP (diagramas de contactos). (CASTRO SILVA, y otros, 2017)

**2.7.3 LabVIEW.** Es un software innovador para la adquisición de datos para el control y análisis. Además, existe una mayor facilidad de programación en este software con respecto a otros sistemas, ya que en LabVIEW se utiliza el lenguaje de programación G (Gráfica) a diferencia del resto que basan su programación en líneas de texto. LabVIEW para la programación presenta dos pantallas; la pantalla de Frontal y la pantalla de Diagrama bloques. (ALBÁN GUERRERO, y otros, 2017)

*Pantalla Frontal:* Es la pantalla que el usuario visualiza cuando el programa se ejecuta. Se realiza todas las maniobras de control y de visualización de datos.

*Pantalla de diagrama de Bloques:* Esta pantalla se visualiza únicamente cuando se está realizando la programación, ya que esta pantalla se realiza toda la programación gráfica.

“NI ofrece tres mecanismos principales para conectar dispositivo Modbus: (1) un servidor OPC de alto nivel, (2) un servidor de E/S Modbus y (3) un API de Modbus introducido en NI LabVIEW 2014 a través de los módulos LabVIEW Real-Time o LabVIEW Datalogging and Supervisory Control (DSC)”. (NATIONAL INSTRUMENTS, 2014)

**2.7.3.1 API de LabVIEW Modbus.** “El API de Modbus de bajo nivel es la opción preferida cuando su aplicación necesita un alto nivel de control en las secuencias y temporización de las solicitudes de Modbus. El API de bajo nivel también es generalmente la elección preferida cuando la flexibilidad es lo más importante. Sin embargo, la flexibilidad y la potencia ofrecida por el API de LabVIEW Modbus también significa que su código de aplicación debe ser más complejo para administrar correctamente el API.” (NATIONAL INSTRUMENTS, 2014)

**2.7.3.2 NI-OPC.** OLE for Process Control u OPC por sus siglas en ingles es un módulo que permite la transferencia de datos en tiempo real desde los equipos de control hacia los HMI y viceversa. Los OPC se encuentran disponibles para todo tipo de controladores lógicos programables. (ALBÁN GUERRERO, y otros, 2017)

“Para aplicaciones complicadas que involucran varios dispositivos esclavos que se comunican a través de diferentes protocolos, las E/S Modbus estándares podrían no ser suficientes. Una solución común es usar un servidor OPC, el cual actúa como un compilador de datos para todos sus sistemas y después usar los servidores de E/S OPC incluidos en el Módulo LabVIEW DSC para comunicarse con ese servidor OPC.” (NATIONAL INSTRUMENTS, 2014)

**2.7.3.3 Datalogging and Supervisory Control.** DSC por sus siglas en el idioma inglés, es un módulo de LabVIEW que se utiliza para personalizar los indicadores y controles del interfaz hombre-máquina (HMI). (ALBÁN GUERRERO, y otros, 2017)

**2.7.3.4 Uso de servidores Modbus E/S.** A través de DSC o módulo Real-Time. “Modbus es un protocolo de mensajería de nivel de aplicación que proporciona la comunicación cliente-servidor entre los dispositivos conectados en diferentes tipos de buses o redes. Puede crear un servidor Modbus o Modbus esclavo de E/S para leer o escribir datos a los dispositivos Modbus. Por ejemplo, puede crear un servidor de E/S Modbus Slave en un controlador en tiempo real de National Instruments y el uso de este controlador como un dispositivo esclavo Modbus. También puede crear un servidor de E/S Modbus en un equipo host y utilizar este servidor como un dispositivo maestro Modbus para comunicarse con un dispositivo esclavo”. (NATIONAL INSTRUMENTS, 2014)

Los elementos de datos del servidor de Modbus y Modbus Slave E/S, usan las siguientes denotaciones:

**D** denota los de primer un entero sin signo de 32 bits.

**F** denota los de primer un número de coma flotante de 32 bits. **S** denota los de primer un entero con signo de 16 bits.

**L** denota los de primer de la longitud de una matriz. El máximo de puntos de datos por cada comando de valores que especifique en la configuración de atributos avanzados cuadro de diálogo determinan la longitud máxima de la matriz. Si los elementos de la matriz son números enteros de 32 bits o números de punto flotante, la longitud máxima de la matriz es la mitad del valor que especifique para el correspondiente máximo de puntos de datos por cada comando de valor.

**SD** denota los de primer un entero con signo de 32 bits.

**A** denota los de primer una matriz. (NATIONAL INSTRUMENTS, 2014).

**Tabla 2-13** Datos que soporta Modbus (resumen).

Ítem Data	Tipo de datos	Modbus		Esclavo Modbus		Descripción	Ejemplo
		Leer	Escribir	Leer	Escribir		
000001-065535	valor booleano	Si	Si	Si	Si	Accede a las bobinas de un solo bit.	000.001 = {000001}
100001-165535	valor booleano	Si	No	Si	Si	Accede a las entradas discretas de un solo bit.	100.002 = {100002}
300.001,1-365535,16	valor booleano	Si	No	Si	Si	Accede a bits individuales de registros de entrada y los interpreta valores VERDADERO o FALSO como lógicas.	300.001,1 = {el primer bit de 300001}
300001-365535	16 bits entero sin signo	Si	No	Si	Si	Accede a la entrada de 16 bits sin signo registra números enteros que van desde 0 a 65.535.	300.001 = {300001}
400.001,1-465.535,16	valor booleano	Si	No	Si	Si	Accede a bits individuales de registros de retención e interpreta valores VERDADERO o FALSO como lógicas. El bit 1 es el menos significativo. El bit 16 es el más significativo.	400.002,16 = {el bit 16 de 400002}
400001-465535	16 bits entero sin signo	Si	Si	Si	Si	Accede a registros de retención de 16 bits como enteros sin signo que van desde 0 a 65.535	400.002 = {400002}

**Realizado por:** Choco Edwin, Álvarez José. 2017

**Fuente:** (NATIONAL INSTRUMENTS, 2014)

**2.7.3.5 Cumpliendo con las necesidades de su aplicación.** “Para comunicación por Modbus, NI ofrece tres opciones principales que brindan un amplio rango de funcionalidad. Primero, un API de bajo nivel ofrece control fino del protocolo, con alto rendimiento, a costa de la facilidad de uso. Los servidores de E/S Modbus son más sencillos y más fáciles para tener acceso o proporcionar datos Modbus, pero disminuyen el control del protocolo que puede ser necesario para algunas aplicaciones. Finalmente, para sistemas grandes y complejos, puede ser beneficioso considerar un servidor OPC con todas las funciones para servir como un compilador de datos. Después, simplemente use una herramienta como un controlador LabVIEW OPC UA o servidores de E/S OPC para que su aplicación tenga acceso a estos datos.” (NATIONAL INSTRUMENTS, 2014)

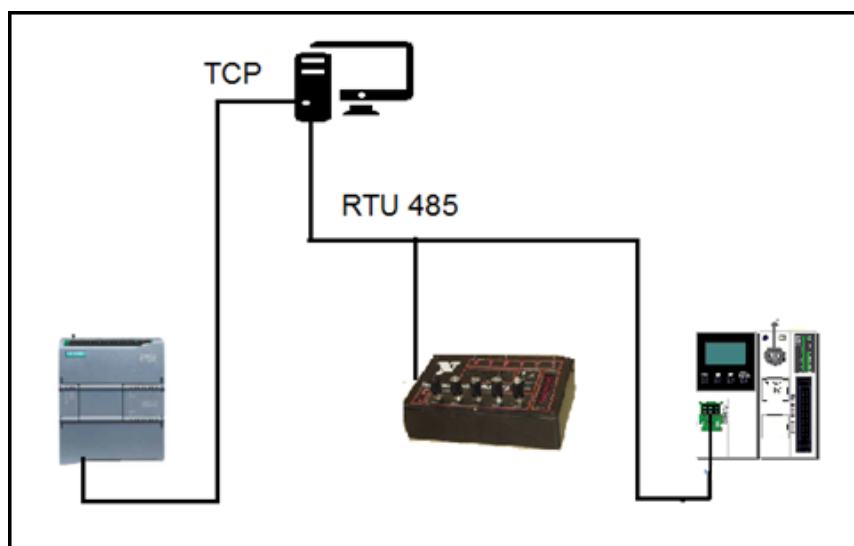


## CAPITULO III

### 3. DISEÑO Y CONSTRUCCIÓN.

#### 3.1. Parámetros de diseño del módulo.

Para el diseño del módulo es imprescindible que los dispositivos puedan comunicarse a través de cualquier formato Modbus, dichos elementos se ilustran en la Figura 3-1 y se describen brevemente a continuación:



**Figura 3-1** Hardware de la red MODBUS TCP/RTU.

**Fuente:** (Autores, 2017).

**Autómata Siemens:** el dispositivo puede comunicarse a través de MODBUS TCP, consta de 8 entradas digitales (24 VCC) y 1 entrada analógica (0 - 10 VCC) y 6 salidas por relé.

**Autómata Twido:** esta gama de controlador programable permite conexión MODBUS RS232 y 485, para nuestra aplicación tenemos una base modular compacta con HMI y con un adaptador serial 485 MODBUS, el número de entradas y salidas utilizables son 12 entradas de 24 VCC Y 8 salidas (0.3 A).

**Módulo MODBUS con entradas y salidas lógicas:** este dispositivo se puede comunicar a través de MODBUS RTU 485, consta de 6 interruptores para sus entradas digitales, y como entradas analógicas posee 4 potenciómetros, 1 sensor de temperatura (LM35) y con

1 sensor de luz (LDR), estos elementos registrarán valores de voltaje comprendidos entre 0 y 5 VCD, y en sus salidas cuenta con 6 LED como salidas digitales y una matriz LED simulado una salida analógica.

### **3.2. Diseño del módulo.**

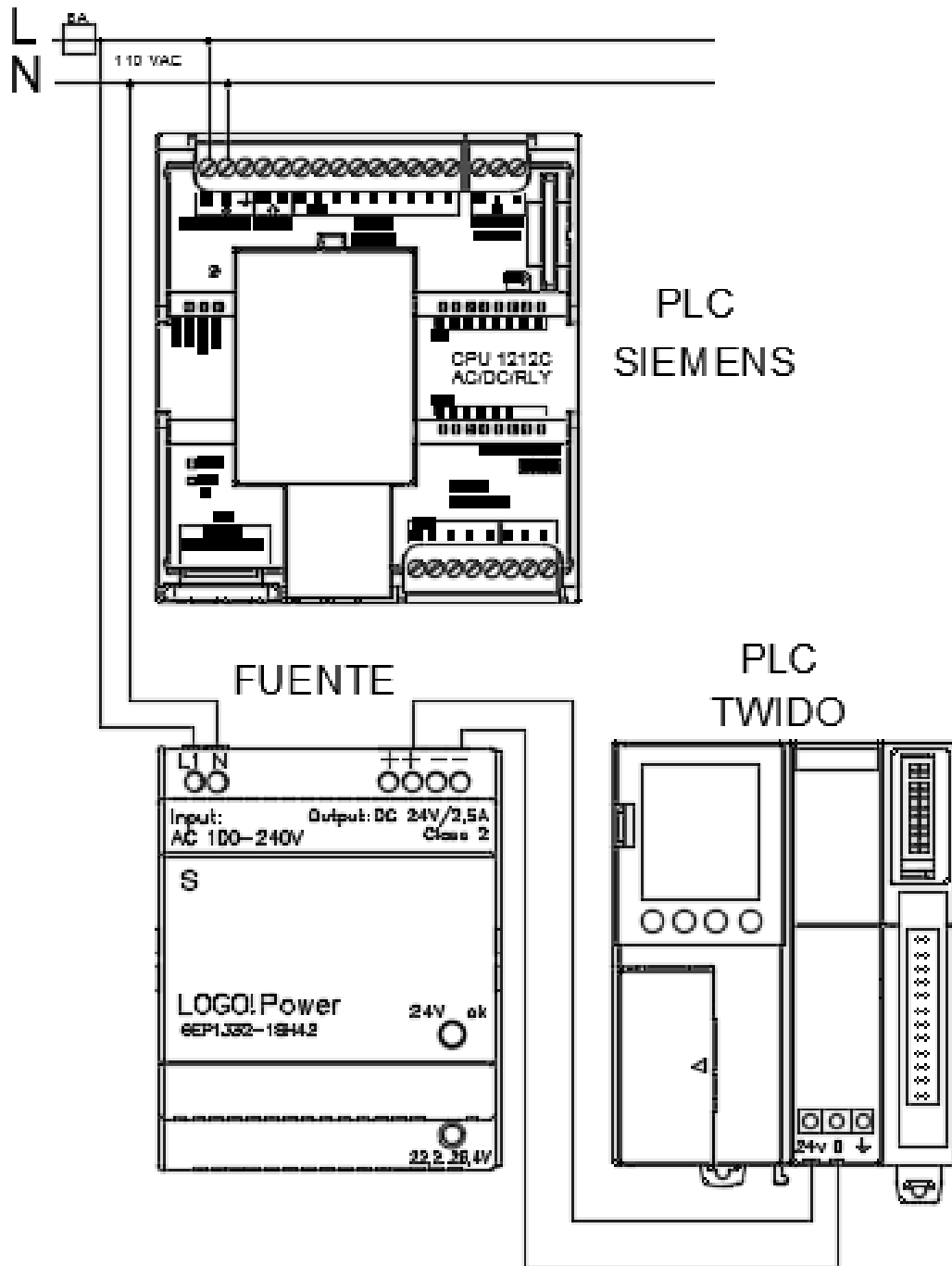
**3.2.1 Módulo con PLC Twido y PLC Siemens.** El módulo constará de una caja eléctrica de plástico para control, en donde se fijará la fuente, las protecciones, los equipos de conexión y los PLC; todos serán colocados sobre rieles de montaje. La alimentación eléctrica de los equipos se realizará con cable AWG #16, para la transmisión de datos a través de MODBUS 485 se utiliza dos cables AWG #18 y cable de red UTP (4 pares de hilos) con terminales RJ45 para MODBUS TCP. La alimentación de este módulo se realizará a 110 VAC.

**3.2.2 Módulo I/O.** El módulo constará de un panel plástico diseñado a medida, donde se fija el microcontrolador Arduino, que se encuentra conectado a la placa electrónica que posee potenciómetros, interruptores, leds, y sensores de temperatura y luz, en la parte lateral de la caja se colocarán los puertos de conexión Modbus TCP y RTU 485, la alimentación de este equipo es 5 VCD provenientes de una fuente externa a la caja.

### **3.3. Construcción del módulo.**

**3.3.1 Carcasa.** La carcasa del módulo de los PLC es un panel plástico comúnmente usado para montaje de equipos eléctricos. La selección de este panel de dimensiones 30 x 40 x 18 cm se realizó en base al espacio necesario calculado según las dimensiones de los equipos y el espacio requerido para realizar sus respectivas conexiones. Para el caso del Módulo I/O la carcasa se realizó a través de impresión 3D en material plástico.

**3.3.2 Montaje de equipos.** El montaje de los equipos se realizó sobre rieles DIN fijados al panel plástico y todos los cables de conexión tanto de alimentación como de transferencia de datos se colocaron dentro de canaletas eléctricas. Se realizaron perforaciones en la parte lateral del panel para el puerto de la alimentación eléctrica del módulo y para los puertos de conexión MODBUS.

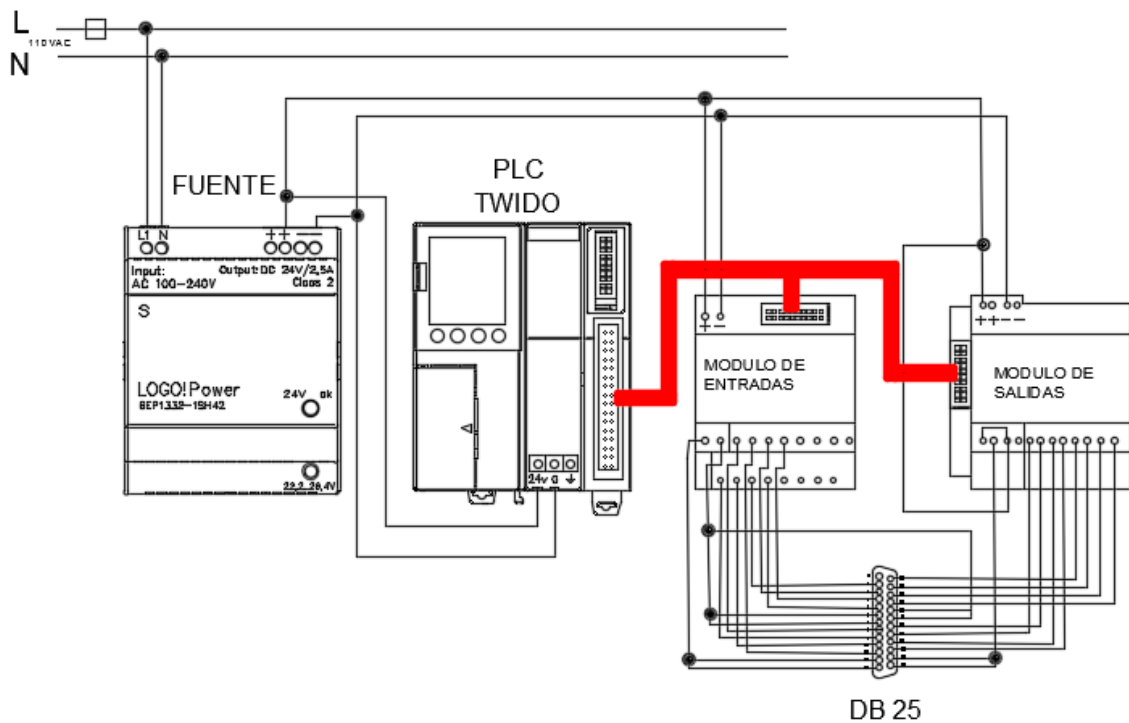


**Figura 3-2** Diagrama conexión eléctrica de alimentación módulo PLC.  
**Fuente:** (Autores, 2017)

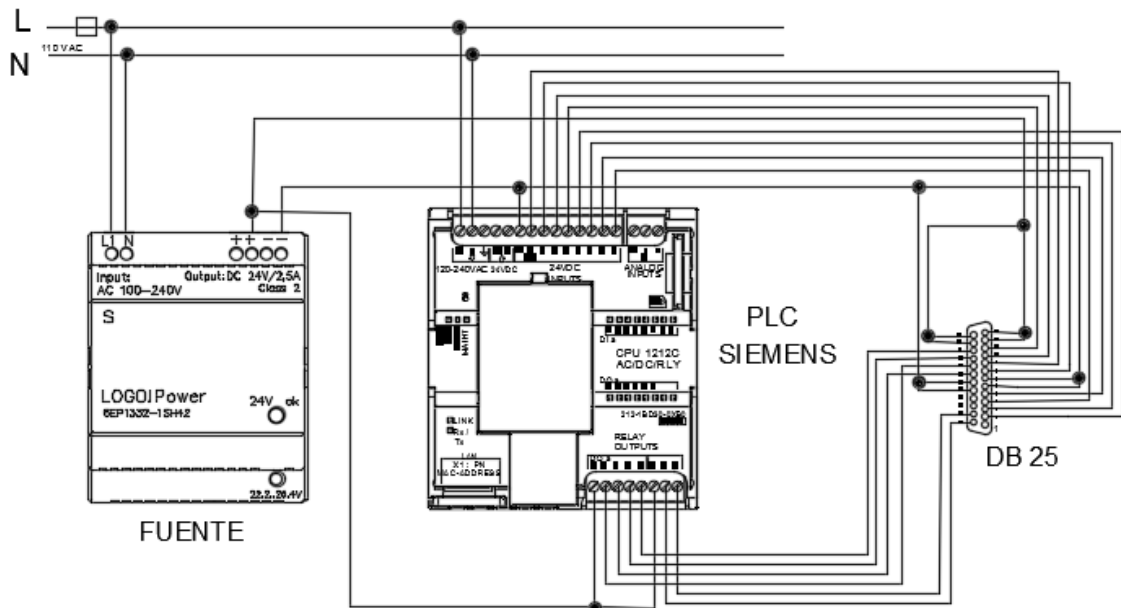
**3.3.3 Cableado de equipos.** En la Figura 3-3 se muestra la conexión de los equipos y sus cables, los cuales se encuentran debidamente protegidos y fijados para evitar su desconexión accidental o problemas de conexionado. En todos los cables han sido colocados terminales y las entradas/salidas de los PLC han sido llevados a terminales DB25 para facilitar su conexión hacia los módulos de procesos existentes en el laboratorio de control. Todo el módulo se encuentra protegido por un breaker de 6 A que fue dimensionado en base al consumo de los equipos.



**Figura 3-3** Conexión de equipos módulo.  
Fuente: (Autores, 2017)



**Figura 3-4** Diagrama eléctrico PLC TWIDO.  
Fuente: (Autores, 2017)



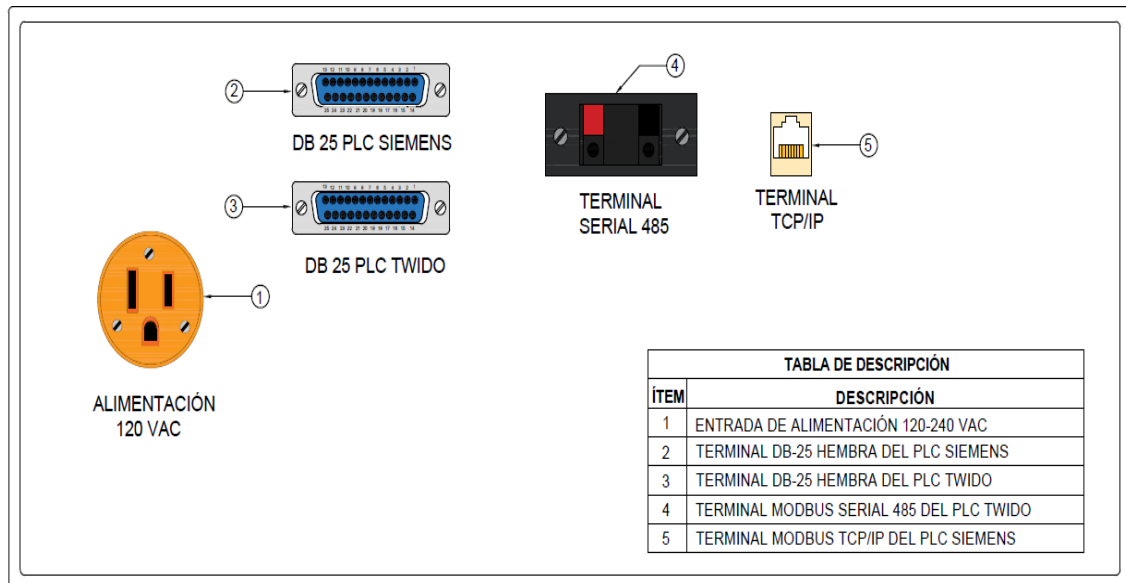
**Figura 3-5** Diagrama eléctrico PLC Siemens.  
**Fuente:** (Autores, 2017)

La conexión de los pines de los puertos DB25 con los PLC se realizó en base a la conexión de los DB25 colocados en los módulos de procesos del laboratorio y se detalla a continuación:

**Tabla 3-1** Conexión entradas/salidas DB-25.

DB 25	Descripción PLC	DB 25	Descripción PLC	Puerto DB25
1	-	14	Q0.4	
2	I0.4	15	Q0.5	
3	I0.5	16	Q0.6	
4	I0.7	17	Q0.7	
5	I0.6	18	0 VA	
6	0 VB	19	0 VA	
7	0 VB	20	Q0.1	
8	I0.1	21	Q0.0	
9	I0.0	22	Q0.2	
10	I0.2	23	Q0.3	
11	I0.3	24	24 VA	
12	24 VB	25	24 VA	
13	24 VB			

**Fuente:** (Autores, 2017)



**Figura 3-6** Vista lateral del módulo MODBUS.  
**Fuente:** (Autores, 2017)

### 3.4. Configuración de la comunicación Modbus.

**3.4.1 Modbus desde Software LabVIEW.** Este programa ofrece tres herramientas principales para la comunicación a través de Modbus.

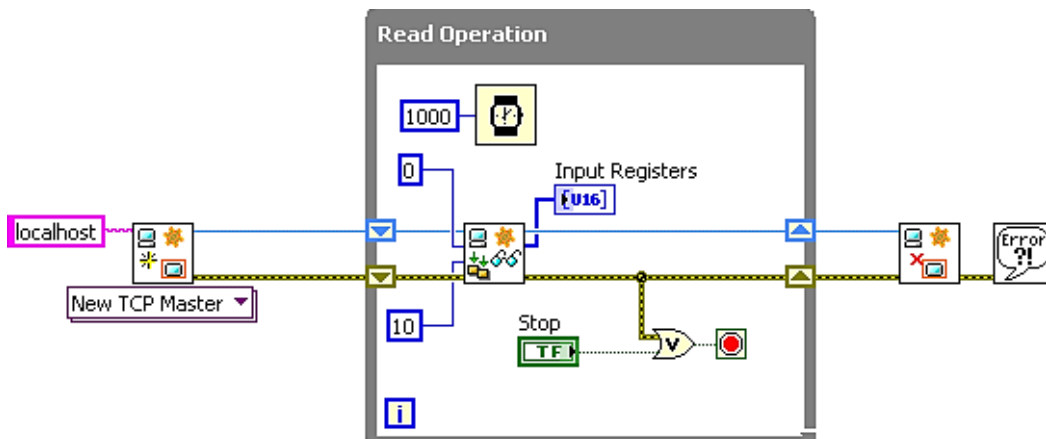
En el nivel más bajo de abstracción, se encuentran disponibles una variedad de bibliotecas y librerías de API (Application Programming Interface) diferentes. Estas bibliotecas brindan por lo general soporte para maestro y esclavo.

Los módulos NI LabVIEW DSC y LabVIEW Real-Time ofertan la herramienta Modbus I/O Server, una biblioteca de bajo nivel, pero con alta flexibilidad y facilidad de uso. Estos servidores de E/S también brindan soporte para maestro y esclavo.

Finalmente, NI OPC Servers es un servidor OPC completamente funcional que puede incluir soporte de controlador para maestros y esclavos Modbus.

**3.4.1.1 Modbus utilizando controladores API de bajo nivel.** Existen muchos controladores de bajo nivel en varios lenguajes para Modbus y que pueden complementar las características de NI OPC Servers y Modbus I/O Servers. “Con estos controladores, usted puede definir de manera explícita lo que sucede cuando una solicitud de Modbus es enviada o recibida”. Además, muchas veces ofertan un rendimiento mayor que los

controladores con más alto nivel, pero su configuración es un poco más complicada ya que se debe escribir una gran cantidad de códigos de procesamiento de datos para que la aplicación interactúe eficazmente con otros dispositivos del sistema. La funcionalidad y el comportamiento de estos códigos variaran dependiendo si el dispositivo a comunicarse es un maestro o un esclavo.



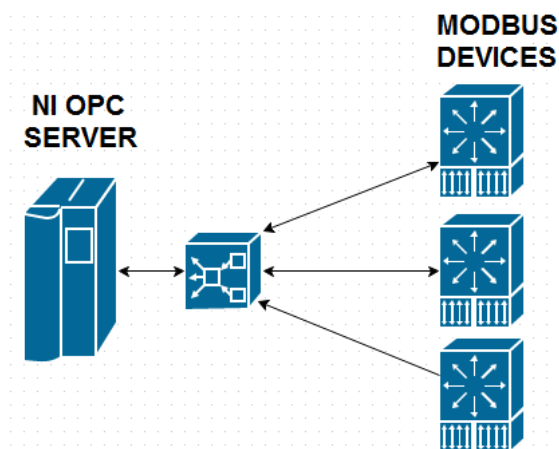
**Figura 3-7** Maestro de bajo nivel controlando temporización y una solicitud leída.  
**Fuente:** (NATIONAL INSTRUMENTS, 2014)

La Figura 3-7 muestra cómo mediante el API de LabVIEW para Modbus un maestro de bajo nivel ofrece el control total de las solicitudes específicas enviadas y la temporización de dichas solicitudes.

**3.4.1.2 Modbus utilizando I/O server.** “Los Modbus I/O servers incluidos en LabVIEW ofrecen una opción intermedia entre la variedad de servidores OPC fáciles de usar y las potentes bibliotecas de bajo nivel que brindan control completo del protocolo”. Los I/O servers permiten una interfaz simple basada en configuración Modbus para comunicarse con un dispositivo e incluyen su propio sistema para agendar solicitudes, lo cual resuelve varios de los problemas de los controladores de bajo nivel en Modbus, como el endianness ("extremidad") que designa el formato en el que se almacenan los datos de más de un byte en un ordenador.

Los servidores de E/S ofrecen una interfaz simple en LabVIEW y brindan a los programas la habilidad de tener un acceso rápido y fácil a datos procesados y sin procesar. Generalmente son usados en aplicaciones donde es muy importante un control supervisorio, mientras que los microcontroladores distribuidos son responsables del control de alta velocidad en el sistema.

**3.4.1.3 Modbus utilizando OPC servers.** “NI OPC Servers es una aplicación de servidor OPC autónoma y con gran funcionalidad que puede ser la base de un sistema SCADA. Como todos los servidores OPC anteriores al lanzamiento de OPC UA, NI OPC Servers es un programa únicamente para Windows y como tal es más adecuado para usarse como un sistema supervisorio que como un sistema de control de alta velocidad de los dispositivos esclavos. NI OPC Servers ofrece un extenso conjunto de controladores que permiten comunicación no solamente con dispositivos Modbus, sino también con dispositivos que utilizan una gran variedad de protocolos específicos de cada proveedor o protocolos estándares como OPC UA.” (NATIONAL INSTRUMENTS, 2014).



**Figura 3-8** Dispositivos Modbus TCP/IP y un servidor NI OPC Servers como host.  
**Fuente:** (NATIONAL INSTRUMENTS, 2014)

Como muchos servidores OPC, NI OPC Servers tiene herramientas para manejar grandes tipos de datos, configuración de ‘endianness’ por dispositivo, funciones para agendar solicitudes y muchas otras características para abstraer algunos detalles de bajo nivel de Modbus. Una vez que los datos están en la aplicación, pueden ser transmitidos usando OPC u OPC UA hacia registradores, interfaces humano-máquina (HMIs) y otras aplicaciones de Windows. (NATIONAL INSTRUMENTS, 2014).

**3.4.2 Pasos para la configuración Modbus de distintos dispositivos.** La comunicación de los dispositivos del módulo se realizará a través de una configuración con los I/O server para Modbus TCP y mediante la librería API NI Modbus para Modbus serial RTU 485.

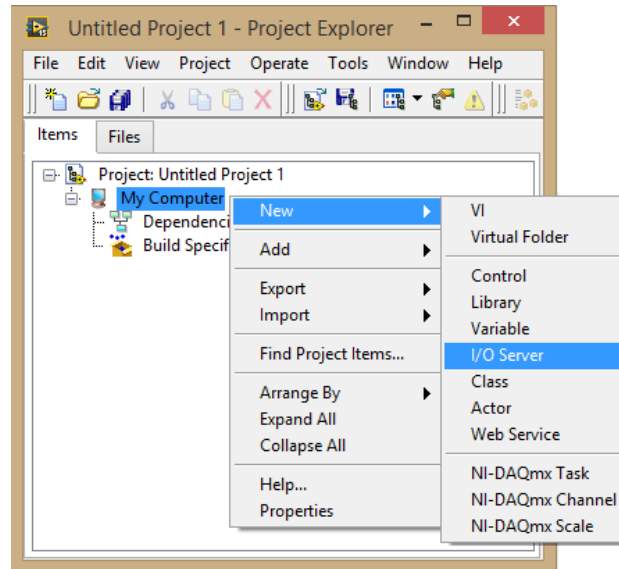
**3.4.2.1 Modbus TCP con I/O SERVER de LabVIEW.** Para la configuración de Modbus en LabVIEW utilizando un servidor E/S Modbus se requiere tener instalada la versión NI LabVIEW Full Development System y el módulo LabVIEW DSC.



### *Pasos para la creación de un servidor Modbus E/S de forma interactiva*

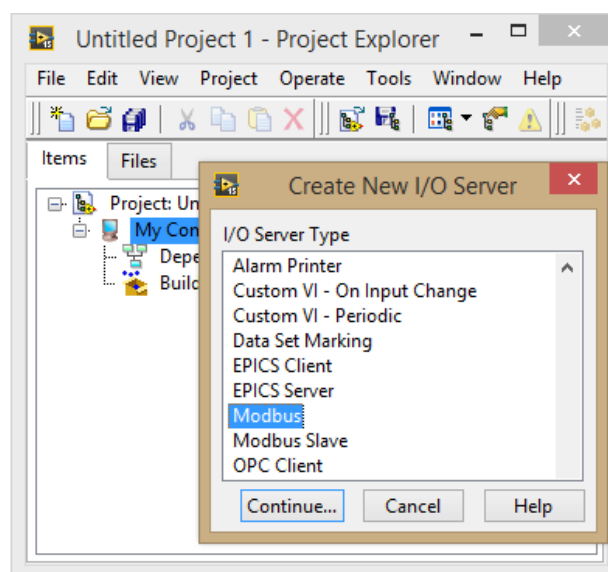
**Paso 1.-** En la ventana Introdutora de LabVIEW, haga clic en Archivo» Nuevo proyecto.

**Paso 2.-** En la ventana explorador de proyectos, haga clic derecho en el objeto mi PC y seleccione Nuevo» E/S del servidor en el menú contextual.



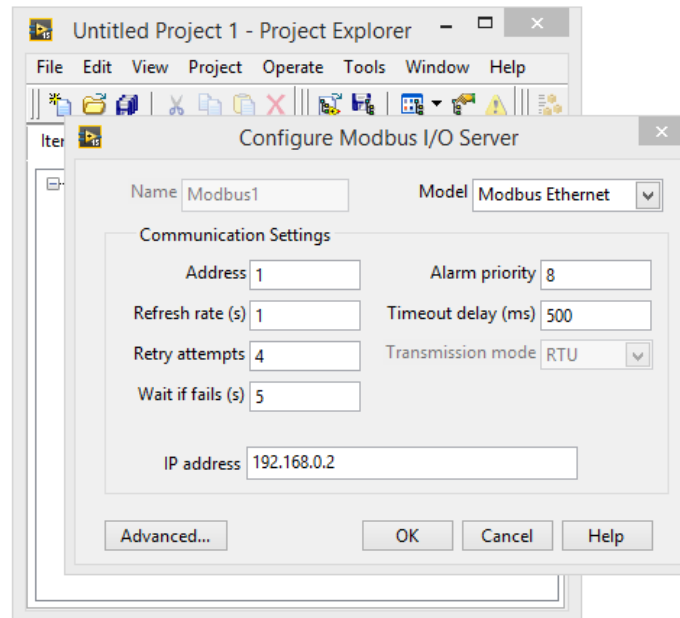
**Figura 3-9** Creación nuevo I/O server.  
Fuente: (Autores, 2017)

**Paso 3.-** En el cuadro de dialogo Crear nuevo servidor de E/S, seleccione Modbus y haga clic en el botón continuar.



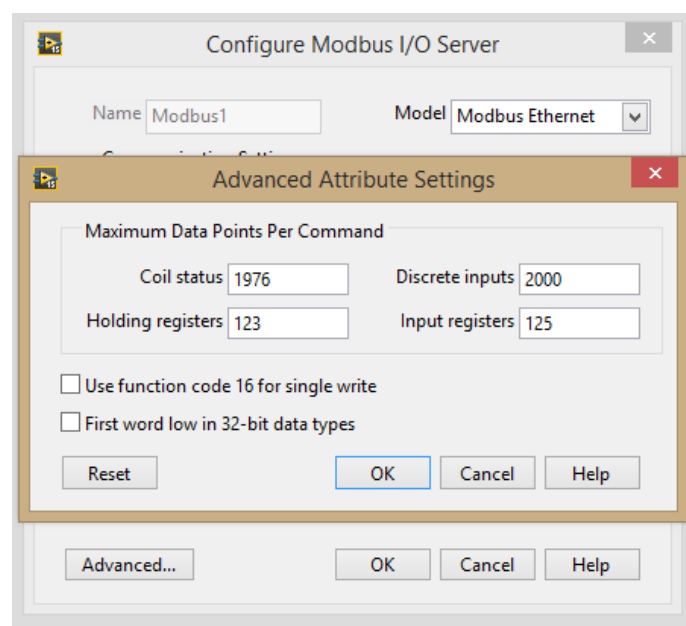
**Figura 3-10** Creación nuevo I/O server Modbus.  
Fuente: (Autores, 2017).

**Paso 4.-** En el cuadro de dialogo Configurar Servidor E/S Modbus seleccione el modelo de comunicación Modbus Ethernet, y llene los campos con los ajustes específicos correspondientes al dispositivo a supervisar. En este paso deberá ingresar la dirección del esclavo o maestro Modbus y la dirección IP del mismo.



**Figura 3-11** Configuración parámetros Modbus I/O server.  
**Fuente:** (Autores, 2017)

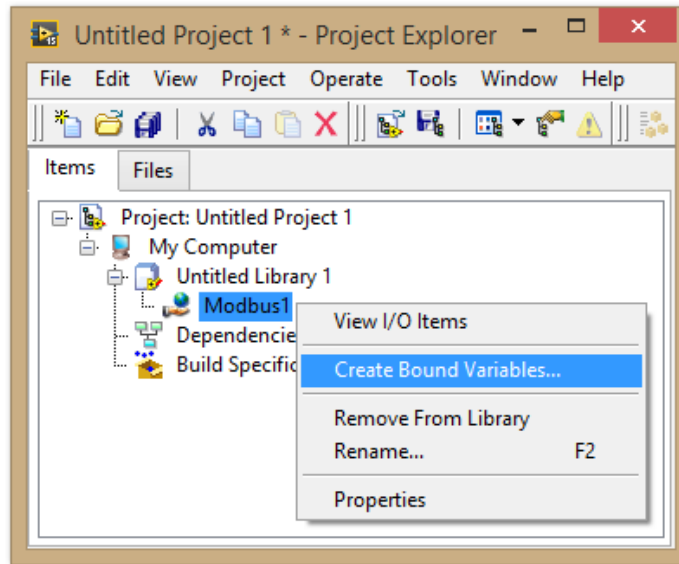
**Paso 4.1.- (Opcional)** Haga clic en el botón Avanzado del cuadro de dialogo Configurar Modbus E/S servidor para configurar el máximo de datos por comando.



**Figura 3-12** Parámetros de configuración avanzada Modbus.  
**Fuente:** (Autores, 2017).

**Paso 5.-** Haga clic en el botón OK, la librería se agregará automáticamente a su proyecto.

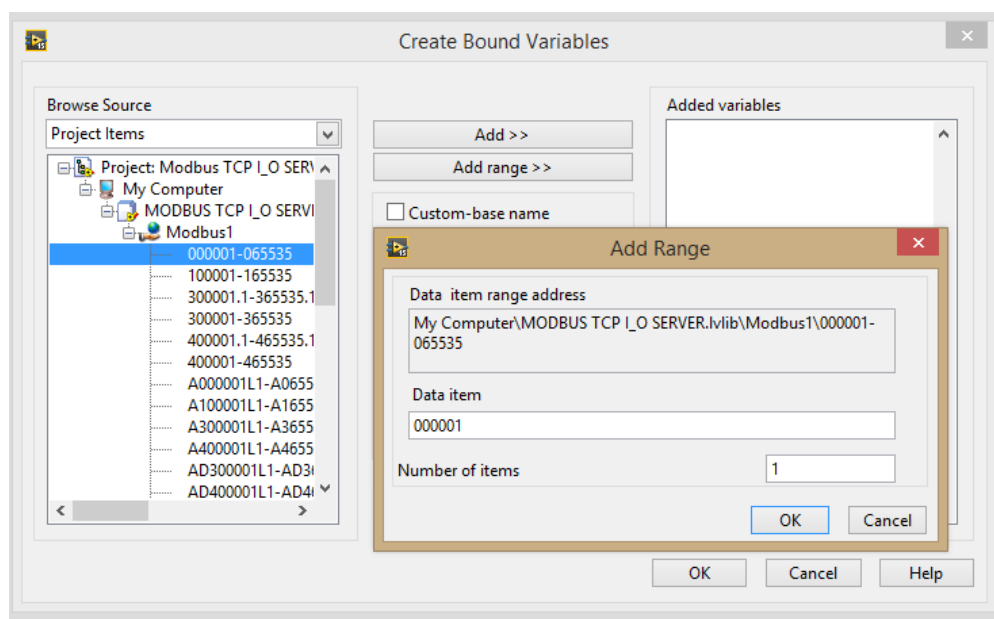
**Paso 6.-** Ahora de clic derecho en la librería agregada y seleccione “Create Bound Variables”.



**Figura 3-13** Creación Bound Variables.

Fuente: (Autores, 2017)

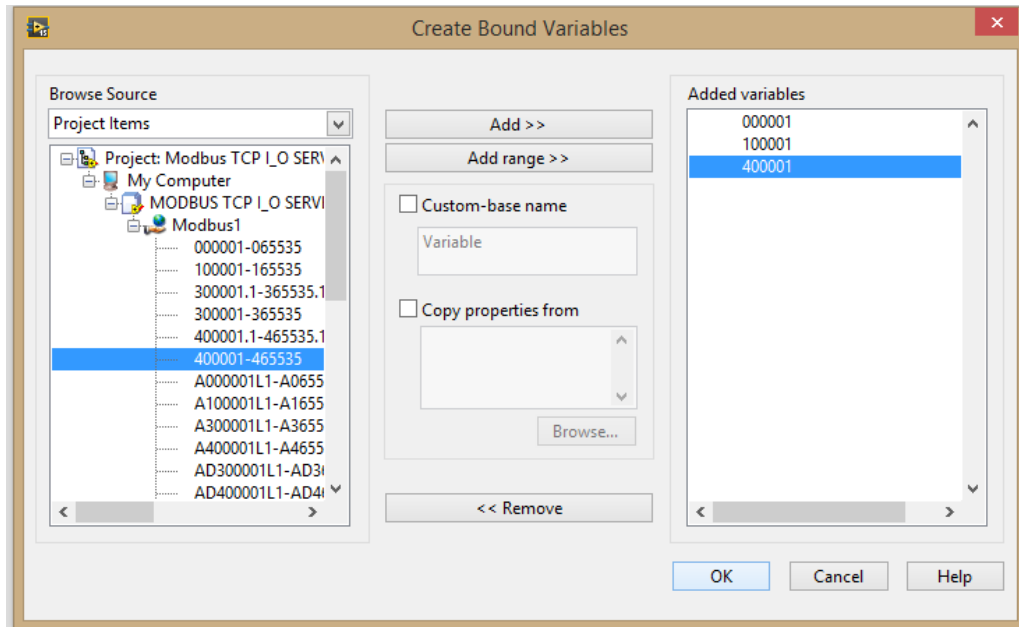
**Paso 7.-** En el menú Create Bound Variables seleccione el tipo de dato deseado y agregue el rango de los mismos que desee leer, escribir o leer/escribir. Las características detalladas del tipo de datos soportados por Modbus se encuentran en la Tabla 13.



**Figura 3-14** Agregar rango de variables.

Fuente: (Autores, 2017)

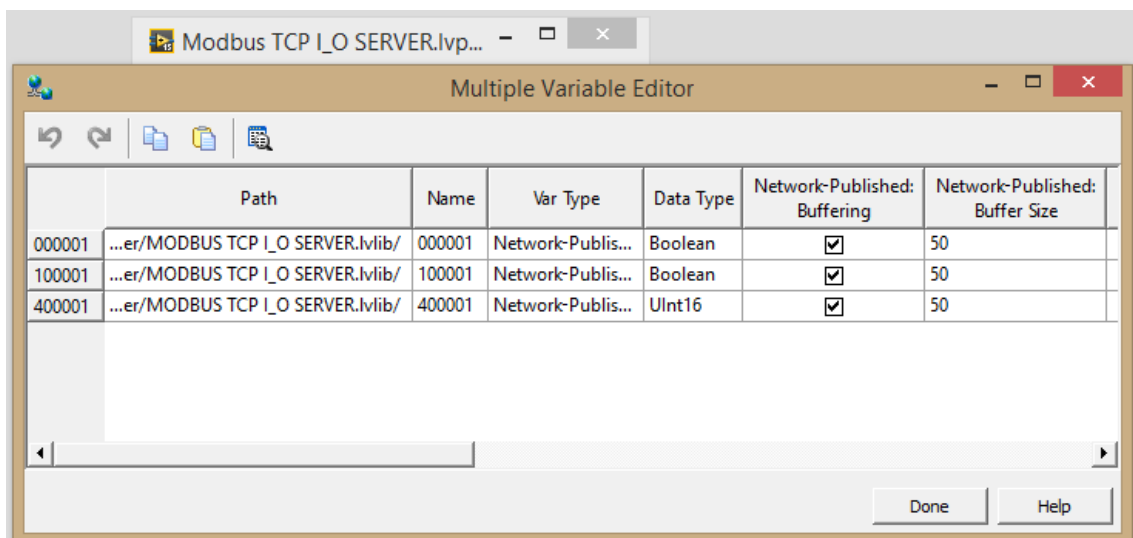
**Paso 8.-** Una vez agregados todos los rangos de direcciones necesarios pulse el botón OK. Como ejemplo se han agregado las direcciones 000001 (Bobina de un solo Bit), 100001(Entrada discreta de un solo Bit), 400001(Registros de retención de 16 bits como enteros sin signo que van desde 0 a 65.535).



**Figura 3-15** Visualización variables agregadas.

**Fuente:** (Autores, 2017).

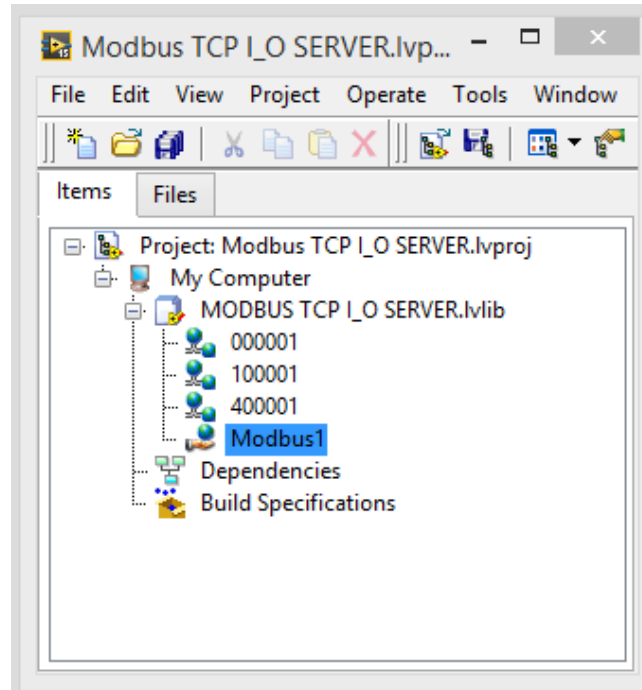
**Paso 9.-** Aparecerá la ventana Editor Múltiple de Variables con todas las variables agregadas, usted podrá revisar, realizar la configuración y edición de parámetros de acceso y de datos de las mismas.



**Figura 3-16** Editor múltiple de variables.

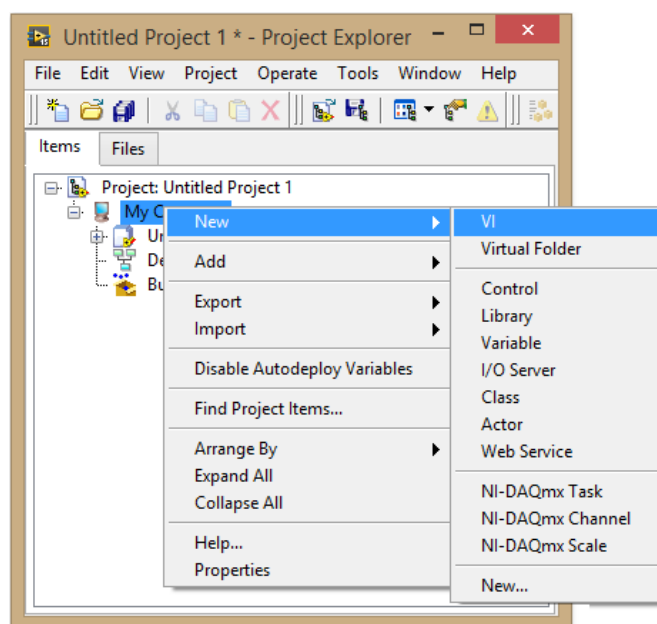
**Fuente:** (Autores, 2017)

**Paso 10.-** Seleccione el botón Done (Hecho) y la configuración de la comunicación Modbus estará completa. Ahora podrá visualizar la librería Modbus agregada en su proyecto.



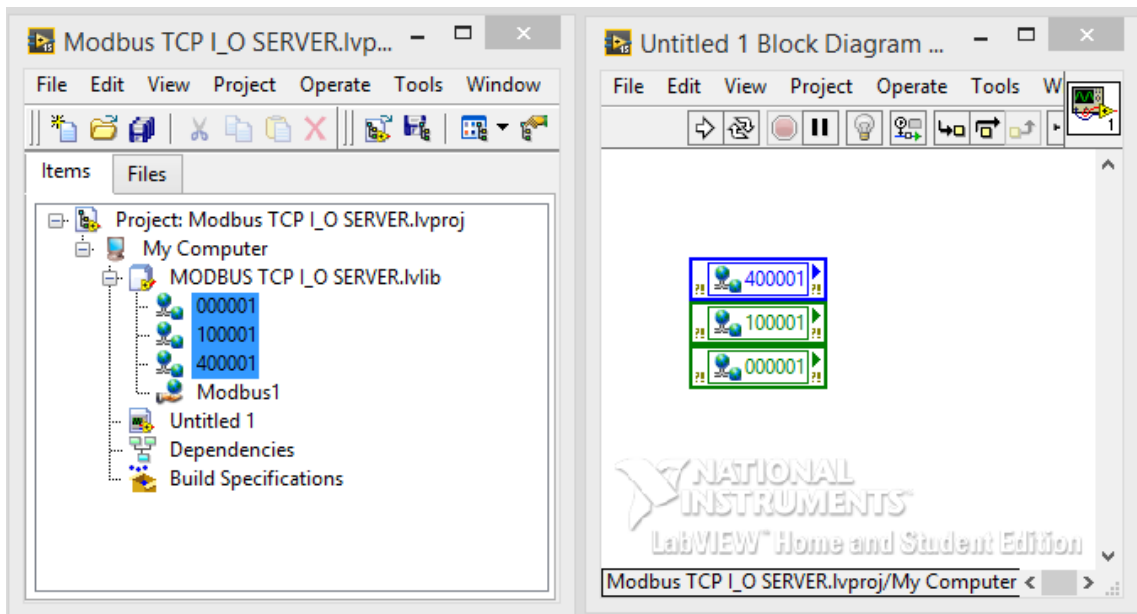
**Figura 3-17** Librería Modbus I/O server agregada al proyecto.  
**Fuente:** (Autores, 2017)

**Paso 11.-** Ahora cree un nuevo VI dentro de su proyecto, de clic derecho en mi PC>> Nuevo>> VI.



**Figura 3-18** Creación de un nuevo VI.  
**Fuente:** (Autores, 2017)

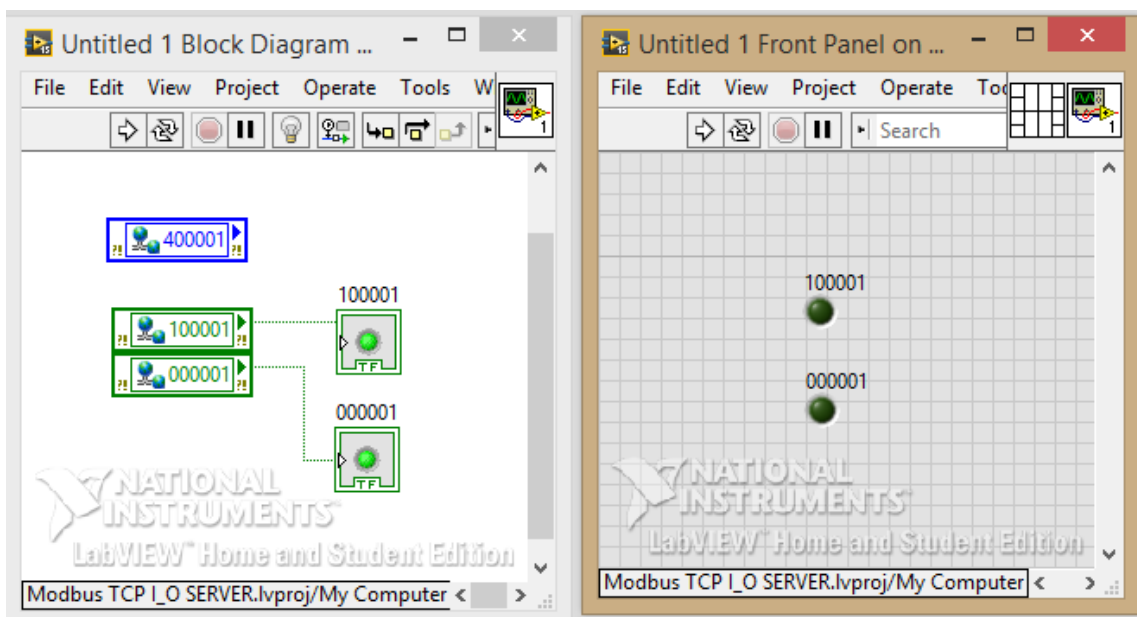
**Paso 12.-** Seleccione las variables requeridas de la librería agregada a su proyecto, y arrástrelas hasta la ventana diagrama de bloques del VI recién creado.



**Figura 3-19** VI con variables Modbus.

Fuente: (Autores, 2017)

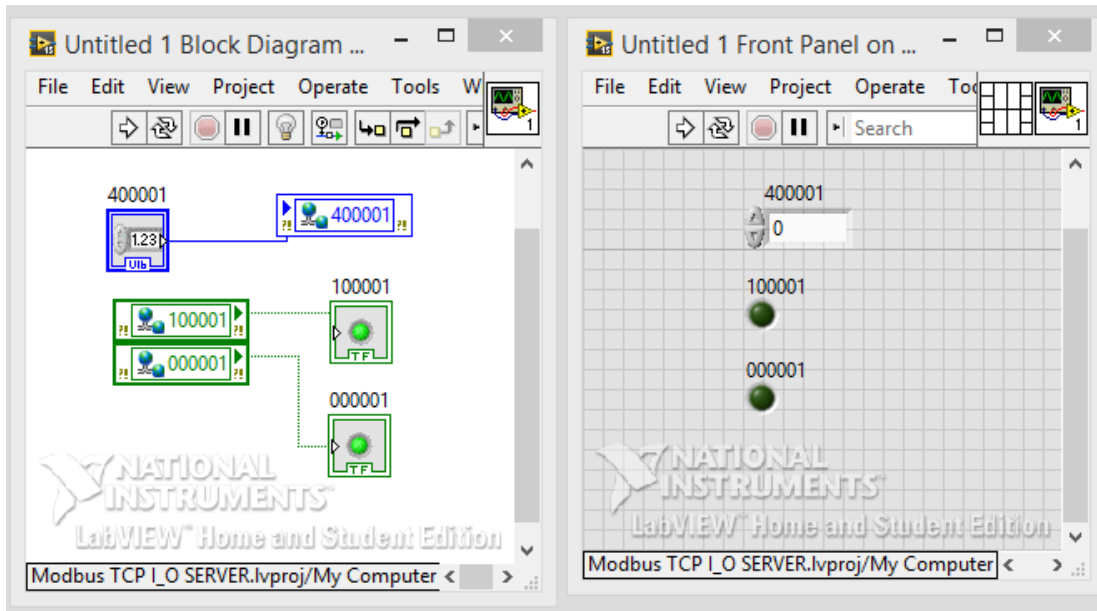
**Paso 13.-** En la ventana diagrama de bloques seleccione las variables Bobina y Entrada de un bit y cree indicadores booleanos para ellas. En este caso usted leerá el estado de sus variables.



**Figura 3-20** Indicadores booleanos conectados con variables Modbus.

Fuente: (Autores, 2017)

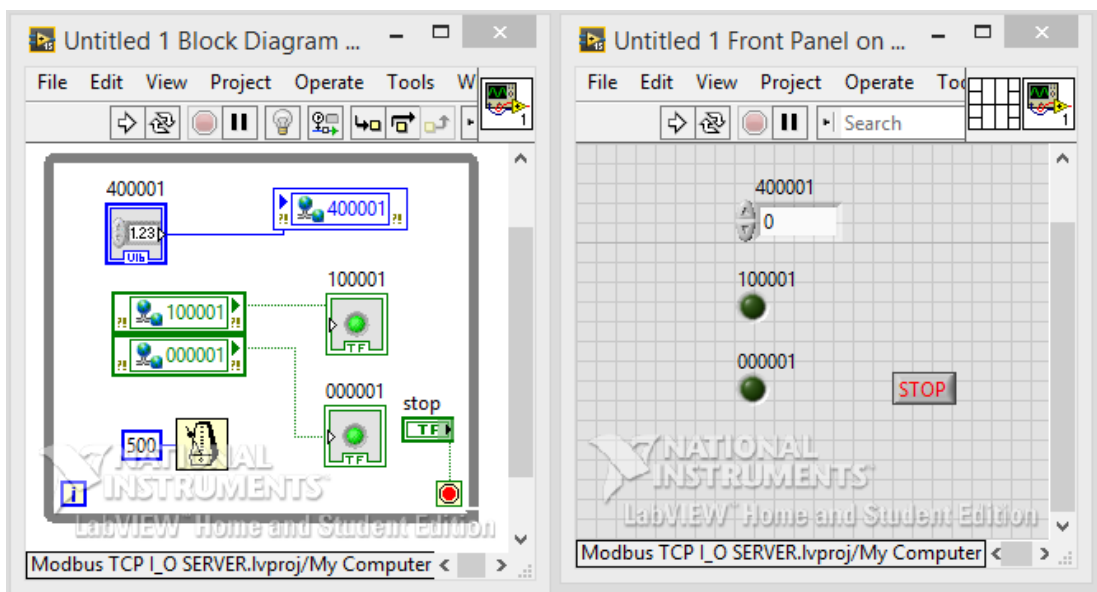
**Paso 14.-** Para la variable numérica UInt16 seleccione el modo de acceso ‘Escribir’ y cree el control adecuado que puede ser del tipo numérico, o construya un arreglo booleano para descomponer el número en bits. En este caso se escribirá desde LabVIEW hacia el dispositivo un número de 16 bits.



**Figura 3-21** Control numérico memoria M101 del PLC.

**Fuente:** (Autores, 2017)

**Paso 15.-** Coloque todos los elementos creados dentro de un While Loop con un botón STOP y establezca el tiempo de ejecución del mismo.

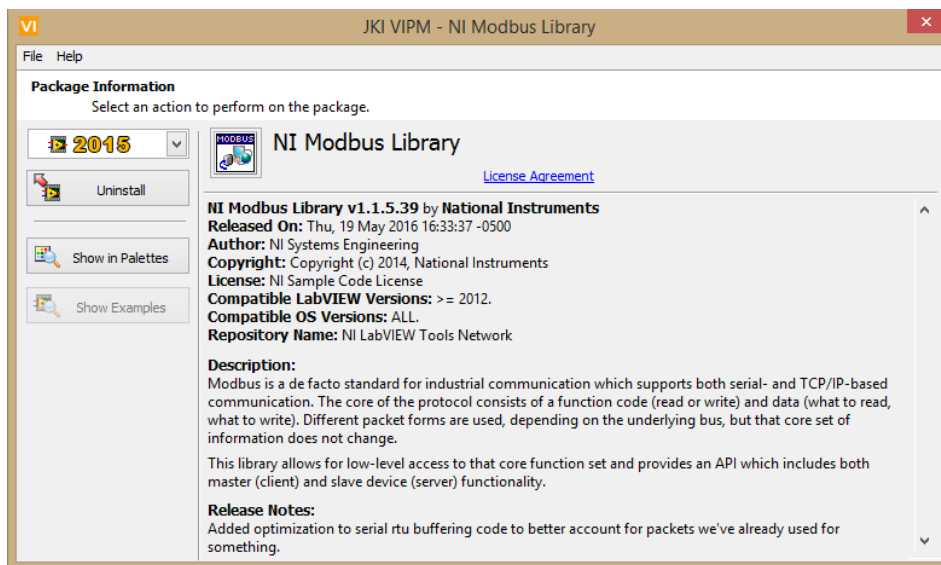


**Figura 3-22** VI completo para comunicación Modbus.

**Fuente:** (Autores, 2017)

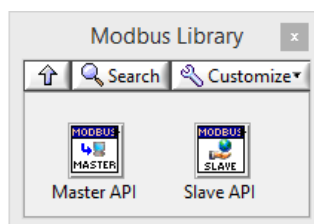
**Paso 16.-** Antes de ejecutar el VI asegúrese que exista conexión del PC con el dispositivo a comunicarse a través del puerto ethernet. Ejecute el VI. Ahora podrá leer y escribir datos en su dispositivo configurado a través de Modbus TCP.

**3.4.2.2 Modbus RTU 485 con API de LabVIEW.** Para la configuración en LabVIEW de Modbus RTU 485 se utilizará un controlador API por lo que usted necesita tener instalada la librería NI Modbus en LabVIEW, que puede ser descargada de la aplicación VI Package Manager.



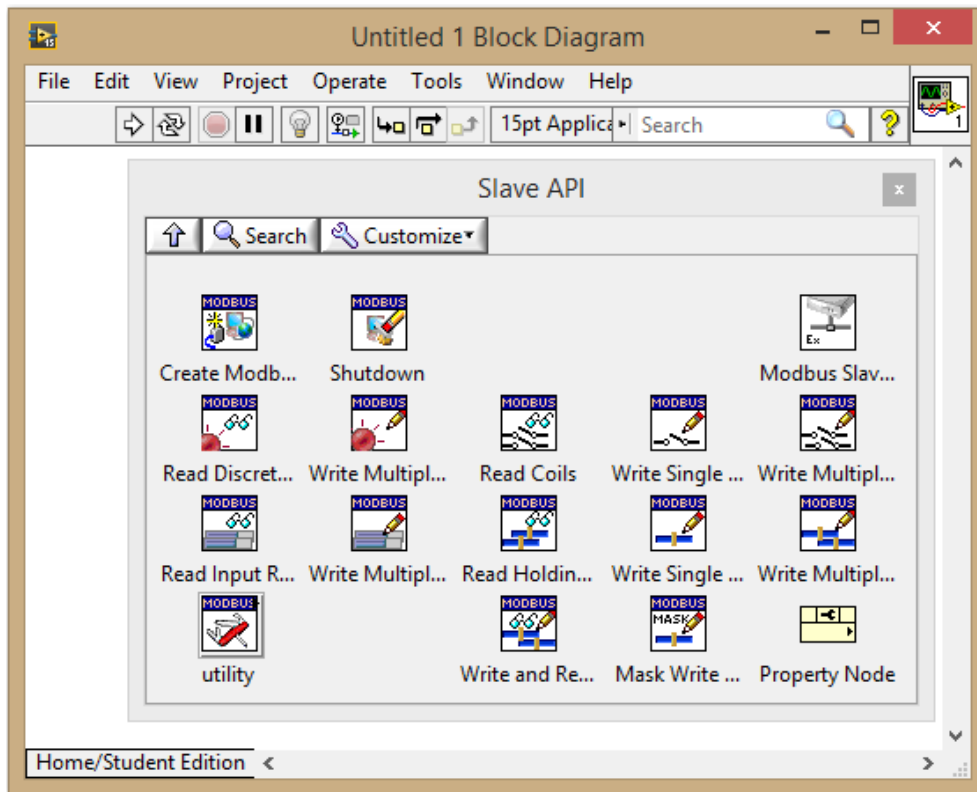
**Figura 3-23** Librería NI MODBUS en VI Packet Manager  
**Fuente:** (Autores, 2017)

Esta librería API de Modbus cuenta con elementos para conexión Maestro y Esclavo Modbus. Y pueden configurarse en Modbus TCP, Modbus Serial RTU y Modbus Serial ASCII. Los elementos más utilizados serán Read Coils, Write Multiple Coils y Read Registers. Cada uno representa una acción de lectura, escritura o ambas. Además, antes de introducir estas acciones se deberá crear o iniciar una instancia Modbus donde se introducirá la configuración para la comunicación.

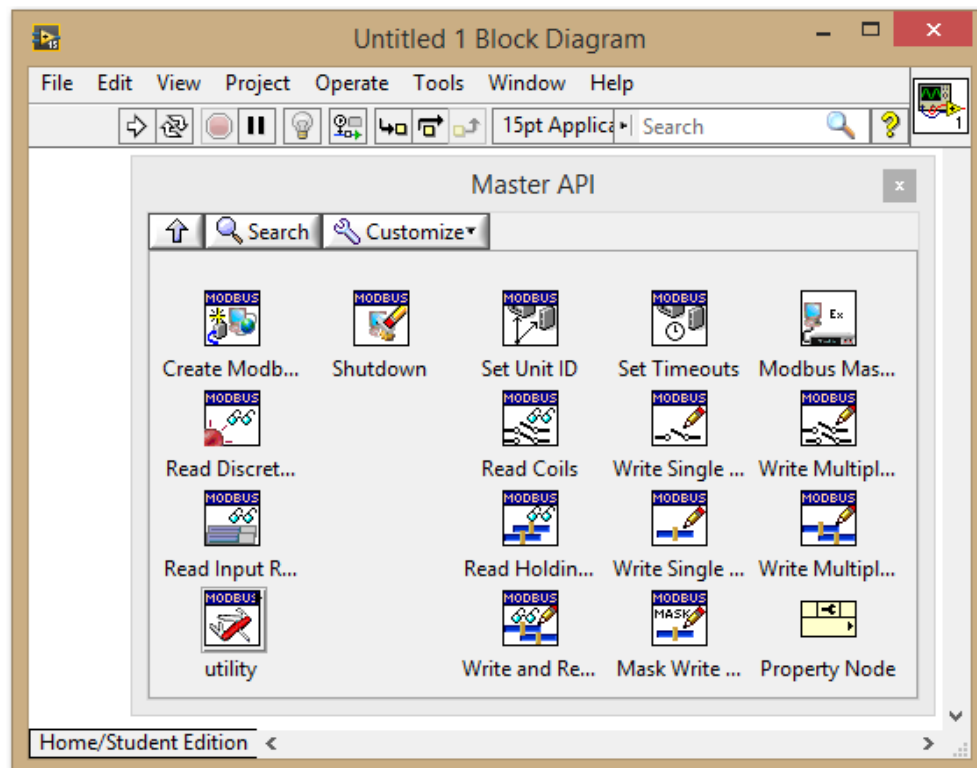


**Figura 3-24** Librería Modbus NI en palettes  
**Fuente:** (Autores, 2017).





**Figura 3-25** Elementos Esclavo API  
**Fuente:** (Autores, 2017)

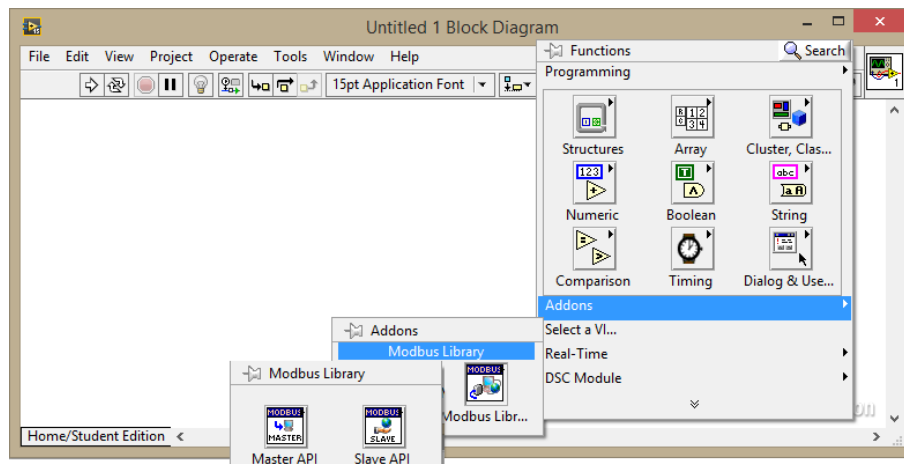


**Figura 3-26** Elementos Maestro API  
**Fuente:** (Autores, 2017)

**Pasos para la creación de un controlador API para comunicación Modbus RTU 485.**

**Paso 1.-** En la pantalla Introdutora de LabVIEW presione clic en file>> seleccione crear nuevo VI.

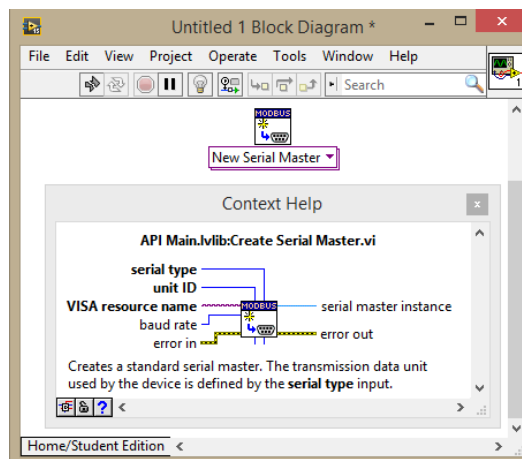
**Paso 2.-** Diríjase a la pantalla Diagrama de Bloques, de clic derecho en ‘Addons’ y seleccione NI Modbus Library. Fije la librería en la pantalla para facilitar su acceso. Seleccione los elementos de Master API.



**Figura 3-27** Librería NI Modbus en LabVIEW.

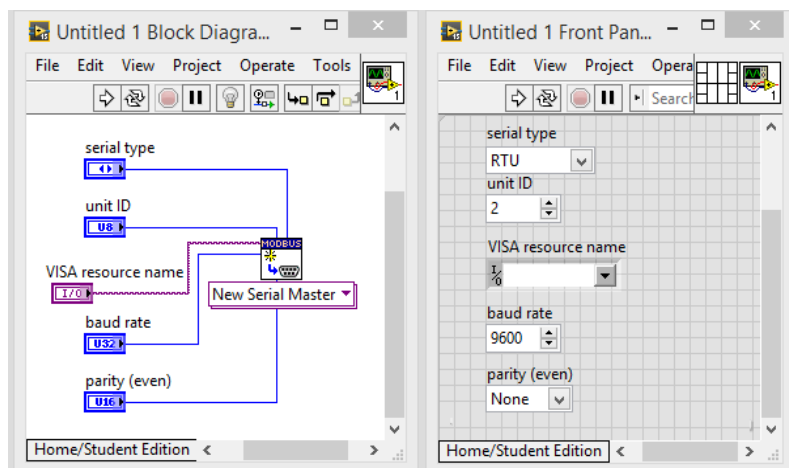
**Fuente:** (Autores, 2017).

**Paso 3.-** Cree una instancia de inicio Modbus. Para este caso seleccione New Master Serial, éste permitirá el acceso a diferentes esclavos Modbus conectados a la red serial. Agregue los controles e introduzca los parámetros de configuración en los mismos. Estos parámetros deberán ser configurados en todos los dispositivos de la red.



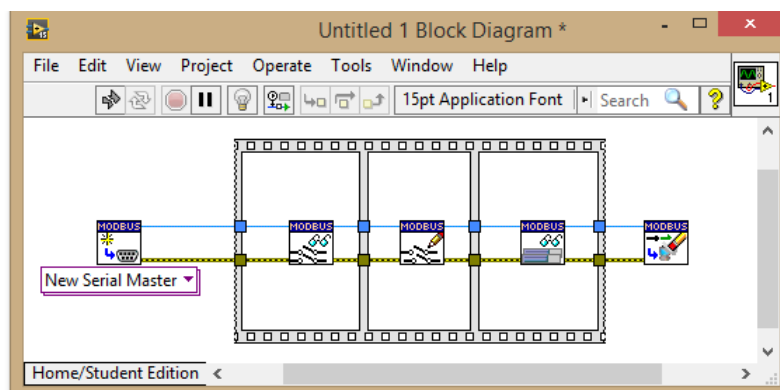
**Figura 3-28** Parámetros a configurar en la instancia de inicio Modbus New Serial Master.

**Fuente:** (Autores, 2017).



**Figura 3-29** Controles creados en la instancia Modbus con sus parámetros.  
**Fuente:** (Autores, 2017).

**Paso 4.-** Cada acción de lectura o escritura se realizará una a continuación de otra, coloque los elementos Read Coils, Write Multiple Coils y Read Registers cada uno dentro de un Frame del Flat Sequence, entrelace cada elemento a la instancia Modbus y el terminal de error. Al final de la secuencia se finaliza la instancia Modbus.



**Figura 3-30** Conexión de elementos librería Modbus.  
**Fuente:** (Autores, 2017)

**Paso 5.-** Cree todos los controles e indicadores para cada elemento de la librería Modbus. De clic derecho sobre cada uno de ellos >> Seleccione Create >> Escoja la opción All Controls and Indicators. Asigne valores según la siguiente información:

**Tabla 3-2** Parámetros WRITE MULTIPLE COILS

Parámetros de los controles del elemento WRITE MULTIPLE COILS.		
<b>WRITE MULTIPLE COILS.</b>	Dirección de inicio	Se coloca un valor entero igual o mayor a 0, este valor corresponderá al primer bit a ser escrito por el array COILS TO WRITE.
	Bobinas por escribir	Se crea por default un arreglo con controles booleanos los cuales enviaran la señal a las bobinas de un bit.

**Fuente:** (Autores, 2017)

**Tabla 3-3** Parámetros READ COILS.

Parámetros de los controles e indicador del elemento READ COILS.		
<b>READ COILS</b>	Dirección de inicio	Se coloca un valor entero igual o mayor a 0, este valor corresponderá al bit de inicio o primer bit a ser leído de la trama.
	Número de entradas	Se coloca un número entero el cual indicará la cantidad de bits a ser leídas a partir del bit de inicio.
	Bobinas	Se crea por default un arreglo con indicadores booleanos que se activaran al recibir la señal del bit correspondiente.

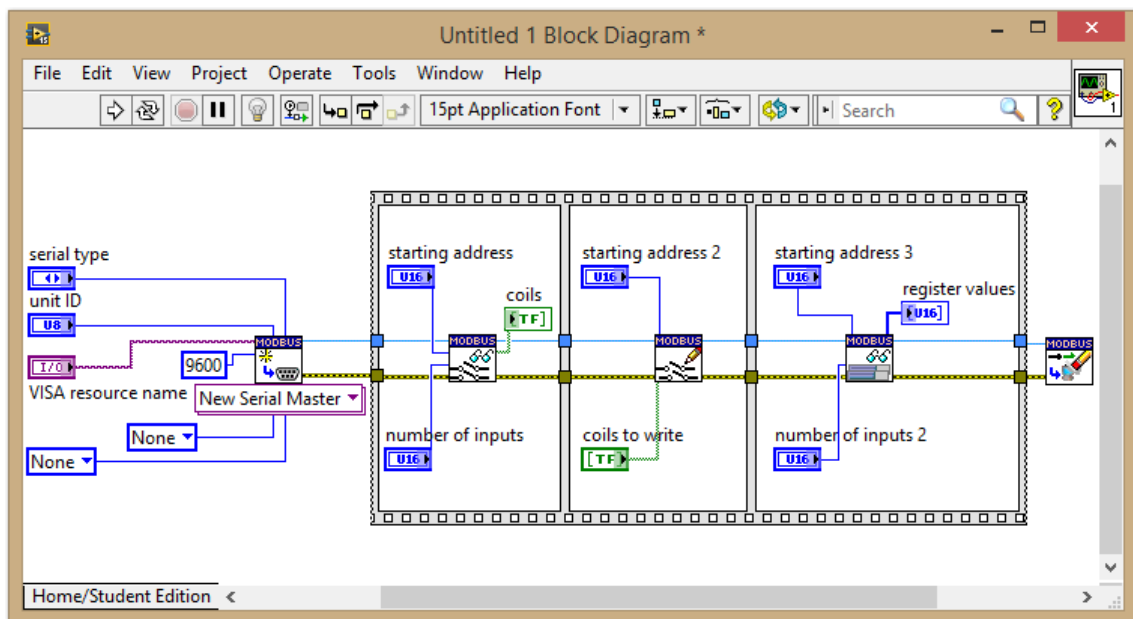
Fuente: (Autores, 2017)

**Tabla 3-4** Parámetros READ INPUT REGISTERS.

Parámetros de los controles e indicador del elemento READ INPUT REGISTERS		
<b>READ INPUT REGISTERS</b>	Dirección de inicio	Se coloca un valor entero igual o mayor a 0, este valor corresponderá al primer Byte a ser leído. Este byte es del tipo UInt16 es decir de 16bits.
	Número de entradas	Se coloca un número entero el cual indicará la cantidad de Bytes a ser leídas a partir del Byte de inicio.
	Valores de registro. (Register values)	Se crea por default un arreglo con indicadores numéricos que visualizaran el valor del Byte correspondiente. También se puede descomponer cada byte en arreglos booleanos mediante la ayuda de los elementos de LabVIEW.

Fuente: (Autores, 2017)

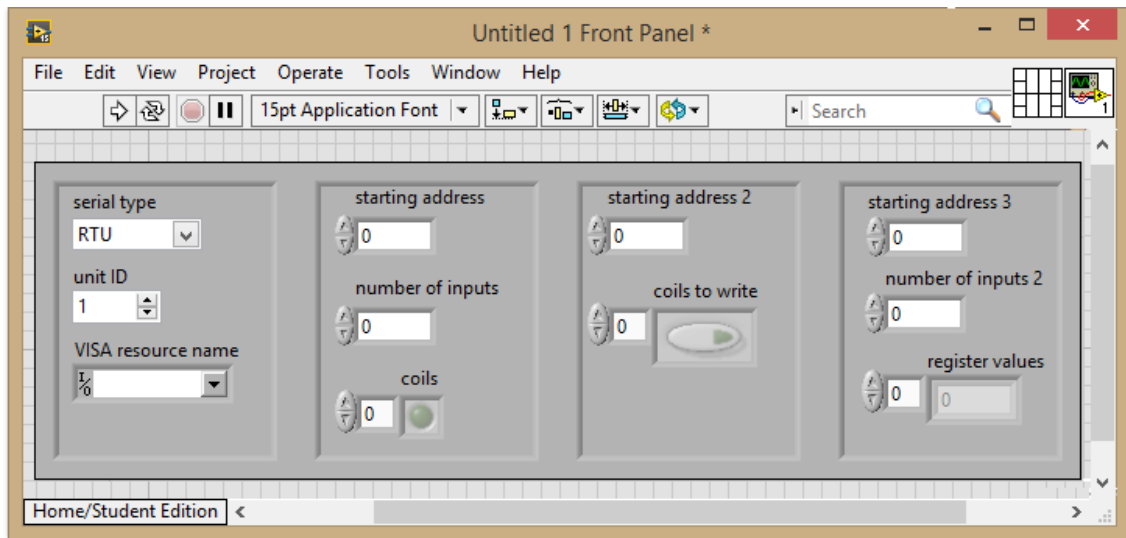
En la Figura 3-31 se muestra el Panel de Diagrama de Bloques de LabVIEW con los elementos Modbus y todos sus controles e indicadores. Todos estos elementos se colocarán dentro de un While Loop con un tiempo de ejecución y un botón Stop.



**Figura 3-31** Creación de todos los indicadores y controles.

Fuente: (Autores, 2017)

**Paso 6.-** Ahora diríjase a la pantalla frontal de LabVIEW, aquí podrá personalizar la ubicación y presentación de sus controles e indicadores. Diríjase a File y guarde el VI.



**Figura 3-32** Panel Frontal LABVIEW.

**Fuente:** (Autores, 2017)

**Paso 7.-** Conecte a su PC el Convertidor Bidireccional USB RS485 y en sus terminales A+ y B- conecte el dispositivo Modbus a supervisar. Diríjase al panel frontal del VI, seleccione la dirección del puerto COM al que conectó el convertidor y el número de esclavo asignado. Ejecute el VI, ahora podrá escribir y leer datos entre el dispositivo y su PC.

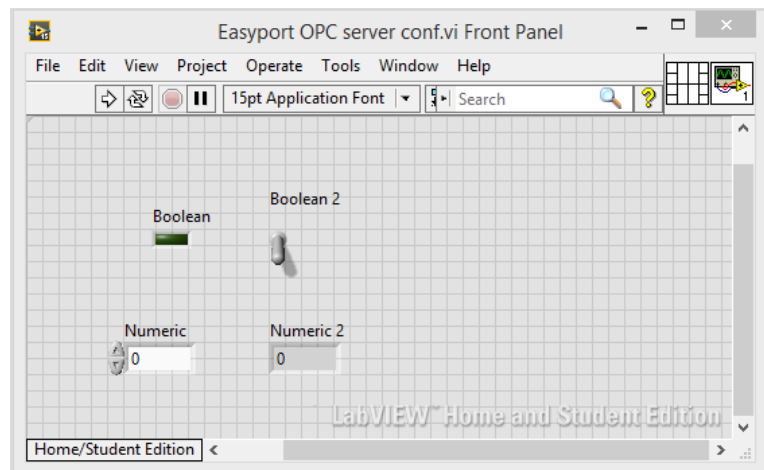
**3.4.3 Comunicación de Easy Port con LabVIEW.** El dispositivo EASY PORT de Festo se puede comunicar con LabVIEW de diversas maneras la primera forma es utilizando elementos de la librería VISA de LabVIEW para establecer comunicación serial con su puerto RS232, el segundo método es a través de los componentes ActiveX, y la tercera forma es utilizando OPC Server. De entre los cuales OPC muestra más facilidad de programación.

**3.4.3.1 Easy Port con LabVIEW mediante la utilización de OPC.** Para la utilización de Easy Port con LabVIEW a través de OPC primero de debe instalar el programa EzOPC de Festo que contiene las direcciones que se utilizará. Una vez instalado y para su utilización solo se debe asignar al control o indicador deseado en LabVIEW la acción que se desea ejecutar.

## *Pasos para la configuración de Easy Port a través de OPC con LabVIEW.*

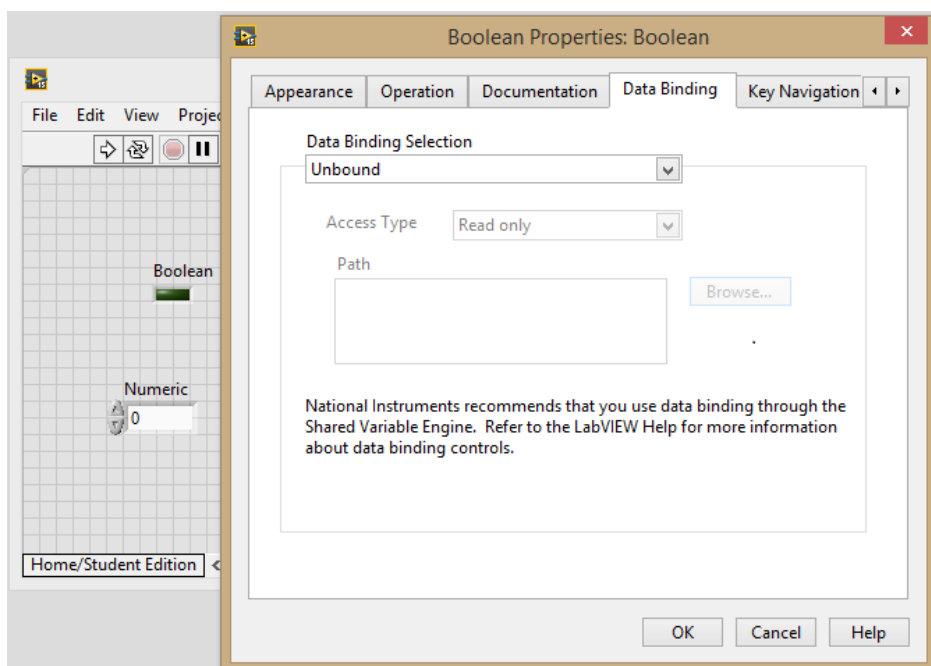
**Paso 1.-** En la pantalla Introdutora de LabVIEW presione clic en File y seleccione crear nuevo VI.

**Paso 2.-** Cree indicadores y controles booleanos y/o numéricos.



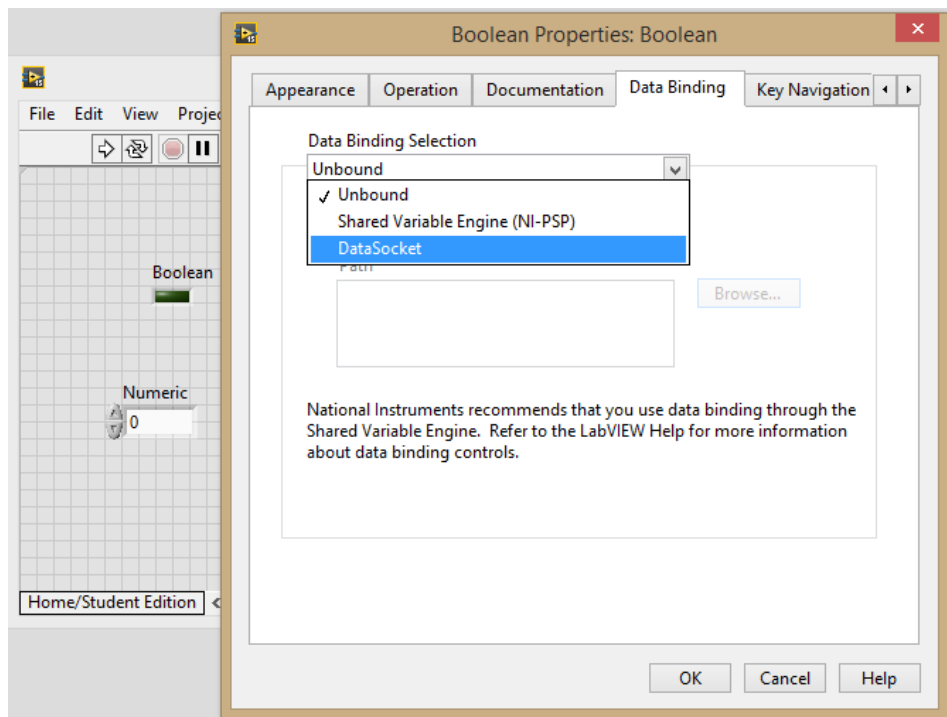
**Figura 3-33** Creación de controles e indicadores en LabVIEW.  
**Fuente:** (Autores, 2017).

**Paso 3.-** Clic derecho sobre cada elemento, clic en propiedades y seleccione Data Binding (enlace de datos).



**Figura 3-34** Enlace de datos.  
**Fuente:** (Autores, 2017)

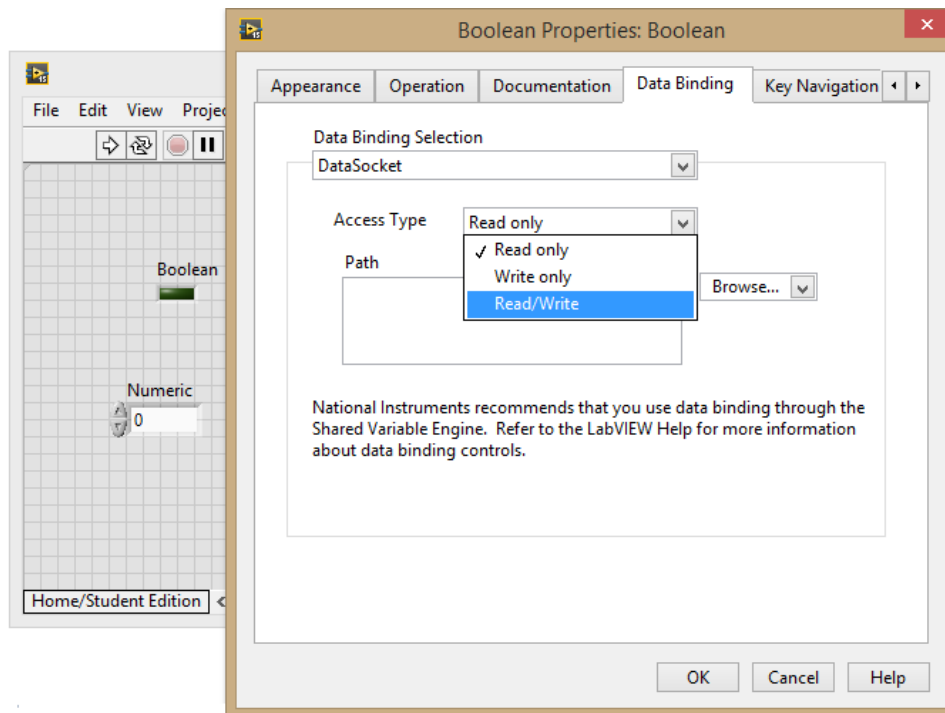
**Paso 4.-** En el casillero desplegable escoger DataSocket (Toma de datos).



**Figura 3-35** Toma de datos.

Fuente: (Autores, 2017)

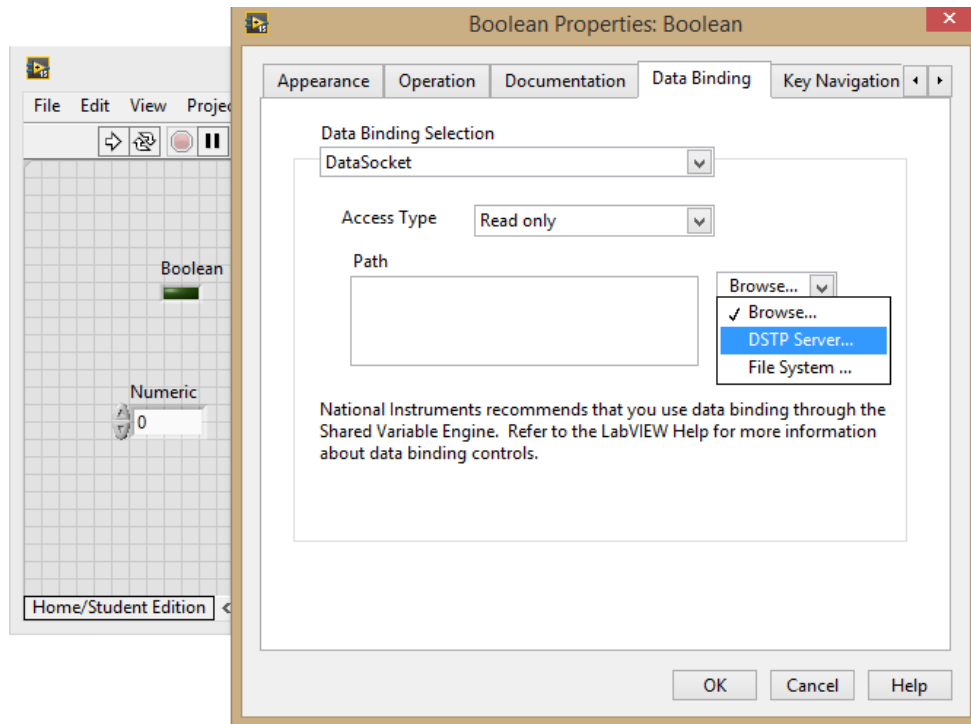
**Paso 5.-** Ahora seleccione el tipo de acceso que le daremos a cada elemento.



**Figura 3-36** Selección tipo de acceso.

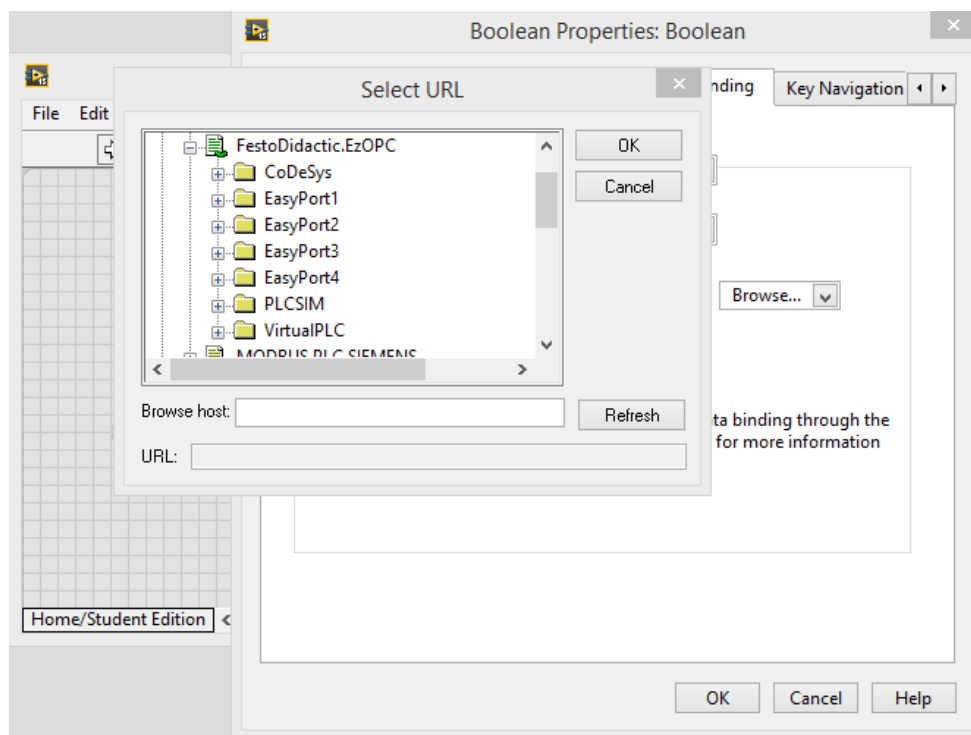
Fuente: (Autores, 2017)

**Paso 6.-** En el casillero Browser escoger DTSP Server esto desplegara una lista con todas las librerías server agregadas, seleccione FestoDidactic.EzOPC.



**Figura 3-37** Selección del acceso al Path.

**Fuente:** (Autores, 2017)

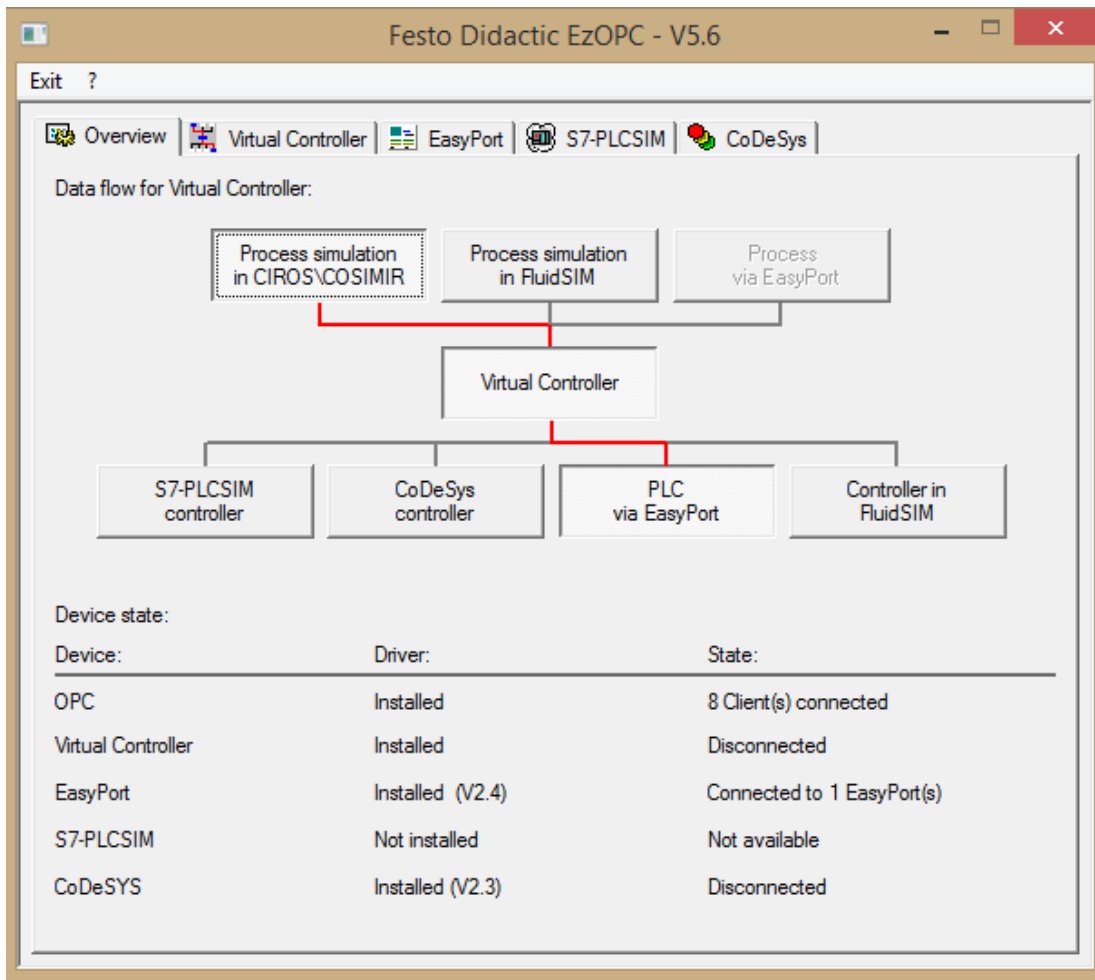


**Figura 3-38** Accesos posibles desde EzOPC.

**Fuente:** (Autores, 2017)



**Paso 7.-** Al seleccionar FestoDidactic.EzOPC se ejecuta dicha aplicación, deberá configurarse el acceso del dispositivo como se muestra en la Figura 3-39.



**Figura 3-39** Configuración de EzOPC para la supervisión y control desde LabVIEW.  
**Fuente:** (Autores, 2017)

**Paso 8.-** Ahora asigne la dirección OPC al elemento correspondiente dependiendo de la acción que realizará, ya sea lectura, escritura o ambas; tanto para elementos booleanos o numéricos.

Para el Puerto 1 y 2 del Easy Port, los controles booleanos en LabVIEW accionarán las salidas digitales, mientras que los indicadores booleanos funcionarán con la señal recibida en las entradas digitales del mismo. Con los elementos numéricos podremos observar y escribir las entradas y salidas analógicas del Puerto 3 del Easy Port.

**NOTA:** Este direccionamiento se puede realizar ya sea a través de los bits, los bytes o words asignados para la acción deseada a ejecutarse.

**Tabla 3-5** Direcciones OPC Puerto 1 del EasyPort.

Direcciones OPC EasyPort Puerto 1			
Entradas digitales		Salidas digitales	
00	InputPort1Bit00	00	OutputPort1Bit00
01	InputPort1Bit01	01	OutputPort1Bit01
02	InputPort1Bit02	02	OutputPort1Bit02
03	InputPort1Bit03	03	OutputPort1Bit03
04	InputPort1Bit04	04	OutputPort1Bit04
05	InputPort1Bit05	05	OutputPort1Bit05
06	InputPort1Bit06	06	OutputPort1Bit06
07	InputPort1Bit07	07	OutputPort1Bit07

Fuente: (Autores, 2017)

**Tabla 3-6** Direcciones OPC Puerto 2 del EasyPort.

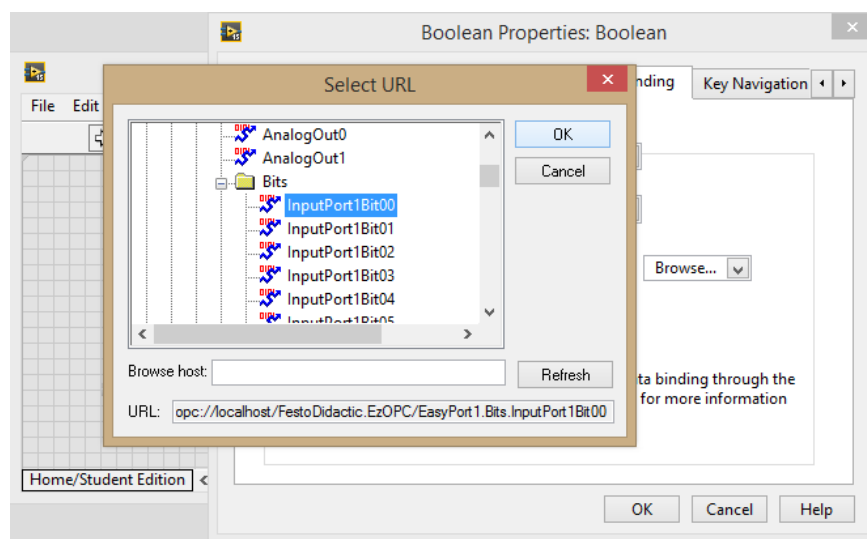
Direcciones OPC EasyPort Puerto 2			
Entradas digitales		Salidas digitales	
00	InputPort2Bit00	00	OutputPort2Bit00
01	InputPort2Bit01	01	OutputPort2Bit01
02	InputPort2Bit02	02	OutputPort2Bit02
03	InputPort2Bit03	03	OutputPort2Bit03
04	InputPort2Bit04	04	OutputPort2Bit04
05	InputPort2Bit05	05	OutputPort2Bit05
06	InputPort2Bit06	06	OutputPort2Bit06
07	InputPort2Bit07	07	OutputPort2Bit07

Fuente: (Autores, 2017)

**Tabla 3-7** Direcciones OPC Puerto 3 EasyPort.

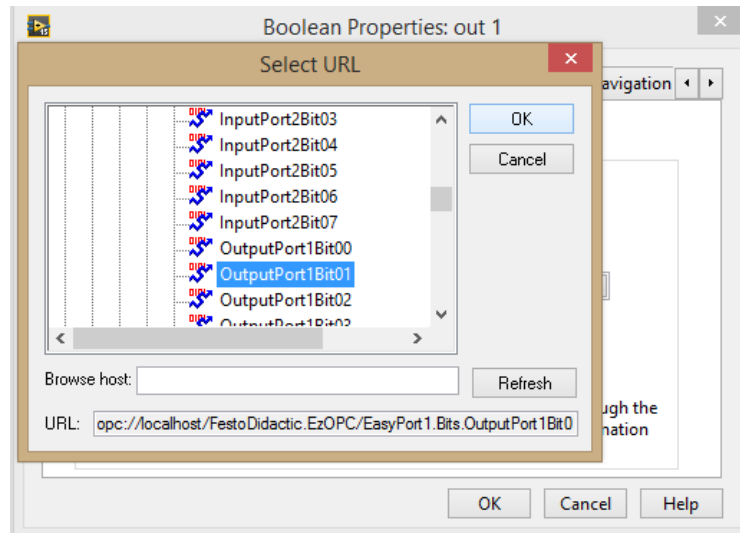
Direcciones OPC EasyPort Puerto 3					
Entradas analógicas				Salidas analógicas	
IN 0	AnalogIn0	IN 2	AnalogIn2	OUT 0	AnalogOut0
IN 1	AnalogIn1	IN 3	AnalogIn3	OUT 1	AnalogOut1

Fuente: (Autores, 2017)

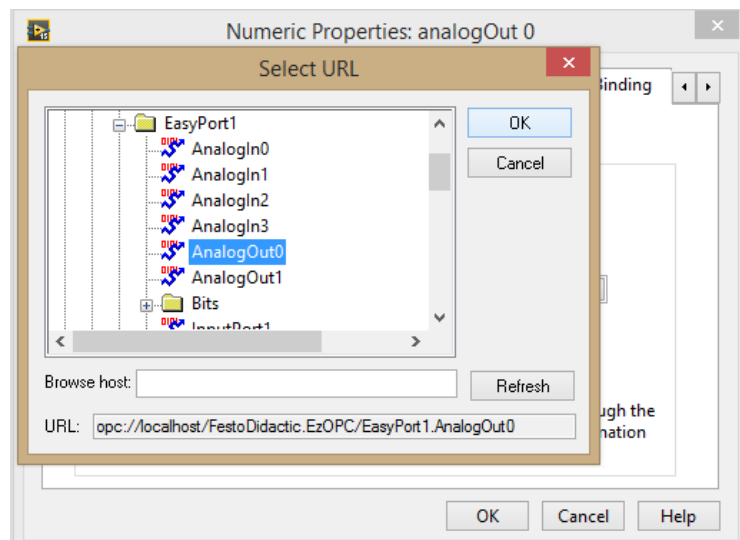


**Figura 3-40** Direccionamiento para la lectura de la entrada 00 del Puerto 1 del Easy Port.

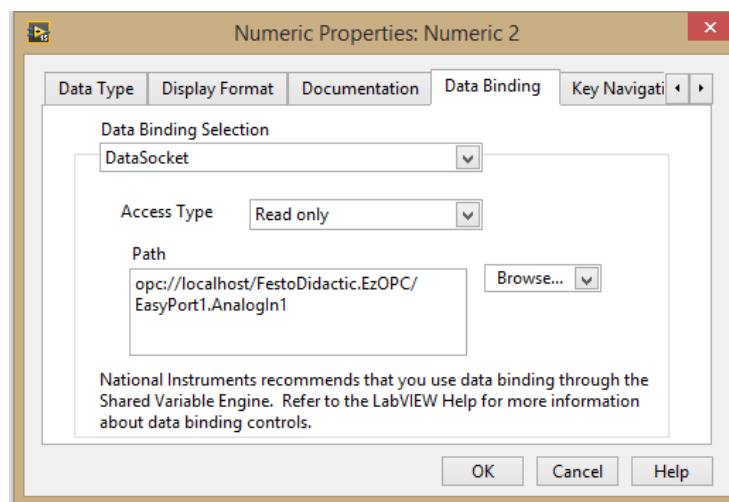
Fuente: (Autores, 2017)



**Figura 3-41** Direccionamiento para la escritura de salida digital 01 del Puerto 1 del Easy Port.  
**Fuente:** (Autores, 2017)

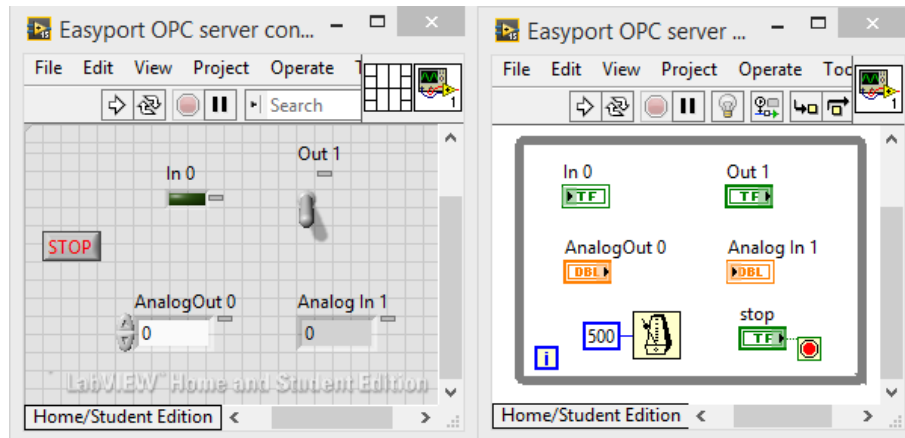


**Figura 3-42** Direccionamiento para salida analógica Out 0 del Puerto 3 del Easy Port.  
**Fuente:** (Autores, 2017)



**Figura 3-43** Direccionamiento para la lectura de la entrada analógica In 1 del Easy Port.  
**Fuente:** (Autores, 2017)

**Paso 9.-** Coloque todos los elementos dentro de un While Loop con su respectivo botón de paro y tiempo de ejecución.



**Figura 3-44** Vista frontal y de diagrama de bloques final.

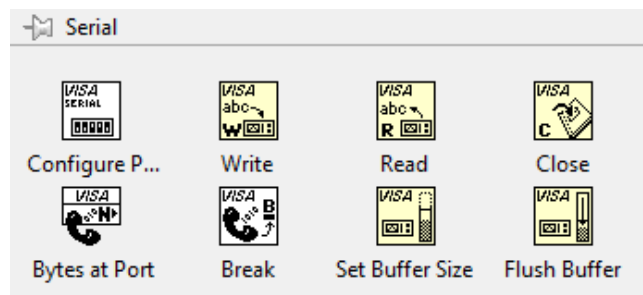
**Fuente:** (Autores, 2017).

**3.4.3.2 EasyPort con LabVIEW a través de comunicación serial.** LabVIEW cuenta con la arquitectura de software de instrumentos virtuales (VISA) un estándar que nos permitirá configurar, programar y solucionar problemas de sistemas. VISA incluye bibliotecas de software, utilidades y programas de configuración. Se aprovechará los mismos para la comunicación serial RS232 entre Easy Port USB y LabVIEW. EasyPort USB transmite y recibe datos bajo una codificación ASCII 115 kBauds, 8, N, 1.

***Pasos para configuración de EasyPort a través de puerto serial RS232 con LabVIEW.***

**Paso 1.-** Cree un nuevo VI y diríjase a la pantalla de diagrama de bloques.

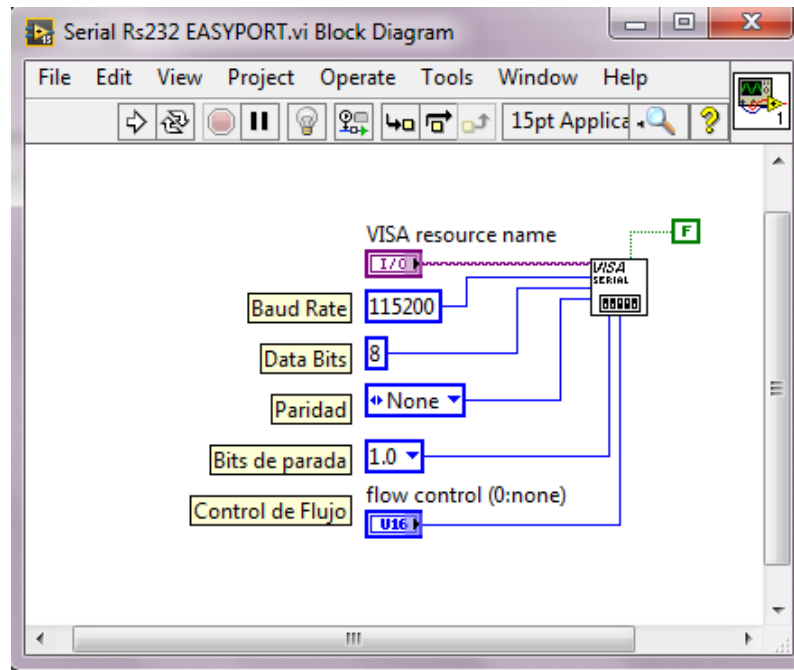
**Paso 2.-** De clic derecho, seleccione Data Communication, ingrese a Protocols, escoja la opción Serial y fije los elementos VISA.



**Figura 3-45** Elementos VISA Serial a ser utilizados.

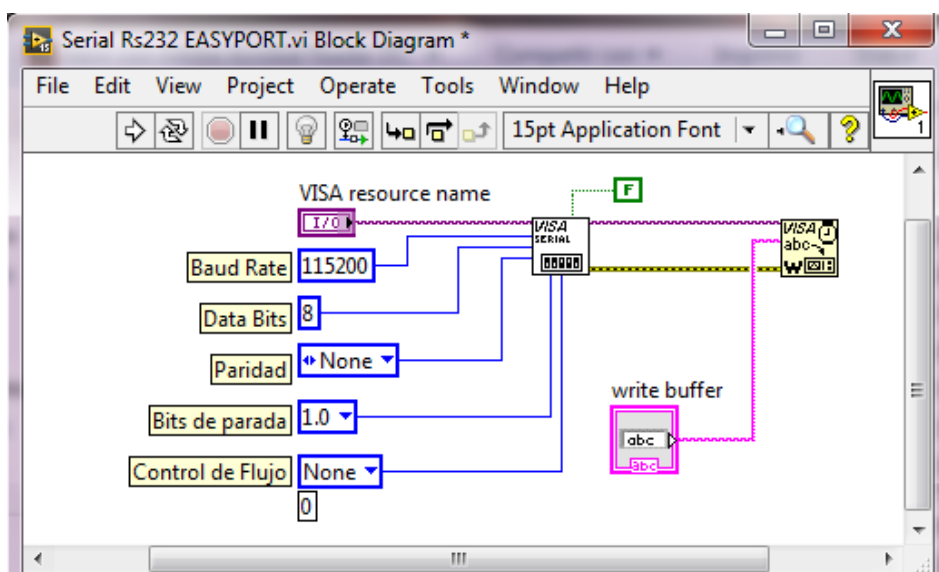
**Fuente:** (Autores, 2017).

**Paso 3.-** Escoja el elemento Configurar Puerto y coloque los parámetros provenientes del Easy Port.



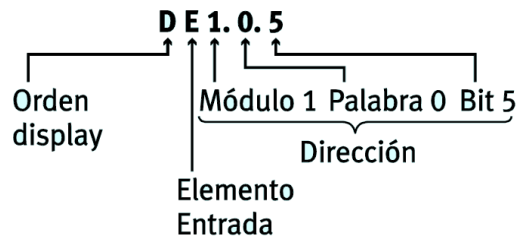
**Figura 3-46** Configuración del puerto de comunicación serial.  
**Fuente:** (Autores, 2017).

**Paso 4.-** Ahora coloque el elemento Escribir y cree un String Control para el mismo. En este String se escribirán los códigos alfanuméricos que contienen la información correspondiente a la acción que se realizará, la ID o dirección asignada al dispositivo y el direccionamiento ya sea a través del bit, byte o Word del EasyPort.

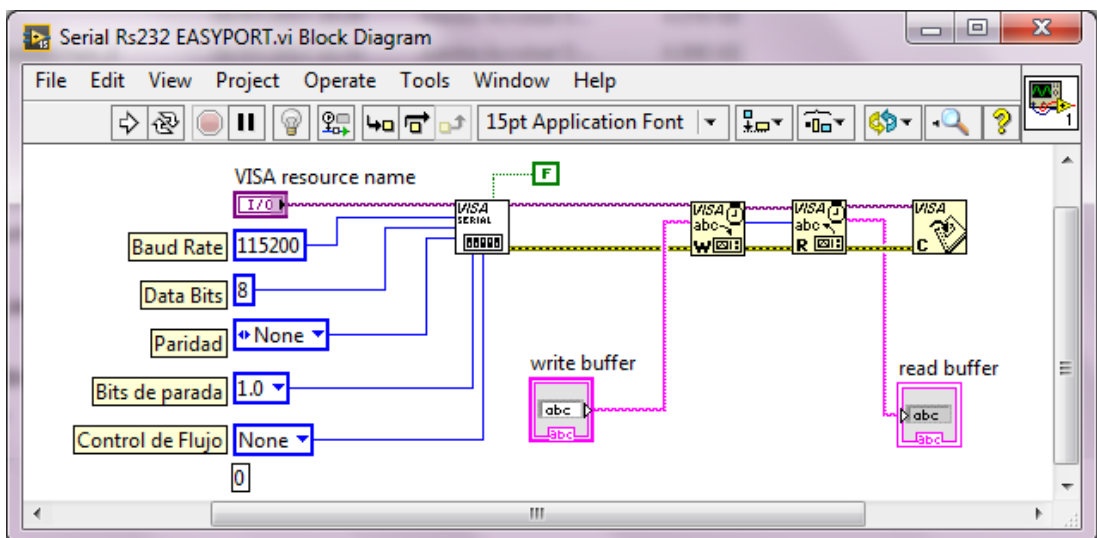


**Figura 3-47** Inserción del elemento escribir.  
**Fuente:** (Autores, 2017).

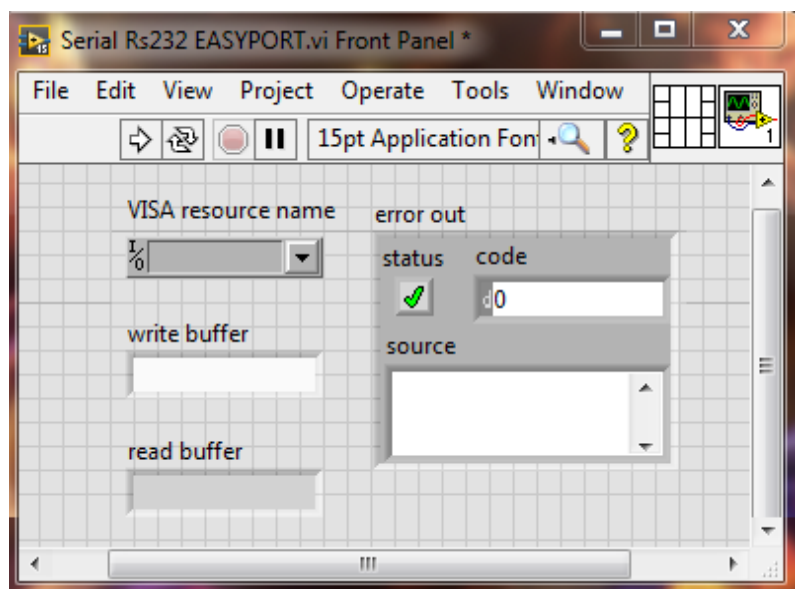
Ejemplo de codificación ASCII:



**Paso 5.-** A continuación, agregue el elemento leer, este le entregará el resultado de la acción realizada. Y finalmente colocar el elemento cerrar VISA.



**Figura 3-48** Inserción elementos Read y Close VISA  
**Fuente:** (Autores, 2017).



**Figura 3-49** Panel Frontal VI para escribir y leer el estado de EasyPort  
**Fuente:** (Autores, 2017).

**Paso 6.-** Coloque todos los elementos dentro de un While Loop con su respectivo Stop y tiempo de ejecución.

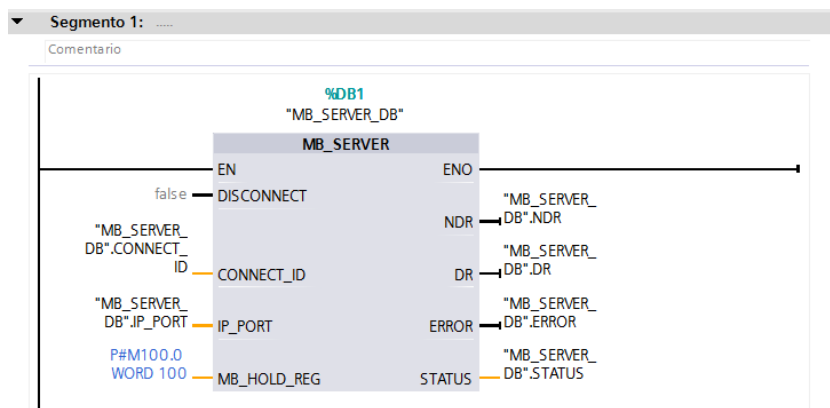
**Paso 7.-** Ejecute el VI, puede ingresar códigos y observar el comportamiento del EasyPort y el código de respuesta que envía al PC. Pruebe con las siguientes direcciones para observar el funcionamiento:

**Tabla 3-8** Ejemplos de códigos para EasyPort USB por puerto serial desde LabVIEW.

CODIGO	ACCIÓN
<b>DEW1.0</b>	Leerá el estado de todas las entradas digitales del EasyPort.
<b>MAW1.0=XX</b>	Escribirá cualquiera de las salidas de ambos puertos dependiendo del valor hexadecimal asignado después del igual. <b>Ejemplos:</b> MAW1.0=8 >> Encenderá la cuarta salida del Puerto 1. MAW1.0=FF >> Encenderá todas las salidas del Puerto 1. MAW1.0=1F >> Encenderá las 5 primeras salidas del Puerto 1. MAW1.0=FFFF >> Encenderá todas las salidas del Puerto 1 y 2. MAW1.0=0 >> Apagará todas las salidas de los Puertos 1 y 2.
<b>MA1.0.0=1</b>	Encenderá la primera salida (salida 0) del Puerto 1 del EasyPort.
<b>MA1.0.0=0</b>	Apagará la primera salida (salida 0) del Puerto 1 del EasyPort.
<b>MA1.0.7=1</b>	Encenderá la octava salida (salida 7) del Puerto 1 del EasyPort.
<b>DE1.0.6</b>	Leerá el valor de la séptima entrada digital del Puerto 1 del EasyPort.

### 3.5. Pruebas y funcionamiento.

**3.5.1 Comunicación Modbus entre PLC Siemens y LabVIEW.** Para verificar el funcionamiento de la configuración Modbus TCP se realizó un programa en el Software TIA PORTAL, dicho programa consta en uno de sus segmentos con el Bloque MODBUS\_SERVER y sus parámetros.



**Figura 3-50** Parámetros MODBUS\_SERVER PLC SIEMENS.

Fuente: (Autores, 2017)

Los parámetros ingresados que se muestran en la Figura 3-50 permitirán que LabVIEW a través de I/O Server tenga acceso a la lectura de todas las entradas físicas del PLC y a la lectura/escritura de las salidas físicas y memorias internas del mismo. (Las memorias del tipo M, MW se leerán a partir del byte 101, es decir M101).

El objetivo de esta práctica es configurar y comprobar la comunicación Modbus al visualizar en LabVIEW el funcionamiento de las entradas y salidas físicas del PLC Siemens y escribir memorias desde el software al PLC que se encargaran del inicio y paro del proceso o programa cargado en el mismo.

Para realizar la comunicación se efectuará los pasos para configuración I/O server en LabVIEW. Crear un proyecto nuevo, dar clic derecho en el icono PC, clic en nuevo, clic en I/O Server. Entonces seleccionar Modbus y configuramos los parámetros correspondientes a nuestro PLC y damos clic en aceptar.

**Tabla 3-9** Parámetros Modbus PLC Siemens.

PLC SIEMENS	
<b>MODEL MODBUS</b>	TCP
<b>Dirección IP</b>	192.168.0.2
<b>DIRECCION MODBUS</b>	1

Fuente: (Autores, 2017)

La librería ya ha sido agregada automáticamente, dar clic derecho sobre la misma, seleccionar Create Bound Variables y agregar el rango de ellas correspondiente a todas las entradas y salidas físicas del PLC, además se debe agregar la variable del Byte de la memoria interna que será escrita desde LabVIEW hacia el autómeta.

**Tabla 3-10** Direccionamiento de entradas y salidas físicas del PLC Siemens en Modbus.

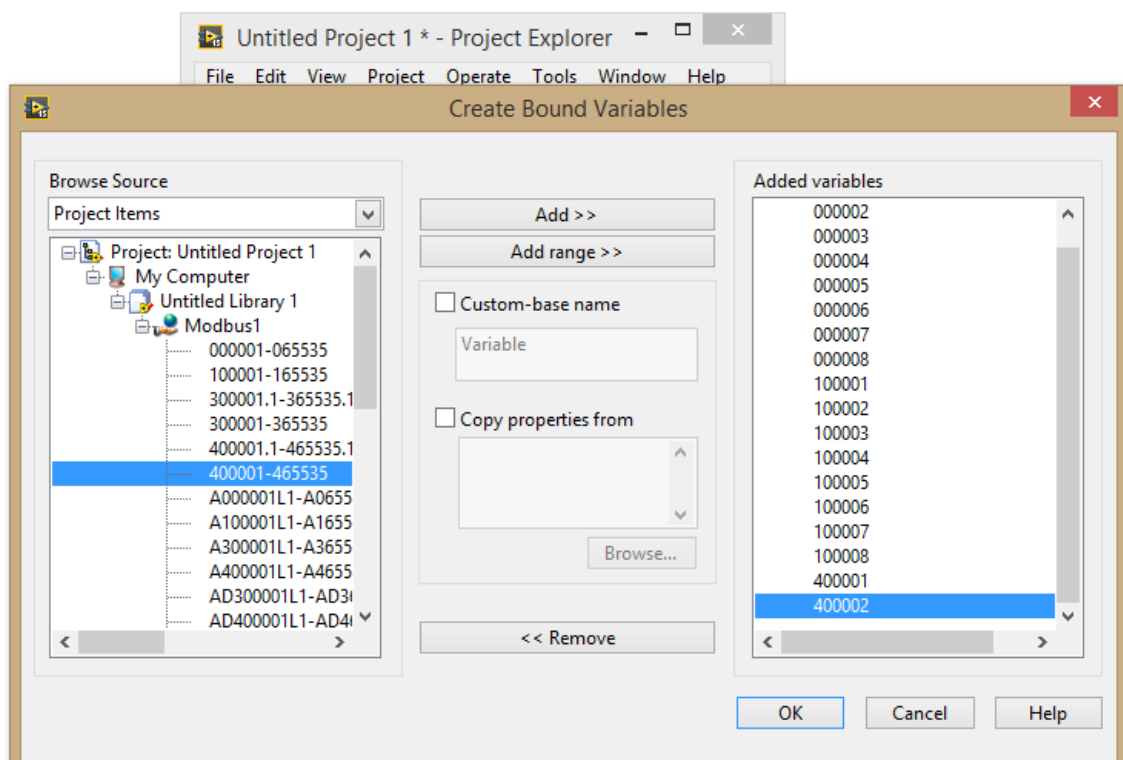
<b>Variable Modbus</b>	<b>Dirección PLC Siemens</b>	<b>Tipo de dato</b>	<b>Modo de acceso permitido desde LabVIEW</b>
000001	Q0.0	Booleano	Lectura/escritura
000002	Q0.1	Booleano	Lectura/escritura
000003	Q0.2	Booleano	Lectura/escritura
000004	Q0.3	Booleano	Lectura/escritura
000005	Q0.4	Booleano	Lectura/escritura



**Tabla 3-11** (Continúa) Direccionamiento de entradas y salidas físicas del PLC Siemens en Modbus.

Variable Modbus	Dirección PLC Siemens	Tipo de dato	Modo de acceso permitido desde LabVIEW
000006	Q0.5	Booleano	Lectura/escritura
100001	I0.0	Booleano	Lectura
100002	I0.1	Booleano	Lectura
100003	I0.2	Booleano	Lectura
100004	I0.3	Booleano	Lectura
100005	I0.4	Booleano	Lectura
100006	I0.5	Booleano	Lectura
100007	I0.6	Booleano	Lectura
100008	I0.7	Booleano	Lectura
400001	M101	WORD	Lectura/escritura

Fuente: (Autores, 2017)



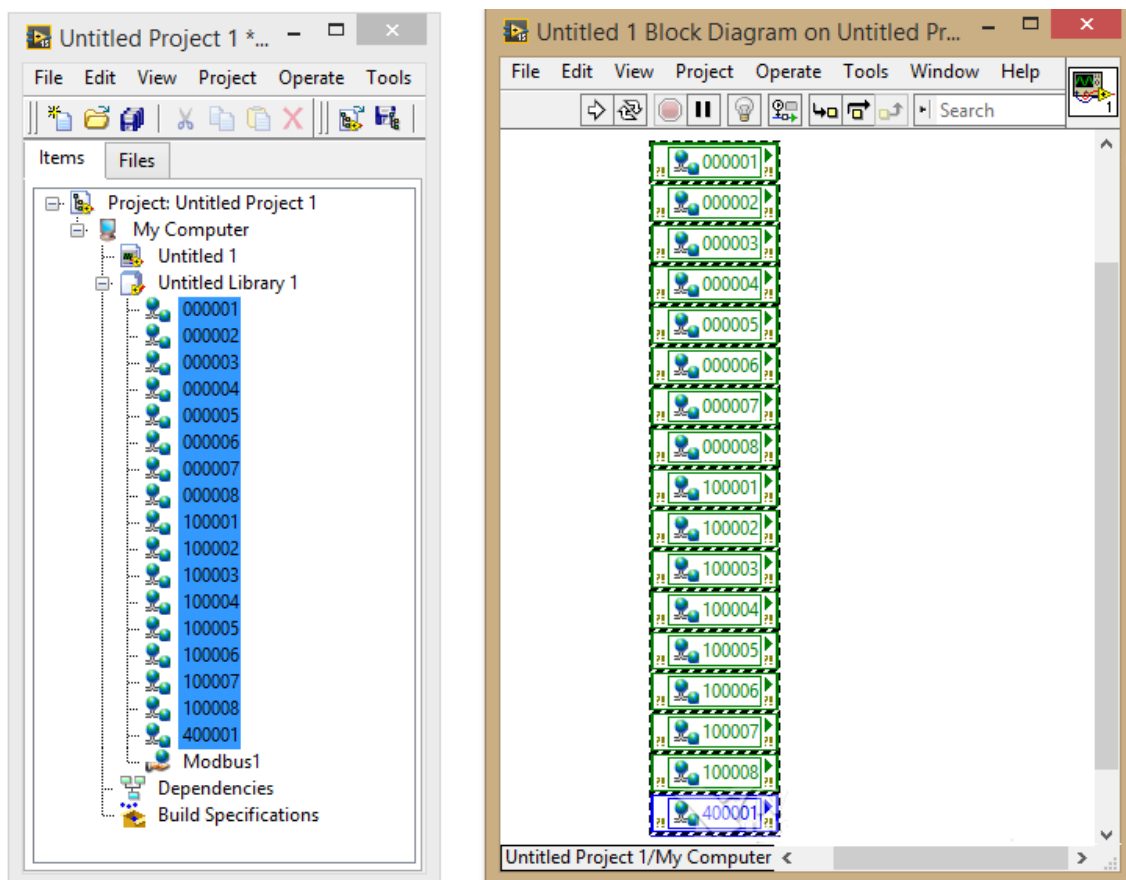
**Figura 3-51** Creación de variables Modbus dentro de la librería.

Fuente: (Autores, 2017)

Una vez agregadas todas las variables seleccionar Ok, entonces se desplegará el editor múltiple de variables donde se visualizarán todas ellas, su tipo de datos, acceso. El siguiente paso es dar clic en el icono PC y crear un VI dentro de nuestro proyecto, después seleccionar las variables creadas y las llevarlas hasta el panel de Diagrama de Bloques de nuestro VI.

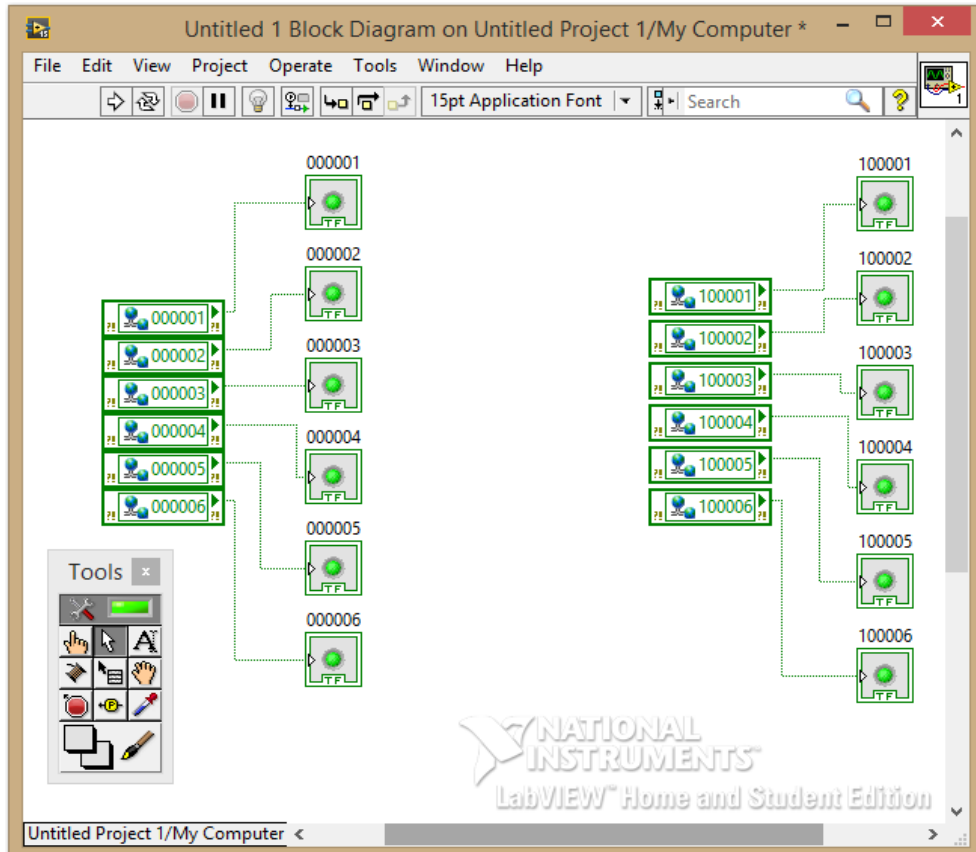
	Path	Name	Var Type	Data Type	ork- Publi Buffering	twor- Publish Buffer Size	Network- Published: Bind to Source	Network- Published: Access Type	Ne
000001	... 1/	000001	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000002	... 1/	000002	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000003	... 1/	000003	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000004	... 1/	000004	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000005	... 1/	000005	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000006	... 1/	000006	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000007	... 1/	000007	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
000008	... 1/	000008	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
100001	... 1/	100001	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100002	... 1/	100002	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100003	... 1/	100003	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100004	... 1/	100004	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100005	... 1/	100005	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100006	... 1/	100006	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100007	... 1/	100007	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
100008	... 1/	100008	Network-Published	Boolean	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read only	Pro
400001	... 1/	400001	Network-Published	UInt16	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro
400002	... 1/	400002	Network-Published	UInt16	<input checked="" type="checkbox"/>	50	<input checked="" type="checkbox"/>	read/write	Pro

**Figura 3-52** Editor múltiple de variables.  
Fuente: (Autores, 2017)



**Figura 3-53** Colocación de variables Modbus dentro del VI.  
Fuente: (Autores, 2017)

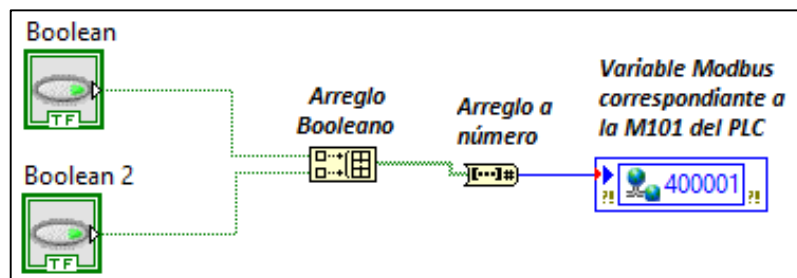
Crear indicadores para todas las variables correspondientes a entradas discretas y bobinas de nuestro PLC. Estos indicadores serán del tipo booleano y se activarán cuando el PLC reciba señales en sus entradas o cuando se activen las cargas o salidas del mismo.



**Figura 3-54** Variables Modbus de un bit con sus indicadores booleanos.

**Fuente:** (Autores, 2017)

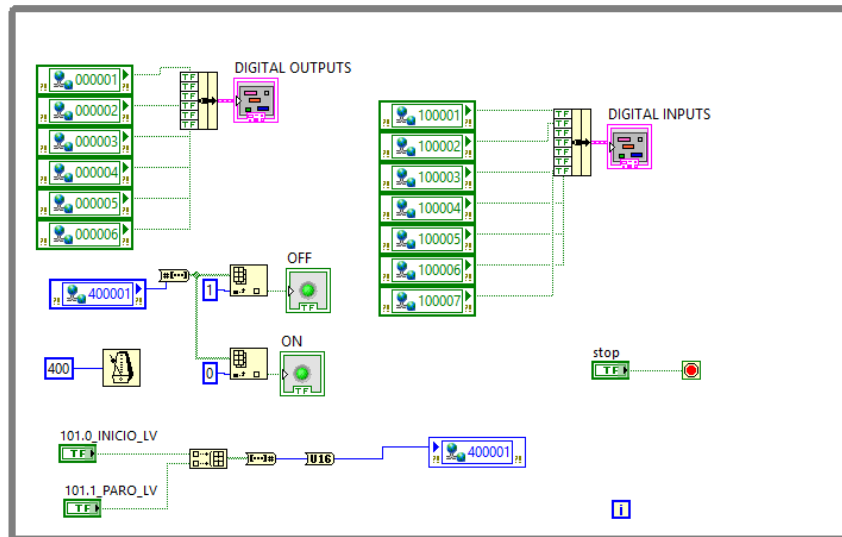
A continuación, realizar un arreglo booleano con dos pulsadores, estos pulsadores equivalen a escribir la memoria M101.0 y M101.1, las mismas que han sido asignadas dentro del proceso para activarlo y desactivarlo respectivamente.



**Figura 3-55** Configuración para la escritura de M101.0 y M101.1 en el PLC.

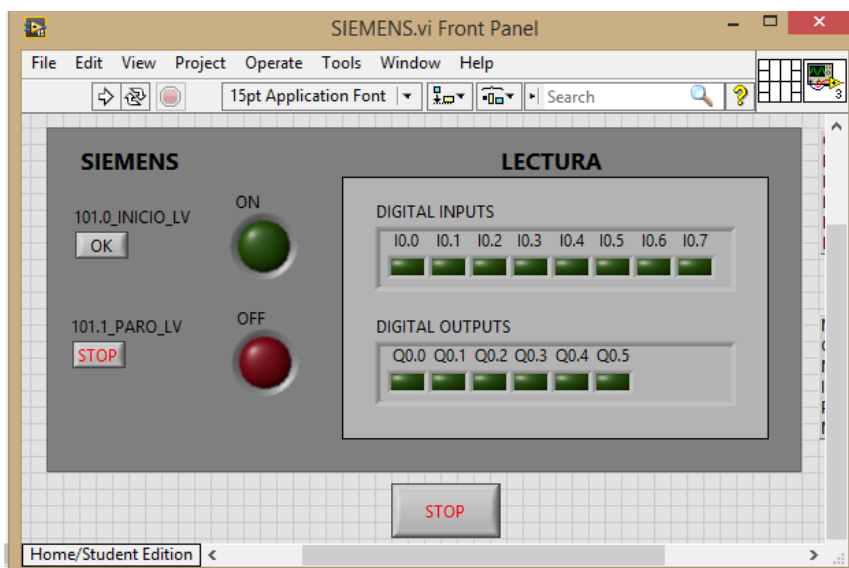
**Fuente:** (Autores, 2017).

Todos los elementos creados se colocan dentro de un While Loop, se agrega un botón de paro y un metrónomo para la ejecución de dicho While Loop.



**Figura 3-56** Panel de Diagrama de Bloques para la comunicación del PLC S7-1200 SIEMENS.  
**Fuente:** (Autores, 2017)

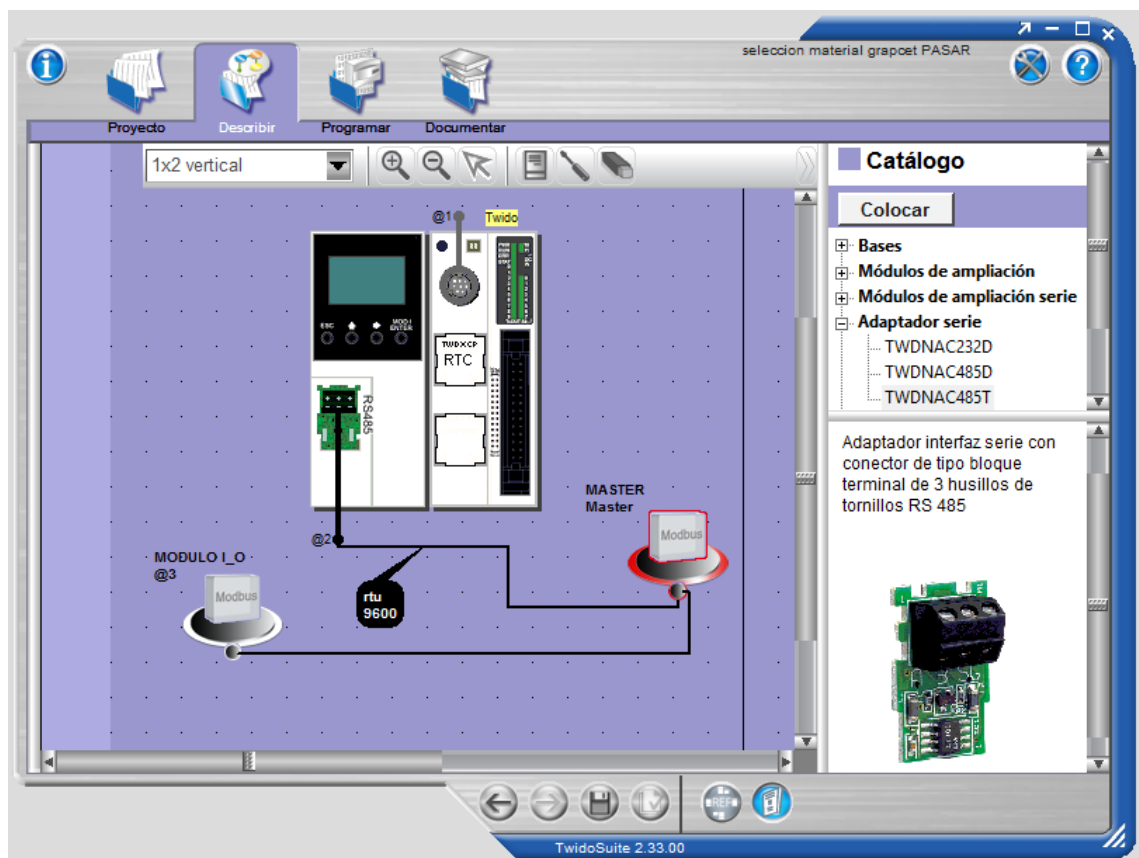
Finalmente dirigirse al panel frontal del VI. En el mismo que se han acomodado los controles e indicadores para dar la apariencia del PLC SIEMENS. Diríjase a file y escoja guardar todo (save all). Esto guardara por completo su VI y las librerías Modbus agregadas.



**Figura 3-57** Panel frontal comunicación Modbus TCP PLC S7-1200  
**Fuente:** (Autores, 2017).

Verificar la conexión del PLC con la PC y ejecutamos el VI. Pulsar el botón inicio o paro para escribir datos en el PLC y observar como los indicadores reciben datos de igual manera.

**3.5.2 Comunicación Modbus entre PLC Twido y LabVIEW.** Para verificar el funcionamiento de la configuración Modbus RTU 485 se realizó un programa en el Software TwidoSuite, en este programa se configura los elementos de la red y los parámetros de la misma. Se asigna una dirección maestro o esclavo al PLC y a todos los elementos Modbus y se carga el programa al PLC.

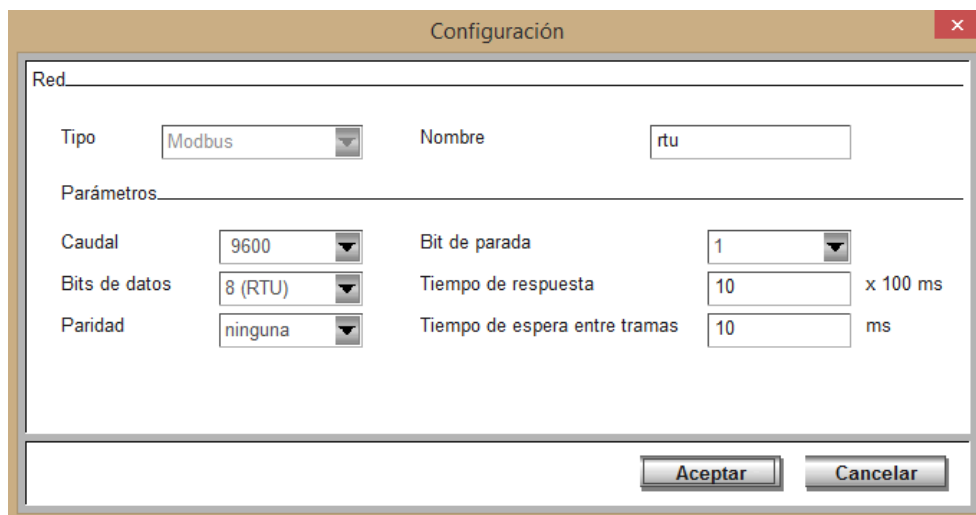


**Figura 3-58** Configuración elementos Modbus 485 en TwidoSuite  
**Fuente:** (Autores, 2017).

**Tabla 3-12** Parámetros MODBUS.

PLC TWIDO.	
<b>Serial type</b>	RTU SERIAL 485
<b>Unit ID</b>	2 (Dirección Esclavo).
<b>Baud rate</b>	9600
<b>Parity</b>	none

**Fuente:** (Autores, 2017)



**Figura 3-59** Configuración parámetros de la red Modbus.

**Fuente:** (Autores, 2017).

Comprobar la comunicación Modbus al visualizar en LabVIEW el funcionamiento de sus memorias internas, permitiendo leerlas y escribirlas con la ayuda del software. Para la lectura y/o escritura de cada entrada y salida física del PLC se asignó dicha entrada/salida a una memoria interna, es decir la memoria asignada se activará o desactivará cuando las entradas y salidas especificadas lo hagan. Además, agregar las memorias internas que se encargarán del inicio y paro del programa del PLC desde LabVIEW.

**Tabla 3-13** Direccionamiento de las Entradas físicas del PLC Twido a Modbus API de LabVIEW.

<b>PLC Twido</b>	<b>Memoria interna asignada</b>	<b>Tipo de dato</b>	<b>Modo de acceso permitido desde LabVIEW</b>
<b>I0.0</b>	M0	Bit Booleano	Lectura
<b>I0.1</b>	M1	Bit Booleano	Lectura
<b>I0.2</b>	M2	Bit Booleano	Lectura
<b>I0.3</b>	M3	Bit Booleano	Lectura
<b>I0.4</b>	M4	Bit Booleano	Lectura
<b>I0.5</b>	M5	Bit Booleano	Lectura
<b>I0.6</b>	M6	Bit Booleano	Lectura
<b>I0.7</b>	M7	Bit Booleano	Lectura
<b>I0.8</b>	M8	Bit Booleano	Lectura
<b>I0.9</b>	M9	Bit Booleano	Lectura
<b>I0.10</b>	M10	Bit Booleano	Lectura
<b>I0.11</b>	M11	Bit Booleano	Lectura

**Fuente:** (Autores, 2017).

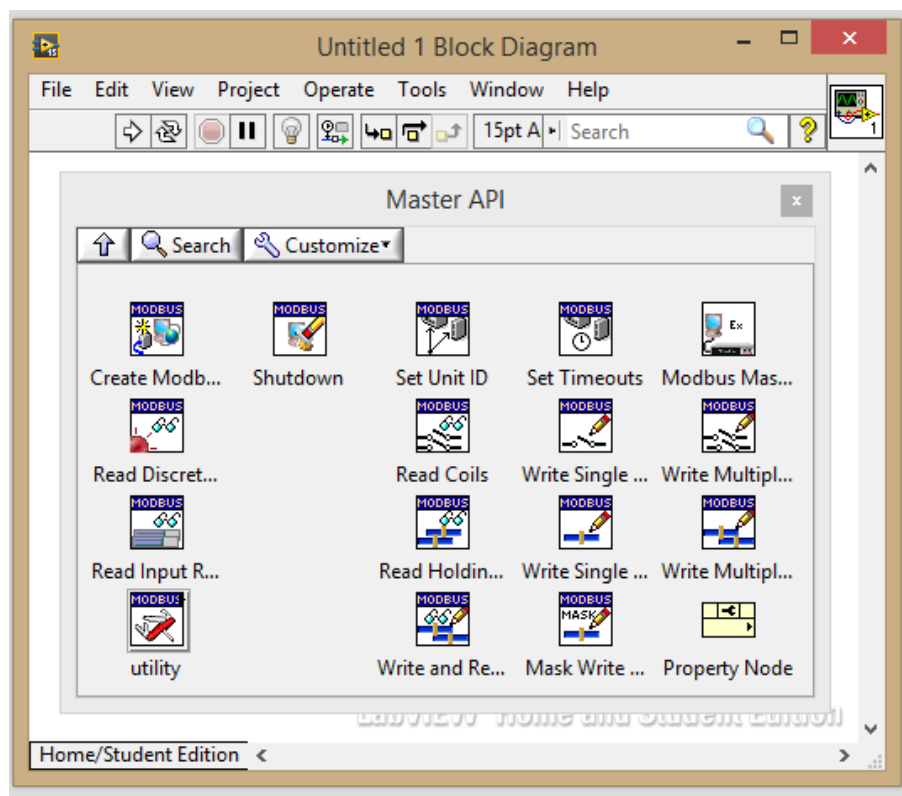
**Tabla 3-14** Direcccionamiento de las Salidas físicas del PLC Twido a Modbus API de LabVIEW.

PLC Twido	Memoria interna asignada	Tipo de dato	Modo de acceso permitido desde LabVIEW
Q0.0	M20	Bit Booleano	Lectura
Q0.1	M21	Bit Booleano	Lectura
Q0.2	M22	Bit Booleano	Lectura
Q0.3	M23	Bit Booleano	Lectura
Q0.4	M24	Bit Booleano	Lectura
Q0.5	M25	Bit Booleano	Lectura
Q0.6	M26	Bit Booleano	Lectura
Q0.7	M27	Bit Booleano	Lectura

Fuente: (Autores, 2017).

**NOTA:** Las memorias del tipo MW se leerán como bytes bajo el formato UInt16.

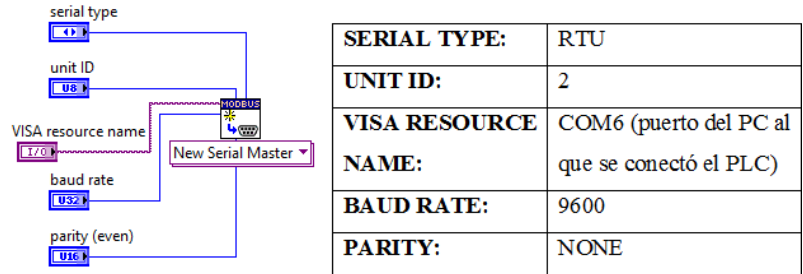
Para realizar esta comunicación efectuamos lo pasos para configuración de un controlador API de LabVIEW. Crear un nuevo VI, luego dirigirse a la ventana Diagrama de Bloques damos clic derecho en la misma, dirigirse a ADDONS y seleccionar la librería Modbus y escoger Maestro API. Fijar dicha librería a la ventana de LabVIEW ya que se agregará varios elementos de la misma a la programación.



**Figura 3-60** Elementos Master API de la librería NI Modbus.

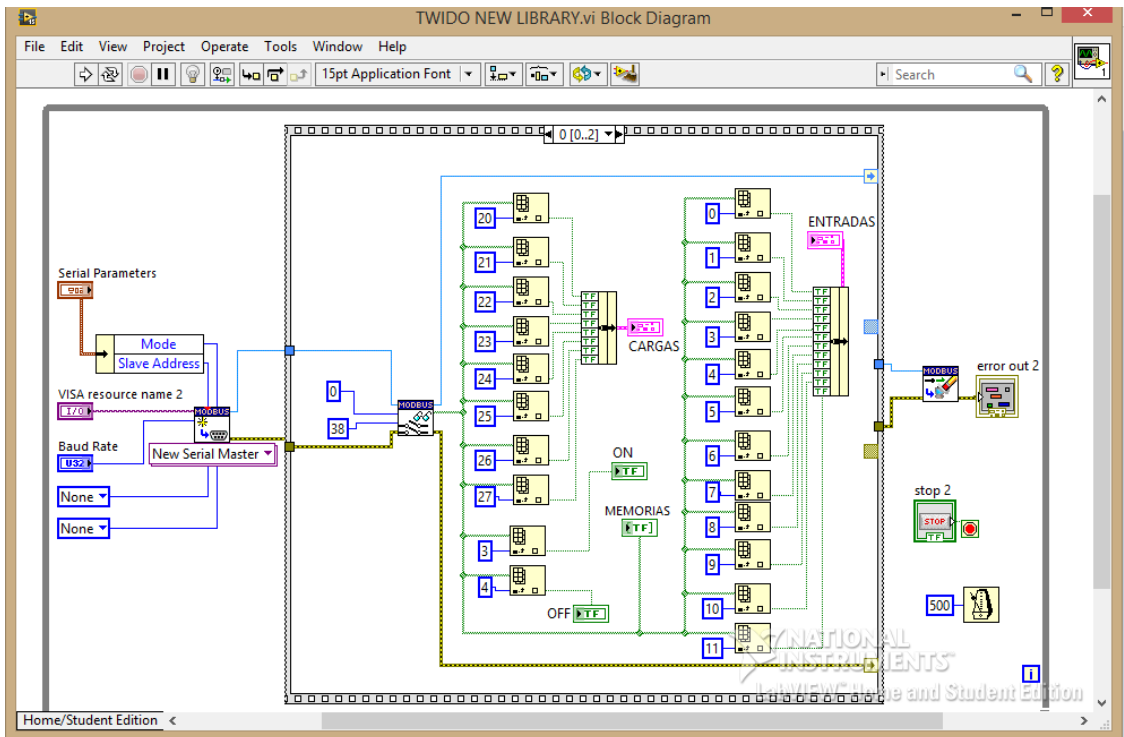
Fuente: (Autores, 2017).

Seleccione el elemento Create Modbus Instance arrástrelo en la pantalla de LabVIEW, en la lista desplegable escoja New Master Serial, cree los controles para el mismo y establezca los valores default en base a los datos de la red configurada.



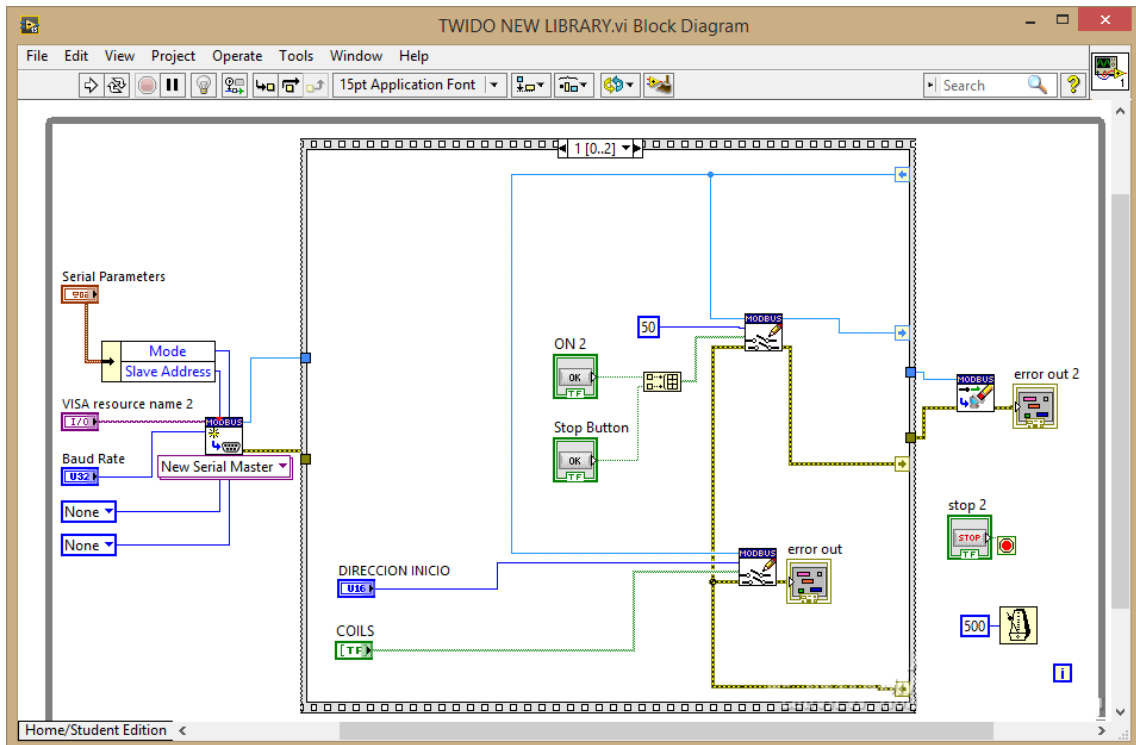
**Figura 3-61** Configuración del elemento new master serial con los parámetros de la red.  
**Fuente:** (Autores, 2017).

Tal y como se indicó en los pasos para configurar Modbus RTU 485 con API de LabVIEW coloque los elementos necesarios de la librería para leer y/o escribir los bits de las memorias M(booleanos) y para leer los Bytes de memorias MW(UInt16). Para facilidad de visualización transforme los tres Frames del Flat Sequence en un Stacked Sequence Structure, el cual despliega una ventana donde seleccionará el Frame deseado. Ahora cree los controles para cada elemento Modbus agregado con las direcciones de inicio correspondientes a las memorias del PLC mencionadas anteriormente.



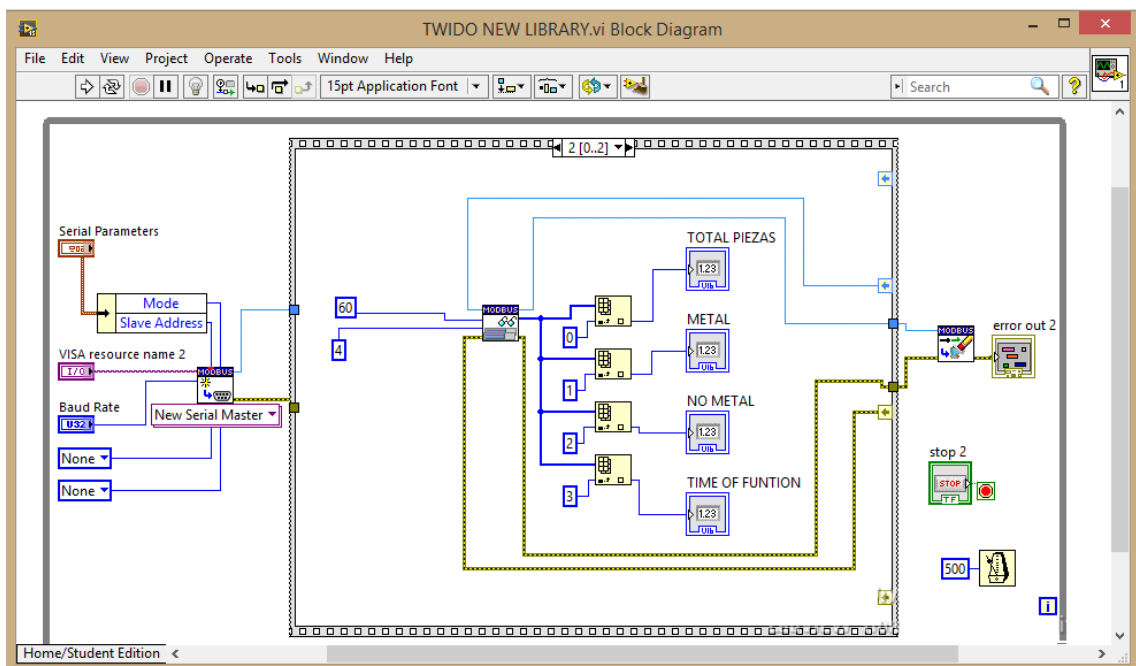
**Figura 3-62** Creación de controles e indicadores en READ COILS.  
**Fuente:** (Autores, 2017).





**Figura 3-63** Creación de controles e indicadores en WRITE MULTIPLE REGISTERS para la escritura de bits.

**Fuente:** (Autores, 2017).

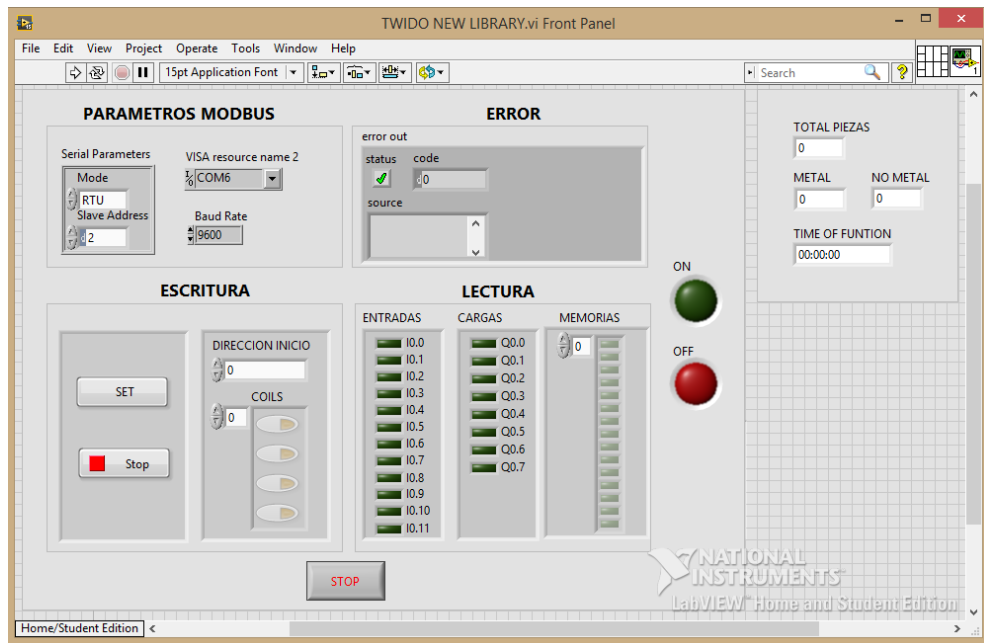


**Figura 3-64** Creación de controles e indicadores en READ REGISTERS.

**Fuente:** (Autores, 2017).

En la última ventana del Stacked Sequence Structure crear un Shut Down Modbus (Finalizar Instancia Modbus). Todos estos elementos fueron agregados dentro de un While Loop, el mismo que consta con un botón de Stop y una constante de tiempo de

ejecución del mismo. En la vista frontal de LabVIEW se ubicaron los elementos de control e indicadores para dar una apariencia similar a un PLC.



**Figura 3-65** Panel frontal del VI en LabVIEW para el PLC Twido.  
**Fuente:** (Autores, 2017).

De clic en File >> Guardar VI. Ahora su programación esta lista para su uso. Antes de ejecutar el VI asegúrese de que los terminales A y B del puerto 485 del PLC se encuentren conectados a los terminales A y B del adaptador 485 conectado a un puerto con de su PC. Coloque el valor del puerto COM al que se encuentra conectado el PLC y la dirección Modbus Esclavo correspondiente.

El PLC Twido debe colocarse en modo RUN para que el software pueda acceder a la comunicación. Ahora puede ejecutar el VI y comprobar el funcionamiento de la comunicación Modbus al visualizar las señales de entradas y salidas físicas del PLC, podrá escribir o visualizar cualquier memoria M de 1bit, es decir enviar y recibir señales booleanas y además podrá leer y/o escribir bytes de memorias del tipo MW(16bits).

**3.5.3 Comunicación Modbus entre el Módulo I/O Modbus y LabVIEW.** Para la configuración en LabVIEW de Modbus RTU 485 para el Módulo I/O Modbus se utilizará el mismo controlador API que se ocupó para la configuración del PLC Twido solo que para este caso ya se tienen definidos todos los parámetros a leer y escribir que corresponderán a todas las entradas y salidas lógicas físicas del módulo.

**Tabla 3-15** Bytes asignados para entradas y salidas digitales del MÓDULO I/O Modbus.

Byte módulo I/O.	Tipo de dato	Modo de acceso permitido desde LabVIEW	Características
0	UInt16	Lectura	Este byte enviara las señales booleanas de los interruptores del Módulo I/O. Para la lectura en LabVIEW se utilizará el elemento READ COILS de la librería API. En la dirección de inicio se coloca el bit desde el cual se quieren empezar a leer los valores y la longitud corresponderá al número de bits que se quieren leer. <b>NOTA:</b> Para el byte 0 el rango de bits a leer va desde el bit 0 al bit 15. (16 datos booleanos).
1	UInt16	Escritura	Este byte recibirá por Modbus las señales booleanas que activaran los indicadores del módulo I/O. En el elemento WRITE MULTIPLE COILS se ingresa un arreglo con controles booleanos, siendo la dirección de inicio el bit en el cual se empezarán a escribir los datos a través de LabVIEW. <b>NOTA:</b> Para el byte 1 el rango de bits que se escribirá va desde el bit 16 hasta el bit 31. (16 datos booleanos).

**Fuente:** (Autores, 2017)

Para la comunicación Modbus de cada entrada analógica se ocupa un byte completo, asignando así cada elemento analógico a partir del byte 2 de nuestro dispositivo.

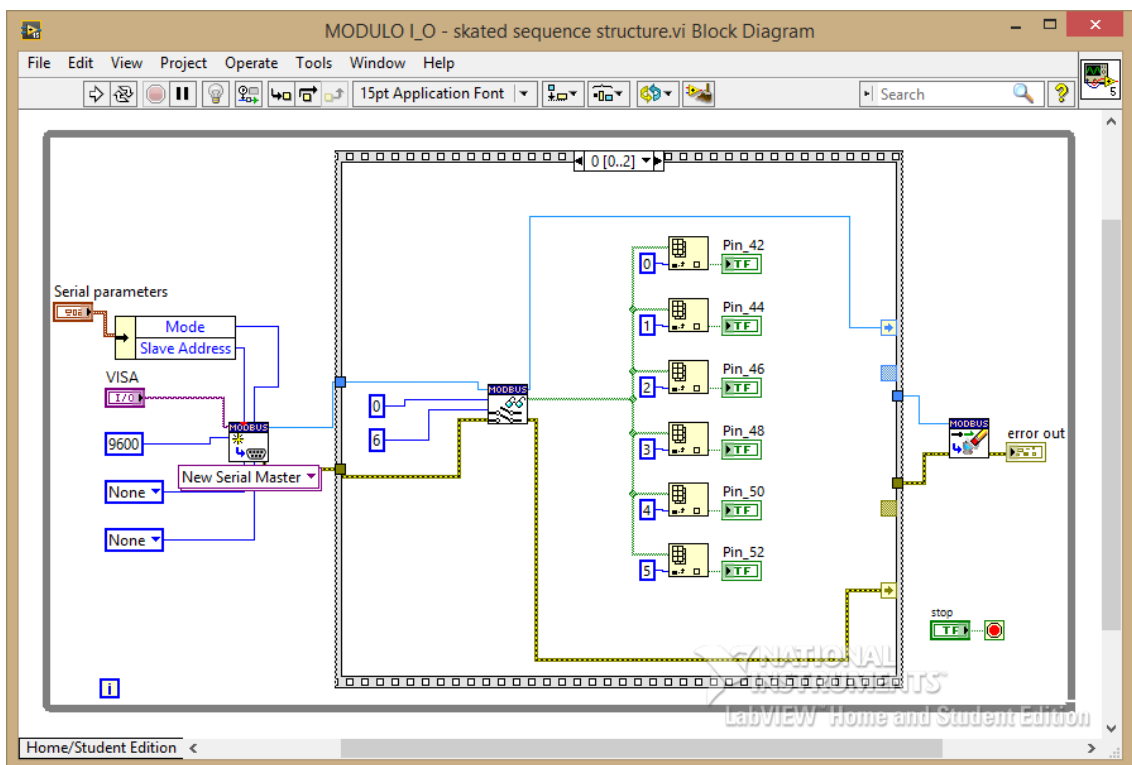
**Tabla 3-16** Bytes asignados para la lectura de las entradas analógicas del Módulo I/O Modbus.

Byte módulo I/O.	Tipo de dato	Modo de acceso permitido desde LabVIEW	Características
2	UInt16	Lectura	Cada byte dentro de este rango enviara una señal analógica de voltaje en formato decimal de un potenciómetro del Módulo I/O. Para la lectura en LabVIEW se utilizará el elemento READ MULTIPLE REGISTERS de la librería API. En la dirección de inicio se coloca el BYTE desde el cual se quieren empezar a leer los datos numéricos y la longitud corresponderá al número de bytes que se quieren leer.
3	UInt16	Lectura	
4	UInt16	Lectura	
5	UInt16	Lectura	
6	UInt16	Lectura	Este byte corresponderá a la señal analógica de voltaje de un sensor de Luz LDR.
7	UInt16	Lectura	Este byte corresponderá a la señal analógica de voltaje de un sensor de Temperatura LM35.

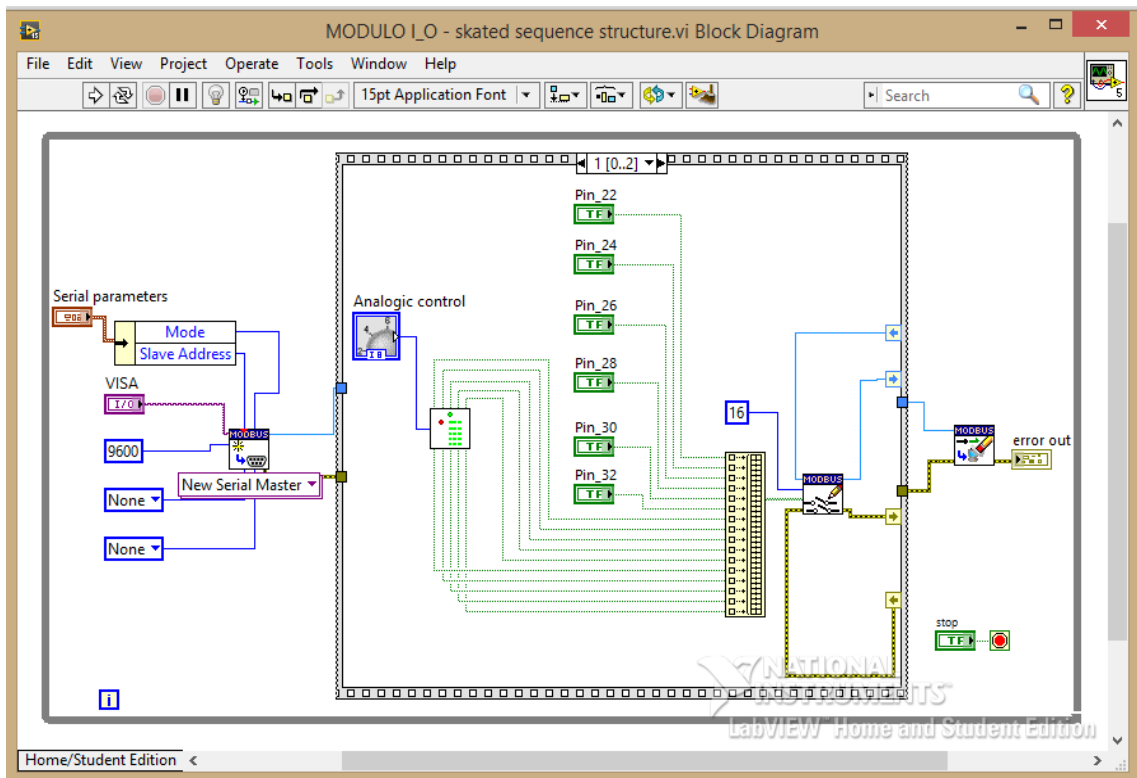
**Fuente:** (Autores, 2017)



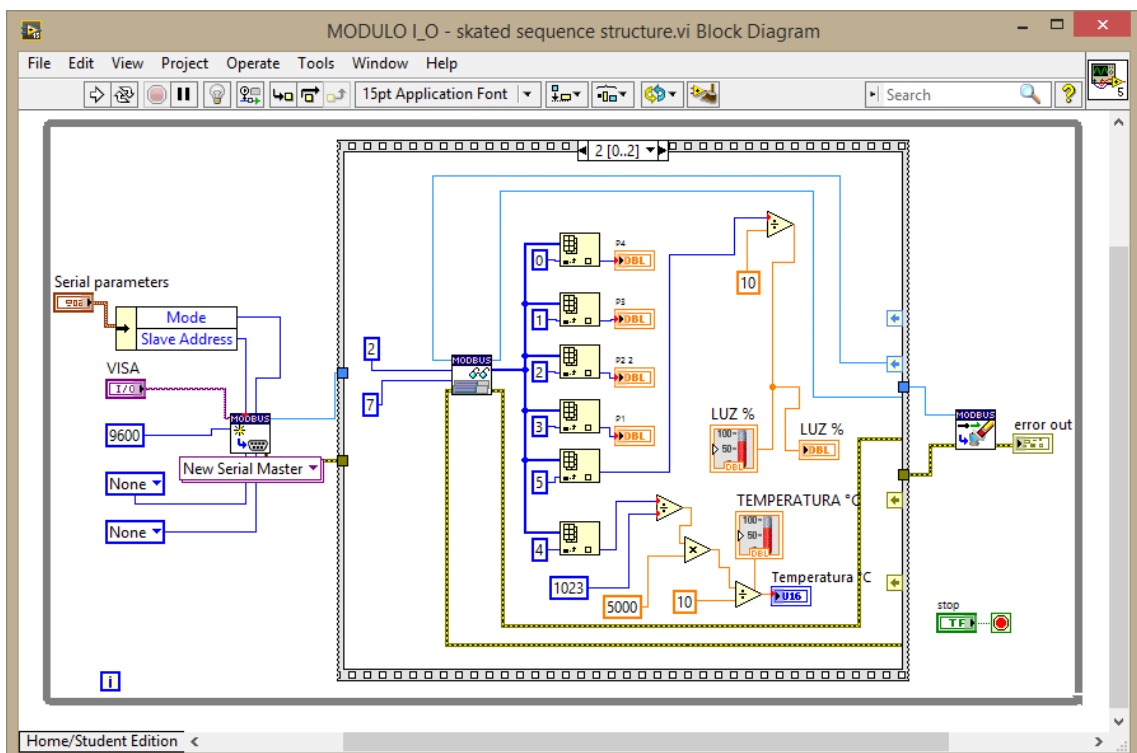
**Figura 3-66** Vista Frontal Módulo I/O Modbus con la asignación de bytes/bits.  
**Fuente:** (Autores, 2017)



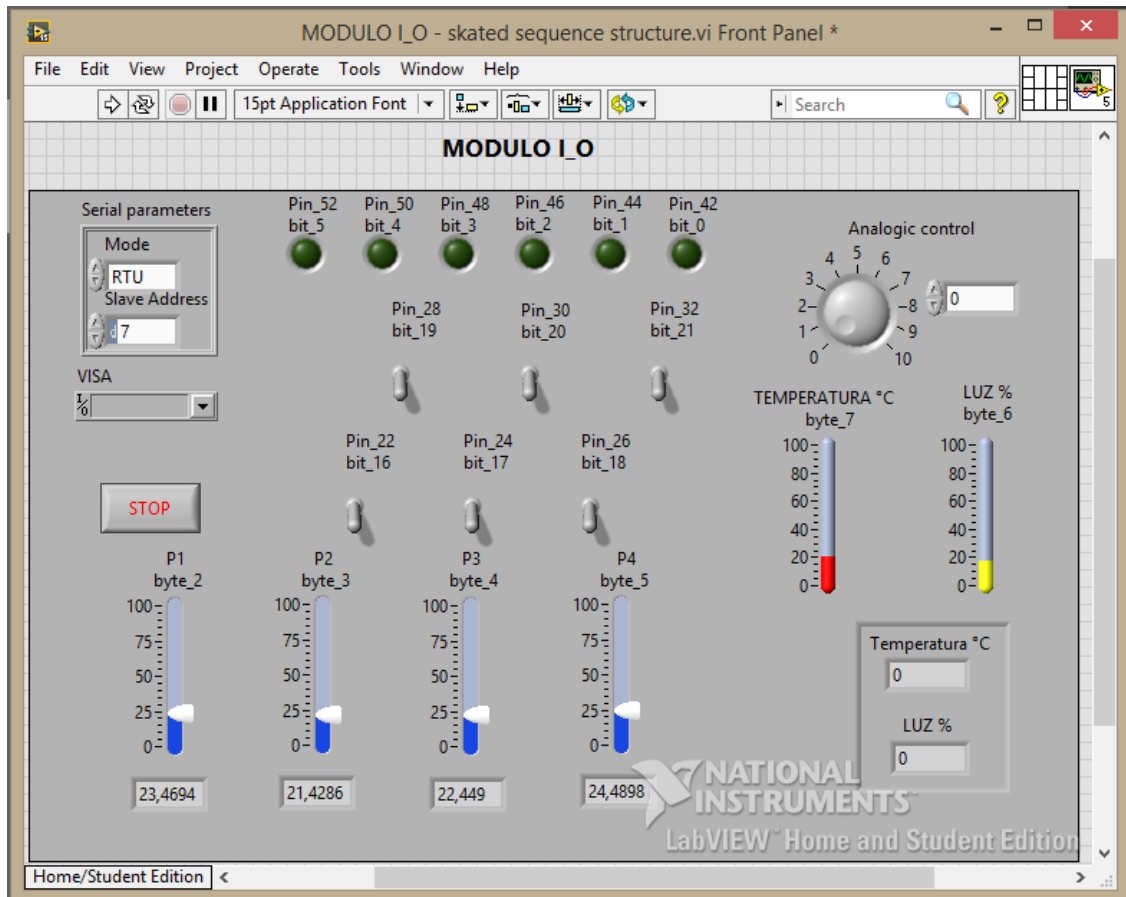
**Figura 3-67** Vista de diagrama de bloques Skated Sequence 0 lectura de bits señales digitales.  
**Fuente:** (Autores, 2017)



**Figura 3-68** Vista de diagrama de bloques Skated Sequence 1 escritura de bits señales digitales.  
**Fuente:** (Autores, 2017)

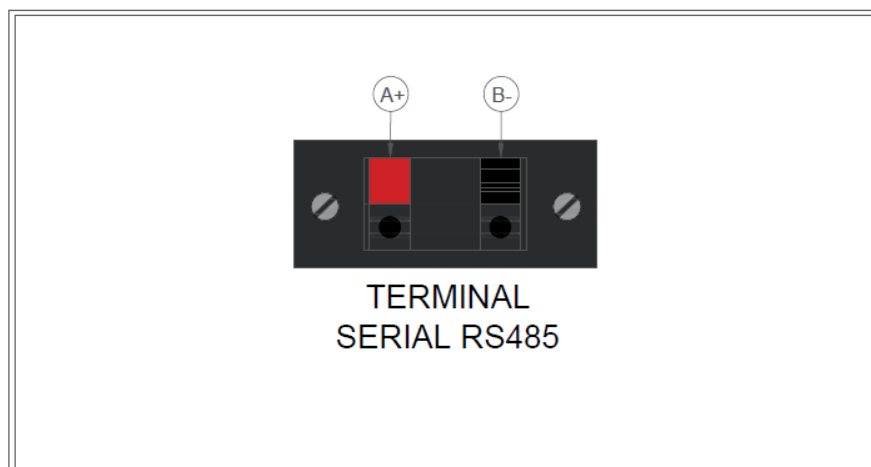


**Figura 3-69** Vista de diagrama de bloques Skated Sequence 2 lectura de señales analógicas.  
**Fuente:** (Autores, 2017)



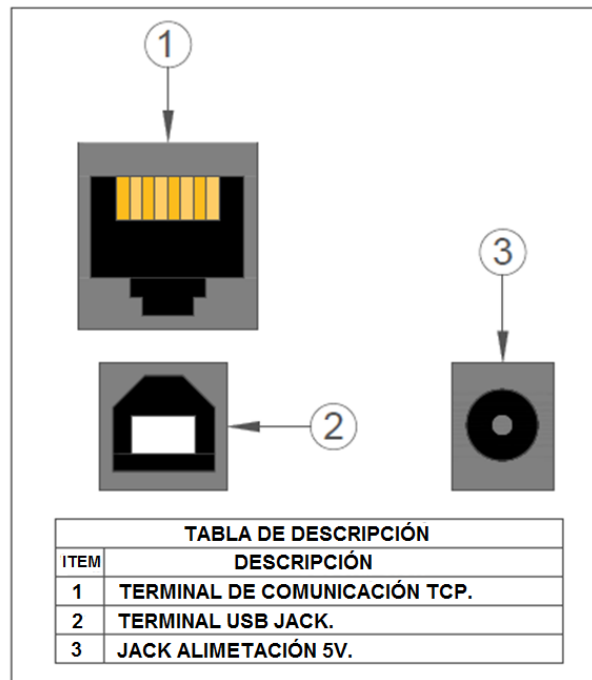
**Figura 3-70** Vista Frontal con controles e indicadores para comunicación Modbus.  
**Fuente:** (Autores, 2017)

Antes de ejecutar el VI asegúrese que los terminales A y B del puerto serial RS 485 se del módulo encuentren conectados a los terminales A y B del adaptador conectado al puerto USB de su PC.



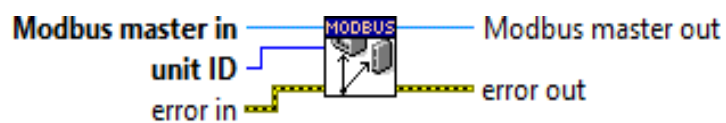
**Figura 3-71** Puerto serial Modulo I/O Modbus.  
**Fuente:** (Autores, 2017)

La alimentación eléctrica del Módulo I/O Modbus puede realizarse a través del Jack de alimentación o por el terminal USB Jack.

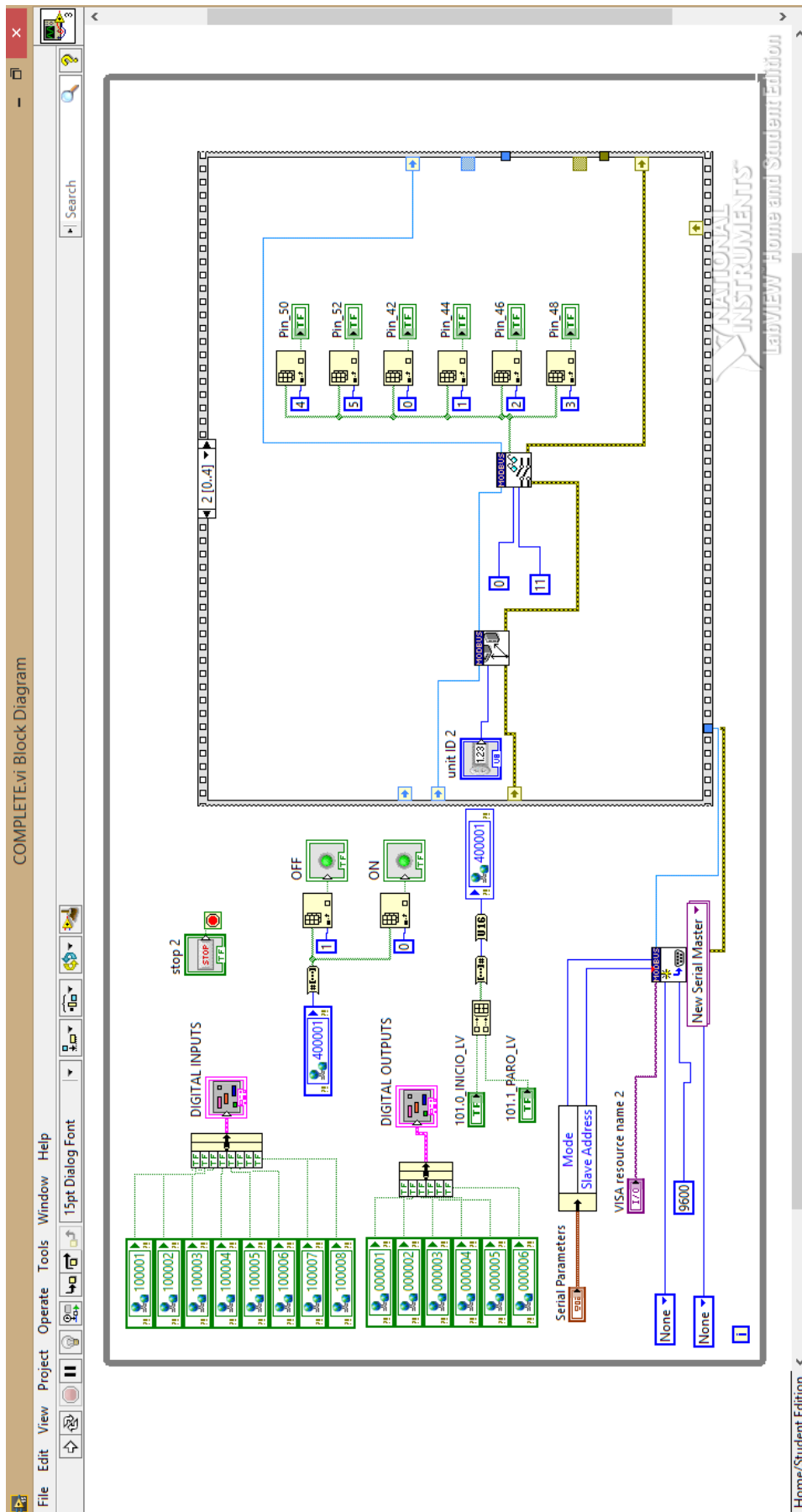


**Figura 3-72** Puertos de conexión/alimentación Módulo I/O Modbus.  
Fuente: (Autores, 2017)

**3.5.4 Comunicación Modbus entre PLCs y Módulo I/O Modbus con LabVIEW.** Para realizar este VI se partirá de los anteriormente creados para cada dispositivo. Primero realizar una copia del proyecto que se creó con la librería I/O server Modbus para el PLC Siemens y crearemos un nuevo VI dentro de este proyecto. Abrir los VI creados para cada dispositivo en su vista de diagrama de bloques y copiar todos los elementos en esta vista hacia el nuevo VI creado. Acoplar el Stacked Sequence copiados del VI del Módulo I/O Modbus al Stacked Sequence del VI del Twido. Utilizar el elemento Set Unit ID al final de la secuencia Modbus del Twido para empezar la secuencia del Módulo I/O Modbus. De esta manera ambos dispositivos se comunicarán al mismo tiempo ya que se establecieron las direcciones de esclavos correspondientes a cada uno.



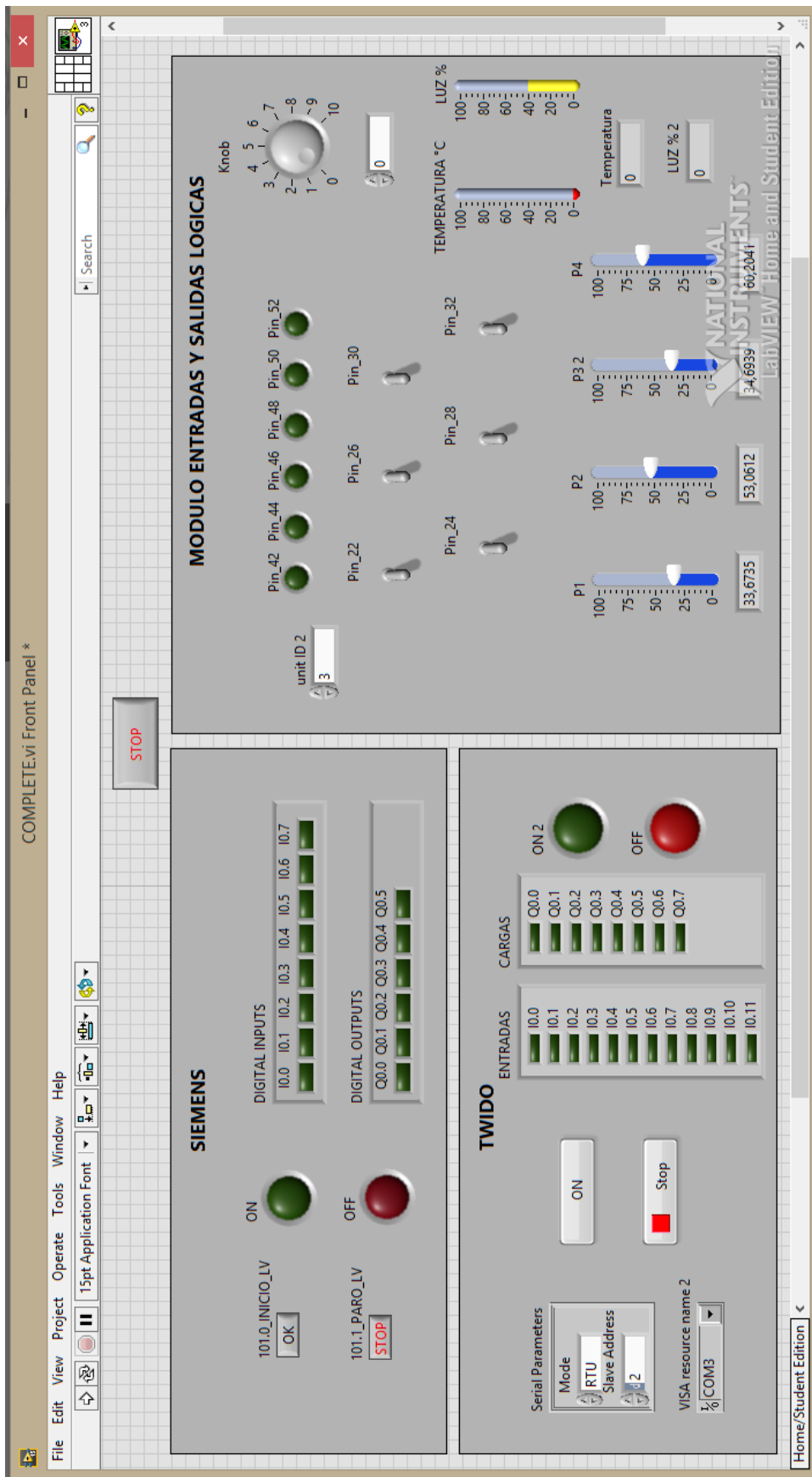
**Figura 3-73** Elemento Set Unit ID.  
Fuente: (Autores, 2017)



**Figura 3-74** Vista de diagrama de bloques VI con PLC Siemens, Twido y Módulo I/O Modbus.

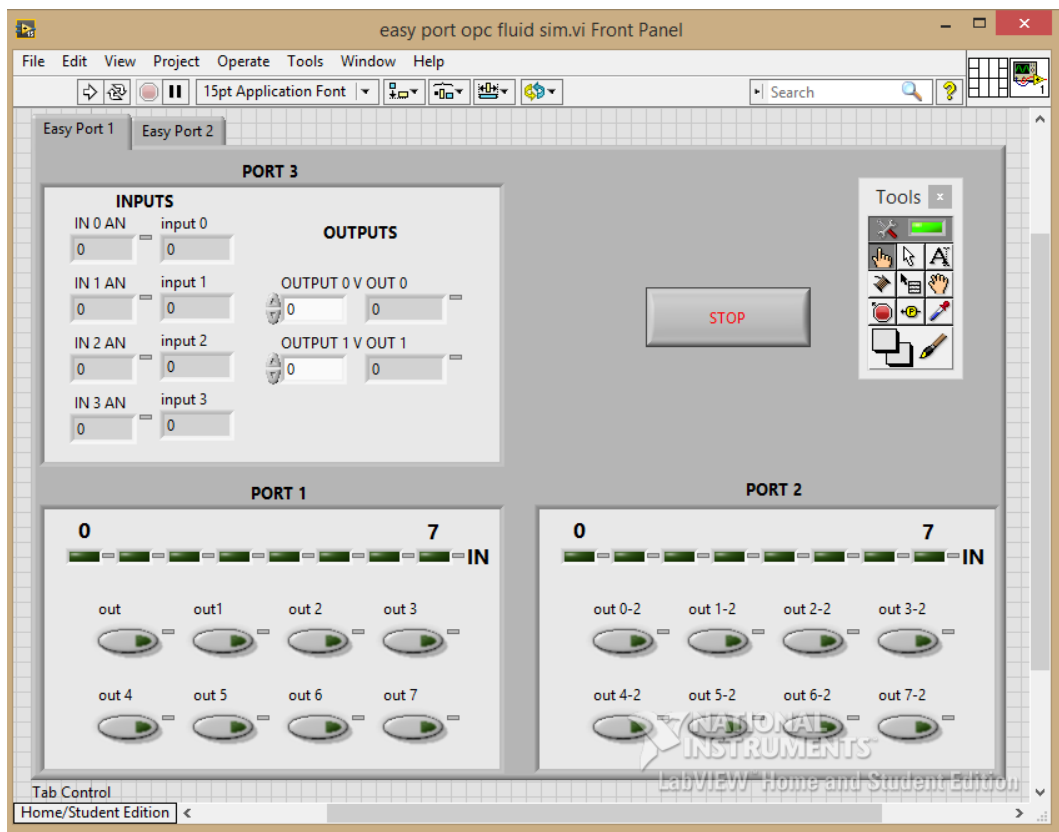
**Fuente:** (Autores, 2017)





**Figura 3-75** Vista frontal VI con PLC Siemens, Twido y Módulo I/O Modbus.  
**Fuente:** (Autores, 2017)

**3.5.5 Comunicación entre Easy Port y LabVIEW por OPC server.** Primero crear un nuevo VI, luego para este caso colocar dentro de la pantalla frontal todos los elementos booleanos y numéricos correspondientes a los elementos de entrada/salida existentes en los puertos del Easy Port. Se enlaza cada elemento con la dirección OPC de Festo, para ello dar clic derecho sobre el elemento, escoger propiedades, luego data binding, seleccionar el tipo de acceso de datos adecuado y en browser se escoge la librería EzOPC y allí elegir la dirección OPC correspondiente a cada elemento del EasyPort.

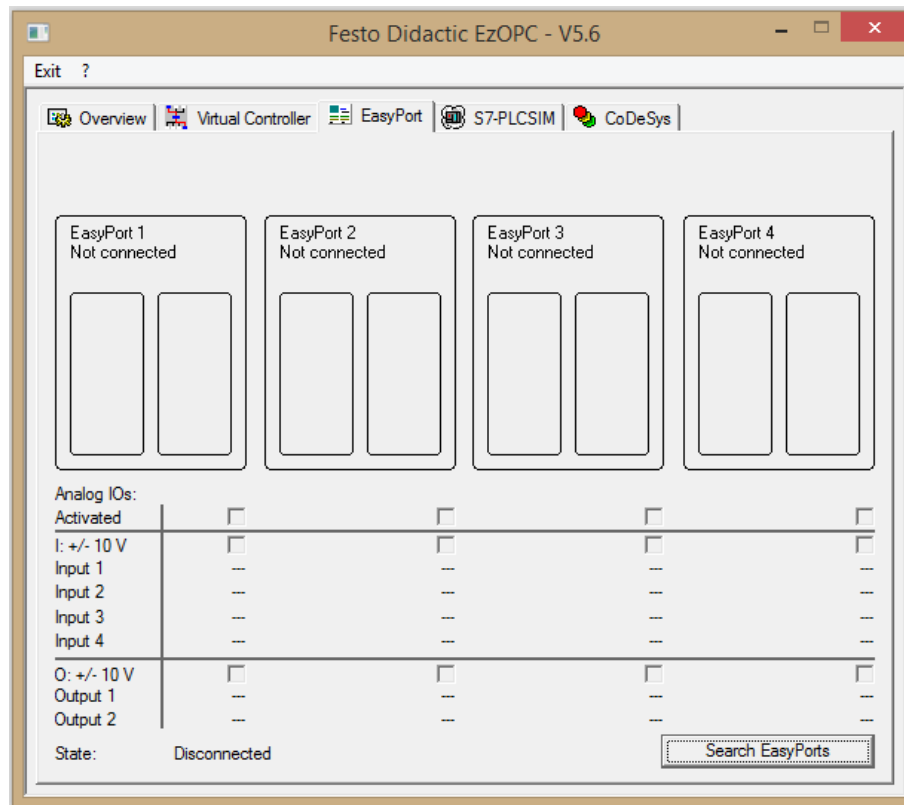


**Figura 3-76** Vista frontal VI Easy Port a través de OPC.

**Fuente:** (Autores, 2017)

Diríjase a la vista de diagrama de bloques y colocar dentro de un while loop todos los elementos.

Las señales analógicas se escriben y leen en formato decimal correspondiente a palabras de 15 bits (32768). Este rango numérico de 0 - 32768 corresponderá al rango de voltaje comprendido entre 0 y 10V, o al rango de -10V a +10V, si así se configurase en la ventana EasyPort del controlador OPC de Festo.



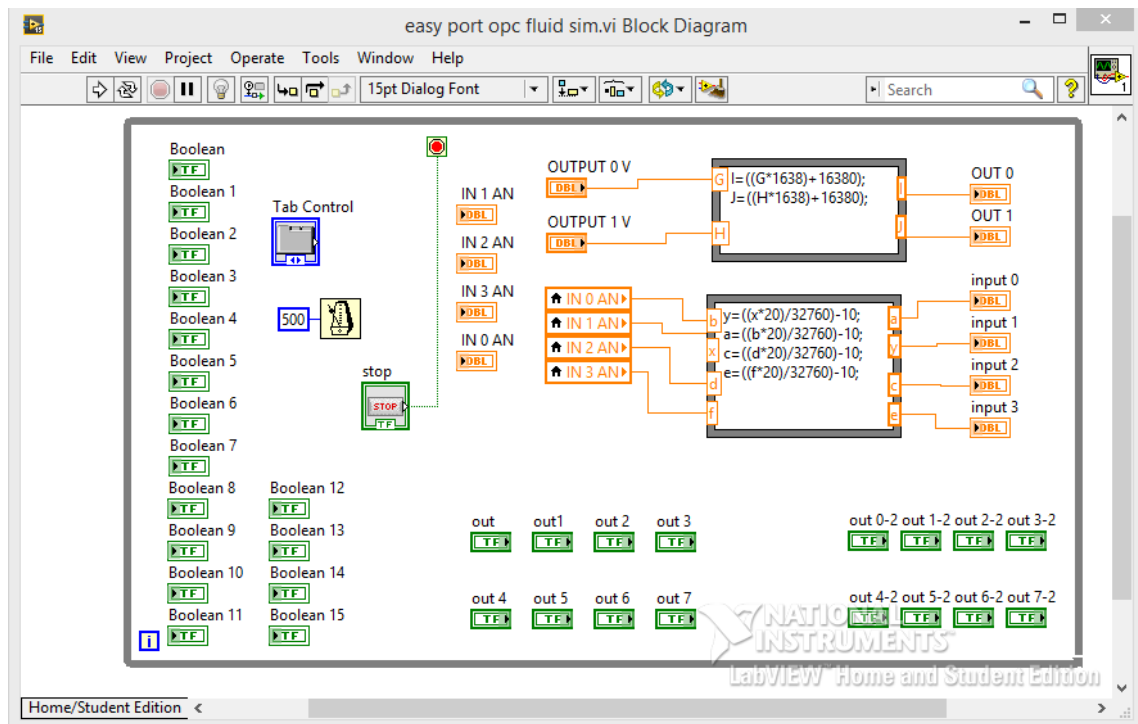
**Figura 3-77** EasyPort en EzOPC.  
**Fuente:** (Autores, 2017)

Para activar la lectura o escritura de las señales analógicas se debe marcar la casilla Activated. Y para configurar las señales para rangos de -10 a 10 V se selecciona la casilla I: +/- 10V para entradas o la casilla O: +/- 10V para las salidas analógicas. Si no se marcarse ninguna se ellas el rango predeterminado será de 0 – 10V.

Analog IOs:	
Activated	<input type="checkbox"/>
I: +/- 10 V	<input type="checkbox"/>
Input 1	--
Input 2	--
Input 3	--
Input 4	--
O: +/- 10 V	<input type="checkbox"/>
Output 1	--
Output 2	--

**Figura 3-78** Entradas/Salidas analógicas EasyPort en EzOPC.  
**Fuente:** (Autores, 2017)

Se realizará en LabVIEW la operación matemática adecuada para transformar los valores numéricos de los analógicos de EasyPort a valores de voltaje.



**Figura 3-79** Vista de diagrama de bloques del VI Easy Port a través de OPC.  
**Fuente:** (Autores, 2017)

Para este VI se realizó la conversión de decimal a voltios seleccionando el rango de -10 a +10V tanto para entradas como para salidas analógicas, teniendo así que el rango numérico de 0 – 32768 corresponderá a 20 voltios (-10 a +10V).

Para lectura de señales analógicas en voltios se tiene que:

$$IV = \left( \frac{(IN \cdot 20)}{32768} \right) - 10$$

**Donde:** IV = Valor en voltios de la entrada analógica.

IN = Valor numérico de los bits de la entrada analógica.

Para la escritura de señales analógicas ingresar el valor en voltios el cual se transformará al formato decimal y se enviará al EasyPort

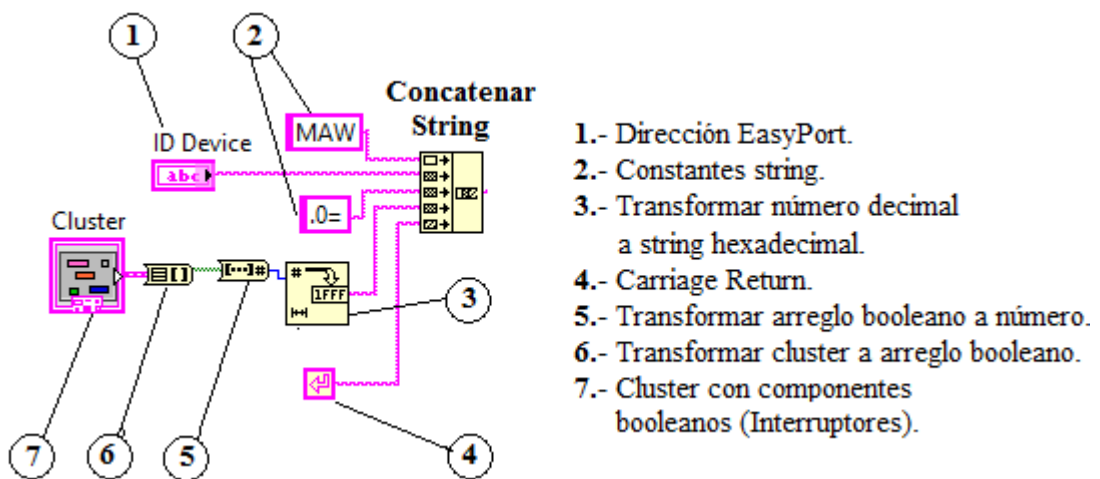
$$OV = (ON \cdot 1638,4) + 16384$$

**Donde:** OV = Valor en voltios a escribir en la salida analógica.

ON = Valor numérico correspondiente al voltaje a enviar al EasyPort.

**3.5.6 Comunicación entre EasyPort y LabVIEW por comunicación Serial.** Para esta comunicación se realizará dos VI diferentes, el primero para leer todas las señales digitales y el segundo para leer las señales analógicas de un EasyPort con cualquier dirección de módulo admisible.

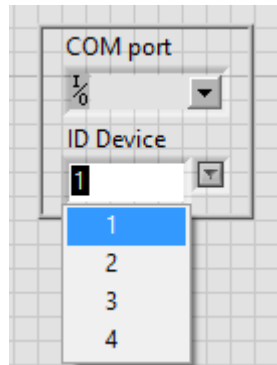
El primer VI nos permitirá leer todas las entradas digitales del EasyPort y escribir/leer todas las salidas digitales del mismo. Para ello se utilizará los códigos DEW1.0 para la lectura y MAW1.0=XX para escritura. El código de lectura se colocará como una constante String y así se obtendrá todas las lecturas de las entradas digitales de los Puertos 1 y 2 del EasyPort, mientras que para el código de escritura mediante la función concatenar se colocará como constante solo partes del código y se ingresará un valor numérico correspondiente al número de módulo y valores alfanuméricos después del igual correspondientes a los bits a escribir/leer. Gracias a las funciones de LabVIEW se puede obtener dichos valores alfanuméricos en formato Hexadecimal a partir de un arreglo o clúster booleano. Para finalizar la función concatenar se debe agregar la opción Carriage Return que enviará el código a ser procesado. Para la función de lectura también se puede ingresar un valor mediante un combo box y la función concatenar para seleccionar la dirección del módulo correspondiente del cual se quieren leer todas las entradas digitales.



**Figura 3-80** Concatenar String para escritura de salidas digitales del EasyPort.  
Fuente: (Autores, 2017)

Mediante la función concatenar se obtiene el String completo en código ASCII correspondiente a los elementos booleanos del Clúster (en este caso interruptores) que activaran las salidas del EasyPort.

El combo Box ID Device permitirá escoger el número de módulo o EasyPort con el cual se realizará la comunicación.



**Figura 3-81** Combo box ID Device.  
**Fuente:** (Autores, 2017)

Cada elemento del Clúster activará una salida digital del EasyPort, debido a que el EasyPort se comunica bajo formato ASCII con códigos de letras para la función a realizar y números hexadecimales para la asignación de valores, transformar los elementos booleanos del clúster a un valor numérico decimal y dicho valor numérico se transformará al formato hexadecimal para ser escrito dentro del String que será enviado al EasyPort.



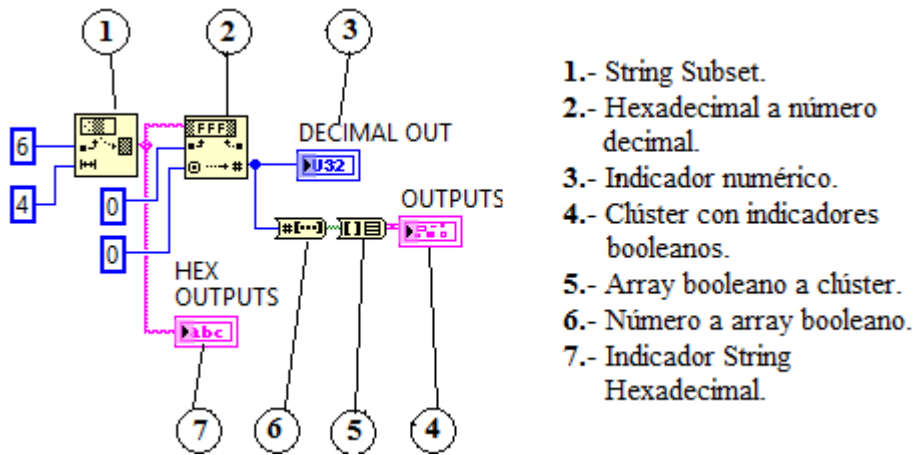
**Figura 3-82** Clúster con elementos booleanos a escribirse  
**Fuente:** (Autores, 2017)

Para la lectura de entradas o salidas digitales del EasyPort se realiza la operación inversa se parte de un valor hexadecimal y se lo descompone en señales booleanas que encenderán indicadores dentro de un clúster.

Para realizar esta conversión primero se extrae el valor hexadecimal del String que devuelve el EasyPort al PC después de ejecutarse cualquier acción en el mismo, para ello ocupamos el elemento String Subset, donde se debe colocar el número correspondiente a la ubicación de un carácter dentro del String y colocamos la cantidad de caracteres a extraer.

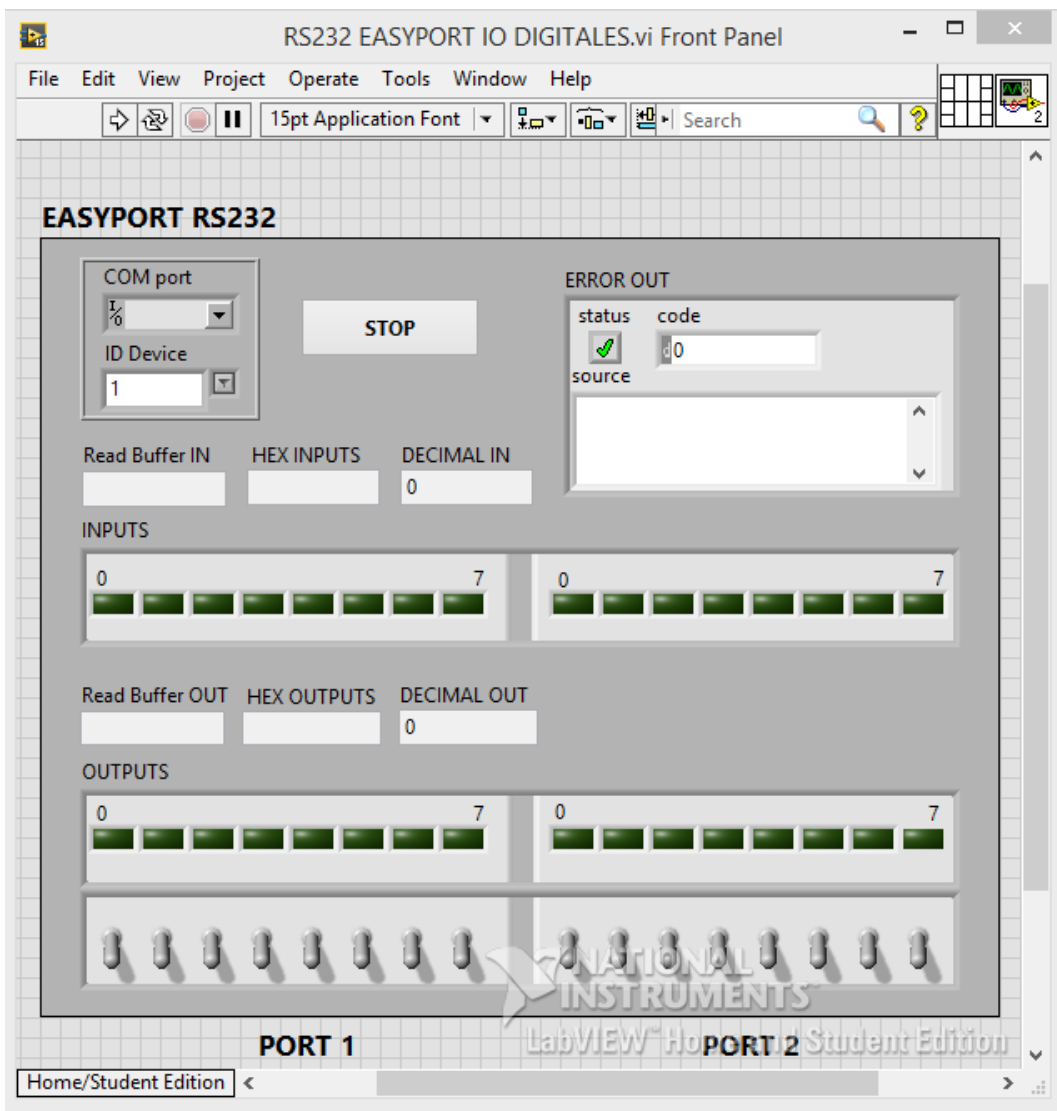
De esta manera se obtendrá otro String solo con los elementos que hayamos seleccionado.

En la figura 99 el clúster llamado OUTPUTS contiene los indicadores booleanos.



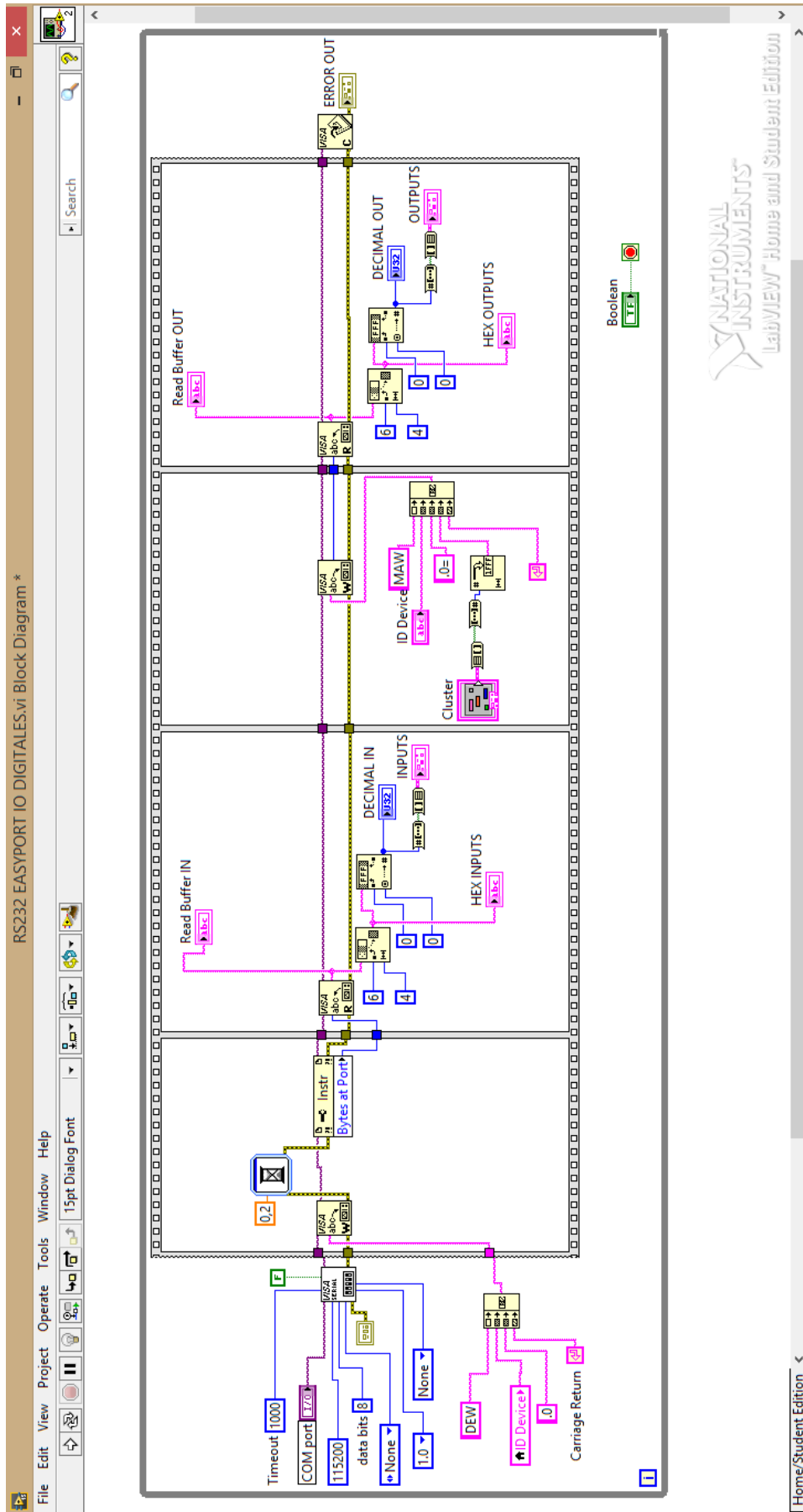
**Figura 3-83** Método convertir Hexadecimal a booleanos.

Fuente: (Autores, 2017)



**Figura 3-84** Vista Frontal VI comunicación EasyPort A través de RS232 para señales digitales

Fuente: (Autores, 2017)



**Figura 3-85** Diagrama de bloques de comunicación EasyPort RS232 para señales digitales.  
**Fuente:** (Autores, 2017)



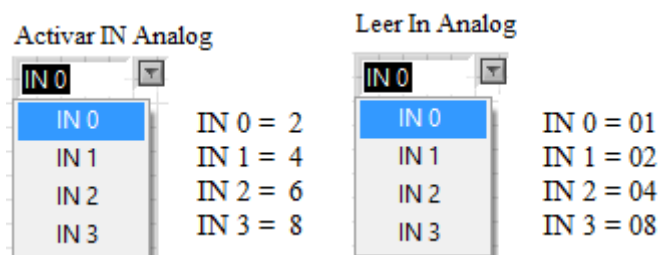
En el segundo VI se puede leer las señales analógicas de los cuatro canales existentes en el EasyPort y escribir cualquiera de las dos salidas analógicas del mismo. Para ello al igual que en el VI de las señales digitales del EasyPort utilizar la función concatenar, pero en este caso los códigos a escribir llevarán la dirección Word correspondiente a cada una de las entradas/salidas analógicas teniendo así que el código del tipo EW1.X activa las entradas analógicas seguido del código MS=XX para la recepción automática de los valores de las mismas y el código del tipo MAW1.X=XX activará el envío de datos hacia el EasyPort.

**Tabla 3-17** Códigos para las señales analógicas del EasyPort.

Orden	Respuesta	Descripción
DEW<n>.2	EW<n>.2=xxxx	Cargar canal de entrada 0.
DEW<n>.4	EW<n>.4=xxxx	Cargar canal de entrada 1.
DEW<n>.6	EW<n>.6=xxxx	Cargar canal de entrada 2.
DEW<n>.8	EW<n>.8=xxxx	Cargar canal de entrada 3.
MS<n>=01		Activar el canal de entradas analógicas 0
MS<n>=02		Activar el canal de entradas analógicas 1
MS<n>=04		Activar el canal de entradas analógicas 2
MS<n>=08		Activar el canal de entradas analógicas 3
MAW<n>.2=xxxx		Transmitir canal de salida 0.
MAW<n>.4=xxxx		Transmitir canal de salida 1.

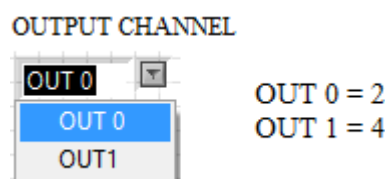
Fuente: (Autores, 2017)

Se seleccionará una entrada a activar y a leer en el combo box Activar IN Analog y Leer IN Analog.



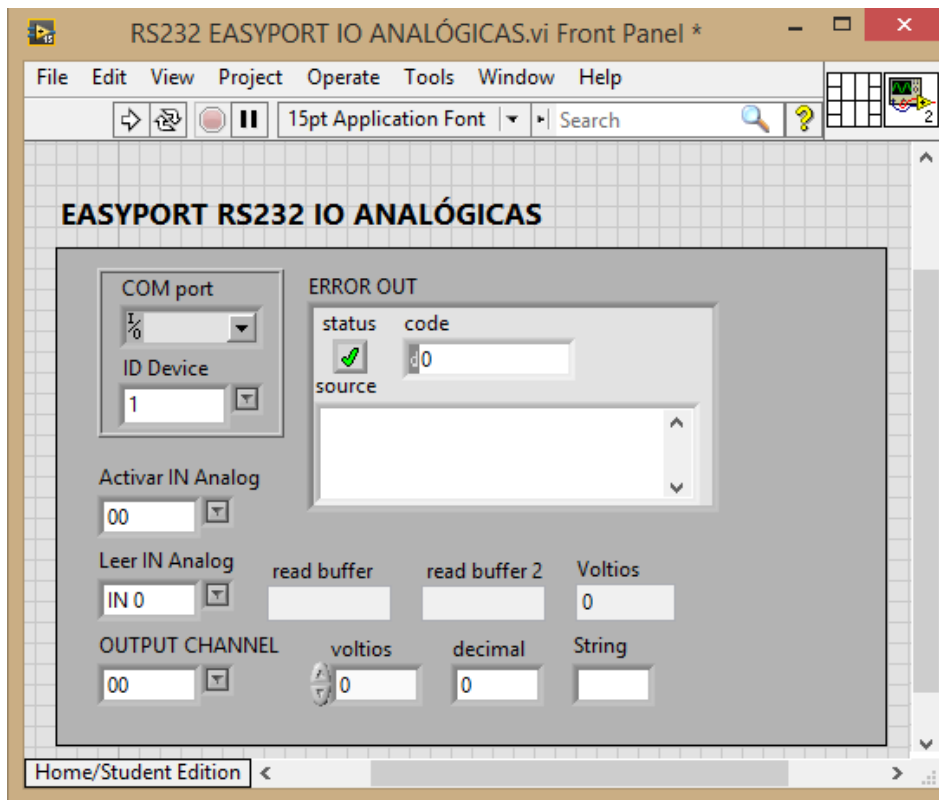
**Figura 3-86** Combo Box Activar y Leer IN Analog.

Fuente: (Autores, 2017)



**Figura 3-87** Combo Box OUTPUT CHANNEL.

Fuente: (Autores, 2017)



**Figura 3-88** Vista Frontal del VI comunicación EasyPort RS232 para señales analógicas.  
**Fuente:** (Autores, 2017)

Todas las señales analógicas se leen en forma de un número decimal de 12bits por lo que realizamos conversiones iguales a las realizadas para las señales analógicas en el VI donde utilizamos OPC de Festo.

### **3.6. Normas de seguridad para el uso del módulo.**

Con el fin de precautelar la seguridad de los usuarios del módulo y el buen desempeño del mismo, se pone a colación las siguientes normas de seguridad:

- Solo deberá ser utilizado por los estudiantes cualificados o bajo la supervisión de un tutor.
- No maniobrar los circuitos eléctricos del módulo o realizar operaciones de mantenimiento cuando este se encuentre energizado.
- No colocar objetos extraños en el interior o sobre el módulo. Podrían provocar daños de consideración y/o poner en riesgo la seguridad de los usuarios.

- Utilizar destornilladores dieléctricos, pinzas con recubrimiento dieléctrico, etc. Los destornilladores con refuerzo metálico, cuchillos y pinzas sin recubrimiento, queda totalmente prohibido su uso en circuitos eléctricos.
- Solo personal autorizado podrá realizar operaciones de mantenimiento. Se deberá utilizar EPP (guantes dieléctricos, protección visual, etc.) cuando se realice trabajos en el equipo energizado.

### 3.7. Desarrollo de Guías de laboratorio.

A continuación, se muestra los temas y objetivos de las guías de laboratorio del protocolo Modbus a ser realizadas por el estudiante, las mismas que se encuentran en su totalidad en el Anexo B.

#### 3.7.1. Guía de Laboratorio 1.

<b>TEMA:</b>	Creación de un VI para la lectura de la señal digital de los bits 0,1,2,3 del Módulo I/O Modbus a través de un controlador API LabVIEW.
<b>OBJETIVOS:</b>	<b>GENERAL:</b> Configurar LabVIEW para la lectura de datos booleanos del dispositivo.
	<b>ESPECÍFICOS</b> <ul style="list-style-type: none"> <li>• Direccionar los bits para la lectura de señales digitales.</li> <li>• Entrenar al estudiante en la utilización de LabVIEW para comunicación Modbus por controladores API.</li> <li>• Establecer la comunicación Modbus RTU.</li> </ul>

Fuente: (Autores, 2017)

#### 3.7.2. Guía de Laboratorio 2.

<b>TEMA:</b>	Creación de un VI para la lectura de la señal analógica del sensor LM35 del Módulo I/O Modbus a través de un controlador API LabVIEW.
<b>OBJETIVOS:</b>	<b>GENERAL:</b> Configurar LabVIEW para la lectura de señales analógicas del dispositivo I/O Modbus.
	<b>ESPECÍFICOS</b> <ul style="list-style-type: none"> <li>• Desarrollar el mapeo de bytes y lectura de señales analógicas.</li> <li>• Entrenar al estudiante en la utilización de LabVIEW para comunicación Modbus por controladores API.</li> <li>• Establecer comunicación a través de Modbus entre los dispositivos.</li> </ul>

Fuente: (Autores, 2017)

### 3.7.3. Guía de Laboratorio 3.

<b>TEMA:</b>	Creación de un VI para la comunicación de Easy Port con LabVIEW a través de OPC.
<b>OBJETIVOS:</b>	<b>GENERAL:</b> Establecer la comunicación entre EASY Port y LabVIEW.
	<b>ESPECÍFICOS</b> <ul style="list-style-type: none"><li>• Realizar el mapeo de bits de la Librería OPC.</li><li>• Ejecutar la lectura y escritura de señales digitales y analógicas entre el PC y el dispositivo Easy Port.</li><li>• Entrenar al estudiante en la utilización de recursos OPC.</li></ul>

Fuente: (Autores, 2017)

### 3.7.4. Guía de Laboratorio 4.

<b>TEMA:</b>	Creación de un VI para la escritura de las salidas físicas del PLC SIEMENS a través de I/O Server de LabVIEW.
<b>OBJETIVOS:</b>	<b>GENERAL:</b> Introducir al estudiante al protocolo Modbus TCP.
	<b>ESPECIFICOS:</b> <ul style="list-style-type: none"><li>• Comprender el mapeo de registro Modbus del PLC Siemens.</li><li>• Configurar los equipos necesarios para la práctica requerida.</li><li>• Entrenar al estudiante en la utilización de LabVIEW para comunicación Modbus a través de I/O server.</li></ul>

Fuente: (Autores, 2017)

### 3.7.5. Guía de Laboratorio 5.

<b>TEMA:</b>	Creación de un VI para la lectura de una memoria analógica del PLC Twido a través de un controlador API LabVIEW.
<b>OBJETIVOS:</b>	<b>GENERAL:</b> Introducir al estudiante a la utilización del protocolo Modbus RTU.
	<b>ESPECIFICOS:</b> <ul style="list-style-type: none"><li>• Comprender el direccionamiento Modbus del Autómata Twido.</li><li>• Configurar los equipos necesarios para la práctica requerida.</li><li>• Entrenar al estudiante en la utilización de LabVIEW para comunicación Modbus mediante controladores API.</li></ul>

Fuente: (Autores, 2017)

## 3.8. Plan de mantenimiento del módulo.

Se deberá dar la atención adecuando en la ejecución del mantenimiento del módulo, ya sea una inspección visual, limpieza o reajustes de componentes, con el fin de mitigar las consecuencias de los modos de fallos.

El plan de mantenimiento plasmado en la Tabla 3-18 se lo ha realizado en base a las recomendaciones de los fabricantes de los dispositivos electrónicos que constituyen el módulo.

**Frecuencia:**

**S:** Semanal.      **M:** Mensual.      **T:** Trimestral.      **SM:** Semestral.      **A:** Anual.

**Tabla 3-18** Plan de mantenimiento.

Ítem	Tarea	Herramientas y Materiales	Frecuencia				
			S	M	T	SM	A
1	Limpieza e inspección general del modulo	<ul style="list-style-type: none"> <li>Guaípe</li> <li>Aire comprimido</li> <li>Aerosol Limpiador de contactos</li> </ul>		X			
2	Inspección visual y limpieza de borneras de los PLCs y fuente de alimentación	<ul style="list-style-type: none"> <li>Aerosol Limpiador de contactos</li> <li>Guaípe</li> </ul>				X	
3	Realizar un reajuste de los tornillos de las borneras de las entradas/salidas de los PLCs y fuente de alimentación.	<ul style="list-style-type: none"> <li>Destornillador plano</li> <li>Destornillador estrella</li> </ul>				X	
4	Verificar el voltaje de entrada y salida de la fuente de alimentación	<ul style="list-style-type: none"> <li>Multímetro</li> </ul>					X
5	Verificar continuidad del cable de conexión entre entradas/salidas y el DB-25.	<ul style="list-style-type: none"> <li>Multímetro</li> </ul>				X	
6	Realizar una inspección del estado del breaker	<ul style="list-style-type: none"> <li>Multímetro</li> </ul>			X		
7	Inspección visual de los conectores DB-25	<ul style="list-style-type: none"> <li>Guaípe</li> </ul>	X				
8	Inspección visual de los puertos de comunicación Modbus	<ul style="list-style-type: none"> <li>Guaípe</li> </ul>			X		

**Fuente:** (Autores, 2017)

**3.8.1 Check List.** Se ha creado con el propósito de realizar las revisiones del módulo, en este documento se registrará el nombre del equipo a revisar, la fecha en que se ejecute la revisión y su encargado, el tiempo que toma realizar dicha revisión y los resultados obtenidos de la misma.

**Tabla 3-19 CHECK LIST**

CHECKLIST				
<b>Equipo:</b>			<b>Frecuencia:</b>	
<b>Fecha</b>			<b>T. Estimado:</b>	
<b>Realizado por:</b>			<b>T. Real:</b>	
Ítem	Descripción	Cumple		Observaciones
		Si	No	
<b>1</b>	¿La carcasa del módulo se encuentra sin daños evidentes (grietas, agujeros, etc.) Reparar o Cambiar si no lo está.			
<b>2</b>	¿El cable de alimentación de energía eléctrica se encuentra en buenas condiciones? Reparar o Cambiar si no lo está.			
<b>3</b>	¿La fuente de alimentación de energía entrega 24 VCD en su salida de tensión? Repara o Cambiar si no es así.			
<b>4</b>	¿Se encuentra ajustados los bornes de entradas y salidas del PLC Siemens? Ajustar si no lo está.			
<b>5</b>	¿Se encuentra ajustados los bornes de entradas y salidas del PLC Twido? Ajustar si no lo está.			
<b>6</b>	¿Se encuentra ajustados los bornes de la fuente de alimentación? Ajustar si no lo está.			
<b>7</b>	¿Presentan continuidad los cables de conexión del DB-25? Reparar si lo está.			
<p><b>Firma:</b></p>   <p style="text-align: center;"><b>Responsable</b></p>				

**Fuente:** (Autores, 2017)

## **CAPITULO IV**

### **4. CONCLUSIONES Y RECOMENDACIONES.**

#### **4.1. CONCLUSIONES.**

El presente documento de titulación no pretende enseñar al lector la programación básica de los PLC utilizados, más bien tiene como objetivo configurar la comunicación y demostrar la utilización del protocolo Modbus RTU y TCP/IP en procesos para supervisión en tiempo real.

Se configuro con éxito la recepción y envió de datos hacia los distintos dispositivos que constituyen el módulo de entrenamiento Modbus que se ha implementado en el laboratorio de manipulación automática, con la ayuda de una PC se realizó la carga de librerías necesarias para que cada dispositivo pueda realizar la comunicación.

Se realizó con éxito la comunicación maestro esclavo mediante Modbus TCP/IP para el envío y recepción de datos entre LabVIEW-PLC Siemens, y a través de Modbus RTU entre LabVIEW-PLC Twido y LabVIEW-Módulo I/O Modbus, así, como la comunicación entre LabVIEW-Easy Port (FESTO) a través de serial RS232 y OPC, cabe recalcar que en todos los casos el software LabVIEW hace de maestro en la red.

Se realizaron las pruebas de funcionamiento de cada elemento del módulo y de la comunicación entre LabVIEW y los diferentes dispositivos utilizados para el desarrollo de este trabajo, obteniendo resultados satisfactorios de cada uno de los procesos simulados.

Las guías de prácticas de laboratorio fueron realizadas y probadas con los equipos disponibles en el módulo de entrenamiento MODBUS, los procesos industriales que se simulan representan una ayuda para la aplicación didáctica de programación, configuración y comunicación por parte del docente, en este documento se pretende incentivar al lector, profesor o alumno a que realice nuevas prácticas con el módulo.

## **4.2. RECOMENDACIONES**

Se recomienda previo a la manipulación del módulo o cualquier equipo que se desee comunicar a través de Modbus una correcta familiarización del uso de todos los elementos que se han utilizado en el desarrollo de este trabajo, así como un estudio de los manuales técnicos de programación de los equipos utilizados, con el fin de minimizar al máximo posibles errores y precautelar el buen desempeño del Módulo de entrenamiento Modbus.

A medida que se programa asignar un nombre específico a cada elemento de LabVIEW correspondiente a la función que desarrolle para facilitar su uso y aplicación.

Verificar que las configuraciones del módulo para adquirir o generar algún tipo de señal sean las correctas para que la comunicación sea óptima y evitar errores o posibles daños a los equipos de procesos.

Para el uso correcto, seguro y fiable de la interface de procesos EasyPort USB, es indispensable conocer y respetar las indicaciones y normas de seguridad.



## BIBLIOGRAFÍA

**ALBÁN GUERRERO, José Eduardo & PAGUAY LLAMUCA, Néstor Jhovany.**

Diseño e implementación de un sistema SCADA con comunicación PROFIBUS para el control [en línea] (tesis) (ingeniería). *Escuela Superior Politécnica de Chimborazo, Mecánica, Ingeniería de Mantenimiento. Riobamba, Ecuador.* 2017. pp. 4-17. [Consulta: 18 de Mayo de 2017]. Disponible en: <http://dspace.espoch.edu.ec/bitstream/123456789/6496/3/25T00292.pdf>

**ARIAN S.A.** *Sistema de supervisión y adquisición de datos* [en línea]. 2006. [Consulta: 23 de Abril de 2017]. Disponible en: <http://www.arian.cl/downloads/ari-s.pdf>

**CANDELAS, Francisco.** *Automatización Avanzada* [en línea]. 2011. [Consulta: 25 de Abril de 2017]. Disponible en: <https://rua.ua.es/dspace/bitstream/10045/18990/1/AA-p3.pdf>

**CASTRO SILVA, Carlos Andrés & ALVARADO FEIJOO, Josie Esteban.**

Implementación de una red MODBUS TCP Wireless con integración de tecnología Siemens [en línea] (tesis) (ingeniería). *Universidad Politécnica Salesiana, Ingeniería Electrónica. Guayaquil, Ecuador.* 2017. pp. 7-47. [Consulta: 18 de Mayo de 2017]. Disponible en: <http://dspace.ups.edu.ec/handle/123456789/14191>

**FESTO.** *EasyPort USB Manual* [en línea]. 2008. [Consulta: 1 de Junio de 2017]. Disponible en: [http://www.festo-didactic.com/ov3/media/customers/1100/721876\\_deene\\_sfr\\_mit lesezeichen\\_4.pdf](http://www.festo-didactic.com/ov3/media/customers/1100/721876_deene_sfr_mit lesezeichen_4.pdf)

**HERNANDEZ TINOCO, Marina.** Desarrollo e implementación de una red de datos basada en Modbus y Ethernet para autómatas industriales [en línea] (tesis). *Universidad de Sevilla, Escuela Técnica Superior De Ingeniería, Dep. Ingeniería de Sistemas y Autómatas. Sevilla, España.* 2016. pp. 7-67. [Consulta: 19 de Mayo de 2017]. Disponible en: <https://idus.us.es/xmlui/bitstream/handle/11441/53165/marhertin.pdf?sequence=1>

**HURTADO, José María.** *Introducción a las redes de comunicación industrial* [en línea]. 2015. [Consulta: 21 de Abril de 2017]. Disponible en: [http://www.infoplcn.net/files/documentacion/comunicaciones/infoPLC\\_net\\_introduccion3b3n-a-las-redes-de-comunicacion3b3n-industrial.pdf](http://www.infoplcn.net/files/documentacion/comunicaciones/infoPLC_net_introduccion3b3n-a-las-redes-de-comunicacion3b3n-industrial.pdf)

**LAGOS, Carolina.** *Protocolos de comunicación industrial* [blogs]. 2006. [Consulta: 20 de abril de 2017]. Disponible en: <http://www.emb.cl/electroindustria/articulo.mvc?xid=562>

**MARTINEZ SALAS, Roció Comanda.** Implementacion de un banco de ensayos para conexiones en red MODBUS controladas por LabVIEW [en línea] (tesis) (ingenieria). *Escuela Superior Politécnica de Chimborazo, Mecánica, Ingeniería de Mantenimiento, Riobamba, Ecuador.* 2009. pp. 3-32. [Consulta: 19 de Mayo de 2017]. Disponible en: <https://idus.us.es/xmlui/bitstream/handle/11441/53165/marhertin.pdf?sequence=1>

**NATIONAL INSTRUMENTS.** *LabVIEW* [en línea]. 2014. [Consulta: 3 de Mayo de 2017]. Disponible en: <http://www.ni.com/labview/labviewdsc/esa/>

**OMRON Electronics SA.** *Modbus tcp* [en línea]. 2007. [Consulta: 22 de Abril de 2017]. Disponible en: [http://www.tecnical.cat/PDF/Omron/PLC/CJ/GR\\_MODBUS\\_TCP.pdf](http://www.tecnical.cat/PDF/Omron/PLC/CJ/GR_MODBUS_TCP.pdf)

**PROSOFT.** *Modbus & Modbus TCP* [en línea]. 2017. [Consulta: 16 de Abril de 2017]. Disponible en: <http://www.prosoft-technology.com/Landing-Pages/Protocol/Modbus-and-Modbus-TCP-Protocol>

**SCHNEIDER ELECTRIC.** *PLC twido* [en línea]. 2008. [Consulta: 17 de Mayo de 2017]. Disponible en: <http://www.schneider-electric.com/>

**SIEMENS.** *PLC siemens* [en línea]. 2015. [Consulta: 15 de Mayo de 2017]. Disponible en: <http://www.siemens.com/>

**TERATRONIX.** *Compatibilidad* [en línea]. 2016. [Consulta: 21 de Mayo de 2017]. Disponible en: [https://www.teratronix.com/new/index.php?option=com\\_content&view=article&id=193&Itemid=2428](https://www.teratronix.com/new/index.php?option=com_content&view=article&id=193&Itemid=2428)

**WEG.** *Manual de la comunicación Modbus RTU* [en línea]. 2012. [Consulta: 15 de Mayo de 2017]. Disponible en: <http://ecatalog.weg.net/files/wegnet/WEG-srw01-manual-de-la-comunicacion-modbus-rtu-10000521680-4.0x-manual-espanol.pdf>

