

UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL

FACULTAD DE INGENIERÍAS

CARRERA:

INGENIERÍA ELECTRÓNICA

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERO ELECTRÓNICO

TEMA:

“DISEÑO E IMPLEMENTACIÓN DE DOS MALETAS DIDÁCTICAS PARA EL FORTALECIMIENTO DE PRÁCTICAS DE COMUNICACIÓN INDUSTRIAL UTILIZANDO EL PROTOCOLO MODBUS, EN EL LABORATORIO DE FABRICACIÓN FLEXIBLE”

NOMBRE DE LOS AUTORES:

JORGE LUIS MACÍAS BURGOS

PETER OSWALDO CARRANZA MAIGUA

TUTOR:

MSc. GARY AMPUÑO AVILÉS

GUAYAQUIL, MARZO DE 2015

DECLARACIÓN DE RESPONSABILIDAD.

Nosotros, Jorge Luis Macías Burgos portador de la cedula de ciudadanía C.I: 092666149-7 y Peter Oswaldo Carranza Maigua portador de la cedula de ciudadanía C.I:092592743-6 estudiantes de la Universidad Politécnica Salesiana declaramos que el presente trabajo de grado es de nuestra autoría y propiedad intelectual de la Universidad Politécnica Salesiana.

Guayaquil, Marzo 2015

(f) _____

Jorge Macías Burgos

C.I. 092666149-7

(f) _____

Peter Carranza Maigua

092592743-6

DEDICATORIAS.

Este trabajo se la dedico a mi Dios quién supo guiarme por el buen camino, darme fuerzas para seguir adelante y no desmayar ante las adversidades, enseñándome a encarar cada una, sin desfallecer en el intento.

A mi familia quienes por ellos soy lo que soy. Para mis padres por su apoyo, consejos, comprensión, ayuda en todo momento, y por ayudarme con los recursos necesarios para estudiar. Me han dado todo lo que soy como persona, mis valores, mis principios, mi carácter, mi empeño, mi perseverancia, mi coraje para conseguir mis objetivos.

Jorge Luis Macías Burgos

A Dios, verdadera fuente de amor y sabiduría. A mi madre y hermanos, porque gracias a ellos sé que la responsabilidad se la debe vivir como un compromiso de dedicación y esfuerzo.

A mi madre, cuyo vivir me ha mostrado que en el camino hacia la meta se necesita de la dulce fortaleza para aceptar las derrotas y del sutil coraje para derribar miedos.

A mi hermano, el incondicional abrazo que me motiva y recuerda que detrás de cada detalle existe el suficiente alivio para empezar nuevas búsquedas.

A mis familiares, viejos amigos y a quienes recién se sumaron a mi vida para hacerme compañía con sus sonrisas de ánimo.

Peter Carranza Maigua

AGRADECIMIENTOS.

El presente trabajo de tesis primeramente me gustaría agradecerle a ti Dios por bendecirme para llegar hasta donde he llegado, porque hiciste realidad este sueño anhelado.

Le doy gracias a mis padres por apoyarme en todo momento, por los valores que me han inculcado, y por haberme dado la oportunidad de tener una excelente educación en el transcurso de mi vida. Sobre todo por ser excelente ejemplo de vida a seguir.

Jorge Luis Macías Burgos

Primero y antes que nada, dar gracias a Dios, por estar conmigo en cada paso que doy, por fortalecer mi corazón e iluminar mi mente y por haber puesto en mi camino a aquellas personas que han sido mi soporte y compañía durante todo el periodo de estudio.

Este trabajo de grado, si bien ha requerido de esfuerzo y mucha dedicación por parte de los autores y director de tesis, no hubiese sido posible su finalización sin la cooperación desinteresada de todas y cada una de las personas que brindaron su apoyo y muchas de las cuales han sido un soporte muy fuerte en momentos difíciles.

Agradecer hoy y siempre a mi familia porque a pesar de no estar presentes físicamente, a pesar de la distancia, el ánimo, apoyo y alegría que me brindan me dan la fortaleza necesaria para seguir adelante.

Peter Carranza Maigua

ÍNDICE GENERAL

1	CAPÍTULO 1: EL PROBLEMA.....	1
1.1	PLANTEAMIENTO DEL PROBLEMA.....	1
1.2	DELIMITACIÓN.....	1
1.2.1	TEMPORAL.....	1
1.2.2	ESPACIAL.....	1
1.2.3	ACADÉMICO.....	1
1.3	OBJETIVOS.....	2
1.3.1	OBJETIVO GENERAL.....	2
1.3.2	OBJETIVOS ESPECÍFICOS.....	2
1.4	JUSTIFICACIÓN.....	2
1.5	VARIABLES E INDICADORES.....	3
1.6	METODOLOGÍA.....	3
1.6.1	OBSERVACIÓN CIENTÍFICA.....	3
1.6.2	EXPERIMENTAL.....	3
1.7	DESCRIPCIÓN DE LA PROPUESTA.....	4
1.8	BENEFICIARIOS DE LA PROPUESTA DE INTERVENCIÓN.....	4
2	CAPÍTULO 2 : MARCO TEÓRICO.....	5
2.1	REDES DE COMUNICACIONES INDUSTRIAL.....	5
2.1.3	VENTAJAS DE UN BUS DE CAMPO.....	7
2.1.4	DESVENTAJAS DE UN BUS DE CAMPO.....	7
2.1.5	PROCESOS DE COMUNICACIÓN POR MEDIO DE BUS.....	8
2.1.6	TIPOS DE BUSES.....	8
2.1.7	TOPOLOGÍA DE REDES INDUSTRIALES.....	9
2.1.8	BENEFICIOS DE UNA RED INDUSTRIAL.....	9
2.2	CONTROLADORES LÓGICOS PROGRAMABLES.....	10
2.2.1	INTRODUCCIÓN.....	10

2.2.2	CLASIFICACIÓN DE LOS PLC's.....	11
2.2.3	VENTAJAS.	11
2.2.4	COMUNICACIÓN.	11
2.3	RED DE COMUNICACIÓN PROTOCOLO MODBUS	12
2.3.1	DESCRIPCIÓN GENERAL.....	13
2.3.2	CAMPOS DE LAS TRAMAS MODBUS	14
2.3.3	DESCRIPCIÓN DE LOS CAMPOS DE LAS TRAMAS MODBUS RTU.	15
2.3.4	MAPA DE REGISTOS MODBUS.....	17
2.3.5	DESCRIPCIÓN DE LOS CÓDIGOS DE FUNCIÓN.....	18
2.3.6	CÓDIGOS ERRORES EXCEPCIONES.....	24
2.3.7	MACROS DE COMUNICACIONES DE TWIDO SUITE.....	26
3	CAPÍTULO 3 : DISEÑO DEL SISTEMA A ESCALA	27
3.1	DESCRIPCIÓN DE ELEMENTOS.....	27
3.1.1	DESCRIPCIÓN DEL AUTÓMATA TWIDO.....	27
3.1.2	DESCRIPCIÓN DEL CABLE DE PROGRAMACIÓN PARA TWIDO 29	
3.1.3	DESCRIPCIÓN DE COMPONENTES DEL ZELIO LOGIC 30	
3.1.4	MÓDULO DE COMUNICACIÓN PARA ZELIO LOGIC..	31
3.1.5	CABLE DE PROGRAMACIÓN ZELIO	33
3.1.7	MOTOR TRIFASICO.....	35
3.1.8	SOFTWARE TWIDOSUITE.	36
3.1.8.1	OPERACIONES BÁSICAS PARA CREAR UN PROYECTO.	37
3.1.9	SOFTWARE ZELIOSOFT 2.....	45
3.2	P&ID DIAGRAMA DE BLOQUES.....	50
3.3	DISEÑO ESTRUCTURAL DE LA MALETA DIDÁCTICA.....	50

3.4	ENSAMBLAJE DE LA ESTRUCTURA	54
4	CAPÍTULO 4: PRÁCTICAS.	58
4.1	ESCRITURA DE UN BIT CONFIGURANDO UN TWIDO MAESTRO Y UN TWIDO ESCLAVO.	58
4.1.1	PROGRAMA PARA EL EQUIPO MAESTRO.....	58
4.1.2	PROGRAMA PARA EL EQUIPO ESCLAVO.	63
4.2	LECTURA DE UN BIT CONFIGURANDO UN TWIDO MAESTRO Y UN TWIDO ESCLAVO.	64
4.2.1	PROGRAMA PARA EL EQUIPO MAESTRO.....	64
4.2.2	PROGRAMA PARA EL EQUIPO ESCLAVO	68
4.3	ESCRITURA DE UNA PALABRA DE MEMORIA CONFIGURANDO UN TWIDO MAESTRO Y UN ZELIO LOGIC ESCLAVO.....	69
4.3.1	PROGRAMA PARA EL EQUIPO MAESTRO.....	70
4.4	COMUNICACIÓN ENTRE MAESTRO PLC TWIDO CON EL ESCLAVO ALTIVAR 312 UTILIZANDO MACROS DRIVE.	74
4.4.1	PROGRAMA PARA EL EQUIPO MAESTRO.....	75
4.5	COMUNICACIÓN ENTRE UN EQUIPO PLC MAESTRO Y DOS EQUIPOS ESCLAVOS.	78

ÍNDICE DE TABLAS.

Tabla 1. Principales códigos de funciones.	18
Tabla 2. Solicitud 01 y 02.	19
Tabla 3. Solicitud 03 y 04.....	20
Tabla 4. Solicitud 05.	21
Tabla 5. Solicitud 06.	22
Tabla 6. Solicitud 15	23
Tabla 7. Solicitud 16	24
Tabla 8. Códigos errores excepciones.	25
Tabla 9. Funciones Macros de comunicación.	26
Tabla 10. Posiciones de perilla.....	30
Tabla 11. Palabras de memoria disponibles en Zelio Logic.	70

ÍNDICE DE FIGURAS

Figura 1. Niveles de una red industrial.	5
Figura 2. Diagrama de un Bus de comunicación.	7
Figura 3. Topología Bus.....	9
Figura 4. Topología Estrella.....	9
Figura 5. Arquitectura del PLC.....	10
Figura 6. Comunicación Autómatas.....	12
Figura 7. Arquitectura de red MODBUS.	12
Figura 8. Relación de Red Maestro-Esclavo.....	13
Figura 9. Esquema de comunicación MODBUS.	14
Figura 10. Trama ASCII.	14
Figura 11. Trama RTU.....	15
Figura 12. Ciclo de Preguntas y Respuestas.	16
Figura 13. Mapa de Registros Modbus.	17
Figura 14. Componentes del Autómata Twido.	28
Figura 15. Convertidor RS485-USB.....	29
Figura 16. Componentes del Zelio Logic.....	30
Figura 17. Módulo de Comunicación.	31
Figura 18. Conexión Zelio y Módulo de comunicación.	32
Figura 19. Adaptador RJ-45/Bornera.....	33
Figura 20. Cable de programación Zelio.....	33
Figura 21. Componentes del Variador de Frecuencia.....	34
Figura 22. Partes del Motor.....	36
Figura 23. Plataforma TwidoSuite.	37
Figura 24. Modos de funcionamiento TwidoSuite.....	38
Figura 25. Administrador de proyectos.....	39
Figura 26. Selección del hardware.	40

Figura 27. Barra de Herramientas Ladder.....	40
Figura 28. Icono agregar sección.	41
Figura 29. Introducción de la primera sección.....	41
Figura 30. Paleta ladder ampliada.....	42
Figura 31. Inserción de elementos de control.	43
Figura 32. Pasos para activación de macros.	43
Figura 33. Selección de macro.....	44
Figura 34. Selección de puerto.....	44
Figura 35. Selección de dirección de esclavo.	44
Figura 36. Lista de funciones Macros.....	45
Figura 37. Menú de opciones ZelioSoft2.....	46
Figura 38. Selección de modelo del módulo Zelio.	47
Figura 39. Ventana de selección de módulo adicional.....	48
Figura 40. Selección de tipo de programación.....	49
Figura 41. Pantalla Programacion ZelioSoft2.....	49
Figura 42. Diagrama P&ID.....	50
Figura 43. Diagrama esquemático del sistema.....	51
Figura 44. Diseño con Medidas.	52
Figura 45. Diseño identificación de equipos.....	53
Figura 46. Estructura terminada.....	53
Figura 47. Montaje de Plancha Metálica.....	54
Figura 48. Montaje y Cableado de equipos.....	55
Figura 49. Mango y base de Maleta.	55
Figura 50. Maleta didáctica.....	56
Figura 51. Maleta con Acabado.	56
Figura 52. Estructura de maleta didáctica terminada.	57
Figura 53. Ventana Catálogo Maestro.	59

Figura 54. Configuración red Maestro.	59
Figura 55. Elemento Modbus.	60
Figura 56. Configuración de red en esclavo.	60
Figura 57. Parámetros de transmisión.	61
Figura 58. Rung 0 escritura de bit.	61
Figura 59. Rung 1 y 2 para la escritura de un bit.	62
Figura 60. Configuración de red del esclavo para escritura.	63
Figura 61. Escribir Bit esclavo.	63
Figura 62. Ventana Catálogo Esclavo.	64
Figura 63. Configuración red Maestro.	65
Figura 64. Elemento Modbus.	65
Figura 65. Configuración de red en esclavo.	66
Figura 66. Parámetros de transmisión.	66
Figura 67. Rung 0 lectura de bit.	67
Figura 68. Rung 2 y 3, lectura de bit.	68
Figura 69. Configuración esclavo.	68
Figura 70. Leer Bit esclavo.	69
Figura 71. Configuración red Maestro.	70
Figura 72. Configuración red Maestro con el Esclavo Zelio Logic.	71
Figura 73. Configuración red esclavo.	71
Figura 74. Parámetros de transmisión.	72
Figura 75. Rung 0 Escritura de una palabra sobre Zelio.	73
Figura 76. Escritura de palabra.	74
Figura 77. Configuración red Maestro.	75
Figura 78. Configuración red Maestro con el ATV 312.	76
Figura 79. Configuración red esclavo ATV 312.	76
Figura 80. Rung 0, 1 y 2 funciones de Macro Drive.	77

Figura 81. Ventana Catálogo Maestro.	79
Figura 82. Configuración red Maestro.	79
Figura 83. Elemento Modbus.	80
Figura 84. Configuración red esclavo Zelio.	80
Figura 85. Configuración red esclavo ATV 312.	80
Figura 86. Configuración red Maestro con el Zelio y el ATV 312.	81
Figura 87. Rung 0 lectura de una palabra.	82
Figura 88. Escritura de palabra.	83
Figura 89. Rung 2 funciones de Macro Drive.	83
Figura 90. Rung 3,4 y 5 funciones de Macro Drive.	84

RESUMEN

AÑO	ALUMNOS	DIRECTOR DE TESIS	TEMA DE TESIS
2015	JORGE LUIS MACÍAS BURGOS PETER OSWALDO CARRANZA MAIGUA	MSc. GARY AMPUÑO AVILÉS	“DISEÑO E IMPLEMENTACIÓN DE DOS MALETAS DIDÁCTICAS PARA EL FORTALECIMIENTO DE PRÁCTICAS DE COMUNICACIÓN INDUSTRIAL UTILIZANDO EL PROTOCOLO MODBUS, EN EL LABORATORIO DE FABRICACIÓN FLEXIBLE”

El presente proyecto de grado “DISEÑO E IMPLEMENTACIÓN DE DOS MALETAS DIDÁCTICAS PARA EL FORTALECIMIENTO DE PRÁCTICAS DE COMUNICACIÓN INDUSTRIAL UTILIZANDO EL PROTOCOLO MODBUS, EN EL LABORATORIO DE FABRICACIÓN FLEXIBLE” consiste en establecer una comunicación entre equipos o controladores industriales, con el fin de implementar un bus con arquitectura maestro/esclavo, basándose en el estándar TIA/EIA-485 y utilizando el protocolo de comunicación abierto Modbus RTU.

Este proyecto tiene como objetivo principal el diseño e implementación dos maletas didácticas compactas, que incorporan controladores, módulos de comunicación, necesarios para construir una red industrial mediante el protocolo Modbus RTU. Estas maletas ayudaran a fortalecer el conocimiento experimental de este tipo de comunicación industrial y promover el uso de este protocolo de manera práctica en el laboratorio Fabricación Flexible de la Universidad Politécnica Salesiana.

Como parte primordial de este proyecto se realizó el estudio y análisis de comandos o códigos de funciones necesarios para la realización del intercambio de datos entre el dispositivo “Maestro” y los dispositivos “Esclavos” que conforman el bus, así como la configuración de parámetros para el establecimiento de la comunicación.

Palabras claves: Protocolo, Modbus RTU, Comunicación Industrial, Maletas didácticas.

ABSTRACT

YEAR	STUDENTS	ADVISOR	TITLE
2015	JORGE LUIS MACÍAS BURGOS PETER OSWALDO CARRANZA MAIGUA	MSc. GARY AMPUÑO AVILÉS	“DISEÑO E IMPLEMENTACIÓN DE DOS MALETAS DIDÁCTICAS PARA EL FORTALECIMIENTO DE PRÁCTICAS DE COMUNICACIÓN INDUSTRIAL UTILIZANDO EL PROTOCOLO MODBUS, EN EL LABORATORIO DE FABRICACIÓN FLEXIBLE”

This draft grade "DESIGN AND IMPLEMENTATION OF TWO CASES FOR TEACHING PRACTICES INDUSTRIAL BUILDING USING MODBUS PROTOCOL COMMUNICATION IN THE LABORATORY OF FLEXIBLE MANUFACTURING" is to establish communication between devices and industrial controllers, in order to implement a bus master / slave architecture, based on the TIA / EIA-485 standard and using the open communication protocol Modbus RTU.

This project's main objective is the design and implementation didactic two compact bags that incorporate controllers, communication modules needed to build an industrial network via the Modbus RTU protocol. These bags help strengthen the experimental knowledge of this type of industrial communication and promote the use of this protocol practical laboratory Flexible Manufacturing Salesian Polytechnic University.

As a key part of this project to study and analyze command or function codes necessary for the realization of data exchange between the "Master" device and "Slave" devices that make up the bus and the configuration parameters was performed establishing communication.

Keywords: Protocol, Modbus RTU, Industrial Communication, Didactic Cases.

INTRODUCCIÓN.

En el área de las comunicaciones industriales (Distefano, 2008), la estandarización de protocolos es un tema en permanente discusión, donde intervienen problemas técnicos y comerciales. Cada protocolo está optimizado para diferentes niveles de automatización y en consecuencia responden al interés de diferentes proveedores (Distefano, 2008). En el presente proyecto se realiza el estudio del protocolo de comunicación industrial Modbus (Instruments, 2014), que contribuye a asimilar la teoría impartida en las aulas y su implementación mediante dos maletas didácticas provistas con equipos de la industria para ofrecer una mejor experiencia al momento de realizar las prácticas en los laboratorios de la Universidad Politécnica Salesiana sede Guayaquil. Las prácticas que se pueden realizar mediante la utilización de equipos autómatas industriales que incorporan las maletas dotan al estudiante de conocimientos experimentales.

En el **Capítulo 1** se plantea el problema, se delimita el trabajo de grado y se describe los objetivos y la metodología implementada. En el **Capítulo 2** se detalla el marco teórico con las bases de los conceptos fundamentales para la realización e implementación del presente proyecto de grado. El **Capítulo 3** se basa en el diseño y construcción del objetivo, el cual también describe los elementos y equipos industriales utilizados en la elaboración de las maletas didácticas. El **Capítulo 4** corresponde a elaboración y validación de las practicas implementadas en las maletas didácticas.

1 CAPÍTULO 1: EL PROBLEMA.

1.1 PLANTEAMIENTO DEL PROBLEMA.

El estudio de las comunicaciones industriales (ETI, 2009) en las carreras técnicas no resulta sencillo dada la diversidad de protocolos existentes. Esta diversidad que afecta y condiciona la toma de decisiones de una empresa a la hora de seleccionar un bus de campo a utilizar, provoca también un dilema en el mundo de la enseñanza técnica a la hora de decidirse por un bus industrial a estudiar.

En la última década se ha visto aparecer buses de campo (Automática, 2010) que estaban llamados a convertirse en el estándar de facto, pero las grandes empresas del sector no se han puesto de acuerdo en la elección de un estándar. Por lo consiguiente existe la necesidad de fortalecer los conocimientos prácticos en cuanto a redes industriales, en el laboratorio de la Universidad Politécnica Salesiana para establecer criterios técnicos en el momento de selección.

1.2 DELIMITACIÓN.

1.2.1 TEMPORAL.

El presente proyecto de grado implementado tuvo un tiempo estimado de seis meses en el periodo 2014 - 2015.

1.2.2 ESPACIAL.

El trabajo de grado para a la obtención del título de “Ingeniero Electrónico”, fue implementado para el laboratorio de fabricación flexible en el bloque B de la Universidad Politécnica Salesiana ubicada en Chambers #227 entre Laura Vicuña y Robles frente (Villa la Joya).

1.2.3 ACADÉMICO.

- El presente trabajo de grado consiste en la implementación de dos maletas didácticas a escala basada en una arquitectura maestro/esclavos, las cuales se comunican por medio del bus de campo Modbus.
- Cada maleta está dotada de un autómatas compacto como Maestro y dos esclavos Modbus con sus respectivos módulos de comunicación.

- Los equipos de campo son convencionales y de la marca Schneider Electric para la demostración de las múltiples alternativas de comunicación de la red.
- La demostración de la red es basada en la comunicación Maestro/Esclavos por medio del envío y recepción de señales discretas y análogas.

1.3 OBJETIVOS.

1.3.1 OBJETIVO GENERAL.

Diseñar e implementar dos maletas didácticas compactas, que incorporen controladores, módulos de comunicación, pulsadores, selectores y actuadores necesarios para establecer una red industrial con arquitectura maestro/esclavos o cliente/servidor utilizando el protocolo de comunicación Modbus.

1.3.2 OBJETIVOS ESPECÍFICOS.

- Construir y equipar dos maletas didácticas con autómatas (Maestros), Relés Inteligentes, variadores de frecuencia (Esclavos) y motores asíncronos como actuadores.
- Instalar los elementos de campo, tales como, pulsadores, selectores, interruptores, luces piloto, motor asíncrono, entre otros con su respectiva identificación.
- Desarrollar una arquitectura de red industrial para comunicación entre los diferentes equipos mediante módulos Modbus.
- Configurar envío y recepción de tramas a través de la red de Modbus.
- Realizar 6 prácticas de programación, configuración y comunicación dentro de una red Modbus.

1.4 JUSTIFICACIÓN.

Se diseñan dos maletas didácticas que contienen los equipos y dispositivos necesarios para la realización de prácticas, adquisición de experiencias y el desarrollo de nuevos proyectos de automatización utilizando el bus de campo Modbus. Permitiendo familiarizarse con la arquitectura maestro/esclavo o cliente/servidor, utilizando los bloques de funciones para el envío y recepción de tramas.

De esta manera, permite aportar en el desarrollo académico práctico de los estudiantes de la Universidad Politécnica Salesiana en su formación como nuevos profesionales.

1.5 VARIABLES E INDICADORES.

1.5.1 VARIABLES.

- Señales discretas: Pulsadores, selectores y contactos de elementos de campo.
- Comunicación de datos.
- Aplicaciones.

1.5.2 INDICADORES.

- Señal discreta: Por medio de la visualización de luces piloto.
- Señal análoga: Por medio del LCD Variador.
- Registros Modbus: Por medio de Software programación.

1.6 METODOLOGÍA.

1.6.1 OBSERVACIÓN CIENTÍFICA.

Se realizó el estudio de los métodos y configuraciones necesarias para la implementación de la comunicación Modbus, mediante tablas estándares de los códigos de peticiones, que nos ayudan a realizar el intercambio de tramas entre el maestro y esclavos.

1.6.2 EXPERIMENTAL.

Se elaboró un listado de asignaciones de señales de los equipos involucrados, para que a través del bus de comunicación envíe y reciba señales al PCL configurado como maestro. Las señales transmitidas por la red nos permitieron implementar una aplicación y demostrar de este modo la comunicación entre el maestro y sus esclavos.

1.7 DESCRIPCIÓN DE LA PROPUESTA.

Se desarrollan e implementan dos maletas didácticas las cuales están dotadas de dispositivos y elementos de la industria tales como autómatas programables o PLCs, relés inteligentes, variadores de frecuencia, motores y elementos de control, con de fin de ambientar un pequeño proceso industrial y establecer un bus de comunicación con arquitectura maestro/esclavo o cliente/servidor, utilizando el protocolo de comunicación abierto Modbus RTU, para fortalecer el conocimiento experimental de las comunicaciones industriales y promover el uso de este protocolo de manera práctica en el laboratorio Fabricación Flexible de la Universidad Politécnica Salesiana.

1.8 BENEFICIARIOS DE LA PROPUESTA DE INTERVENCIÓN.

- Los estudiantes de la carrera de Ingeniería Electrónica, que pueden hacer uso de las maletas didácticas para la realización de prácticas lo cual contribuirá a enriquecer su formación académica.
- Docentes de la carrera de Ingeniería Electrónica los cuales podrán explicar de una manera más didáctica y experimental la arquitectura, configuración y funcionamiento del protocolo de comunicación Modbus.

2 CAPÍTULO 2 : MARCO TEÓRICO

2.1 REDES DE COMUNICACIONES INDUSTRIAL

Hoy en día las comunicaciones industriales (Distefano, 2008) adquieren una gran importancia en nuestro sistema de automatización. Los equipos tienen la necesidad de comunicarse entre sí de una manera segura y basándose en los últimos estándares de comunicación.

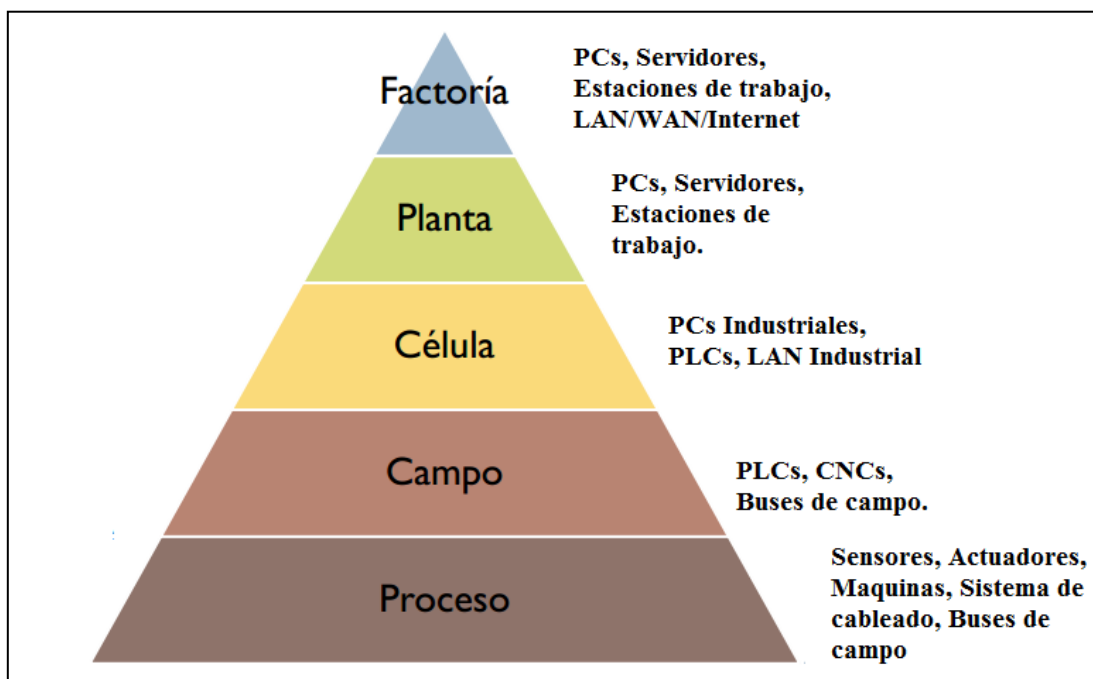


Figura 1. Niveles de una red industrial.

Esta estructura no es universal, puede variar en cantidad de niveles dependiendo directamente del tamaño del proceso (ETI, 2009)

Las redes de comunicaciones industriales deben su origen a Foundation FieldBus según (International, 2006). Foundation FieldBus desarrolló un nuevo protocolo de comunicación para la medición y el control de procesos donde todos los instrumentos puedan comunicarse una misma plataforma.

Según lo indica (Sanchez, 2006) las comunicaciones entre los instrumentos de proceso y el sistema de control se basan principalmente en señales analógicas (neumáticas de 3 a 15 psi en las válvulas de control y electrónicas de 4 a 20 mA cc). Pero ya existen instrumentos digitales capaces de manejar gran cantidad de datos y guardarlos históricamente; su precisión es diez veces mayor que la de la señal típica de 4-20 mA cc. En vez de transmitir cada variable por un par de hilos, transmiten

secuencialmente las variables por medio de un cable de comunicaciones llamado bus.

2.1.1 DEFINICIÓN DE PROTOCOLO DE COMUNICACIÓN.

Un protocolo de comunicación industrial es un conjunto de normas o mecanismos de arbitraje que engloba todas las reglas que deben seguir un conjunto de dispositivos, para poder intercambiar información según (Carolina Lagos, 2006).

Un importante número de empresas en nuestro país presentan la existencia de islas automatizadas (células de trabajo sin comunicación entre sí), siendo en estos casos las redes y los protocolos de comunicación Industrial indispensables para realizar un enlace entre las distintas etapas que conforman el proceso (Carolina Lagos, 2006).

La irrupción de los microprocesadores en la industria ha posibilitado su integración a redes de comunicación con importantes ventajas, entre las cuales figuran:

- Mayor precisión derivada de la integración de tecnología digital en las mediciones.
- Mayor y mejor disponibilidad de información de los dispositivos.
- Diagnóstico remoto de componentes

2.1.2 TECNOLOGÍA DE BUS DE CAMPO.

Físicamente podemos considerar a un bus (Automática U. d., 2010) como un conjunto de conductores que conectan conjuntamente varios circuitos para permitir el intercambio de datos. Contrario a una conexión punto a punto donde solo dos dispositivos intercambian información, un bus consta normalmente de un número de usuarios superior, además que generalmente un bus transmite datos en modo serial, a excepción de algún protocolo de bus particular como SCSI o IEEE-488, utilizado para interconexión de instrumentos de medición, que no es el caso de los buses tratados como buses de campo.

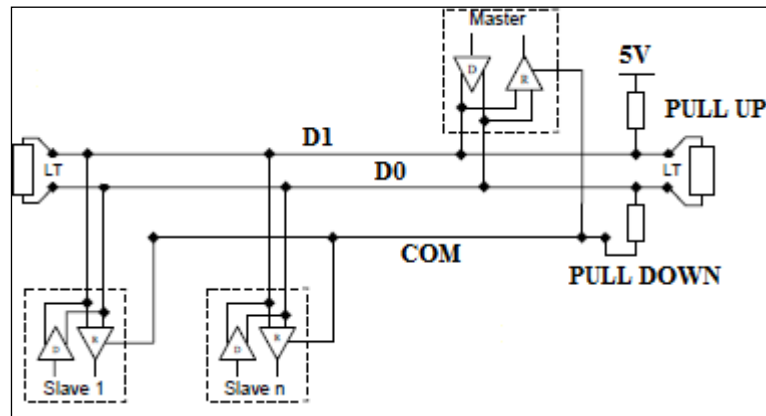


Figura 2. Diagrama de un Bus de comunicación.
 El estándar TIA/EIA-485 define un bus para la comunicación serie multipunto. (IDA, MODBUS OVER SERIAL LINE, 2006)

Para una transmisión serial es suficiente un número de cables muy limitado, generalmente dos o tres conductores y la debida protección contra las perturbaciones externas para permitir su tendido en ambientes de ruido industrial. (Automática U. d., 2010)

2.1.3 VENTAJAS DE UN BUS DE CAMPO.

- El intercambio puede llevarse a cabo por medio de un mecanismo estándar.
- Flexibilidad de extensión.
- Conexión de módulos diferentes en una misma línea.
- Posibilidad de conexión de dispositivos de diferentes procedencias.
- Distancias operativas superiores al cableado tradicional.
- Reducción masiva de cables y costo asociado.
- Simplificación de la puesta en servicio.

2.1.4 DESVENTAJAS DE UN BUS DE CAMPO.

- Necesidad de conocimientos superiores.
- Inversión de instrumentación y accesorios de diagnóstico.
- Costos globales inicialmente superiores.

2.1.5 PROCESOS DE COMUNICACIÓN POR MEDIO DE BUS.

El modo más sencillo de comunicación con el bus es el sondeo cliente/servidor. Más eficiente pero también más costoso es el Token bus (IEEE 802.4), que desde el punto de vista físico tenemos un bus lineal, y desde el punto de vista lógico un token ring.

2.1.6 TIPOS DE BUSES.

La mayoría de los buses trabajan en el nivel 1 con interfaz RS 485.

- **ASI (Actuator Sensor Interface):** Es el bus más inmediato en el nivel de campo y más sencillo de controlar. Consiste en un bus cliente/servidor con un máximo de 31 participantes que transmite por paquetes de solo 4 bits de datos. Es muy veloz, con un ciclo de 5 ms aproximadamente. Alcanza distancias de 100 m o hasta 300 m con ayuda de repetidores. (Ingeniería, 2011)
- **Bitbus:** Es el más difundido en todo el mundo; es del tipo cliente/servidor y admite como máximo 56 clientes; el paquete puede transmitir hasta 43 bytes de datos. (Ingeniería, 2011)
- **Profibus:** Es el estándar europeo en tecnología de buses; se encuentra jerárquicamente por encima de ASI y BITBUS, trabaja según procedimiento híbrido token passing, y dispone de 31 participantes hasta un máximo de 127. Su paquete puede transmitir un máximo de 246 bytes, y el ciclo para 31 participantes es de aproximadamente 90 ms. Alcanza una distancia de hasta 22300 m. (Ingeniería, 2011)
- **FieldBus en OSI:** En la arquitectura OSI, fieldbus ocupa los niveles 1 (Físico), 2 (Enlace de Datos) y 7 (Aplicación); teniendo en cuenta que este último no solo se encarga de la interfaz de usuario sino de aplicaciones específicas, dependiendo de cada aplicación. (Ingeniería, 2011)
- **Modbus:** Modbus es un protocolo de mensajería capa de aplicación, situado en el nivel 7 del modelo OSI, que proporciona la comunicación maestro / esclavo o cliente / servidor entre dispositivos conectados al bus. (Kryon_Ingeniería, 2014)

2.1.7 TOPOLOGÍA DE REDES INDUSTRIALES.

Los sistemas industriales usualmente consisten en dos o más dispositivos. Como un sistema industrial puede ser bastante grande, debe considerarse la topología de la red. Las topologías más comunes son: red bus, red estrella y red híbrida (Electric, 2010).

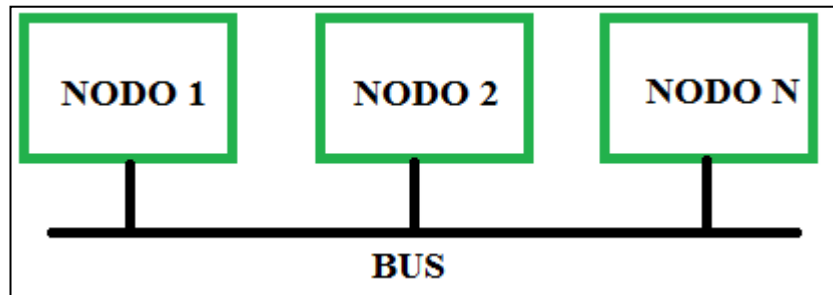


Figura 3. Topología Bus.

Todos los dispositivos comparten el mismo canal para comunicarse entre sí. (Electric, 2010)

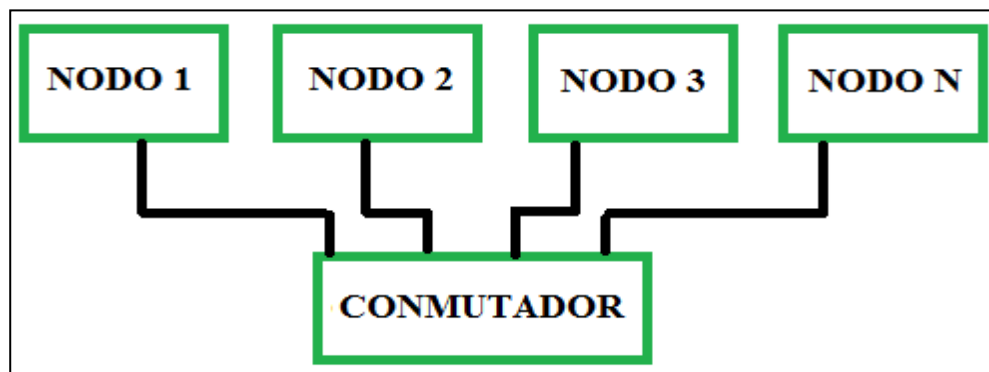


Figura 4. Topología Estrella.

Reduce la posibilidad de fallo de red conectando todos los nodos a un nodo central. (Electric, 2010)

2.1.8 BENEFICIOS DE UNA RED INDUSTRIAL.

- Reducción de cableado (físicamente).
- Dispositivos inteligentes (funcionalidad y ejecución).
- Control distribuido (flexibilidad).
- Simplificación de cableado de las nuevas instalaciones.
- Reducción de costo en cableado y cajas de conexión.
- Aplicable a todo tipo de sistema de manufactura.

- Incremento de la confiabilidad de los sistemas de producción.
- Optimización de los procesos existentes.

2.2 CONTROLADORES LÓGICOS PROGRAMABLES.

2.2.1 INTRODUCCIÓN.

De acuerdo a (SIEMENS, 2015) un Controlador Lógico Programable, o P.L.C. es un computador especialmente diseñado para automatización industrial, para el control de una maquina o proceso industrial.

El PLC nace como solución al control de circuitos complejos de automatización. A él se conectan los captadores (finales de carrera, pulsadores, etc.) por una parte, y los actuadores (bobinas de contactores, lámparas, pequeños receptores, etc.) por otra (SIEMENS, 2015).

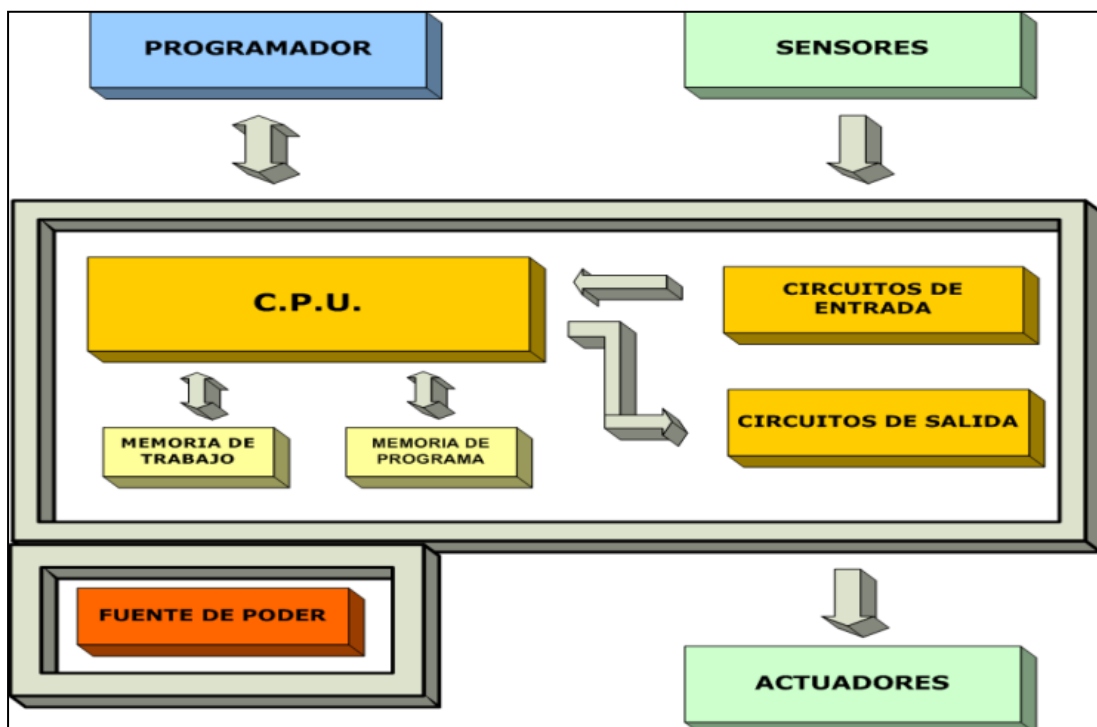


Figura 5. Arquitectura del PLC.

Los diferentes elementos que conforman esta arquitectura pueden formar parte de una base compacta del Autómata o ser módulos que se agreguen según el requerimiento. (Automática U. d.-I., 2010)

Un PLC permite controlar o proteger un proceso industrial, posibilitando además las opciones de monitoreo y diagnóstico de condiciones (alarmas), presentándolas en un HMI (Human-Machine Interface) o pantalla de operación (SIEMENS, 2015), o

presentándolas a una red de control superior. Es un ejemplo de control en tiempo real, pues reacciona automáticamente ante las condiciones variables que está vigilando. Puede ser parte de un sistema de control distribuido (DCS). O puede ser parte de un SCADA. O puede ser parte del sistema de seguridad. (MEX, 2008)

2.2.2 CLASIFICACIÓN DE LOS PLC's.

Compacto: Suelen integrar en el mismo bloque la alimentación, entradas y salidas y/o la CPU. Se expanden conectándoles a otros con parecidas características.

Modulares: Están compuestos por módulos o tarjetas adosadas al rack con funciones definidas (CPU, fuente alimentación, módulos E/S, módulos de comunicación, etc.) (ORG, 2008)

2.2.3 VENTAJAS.

Hablar sobre las ventajas que ofrece un PLC es un tema largo, pero aquí te presentare las más importantes:

- Menor tiempo empleado en su elaboración.
- Podrás realizar modificaciones sin cambiar cableado.
- La lista de materiales es muy reducida.
- Mínimo espacio de aplicación.
- Menor costo.
- Mantenimiento económico por tiempos de paro reducidos.

2.2.4 COMUNICACIÓN.

Las habilidades de comunicación del PLC comenzaron a aparecer en 1973 aproximadamente. El primer sistema fue el bus Modicon (Modbus). El PLC podía ahora dialogar con otros PLC y en conjunto podían estar aislados de las máquinas que controlaban. También podían enviar y recibir señales de tensión variables, entrando en el mundo analógico. Desafortunadamente, la falta de un estándar acompañado con un continuo cambio tecnológico hizo que la comunicación de PLC

sea un maremágnum de sistemas físicos y protocolos incompatibles entre sí (OMRON, 2010).

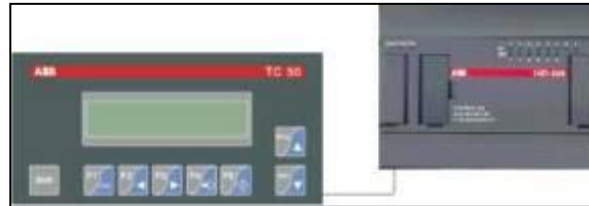


Figura 6. Comunicación Autómatas.
Tipo de comunicación serial entre diferentes dispositivos. (ORG, 2008)

2.3 RED DE COMUNICACIÓN PROTOCOLO MODBUS

Como indica (IDA, MODBUS APPLICATION PROTOCOL SPECIFICATION, 2012), el protocolo de comunicaciones industriales MODBUS fue desarrollado en 1979 por la empresa norteamericana MODICON y debido a que es público, relativamente sencillo de implementar y flexible se ha convertido en uno de los protocolos de comunicaciones más populares en sistemas de automatización y control. A parte de que muchos fabricantes utilizan este protocolo en sus dispositivos, existen también versiones con pequeñas modificaciones o adaptadas para otros entornos.

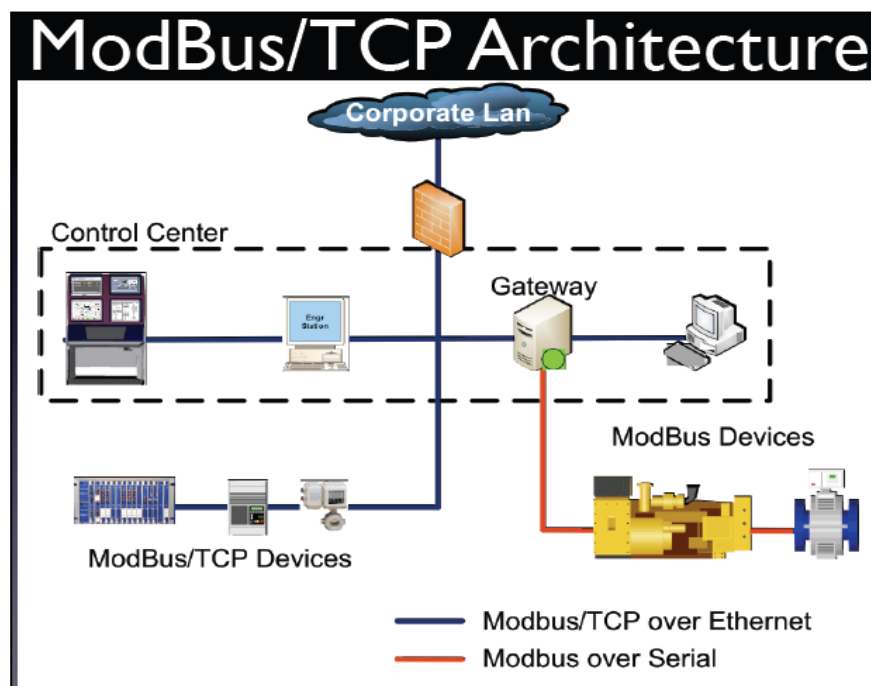


Figura 7. Arquitectura de red MODBUS.
Modbus TCP/IP es Modbus RTU encapsulado en TCP/IP, se requiere una pasarela que encapsule la trama Modbus RTU. (Barragan, 2013)

MODBUS especifica el procedimiento que el controlador y el esclavo utilizan para intercambiar datos, el formato de estos datos, y como se tratan los errores. No especifica estrictamente el tipo de red de comunicaciones a utilizar, por lo que se puede implementar sobre redes basadas en Ethernet, RS-485 y RS-232 (Bartolomé, 2011).

2.3.1 DESCRIPCIÓN GENERAL

MODBUS funciona siempre en modo maestro-esclavo (cliente - servidor), siendo el maestro (cliente) quien controla en todo momento las comunicaciones con los esclavos que pueden ser hasta 247. Los esclavos (servidores) se limitan a retornar los datos solicitados o a ejecutar la acción indicada por el maestro (Bartolomé, 2011). La comunicación del maestro hacia los esclavos puede ser de dos tipos:

- “Peer to peer”: en que se establece comunicación “maestro - esclavo”, el maestro solicita información y el esclavo responde.
- “Broadcast”: en que se establece comunicación “maestro - todos los esclavos”, el maestro envía un comando a todos los esclavos de la red sin esperar respuesta.

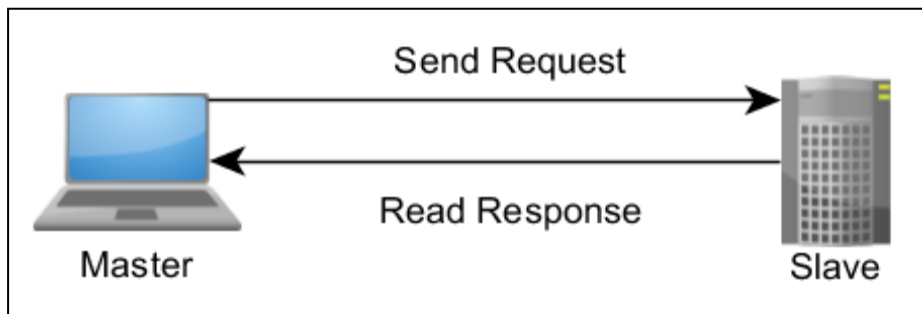


Figura 8. Relación de Red Maestro-Esclavo.

El maestro inicia una petición utilizando los códigos de funciones y a su vez el esclavo responde. (Instruments, 2014)

Como se puede ver, la secuencia básica en las comunicaciones MODBUS consiste siempre en una trama de pregunta, seguida de su correspondiente trama de respuesta:

- Pregunta: con el código de función que indica al esclavo que operación ha de realizar, y los bytes necesarios (datos, comprobación) para su ejecución.
- Respuesta: con la confirmación o datos resultantes de la ejecución de la función.

Existe algún caso concreto, en que hay más de una trama de respuesta para una trama de pregunta, como por ejemplo. cuando el maestro envía una operación cuya respuesta puede llevar al esclavo un tiempo elaborar. En estas situaciones el esclavo envía una primera respuesta indicando que aún no tiene los datos y tardará un tiempo en disponer de ellos, y otra segunda con los datos o confirmación de la operación.

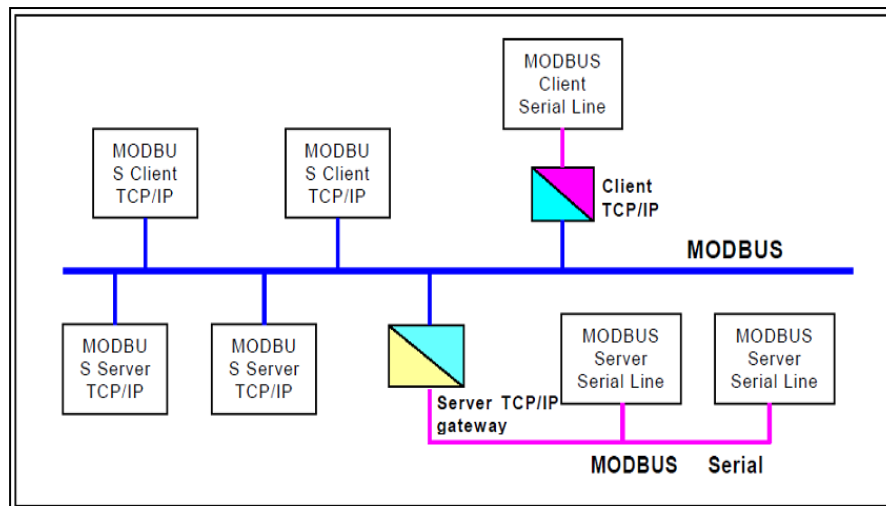


Figura 9. Esquema de comunicación MODBUS.

Se describen las arquitecturas Maestro/Esclavo y Cliente/Servido para Modbus TCP. (IDA, MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE, 2006)

Las comunicaciones MODBUS se pueden realizar en modo ASCII o en modo RTU. Para el modo ASCII los bytes se envían codificados en ASCII, dicho de otra manera, por cada byte a transmitir se envían dos caracteres ASCII (2 bytes) con su representación hexadecimal. Para el modo RTU (el más usado) se envían en binario.

2.3.2 CAMPOS DE LAS TRAMAS MODBUS

El número de campos de las tramas MODBUS varía ligeramente dependiendo de si utilizamos la codificación ASCII o RTU:

2.3.2.1 CODIFICACIÓN ASCII:

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

Figura 10. Trama ASCII.

Estructura típica de la trama Modbus en formato ASCII. (IDA, MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE, 2006)

2.3.2.2 CODIFICACIÓN RTU (EN EL FORMATO BINARIO):

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low CRC Hi

Figura 11. Trama RTU.

Estructura típica de la trama Modbus en formato RTU. (IDA, MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE, 2006)

2.3.3 DESCRIPCIÓN DE LOS CAMPOS DE LAS TRAMAS MODBUS RTU.

Número de Esclavo (1byte): En el caso de las tramas enviadas por el máster, el campo de número de esclavo indica la dirección del destinatario de esta trama. Permite direccionar hasta 247 esclavos. Las tramas broadcast, no tienen asociada respuesta, y algunas implementaciones de MODBUS no admiten la trama de broadcast. (Bartolomé, 2011).

Código de Operación o Función (1byte): Indica el tipo de operación que queremos realizar sobre el esclavo. Las operaciones se pueden clasificar en dos tipos:

De lectura / escritura en memoria: para consultar o modificar el estado de los registros del mapa de memoria del esclavo.

Órdenes de control del esclavo: para realizar alguna actuación sobre el esclavo.

El código de operación puede tomar cualquier valor comprendido entre el 0 y el 127. Cada código se corresponde con una determinada operación. Algunos de estos códigos se consideran estándar y son aceptados e interpretados por igual por todos los dispositivos que dicen ser compatibles con MODBUS, mientras que otros códigos son implementaciones propias de cada fabricante. Es decir que algunos fabricantes realizan implementaciones propias de estos códigos “no estándar”. (Bartolomé, 2011).

Es también mediante el código de función que el esclavo confirma si la operación se ha ejecutado correctamente o no. Si ha ido bien responde con el mismo código de operación que se le ha enviado, mientras que si se ha producido algún error, responde también con el mismo código de operación pero con su bit de más peso a 1 y un byte en el campo de datos indicando el código de error que ha tenido lugar. (Mista, 2012).

Dirección, datos y subfunciones (n bytes): Este campo contiene la información

necesaria para realizar la operación indicada en el código de operación. Cada operación necesitará de unos parámetros u otros, por lo que el número de bytes de este campo variará según la operación a realizar. En el caso del esclavo, este puede responder con tramas con o sin campo de datos dependiendo de la operación. En los casos en que se produzca algún error es posible que el esclavo responda con un byte extra para especificar el código de error. (Kryon_Ingeniería, 2014).

Al establecer la dirección de una variable u otro elemento en el mapa de direcciones Modbus, direccionamos con 1 unidad menos a la del registro al que queremos acceder, para una mejor comprensión se detallan los siguientes ejemplos:

- El relé número 1 de un controlador se direccionaría con el valor 0000 en el campo de dirección de un mensaje MODBUS.
- El relé 0x007F (127d) de un controlador se direccionaría con el valor 0x007E (126d) en el campo de dirección de un mensaje MODBUS.
- El Holding Register 40108 es accedido leyendo de la dirección 0x006B (107d).

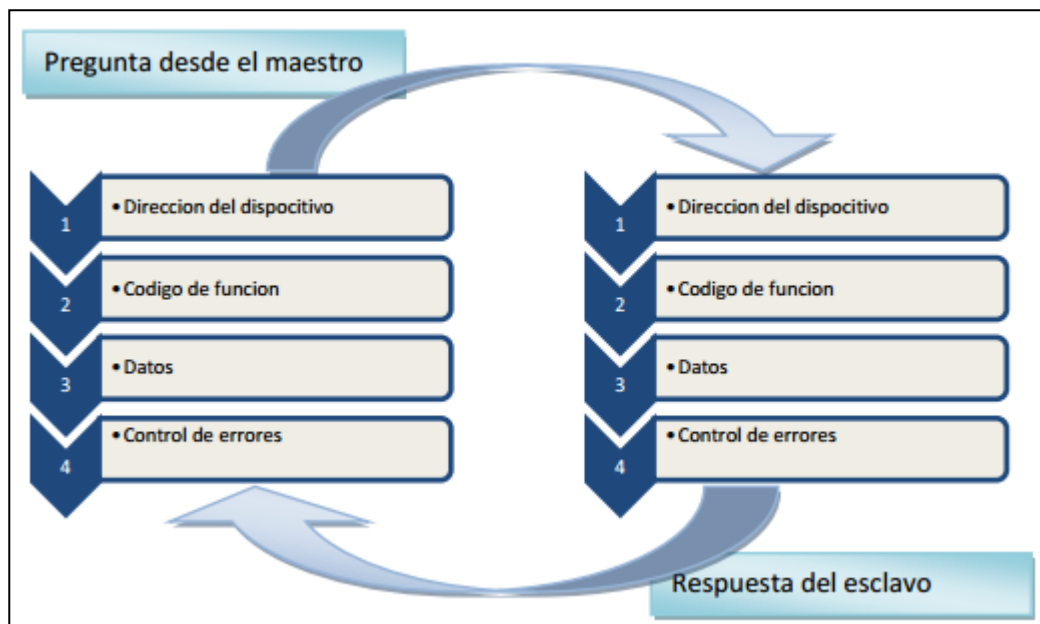


Figura 12. Ciclo de Preguntas y Respuestas.
Se evidencia el clico la trama en el proceso de petición y respuesta. (Mista, 2012)

2.3.4 MAPA DE REGISTOS MODBUS.

En MODBUS cada tipo de dato se mapea en un rango de memoria concreto:

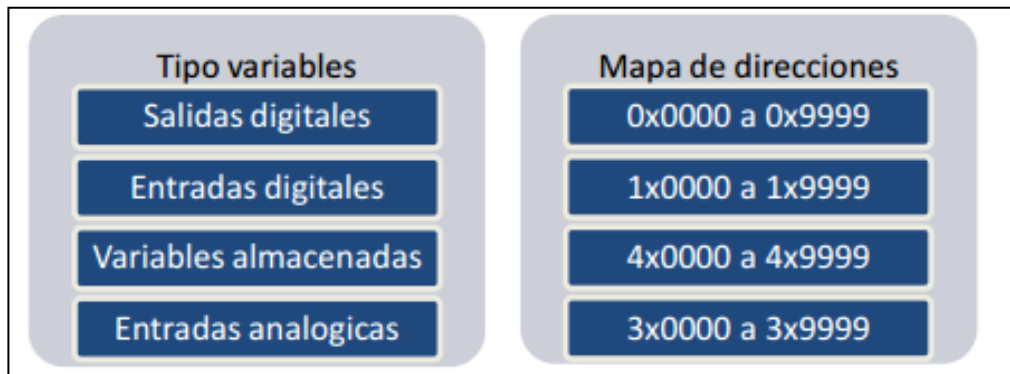


Figura 13. Mapa de Registros Modbus.

Especifica el espacio de memoria designado en el PLC a cada tipo de variable. (Mista, 2012)

1-10000 (Salidas digitales): 1 bit por dirección para indicar el estado de una salida o relé (0 desactivado, 1 activado). Las direcciones de este rango se suelen acceder mediante las funciones 1 para lectura, 5 para escritura, 15 de escritura múltiple, según (SIEMENS_AG, 2010).

10001-20000 (Entradas digitales): 1 bit por dirección para leer el estado de una entrada digital (0 desactivada, 1 activada). Las direcciones de este rango se suelen acceder con la función 2 para lectura y llevan implícita la dirección 10001 como dirección base (para acceder a una dirección bastará con especificar la distancia entre esta y la dirección base), en base a (SIEMENS_AG, 2010).

20001-30000: el protocolo MODBUS estándar no hace uso de este rango de direcciones. Según (Bartolomé, 2011)

30001-40000 (Entradas analógicas): 16 bits por dirección con el estado de las medidas o entradas analógicas. Dependiendo del dispositivo, este puede hacer uso de más de un registro para almacenar la información de la medida, así con 2 registros consecutivos podríamos almacenar medidas de 32 bits. Las direcciones de este rango se acceden mediante la función 4 para lectura y llevan implícita la dirección 30001 como dirección base (para acceder a una dirección bastará con especificar la distancia entre esta y la dirección base), según indica (Mista, 2012).

40001-50000 (Salidas Analógicas): 16 bits con los registros de salidas analógicas o de propósito general (Output Registers – Holding Registers). Se acceden con las funciones 3 para lectura, 6 para escritura o 16 de escritura múltiple y llevan implícita la dirección 40001 como dirección base (para acceder a una dirección bastará con especificar la distancia entre esta y la dirección base), indicado por (Bartolomé, 2011).

2.3.5 DESCRIPCIÓN DE LOS CÓDIGOS DE FUNCIÓN.

Los siguientes códigos son algunos de los códigos de función MODBUS más extendidos, soportados por todos los dispositivos que cumplen con las especificaciones del estándar.

Tabla 1. Principales códigos de funciones.

FUNCIÓN	CÓDIGO	DESCRIPCIÓN
1	01H	Lectura de múltiples bits (bobinas) de salida o internos
2	02H	Lectura de múltiples (bobinas) bits de entradas
3	03H	Lectura múltiples registros (palabras) de salida o internos
4	04H	Lectura múltiples registros (palabras) de entrada
5	05H	Escritura de una bobina (Bit)
6	06H	Escritura de un registro (palabra)
15	0FH	Escritura de múltiples bits (bobinas)
16	10H	Escritura de múltiples registros (palabras)

Nota: Solo se describen los principales códigos de funciones. (Barragan, 2013)

2.3.5.1 FUNCIÓN 1 O 2 (1 LEER BOBINAS - 2 LEER ENTRADAS DISCRETAS):

Permite realizar la lectura del estado de las DIs (@1XXXX el comando 2-Read input status) o DOs (@0XXXX el comando 1-Read Coil Status). Para ello el maestro solicita el número de bits que desea leer a partir de una determinada dirección. Cada dirección se corresponde con un registro de 1 bit con el estado de la entrada digital. El esclavo responde indicando el número de bits que retorna y sus valores. En la trama de respuesta se aprovechan todos los bits del byte, y puede haber hasta 256 bytes. (SIEMENS_AG, 2010)

Tabla 2. Solicitud 01 y 02.

	Table Index	Most significant byte	Least significant byte
Control table	0	01 (Transmission/rece	06 (Transmission length) (*)
	1	03 (Reception	00 (Transmission offset)
Transmission table	2	Slave@(1..247)	01 or 02 (Request code)
	3	Address of the first bit to read	
	4	N ₁ = Number of bits to read	
	5	Slave@(1..247)	01 or 02 (Response code)
Reception table	6	00 (byte added by Rx Offset action)	N ₂ = Number of data bytes
	7	Value of the 1 st byte (value = 00 or 01)	Value of the 2 nd byte (if N ₁ >1)
	8	Value of the 3 rd byte (if N ₁ >1)	
	...		
	(N ₂ /2)+6 (if N ₂ is even) (N ₂ /2+1)+6 (if N ₂ is odd)	Value of the N ₂ th byte (if N ₁ >1)	

Nota: (*) Este byte recibe también la longitud de la cadena de transmisión después de la respuesta. (Schneider-Electric, 2011)

2.3.5.2 FUNCIÓN 3 O 4 (3 LEER MÚLTIPLES REGISTROS – 4 LEER REGISTROS DE ENTRADA):

Permite realizar la lectura del valor de las AIs (@4XXXX el comando 3 Read Holding Registers) o AOs (@3XXXX el comando 4 Read Input Registers). El máster indica la dirección base y número de palabras a leer a partir de esta, mientras que el esclavo indica en la respuesta el número bytes retornados, seguido de estos valores. Aunque en realidad se está escribiendo en el rango de registros o valores numéricos, los registros son direccionados a partir de la dirección. (Kryon_Ingeniería, 2014).

Tabla 3. Solicitud 03 y 04.

	Table Index	Most significant byte	Least significant byte	
Control table	0	01 (Transmission/reception)	06 (Transmission length) (*)	
	1	03 (Reception Offset)	00 (Transmission offset)	
Transmission table	2	Slave@(1..247)	03 or 04 (Request code)	
	3	Address of the first word to read		
	4	N = Number of words to read		
	Reception table	5	Slave@(1..247)	03 or 04 (Response code)
		6	00 (byte added by Rx Offset action)	2*N (number of bytes read)
		7	First word read	
		8	Second word read (if N>1)	
		...		
		N+6	Word N read (if N>2)	

Nota: (*) Este byte recibe también la longitud de la cadena de transmisión después de la respuesta. La Rx compensada en tres añadirá un byte (valor = 0) en la tercera posición en la tabla de la recepción. Esto asegura un buen posicionamiento del número de bytes leídos y de los valores de las palabras leídas en esta tabla. (Schneider-Electric, 2011)

2.3.5.3 FUNCIÓN 5 (ESCRIBIR UNA BOBINA):

Permite modificar el estado de una DO del esclavo (mando o relé). Es decir mediante este comando podemos modificar algún bit de alguna de las variables internas del esclavo u ordenar la ejecución o activación de un mando. Actúa sobre la zona de memoria de los DOs @0XXXX. El Maestro especifica la dirección del bit o mando que quiere modificar seguido de 0x00 para ponerlo a 0 o 0xFF para ponerlo a 1. El esclavo responde con una trama similar indicando la dirección que ha modificado y el valor que ha establecido en el bit o mando. (Kryon_Ingeniería, 2014).

Tabla 4. Solicitud 05.

	Table Index	Most significant byte	Least significant byte
Control table	0	01 (Transmission/reception) (*)	06 (Transmission length)
	1	00 (Reception offset)	00 (Transmission offset)
Transmission table	2	Slave@(1..247)	05 (Request code)
	3	Address of the bit to write	
	4	Bit value to write	
Reception table	5	Slave@(1..247)	05 (Response code)
	6	Address of the bit written	
	7	Value written	

Nota: (*) Este byte recibe también la longitud de la cadena de transmisión después de la respuesta. Esta petición no necesita el uso de offset. Para escribir 1 al bit, la palabra asociada en la tabla de emisión debe contener el valor FF00H, y 0 para escribir 0. (Schneider-Electric, 2011)

2.3.5.4 FUNCIÓN 6 (ESCRIBIR UN REGISTRO):

Permite la escritura en las AOs del esclavo (ya sea una señal o valor interno del equipo), y por tanto actúa sobre la zona de memoria de las AOs (@4XXXX). Debemos indicar la dirección del valor que queremos modificar y la magnitud que queremos asignarle. Luego el esclavo debería responder con la dirección del dato que ha modificado y el valor que le ha asignado, que debería coincidir con el enviado. Aunque en realidad se está escribiendo en el rango de AOs, los registros son direccionados a partir de la dirección 0. (SIEMENS_AG, 2010)

Tabla 5. Solicitud 06.

	Table Index	Most significant byte	Least significant byte
Control table	0	01 (Transmission/reception)	06 (Transmission length) (*)
	1	00 (Reception offset)	00 (Transmission offset)
Transmission table	2	Slave@(1..247)	06 (Request code)
	3	Address of the word to write	
	4	Word value to write	
Reception table	5	Slave@(1..247)	06 (Response code)
	6	Address of the word written	
	7	Value written	

Nota: (*) Este byte recibe también la longitud de la cadena de transmisión después de la respuesta. Esta petición no necesita el uso de offset. La trama de respuesta es la misma que la trama de petición aquí. (Schneider-Electric, 2011)

2.3.5.5 FUNCIÓN 15 (ESCRIBIR MÚLTIPLES BOBINAS):

Permite la modificación simultanea de varios bits de DOs en el esclavo, pasándolos a OFF ('0') o a ON ('1') según convenga. Actúa sobre la zona de memoria de las DOs (@0XXXX). (Kryon_Ingeniería, 2014)

El Maestro especifica la dirección del primer bit o bobina que se quiere modificar seguido del número total de bit a escribir. Luego se coloca consecutivamente los valores que se les va a escribir a cada bit (0x00 para ponerlo a 0 o 0xFF para ponerlo a 1). El esclavo responde con una trama similar indicando las direcciones que han modificado.

Tabla 6. Solicitud 15

	Table Index	Most significant byte	Least significant byte
Control table	0	01 (Transmission/receptio	8 + number of bytes(transmission)
	1	00 (Reception Offset)	07 (Transmission offset)
Transmission table	2	Slave@(1..247)	15 (Request code)
	3	Number of the first bit to write	
	4	N ₁ = Number of bits to write	
	5	00 (byte not sent, offset effect)	N ₂ = Number of data bytes to write = $[1+(N_1-1)/8]$,
	6	Value of the 1 st byte	Value of the 2 nd byte
	7	Value of the 3 rd byte	Value of the 4 th byte
	...		
	(N ₂ /2)+5 (if N ₂ is even) (N ₂ /2+1)+5 (if N ₂ is odd)	Value of the N ₂ th byte	
		Slave@(1..247)	15 (Response code)
Reception table		Address of the 1 st bit written	
		Address of bits written (= N ₁)	

Nota: El Tx Offset = 7 suprimirá el séptimo byte de la trama enviada. Esto también permite una buena correspondencia de valores de las palabras en la tabla.

Aunque el estado de las DOs se especifica bit a bit, las tramas se componen de bytes, y esto obliga a enviar los estados en grupos de 8. El esclavo no debería hacer caso a los bits sobrantes, es decir, no debería considerar los que queden por encima del último bit indicado en el campo “cantidad de bits a modificar”. (Kryon_Ingeniería, 2014).

2.3.5.6 FUNCIÓN 16 (ESCRIBIR MÚLTIPLES REGISTROS):

Permite realizar la escritura en un grupo de registros, y por tanto actúa sobre la zona (@4XXXX). Se debe especificar la dirección a partir de la que queremos comenzar a actualizar valores, el número de valores que queremos actualizar, y la lista de valores que queremos asignar a estos registros. Estos registros son direccionados a partir de la dirección 0 (es decir el registro @40001 se direcciona 0). (SIEMENS_AG, 2010).

Tabla 7. Solicitud 16

	Table Index	Most significant byte	Least significant byte	
Control table	0	01 (Transmission/reception)	8 + (2*N) (Transmission length)	
	1	00 (Reception offset)	07 (Transmission offset)	
Transmission table	2	Slave@(1..247)	16 (Request code)	
	3	Address of the first word to write		
	4	N = Number of words to write		
	5	00 (byte not sent, offset effect)	2*N = Number of bytes to write	
	6	First word value to write		
	7	Second value to write		
	...			
	N+5	N values to write		
	Reception table	N+6	Slave@(1..247)	16 (Response code)
		N+7	Address of the first word written	
N+8		Address of words written (= N)		

Nota: El Tx Offset = 7 suprimirá el séptimo byte de la trama enviada. Esto también permite una buena correspondencia de valores de las palabras en la tabla. (Schneider-Electric, 2011)

2.3.6 CÓDIGOS ERRORES EXCEPCIONES.

Cuando se produce un error en la ejecución de un comando en el esclavo, este responde poniendo a 1 el bit de más peso del código de función. Con este bit el maestro sabe que se ha producido un error, pero para obtener más detalle sobre el tipo de error, ha de comprobar el campo de datos. (Bartolomé, 2011)

Para las redes Modbus se utiliza dos clases de métodos de revisión de error. El contenido del campo de revisión de error depende de que método se esté utilizando. (Instruments, 2014)

Cuando se utiliza el modo ASCII, el campo de revisión de error contiene dos caracteres, los cuales son resultado del cálculo de un Chequeo de Redundancia Longitudinal (LRC), que es realizado en el contenido del mensaje y se añaden al mismo como último campo de la trama. (Bartolomé, 2011)

Para el modo RTU, el campo de revisión de error utiliza 16 bits implementados en dos

bytes. El valor de la revisión de error es el resultado del cálculo de un Chequeo de Redundancia Cíclica (CRC) que es realizado en el contenido del mensaje y se añade como último campo de la trama. El byte de orden inferior del campo es añadido primero, seguido del byte de orden superior. (Manuel Jiménez Buendía, 2010)

Tabla 8. Códigos errores excepciones.

CÓDIGO	NOMBRE	SIGNIFICADO
1	ILLEGAL FUNCTION	El código de función recibido no se corresponde a ningún comando disponible en el esclavo.
2	ILEGAL DATA ADDRESS	La dirección indicada en la trama no se corresponde a ninguna dirección válida del esclavo.
3	ILLEGAL DATA VALUE	El valor enviado al esclavo no es válido.
4	SLAVE DEVICE FAILURE	El controlador no responde o ha ocurrido un error.
5	ACKNOWLEDGE	Se ha aceptado la función y se está procesando.
6	SLAVE DEVICE BUSY	El esclavo está ocupado realizando otra tarea y no puede atender a esa petición en ese instante por lo que el máster tendrá que reintentarlo más adelante.

Nota: El diagrama de estado para cada código de función debe de cubrir al menos el código de excepción 01. (Instruments, 2014)

Las redes Modbus utilizan dos tipos de revisión de error. La revisión de paridad (par o impar) es opcional. La revisión de trama (LRC o CRC) es aplicada al mensaje entero. Ambas revisiones son generadas en el dispositivo maestro y aplicadas al contenido del mensaje antes de la transmisión.

El dispositivo esclavo revisa cada carácter y la trama del mensaje entero durante la recepción. El maestro es configurado por el usuario para esperar un intervalo de tiempo antes de abortar la transacción. Este intervalo es fijado para ser lo suficientemente largo para cualquier respuesta normal del esclavo.

2.3.7 MACROS DE COMUNICACIONES DE TWIDO SUITE.

La plataforma de programación de Schneider Electric “TwidoSuite” utilizada para la elaboración y carga de programas al PLC Twido, facilita el proceso de configuración de las tablas de peticiones Modbus, por medio de “Macros de Comunicaciones” establecidas por este fabricante, con el fin de reducir el número de líneas de programa y el tiempo de elaboración.

Las macros de comunicación para Twido nos permiten realizar el intercambio de datos con cualquier tipo de esclavo Modbus.

Se pueden implementar para cualquiera de los puertos Modbus que acepta Twido:

- **Puerto 1:** puerto serie integrado en Twido que permite comunicar en protocolo Modbus.
- **Puerto 2:** puerto serie opcional para comunicación Modbus RS-485.
- **Puerto 3:** Puerto integrado en ciertas CPUs para comunicación Modbus TCP/IP.

Es posible realizar la escritura/lectura de un bit o palabras de memoria, accediendo a un registro o múltiples registros.

Tabla 9. Funciones Macros de comunicación.

MACRO	FUNCIÓN
C_RD1B	Realiza lecturas de 1 bit (bobina)
C_RD1W	Realiza lecturas de 1 registro (palabra)
C_WR1B	Realiza escritura de 1 bit (bobina)
C_WR1W	Realiza escritura de 1 registro (palabra)
C_RDNW	Realiza lectura de Múltiples registros (palabras)
C_WRNW	Realiza escritura de Múltiples registros (palabras)

Nota: Cada macro ocupa un determinado espacio de memoria del Twido.

3 CAPÍTULO 3 : DISEÑO DEL SISTEMA A ESCALA

3.1 DESCRIPCIÓN DE ELEMENTOS

En el desarrollo de las dos maletas didácticas, para las prácticas de la comunicación entre dispositivos industriales mediante protocolo Modbus RTU, se utilizan autómatas, relés inteligentes, variadores de frecuencia y motores, como equipos principales, los cuales serán descritos en este capítulo.

Para la implementación de este proyecto se utilizan controladores de la marca Schneider Electric, los cuales están dotado con una interface de comunicación tipo serial que admite el protocolo Modbus RTU. Mediante el cableado a 2 hilos según el estándar TIA/EIA-485 que define un bus para la transmisión serie multipunto se realiza la transferencia de datos entre los equipos.

3.1.1 DESCRIPCIÓN DEL AUTÓMATA TWIDO.

Esta gama de controlador programable es de tipo compacto, ofrece una solución “todo en uno” con unas dimensiones reducida. Esta base compacta utiliza una alimentación de corriente alterna comprendida entre a 100 y 240 VAC.

Este autómata cuenta con una capacidad de 14 entradas digitales y 10 salidas digitales las cuales sirven para realizar prácticas con actuadores del tipo ”todo o nada”, adicional se puede instalar un puerto serie para comunicación con protocolo Modbus RTU.

En el desarrollo del proyecto se consideró utilizar este autómata como un elemento maestro de la red Modbus RTU implementada dentro de cada maleta didáctica.

A continuación se detalla cada uno de los elementos principales del autómata Twido:

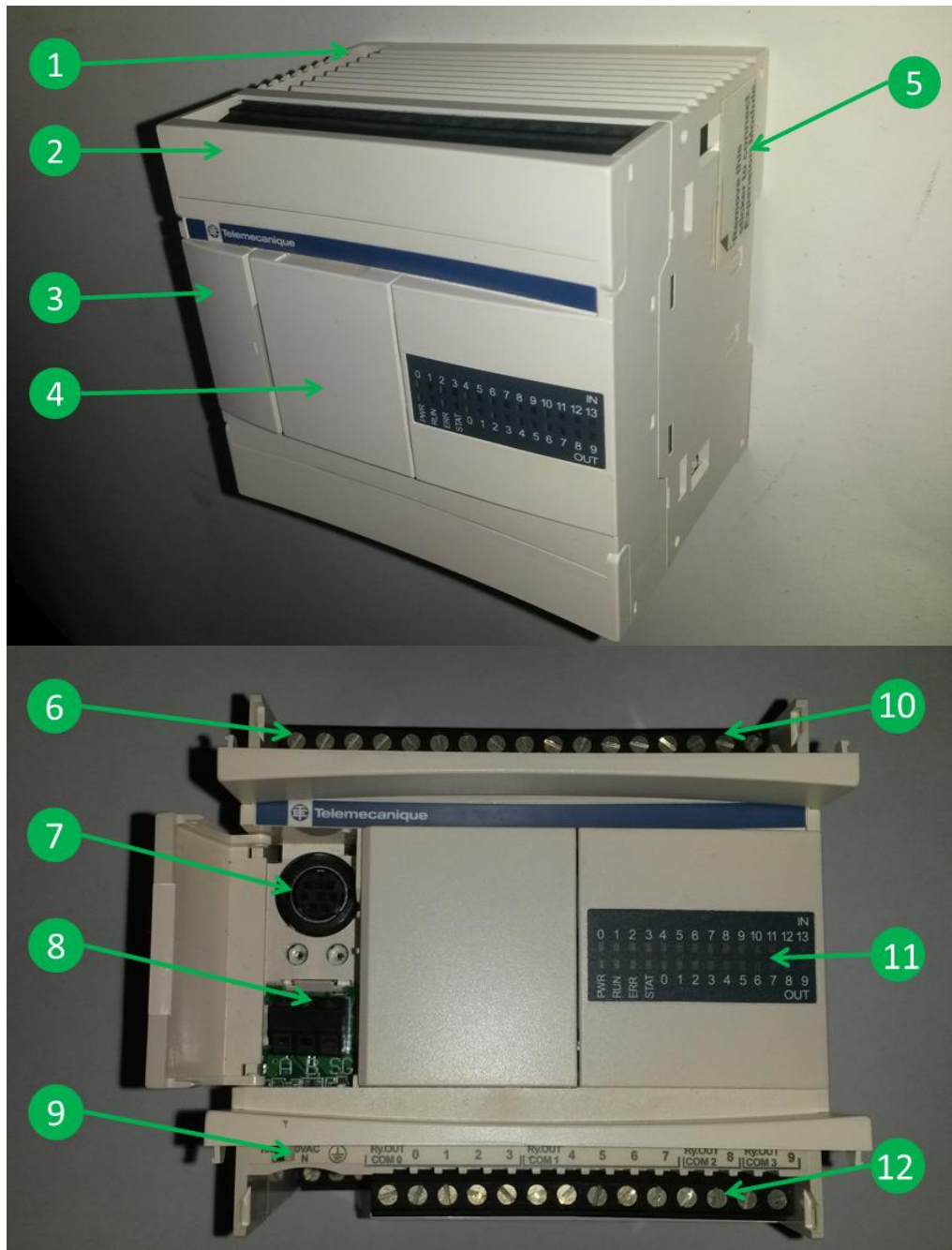


Figura 14. Componentes del Automata Twido.

Se indica cada una de las partes del automata para así obtener un mejor manejo del equipo. (Schneider Electric, 2011)

- 1: Orificio de montaje.
- 2: Cubierta de terminal
- 3: Puerta de acceso
- 4: Cubierta extraíble del conector del HMI
- 5: Conector de ampliación (en las series 24DRF y 40DRF)

- 6: Terminales de potencia del sensor
- 7: Puerto serie 1
- 8: Conector de puerto serie 2.
- 9: Terminales de fuente de alimentación
- 10: Terminales de entradas
- 11: Indicadores LED
- 12: Terminales de salidas

3.1.2 DESCRIPCIÓN DEL CABLE DE PROGRAMACIÓN PARA TWIDO

El convertidor TSXCUSB485 es un dispositivo de comunicación multifuncional que convierte las señales de serie, a través de una conexión USB, en señales RS485. Proporciona el vínculo entre un PC equipado con un puerto USB tipo A y varios otros dispositivos.



Figura 15. Convertidor RS485-USB.
Utilizado para las descargas de programas y monitoreo de las variables. (Schneider Electric, 2011)

Mediante este cable se realizan las descargas de los programas al autómata Twido para la realización de prácticas. Este cuenta con una perilla de cuatro posiciones la cual nos ayuda a seleccionar su modo de funcionamiento.

Tabla 10. Posiciones de perilla

POSICIÓN	FUNCIÓN	USO
0	TER MULTI -Modo multipunto	NO USADO
1	OTHER MULTI - Modo multipunto. Otras comunicaciones	NO USADO
2	TER DIRECT - Modo Punto a Punto.	USADO
3	OTHER DIRECT - Modo Punto a Punto. Otro tipo de comunicación (Modbus)	USADO

Nota: Se describen los modos de funcionamiento del cable.

3.1.3 DESCRIPCIÓN DE COMPONENTES DEL ZELIO LOGIC

Incluyendo 4 entradas digitales, 2 entradas análogas de 0-10V y 4 salidas además posee reloj, la tensión de alimentación es de 24 VCC. La alimentación del módulo de comunicación Modbus se hace a través del relé inteligente al cual está conectado. A continuación se describen los componentes más importantes del relé Zelio Logic:

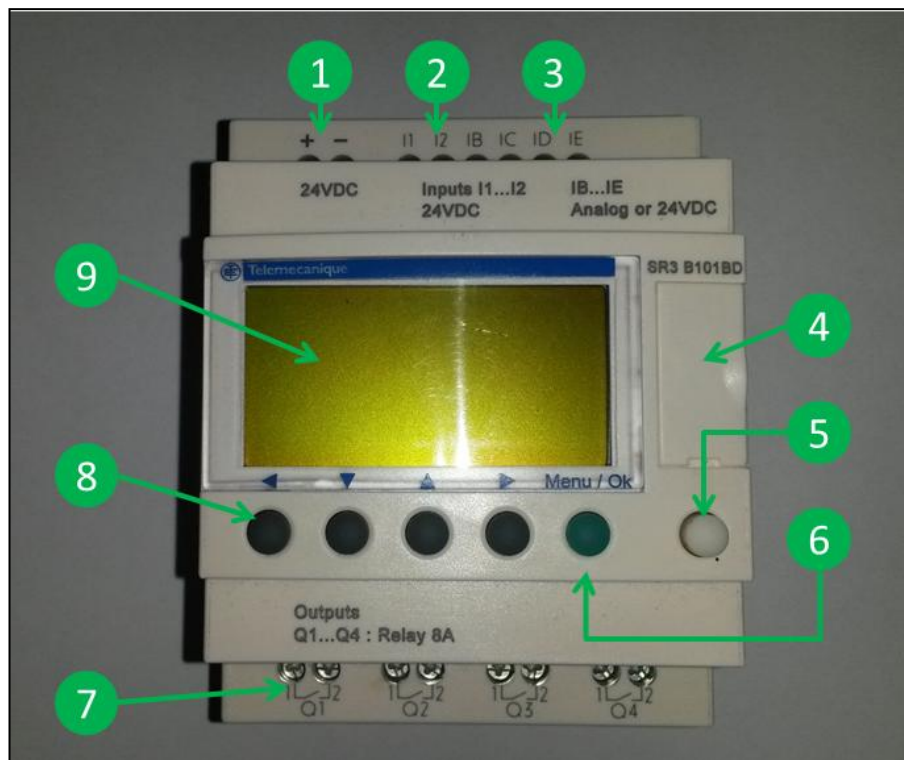


Figura 16. Componentes del Zelio Logic.

Se indica cada uno de los componentes del relé inteligente para así, obtener un mejor manejo del equipo. (Schneider Electric, 2011)

- 1: Bloque terminal de alimentación.
- 2: Bloque terminal de entradas Digitales.
- 3: Bloque terminal de entradas analógicas 0-10 voltios utilizables en entradas DIG según el modelo.
- 4: Alojamiento de la memoria de copia de seguridad o cable conexión para PC.
- 5: Tecla Mayúscula (blanca).
- 6: Tecla Menú/Aceptar (verde) de selección y validación.
- 7: Bloque terminal de salida para relé.
- 8: Teclas de navegación (grises) o, después de la configuración, botones pulsadores.

3.1.4 MÓDULO DE COMUNICACIÓN PARA ZELIO LOGIC.

Para poder integrar el relé Zelio Logic a la red Modbus RTU, necesita de una interfaz de comunicación. Mediante esta interfaz el equipo podrá intercambiar datos y recibir peticiones desde un maestro Modbus. La alimentación de la interfaz de comunicación se suministra a través del mismo relé modular.

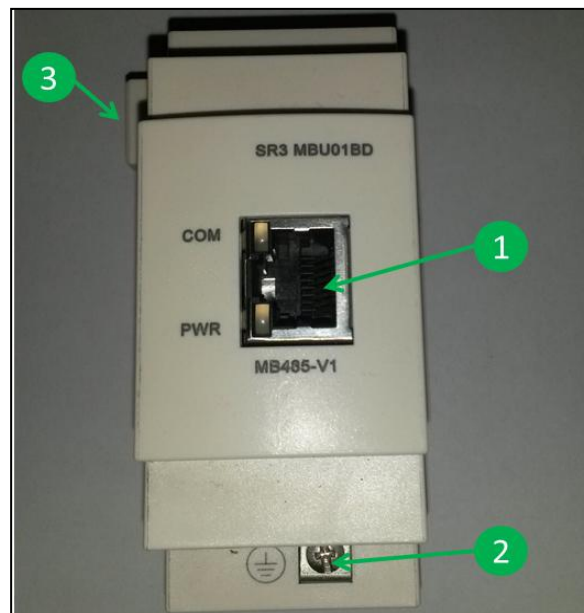


Figura 17. Módulo de Comunicación. Indica cada componente del módulo de comunicación para así, comprender de mejor manera su funcionamiento. (Schneider Electric, 2011)

1: Conexión de red Modbus (blindado RJ45 conector hembra) con 2 LED de visualización.

2: Tornillo-en tablero de bornes para la conexión a tierra de protección

3: Puerto de Conexión



Figura 18. Conexión Zelio y Módulo de comunicación.

Se visualiza como queda el bloque compacto que resulta tras la conexión del módulo de comunicación al Zelio Logic. (Schneider Electric, 2011)

Como se puede observar la interfaz de comunicación Modbus tiene un puerto RJ-45, contrario al autómatas asignado como maestro para la mayoría de las prácticas, que cuenta con terminales de bornera. Es por este motivo que se desarrolló un adaptador que permite pasar de RJ-45 a terminales de bornera y así mayor facilidad al momento de cablear los pines que corresponden entre los equipos a comunicarse.



Figura 19. Adaptador RJ-45/Bornera.
Solo se utilizan tres hijos, D1, D0 y GND. Se colocó termoencogible para protección.

3.1.5 CABLE DE PROGRAMACIÓN ZELIO

Esta herramienta de programación permite conectar el relé Zelio Logic al PC y mediante el software Zelio Soft 2 podremos realizar la lógica de programación y descargarla al equipo.



Figura 20. Cable de programación Zelio.
Se utiliza tanto para la programación del Zelio como para la descarga y actualización del firmware. (Schneider Electric, 2011)

3.1.6 VARIADOR DE FRECUENCIA ATV312.

Se utilizara para cada maleta didáctica un variador de frecuencia de la marca Schneider Electric que cumple con las siguientes características técnicas:

- Capacidad de 1 HP (0.75 KW).
- Tensión nominal de alimentación de 200 V a 240 V.
- Frecuencia de alimentación de 50 Hz a 60 Hz.
- Monofásico.
- Corriente de línea 7.5 A - 8.9 A.
- Disipación de potencia de 60W a carga nominal.
- Puerto de comunicación Modbus RTU.

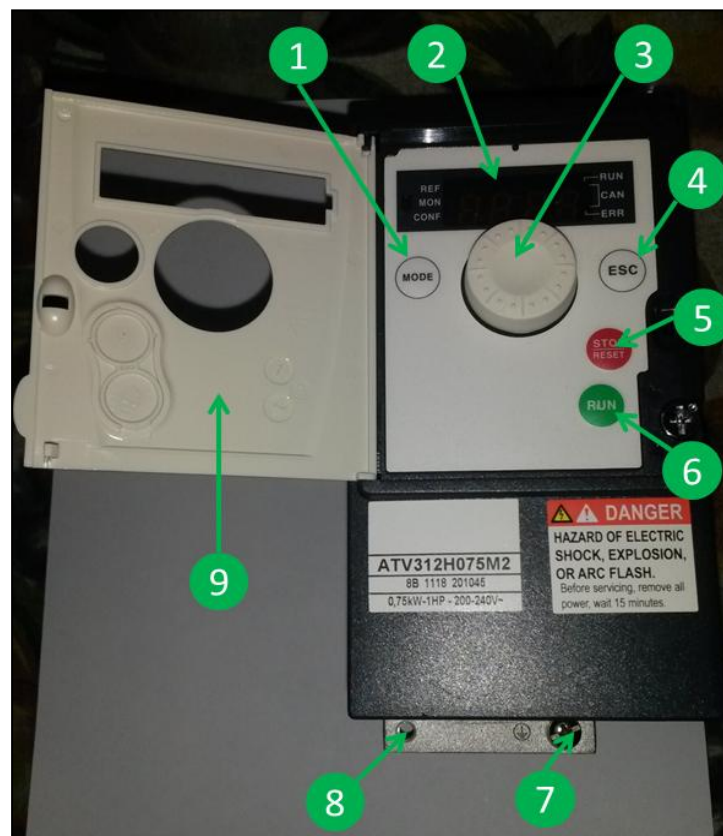


Figura 21. Componentes del Variador de Frecuencia. Se describe cada elemento que lo conforma para su correcto manejo y operación. (Schneider Electric, 2011)

- 1: Botón MODE, selecciona velocidad de referencia.
- 2: Pantalla gráfica.
- 3: Botón ESC, se utiliza para salir de un menú / parámetro.
- 4: Rueda de navegación entre los diferentes Menús.
- 5: botón STOP, se utiliza para detener el motor y realizar un reinicio.
- 6: Botón RUN, ejecuta la función suponiendo que ha sido configurado.
- 7: Borne de aterramiento del Variador de Frecuencia.
- 8: Orificio de anclaje.

Este variador cuenta con una interfaz Modbus RTU integrada, la cual permitirá configurarlo para establecer la comunicación y por medio de registros o espacios de memorias Modbus asignados, poder realizar las peticiones desde algún maestro dentro de la red.

Esto nos permitirá más adelante realizar prácticas de “Marcha” y “Paro” del motor el cual esta controlador por el variador. También podremos controlar la frecuencia del equipo mediante el acceso al registro asignado por el fabricante el cual lo podremos encontrar en la hoja de técnica.

3.1.7 MOTOR TRIFASICO.

Cada maleta didáctica cuenta con un motor de 0.5 HP con alimentación trifásica y tensión a 220VAC, el cual se utilizará para ser controlado desde un variador de frecuencia que estará integrado a un bus de comunicación Modbus RTU.

Cada motor se instaló con sus debidas protecciones tomando en cuenta su corriente nominal y basándose en que para efectos didácticos este trabajará en vacío. El peso aproximado de cada motor es de 24 libras por este motivo se consideró reforzar la base de la maleta didáctica con una plancha metálica.

También fue necesario colocar un acrílico protector cubriendo el eje del motor, esto como medida de seguridad para evitar accidentes.

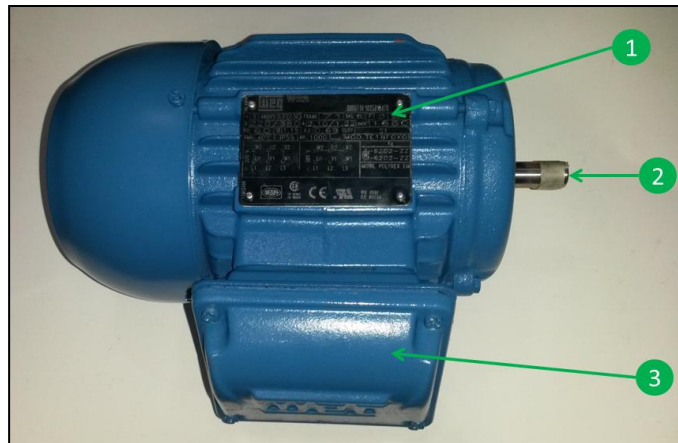


Figura 22. Partes del Motor.

Se detallan las partes básicas del motor trifásico, que será empleado en las prácticas. (WEG, Weg Motores Electricos, 2014)

1: Placa del motor.

2: Eje del motor.

3: Caja de conexiones.

Debido a las dimensiones y el peso del motor se montó sobre la base de cada maleta una plancha de metal galvanizado para dar mayor rigidez y soporte. La sujeción de esta máquina se la realizo con pernos avellanados y anillos de presión con el fin de dar un buen ajuste en la base.

3.1.8 SOFTWARE TWIDOSUITE.

TwidoSuite es el primer software que está organizado según el ciclo de desarrollo del proyecto. La navegación por el software es tan sencilla que se convierte en innata.

TwidoSuite es un entorno de desarrollo gráfico, lleno de funciones para crear, configurar y mantener aplicaciones de automatización para los autómatas programables Twido de Schneider Electric. TwidoSuite permite crear programas con distintos tipos de lenguaje, después de transferir la aplicación para que se ejecute en un autómata.

El TwidoSuite es un software gratuito que se puede descargar desde la página de Schneider Electric.



Figura 23. Plataforma TwidoSuite.
Programa basado en Windows de 32 bits para un ordenador personal. (Schneider Electric, 2011)

Las principales funciones del software TwidoSuite son:

- Interface de usuario intuitiva y orientada a proyectos.
- Diseño de software sin menús. Las tareas y funciones del paso.
- Selección de un proyecto siempre se encuentra visible.
- Soporte de programación y configuración.

Por medio de esta plataforma de programación diseñada para PLCs Twido de Schneider Electric, se realizan las diversas prácticas, en las que se elaboran las tablas de petición para el envío de tramas, utilizando los códigos de funciones estudiados según la acción que se requiera realizar sobre uno o varios esclavos Modbus.

3.1.8.1 OPERACIONES BÁSICAS PARA CREAR UN PROYECTO.

Al abrir el software TwidoSuite, aparecen tres opciones principales:



Figura 24. Modos de funcionamiento TwidoSuite.
Se muestra un menú con tres opciones de funcionamiento del software.
(Schneider Electric, 2011)

Modo “Programación”: Modo estándar para la creación de una aplicación.

Modo “Vigilancia”: Este modo permite conectarse a un autómata en modo vigilancia, donde podrá comprobar su funcionamiento sin necesidad de sincronizar su aplicación con la que hay cargada en la memoria del autómata.

Actualización de autómatas: Es un programa que indica todos los pasos necesarios para actualizar el Firmware.

Para crear un proyecto, seleccionar el “Modo Programación” y aparecerán las opciones principal de TwidoSuite. Luego aparecerá por defecto la ventana de proyecto, donde se podrá realizar la gestión de crear, abrir, guardar y cerrar un proyecto.

Pulsar en “Crear un proyecto nuevo”, acto seguido rellenar los campos de información general del proyecto, como puede ser: el nombre del proyecto, la ruta donde lo deseamos guardar, el autor, la versión, la compañía etc.

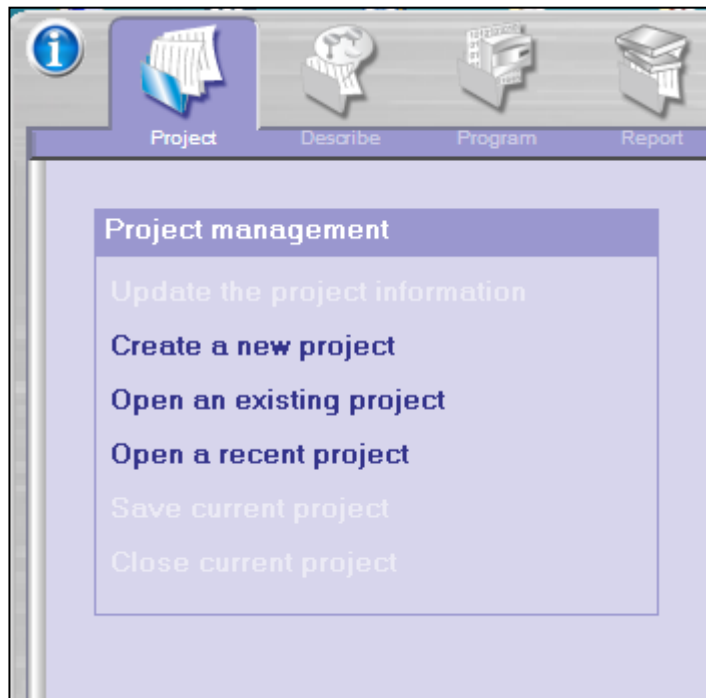


Figura 25. Administrador de proyectos.
Se muestra la ventana con las opciones para administrar los proyectos. (Schneider Electric, 2011)

La navegación por el software TwidoSuite es muy sencilla, intuitiva y gráfica, ya que sigue los pasos de ciclo de desarrollo natural de una aplicación de automatización.

El primer paso que se realiza cuando se inicia un proyecto de automatización es la configuración o descripción del hardware que se necesitará para dicho propósito, por lo cual en función de ciertas premisas como son: El número de entradas y salidas y el tipo, la necesidad de memoria y velocidad en la CPU, necesidad de interfaz de comunicación, etc.

En la pantalla llamada “Describe” se puede agregar el hardware a utilizar para el proyecto. Aquí se encontrará una ventana llamada “Catálogo” donde se escogerá cada elemento y se colocará en la pantalla para el armado del hardware como se puede ver en la siguiente imagen.

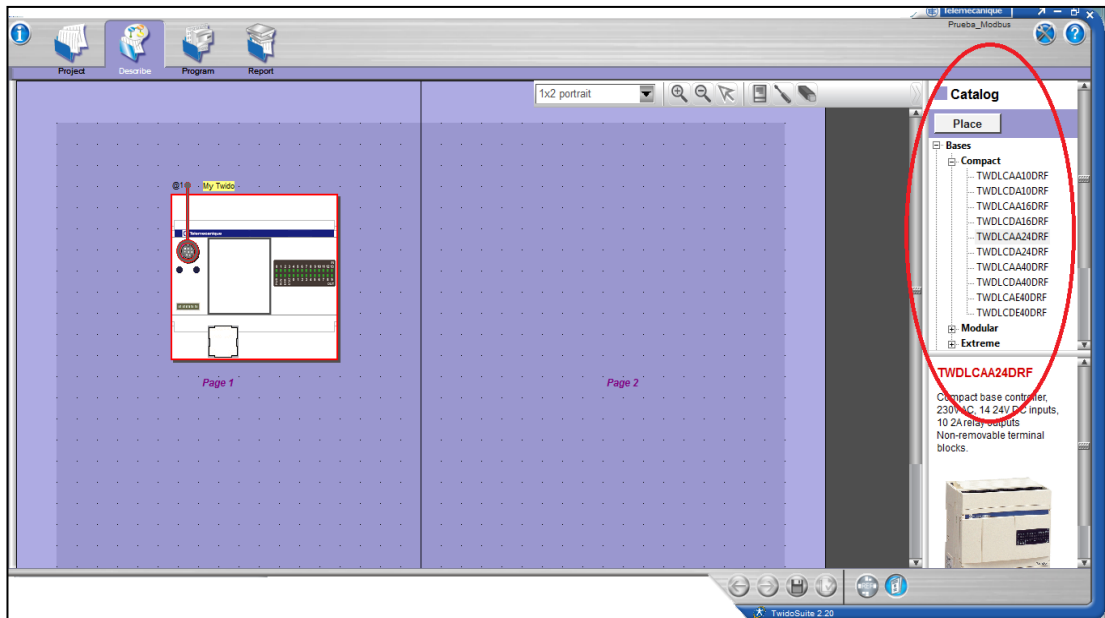


Figura 26. Selección del hardware.

Del lado izquierdo de la pantalla se encuentra el catálogo de equipos.

Para abrir el editor de programas ingresamos a la pestaña “Programar” en la barra de pasos de la aplicación, luego se irá dentro en la barra de pasos del programa a la pestaña “Programa”, aparece la ventana para la edición de programa. Es en esta última pantalla donde realizaremos la lógica de programación.

Existen tres barras de herramientas de programación de Ladder disponibles que permiten editar programas e introducir instrucciones gráficamente de forma rápida y sencilla:

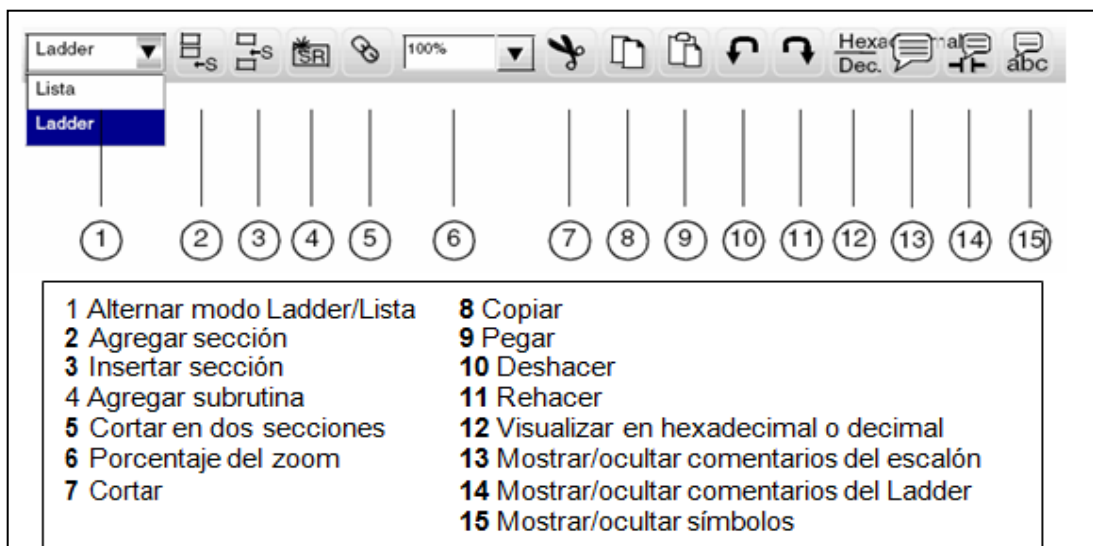


Figura 27. Barra de Herramientas Ladder.

Se describen cada icono de la barra. (Schneider-Electric, Manual Twido, 2008)

Para comenzar a programar se tiene que introducir la primera sección del programa para ello en la barra de herramientas pulsaremos el icono de “agregar una sección”.

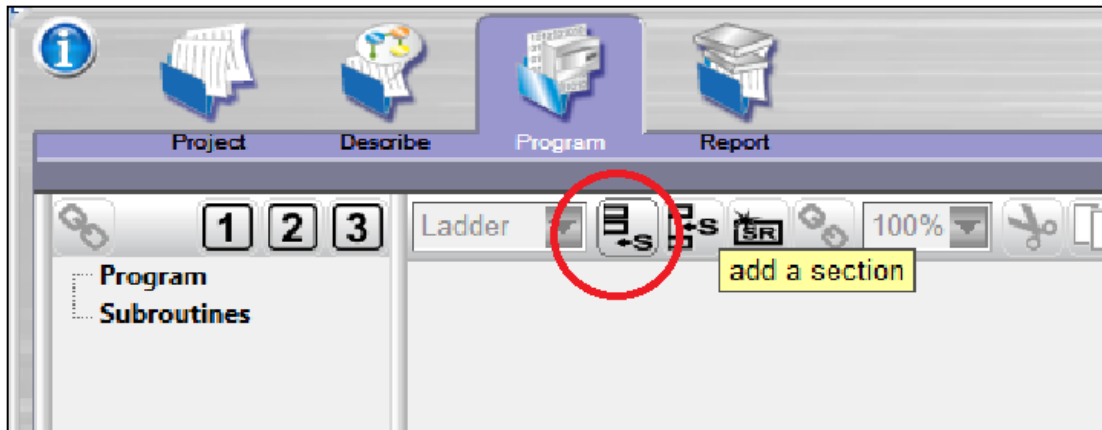


Figura 28. Icono agregar sección.

Se inserta la primera sección de programa haciendo clic en el ícono indicado con el círculo rojo. (Schneider-Electric, Manual Twido, 2008)

La primera sección se inserta en el Editor de Ladder Logic, de forma que muestra el primer escalón vacío, como figura a continuación. Tenga en cuenta que un programa debe contener al menos una sección. (Schneider-Electric, Manual Twido, 2008).

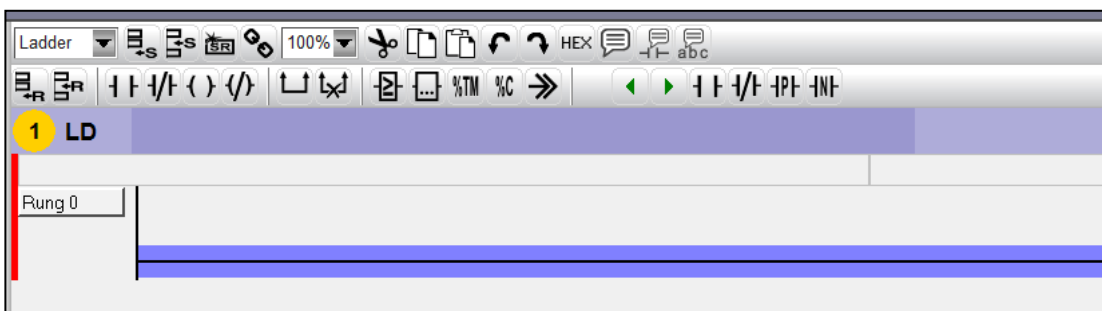


Figura 29. Introducción de la primera sección.

La conexión seleccionada se marca en color azul. (Schneider-Electric, Manual Twido, 2008)

Hacer doble clic en el encabezado de la sección para introducir el título de la sección y los comentarios.

Hacer doble clic en la cabecera de escalón para introducir título de escalón, comentarios y declaración de tipos de escalón.

Seleccione la conexión horizontal vacía que se ejecuta entre las líneas de

alimentación derecha e izquierda del escalón.

Luego de haber insertado la primera sección, para la inserción de las instrucciones básicas en la pantalla de programación se dispone de una barra de herramientas de la paleta de Ladder.

Las instrucciones más utilizadas en un programa ladder son:

- Contacto normalmente abierto
- Contacto normalmente cerrado
- Bobina
- Bloque de comparación
- Temporizadores
- Contadores

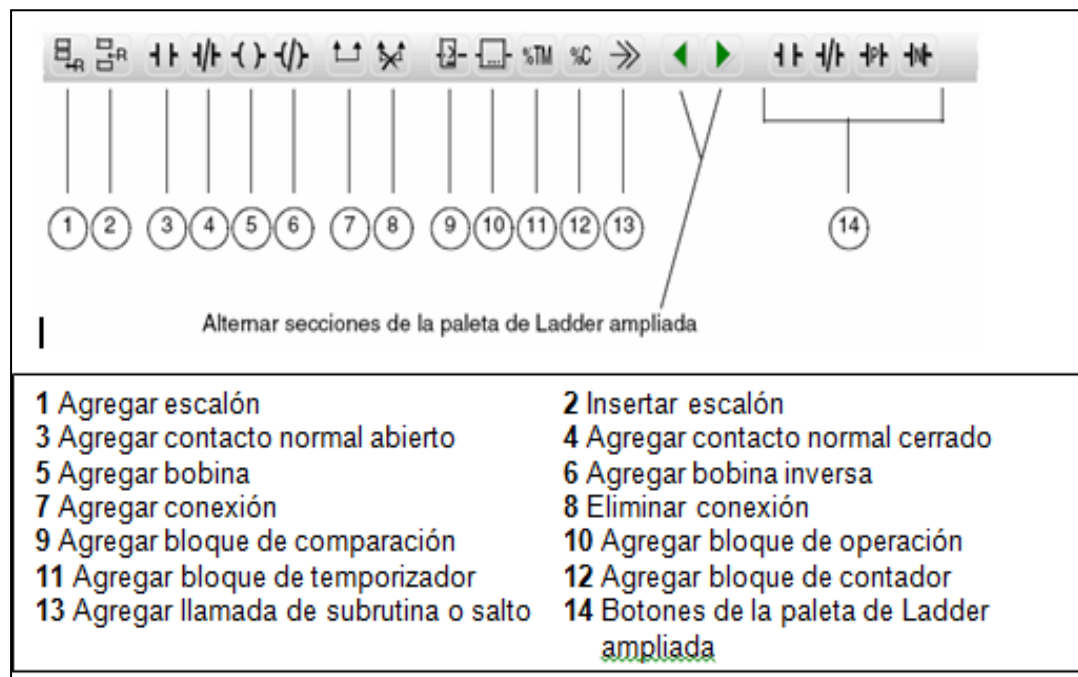


Figura 30. Paleta ladder ampliada.

En esta se encontrará las herramientas necesarias para la creación de un nuevo programa de una manera sencilla en lenguaje Ladder o escalera. (Schneider-Electric, Manual Twido, 2008).

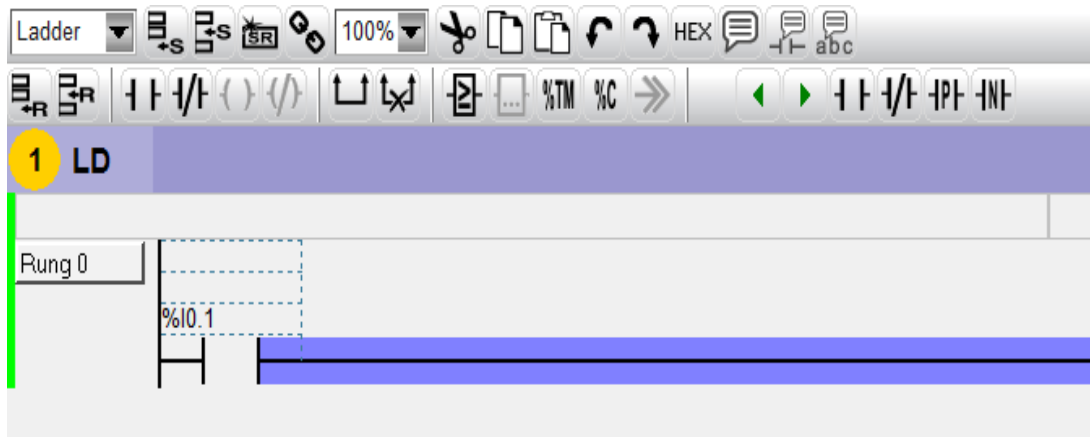


Figura 31. Inserción de elementos de control.

Se agrega un contacto seco normalmente abierto en la primera línea de programación.

3.1.8.2 CONFIGURACIÓN DE LAS MACROS

El primer paso que se debe realizar es configurar la macro que se va a utilizar, para lo cual se debe acceder las pestañas, programar, configurar, configurar los datos y dentro de esta, objetos avanzados, macros de comunicación.

A continuación podremos activar hasta 32 macros aunque solo podrán ejecutarse una con clic de scan. Cada una de ellas puede estar asociada a un esclavo distinto.



Figura 32. Pasos para activación de macros.

Ventana “Categoría de Objetos” en TwidoSuite para activación de macros. (Schneider-Electric, Manual Twido, 2008)

Luego de ingresar a la ventana “Macros de comunicaciones” seleccionamos las macros que vamos a utilizar.

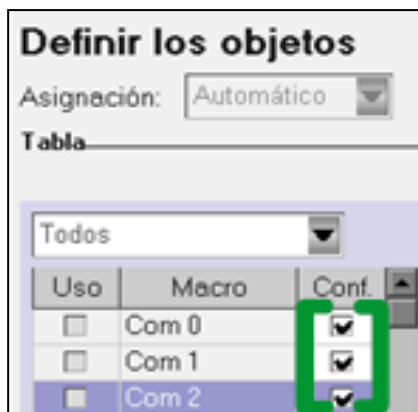


Figura 33. Selección de macro.
Se marca en las casillas para activarlas.

Una vez seleccionada la macro, elegimos el puerto de comunicación que va a utilizar.



Figura 34. Selección de puerto.
En la sección “General” se elige el puerto que se va a asignar a la macro.

El siguiente paso es escoger la dirección del esclavo a asignar a la macro.

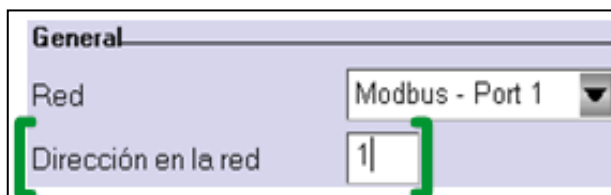


Figura 35. Selección de dirección de esclavo.
Se asigna el puerto a una dirección de esclavo en la red.

Por ultimo habilitamos las funciones que vayan a ser utilizadas.

Funciones			
<input type="checkbox"/> Función PRE		<input type="checkbox"/> Función POST	
Nombre de función	Dirección de inicio	Número de palabras	Símbolos
C_RD1B	0	8	<input type="checkbox"/>
C_RD1W	0	8	<input checked="" type="checkbox"/>
C_WR1B	0	8	<input type="checkbox"/>
C_WR1W	10	8	<input checked="" type="checkbox"/>
C_RDNW	0	N+7	<input type="checkbox"/>
C_WRNW	0	N+9	<input type="checkbox"/>

Figura 36. Lista de funciones Macros.

Se coloca un visto para activar las funciones que se va a utilizar.

Cada función necesita una cantidad de palabras para su uso, en caso de activar varias funciones, aparecerá un mensaje de error si su “Dirección de inicio” hace que se solapen con las direcciones de otras funciones. (Schneider-Electric, Manual Twido, 2008)

3.1.9 SOFTWARE ZELIOSOFT 2.

El software ZelioSoft2 facilita la configuración de los relés inteligentes Zelio Logic mediante una programación rápida y segura gracias a las pruebas del programa, además consta de las siguientes características:

- Programación real con FBD o lenguaje de contacto (LADDER).
- Detección de errores de programación con su función “prueba de coherencia”.
- Ventanas de supervisión: permite ver los estados de las E/S del relé inteligente en su entorno de aplicación.
- Descarga y carga de programas.
- Compilación automática de programas.

Al abrir el software se despliega un menú con diferentes opciones a elegir, entre las cuales se tiene “Crear nuevo programa”, “abrir un programa existente”, “abrir un programa utilizado recientemente”, “Telecargar un programa desde un módulo” etc.

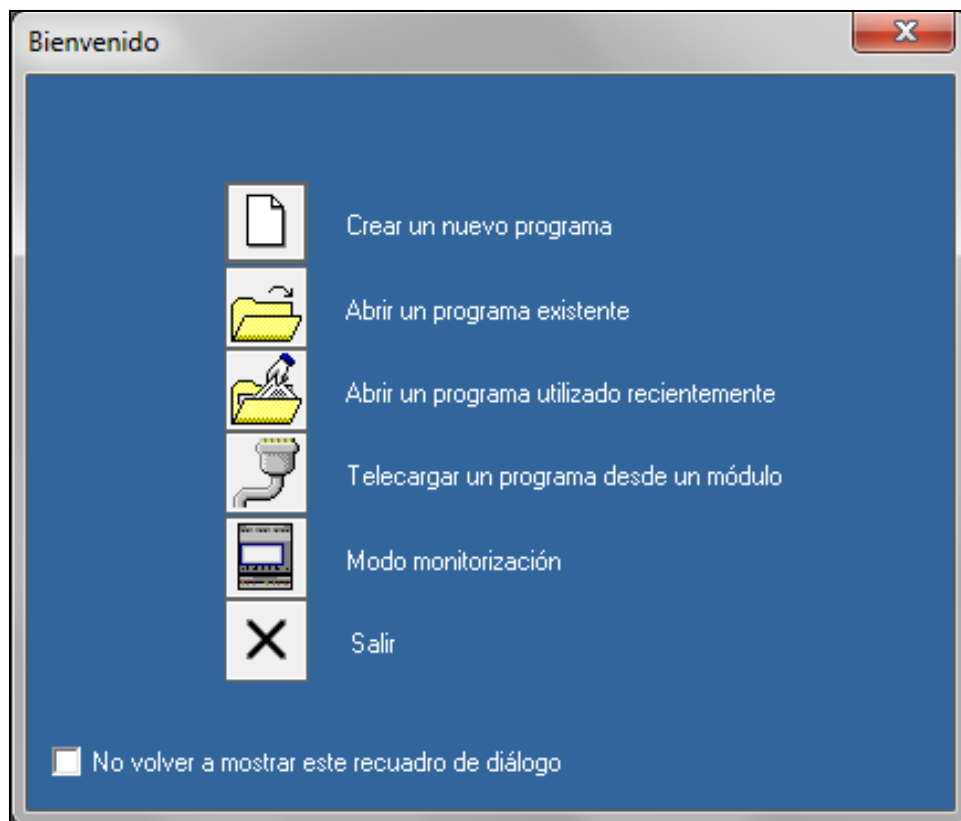


Figura 37. Menú de opciones ZelioSoft2.

En esta ventana se elige que se desea realizar según las opciones que permite el software.

Para la creación de un nuevo programa se elige a la opción “Crear nuevo programa”, luego aparecerá una nueva ventana la cual permitirá elegir el modelo de módulo Zelio Logic que se usará. También se cuenta con las opciones “Abrir un programa existente”, “Abrir un programa recientemente”, “Modo monitorización” o en su defecto la opción “Salir”.

El programa ZelioSoft2 brinda la posibilidad de acceder a “Modo monitorización” en el cual se podrá visualizar en línea el control que se ejerce y realizar un sondeo en tiempo real de cada variable del proceso, facilitando la comprobación de programas y verificando que los resultado sean satisfactorios.

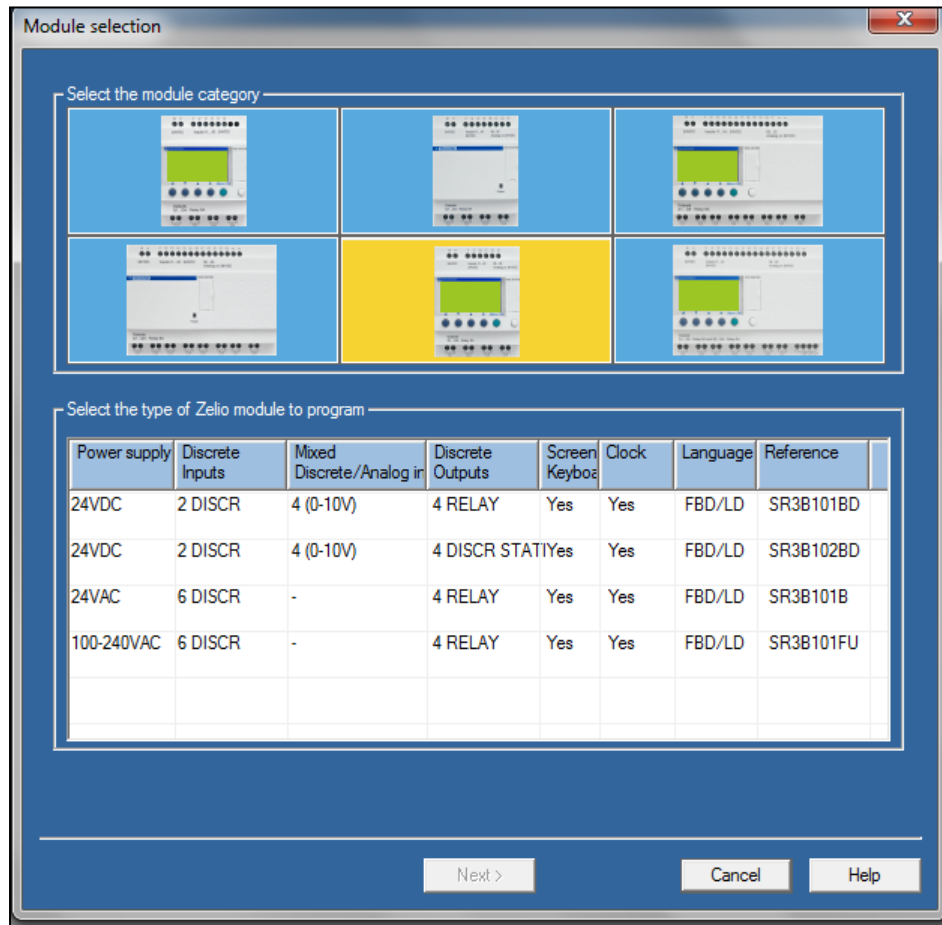


Figura 38. Selección de modelo del módulo Zelio.

Se despliega una lista con todos los modelos de módulos Zelios disponibles.

Se recomienda obtener la última versión del software ZelioSoft2 la cual se encontrará disponible en la página de Schneider Electric, esto para evitar problemas al momento de realizar la configuración del hardware y elegir los modelos de módulos o versiones de firmware.

Una vez elegido el modelo se procede a dar clic en el botón siguiente donde aparecerá una nueva ventana la cual permitirá agregar alguna extensión adicional al módulo, por ejemplo una extensión para comunicación Modbus o si se desea agregar más entradas y salidas al módulo.

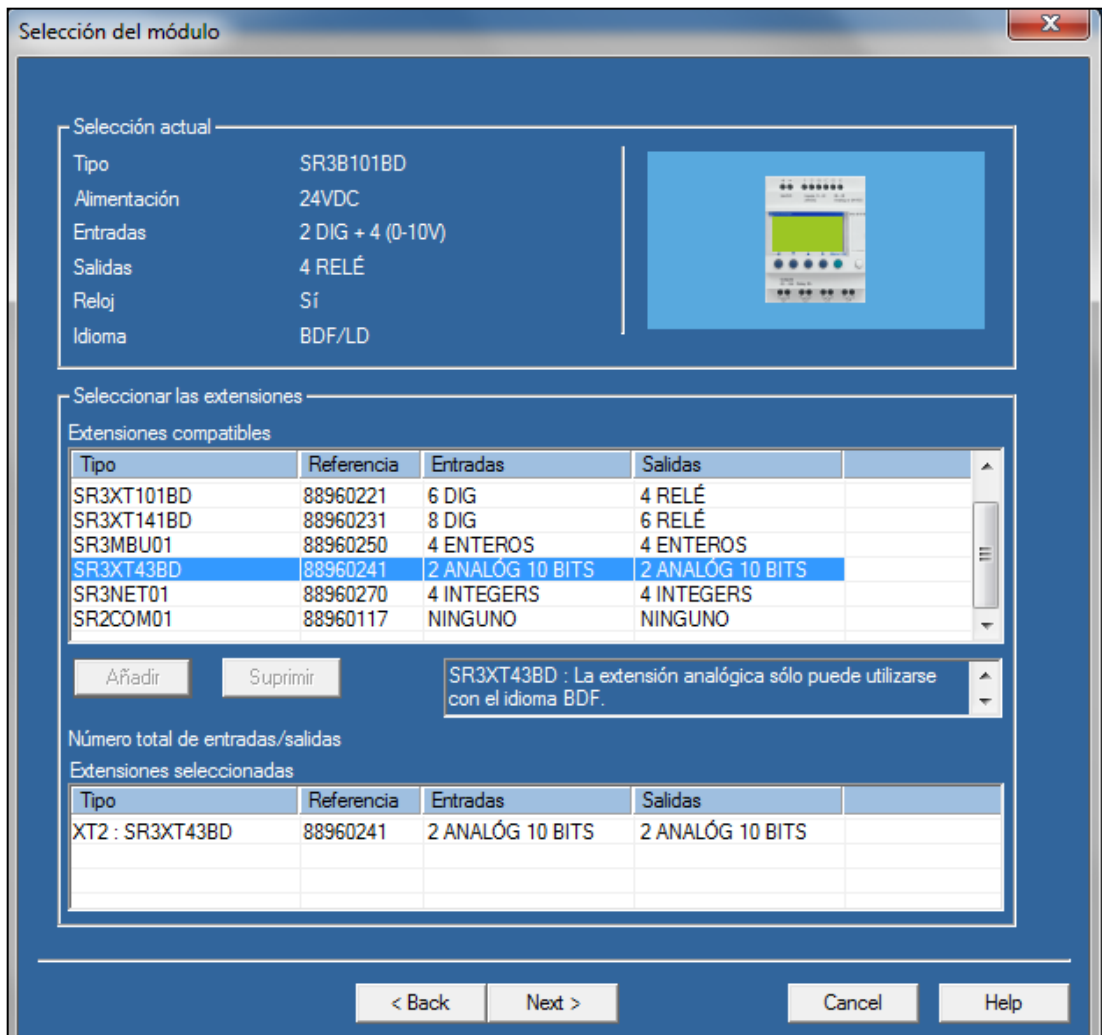


Figura 39. Ventana de selección de módulo adicional.

Se despliega una lista de los módulos posibles.

Para agregar los módulos requeridos se da clic en el botón añadir y se irá llegando la lista inferior con las extensiones cargadas.

Después de agregar las extensiones, si fuese necesario, se da clic en el botón siguiente para que se abra una nueva ventana en la cual se desplegara un resumen de todo lo configurado anteriormente a modo de verificación. Adicionalmente, por medio de esta ventana podremos elegir el tipo de programación a realizar, las cuales pueden ser Ladder o BDF (Modo gráfico).

Por último se finaliza la configuración dando clic en el botón siguiente, para que se despliegue la pantalla de programación elegida y ya poder a realizar el proyecto.

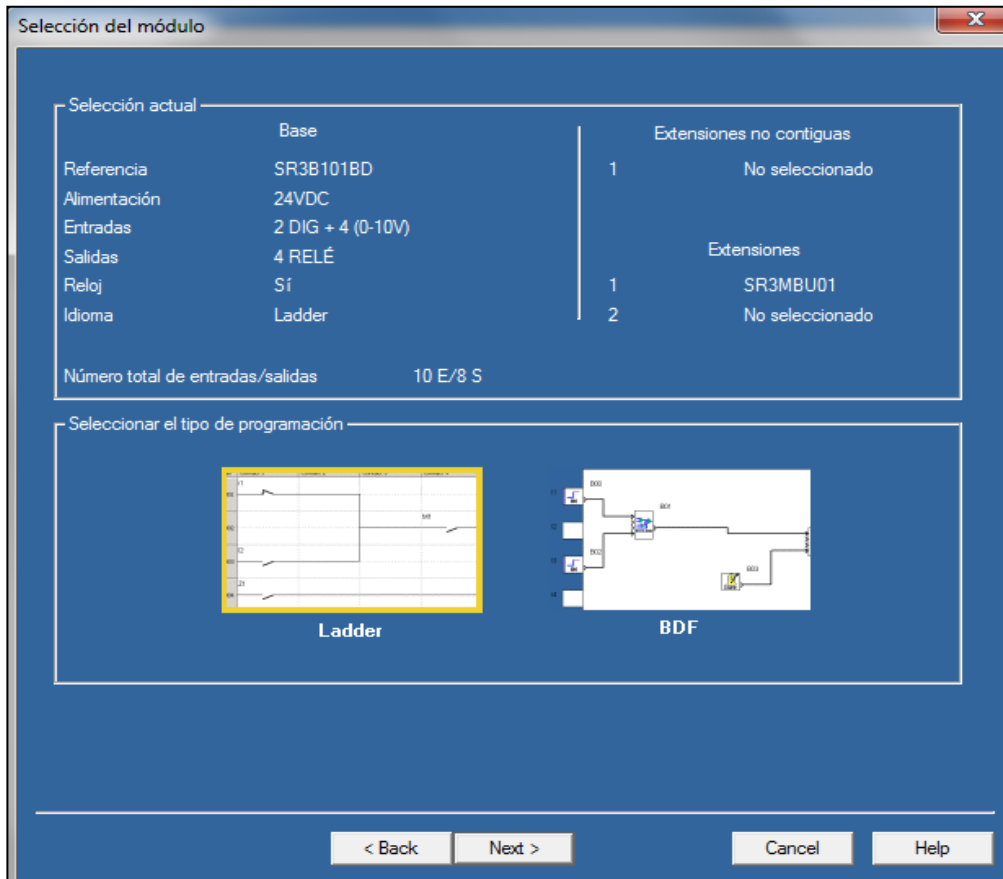


Figura 40. Selección de tipo de programación.
Por lo general la programación se realiza en lenguaje Ladder.

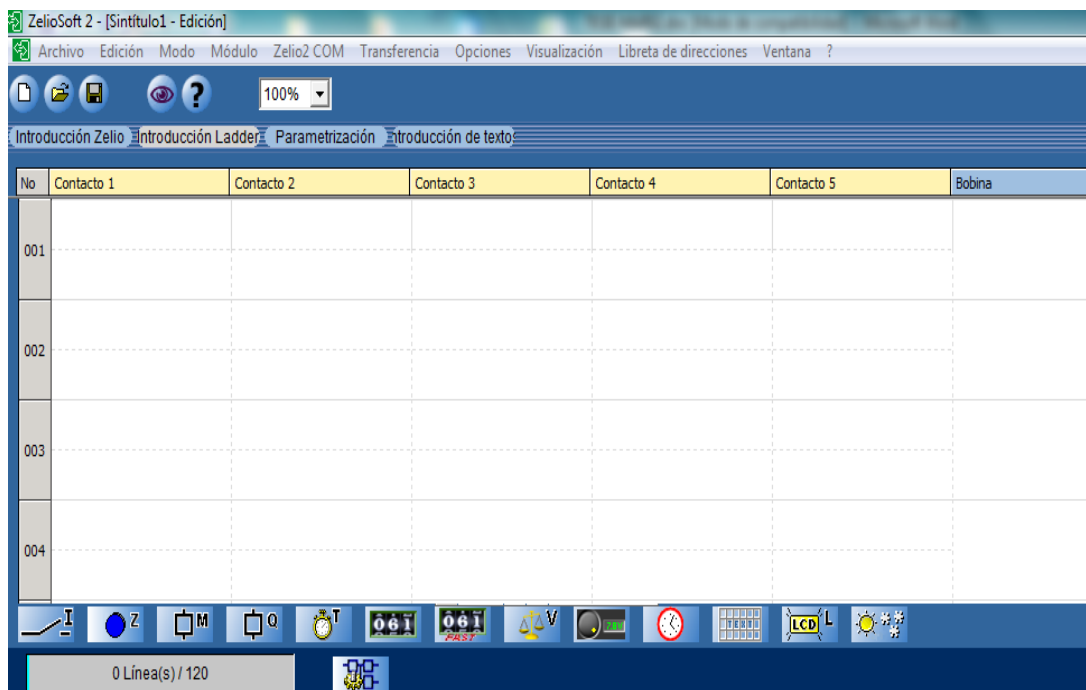


Figura 41. Pantalla Programación ZelioSoft2.
Por medio de contactos, marcas, temporizadores y bobinas se realiza la programación en este modo.

3.2 P&ID DIAGRAMA DE BLOQUES

Para poder identificar cada uno de los equipos e instrumentos del proyecto de una manera sencilla y poder tener una idea más clara de las condiciones de diseño de las maletas nos guiaremos por medio de una herramienta que se conoce comúnmente por las siglas P&ID.

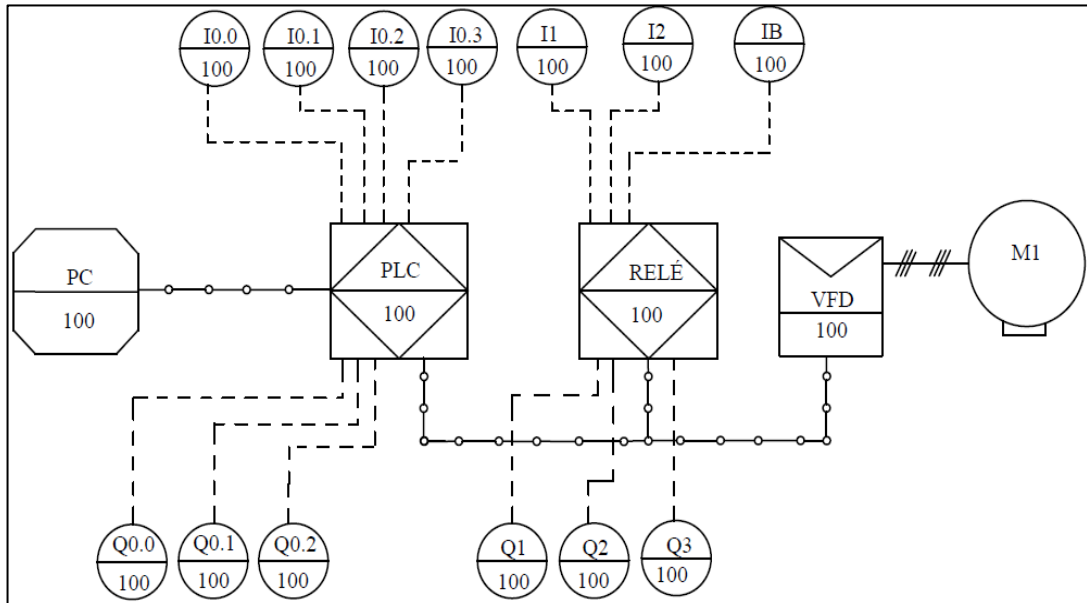


Figura 42. Diagrama P&ID.

Se detalla cada uno de los elementos y dispositivos que conforman las maletas.

3.3 DISEÑO ESTRUCTURAL DE LA MALETA DIDÁCTICA.

Primeramente se realizó el diseño eléctrico y de comunicación de los equipos dentro de la maleta, para lo cual se establecieron tipos y calibres de cables, selección de protecciones, fuentes de poder, conectores, terminales, canaletas, y demás materiales a utilizar.

Se elaboró un pequeño tablero de control con selectores pulsadores y luces piloto con el fin de manipular entradas y salidas de los equipos.

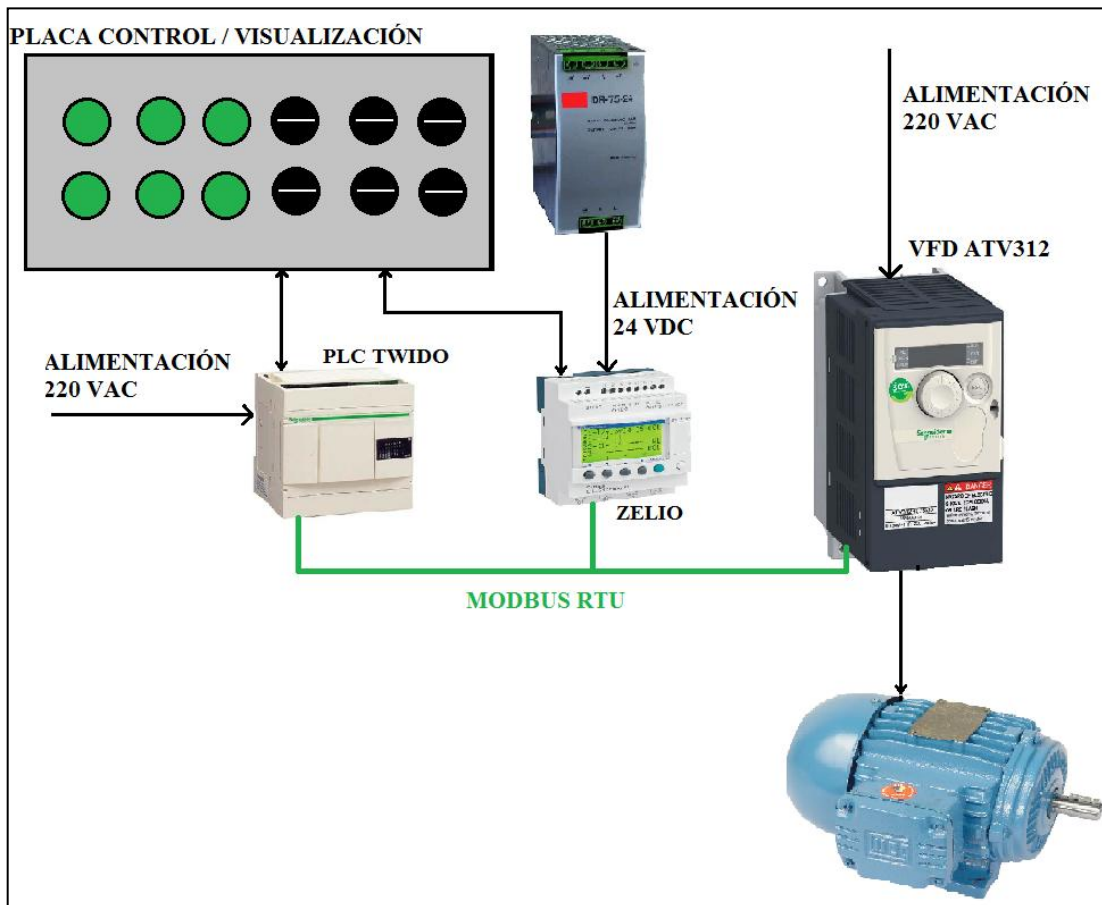


Figura 43. Diagrama esquemático del sistema. Muestra las conexiones entre los diferentes equipos instalados en la maleta didáctica.

Para proceder a implementar el proyecto se realizó el diseño estructural de la maleta didáctica para lo cual se tomó las medidas de cada uno de los equipos y elementos a ser utilizados, luego se empleó la herramienta AutoCAD para desarrollar el dibujo a escala real.

Se elaboró una placa en la cual se mostraran selectores, pulsadores y luces piloto, con la finalidad de simular entradas y salidas digitales hacia los diferentes equipos controladores.

Se cuenta con una alimentación de 220 VAC la cual es suministrada al variador de frecuencia y al PLC Twido. Para el relé Zelio se incorporó una fuente de alimentación de 24 VDC.

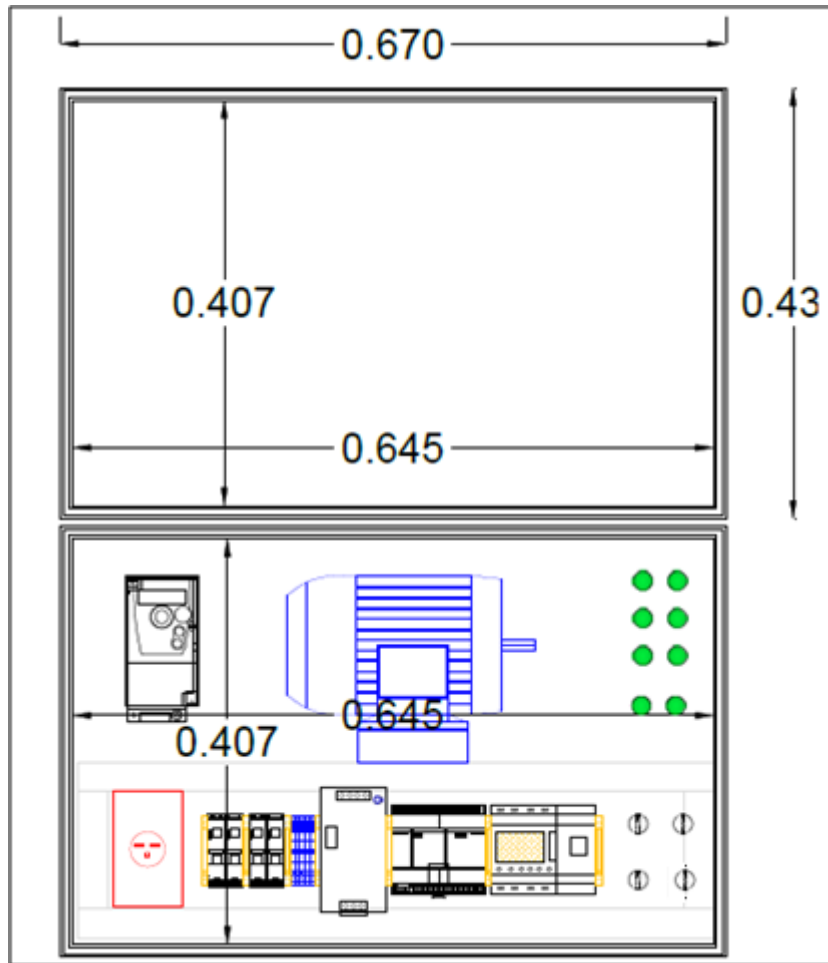


Figura 44. Diseño con Medidas.
 Cabe indicar que estas medidas estas dadas en milímetros.

El diseño de la maleta didáctica está enfocado a brindar facilidades al alumno tanto para realización de prácticas como para el transporte de la misma ya que cuenta con un diseño compacto referenciado en maletas de viaje, por lo cual se añadió un mango de viaje y ruedas para su rodamiento.

La elaboración de la maleta didáctica se realizó bajo el diseño en AutoCAD de cada uno de los elemento.

Las ubicaciones se establecieron de acuerdo a criterio de ventilaciones de equipos, y una mejor distribución del peso los equipos en la maleta. La posición del tablero de control (luces, selectores y pulsadores) se determinó de acuerdo al mejor manejo del estudiante.

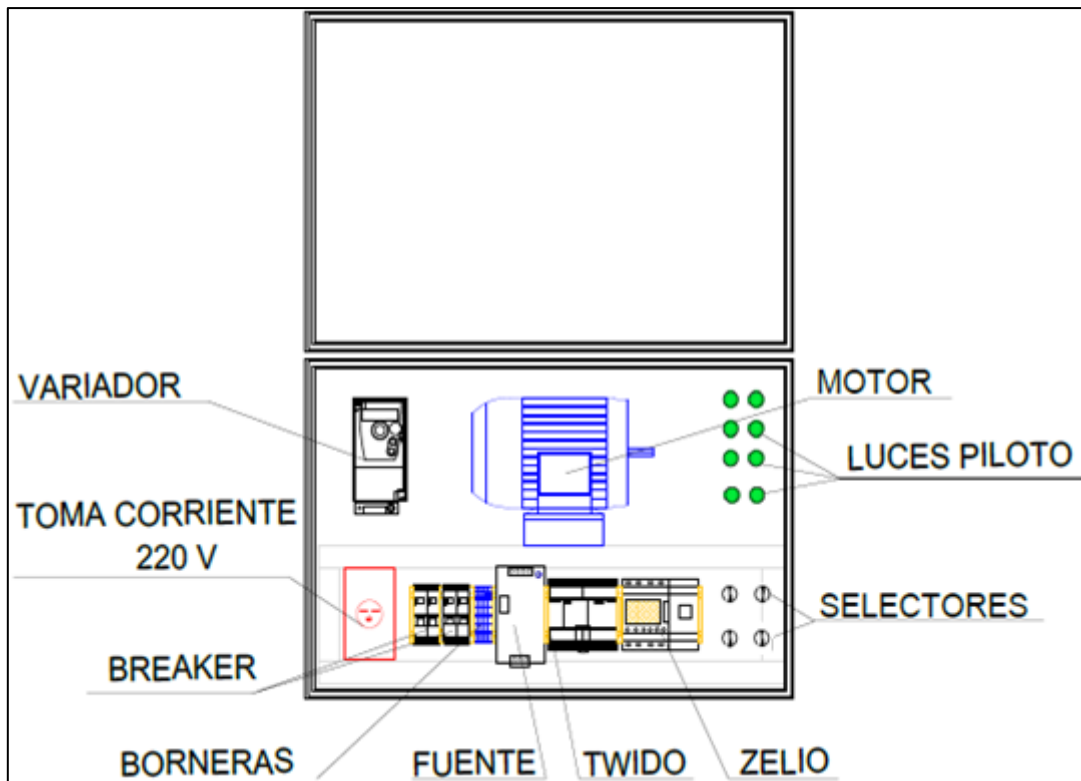


Figura 45. Diseño identificación de equipos.

Se señalan los equipos y elementos distribuidos dentro de la maleta didáctica.

Se colocó un tomacorriente “chino” para 220VAC, este alimenta toda la maleta, internamente se dividen los circuito con sus respectivas protecciones.

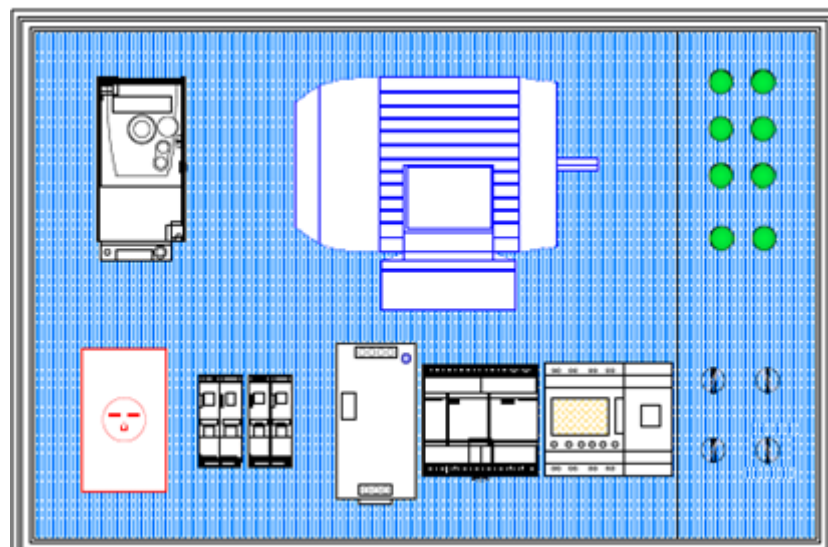


Figura 46. Estructura terminada.

Se realiza el acabado colocando una cubierta de PVC.

3.4 ENSAMBLAJE DE LA ESTRUCTURA

La estructura de la maleta didáctica está compuesta de madera con perfiles y ángulos de aluminio para brindar mayor robustez y estabilidad. Adicional se montó una plancha de metal galvanizado en el fondo para obtener una base más firme al momento de anclar los equipos.



Figura 47. Montaje de Plancha Metálica.

Se muestra la plancha metálica fija en el fondo con el objetivo de darle mayor robustez al anclaje de los equipos.

Sobre la plancha metálica se fija un riel din para el agarre de los equipos. Para instalación de motor se utilizó pernos avellanados debido al peso de la máquina. Se colocaron canaletas rasuradas de PVC con medidas de 25x25mm para el cableado de control y de comunicación.

Para alimentación del equipo Zelio Logic y señales de control, se colocó una fuente de alimentación de 24VDC 3 A.



Figura 48. Montaje y Cableado de equipos.
Se cablean los equipos tanto en su parte eléctrica como de comunicación.

Se cableo con cable calibre 16 esto se determinó de acuerdo a la carga total que maneja cada circuito. También se utilizó canaleta plástica de 25x25 mm para el manejo del cableado eléctrico y de control.



Figura 49. Mango y base de Maleta.
Con el fin de dar facilidad de transporte se instaló una base con ruedas de goma y un mango para su agarre.



Figura 50. Maleta didáctica.
Revisión de la maleta en la UPS

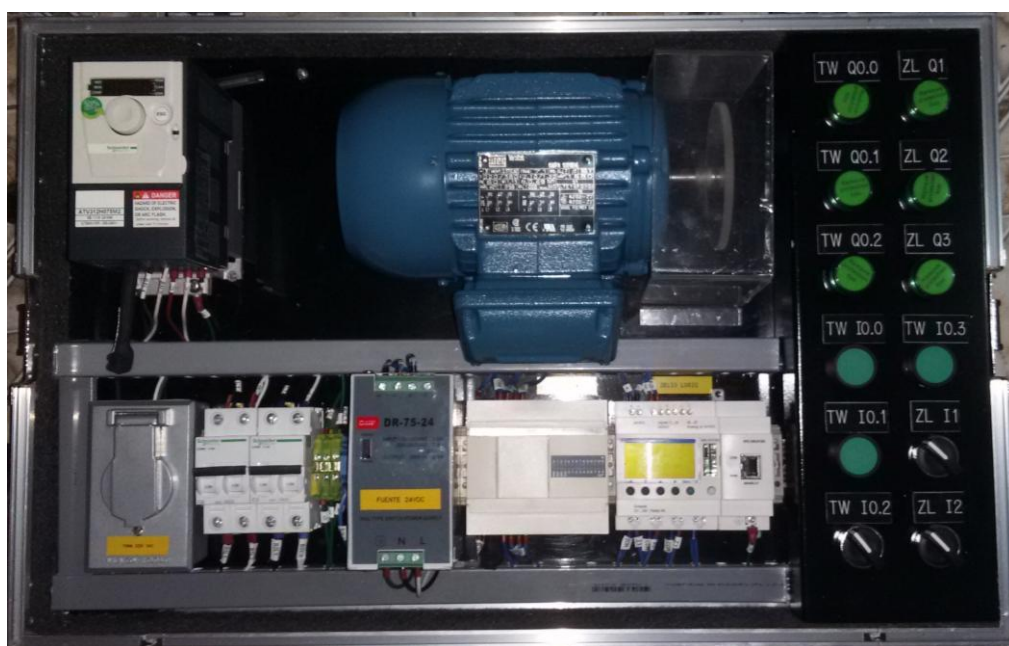


Figura 51. Maleta con Acabado.
Se instaló una lámina de PVC para darle un mejor acabado.

Luego de terminar con el armado de la estructura y montaje de los equipos se realizaron las pruebas funcionales de cada equipo, en las cuales se revisó el voltaje de alimentación principal, protecciones, contactos, selectores y luces piloto, las cuales dieron un resultado satisfactorio.



Figura 52. Estructura de maleta didáctica terminada.

Se muestra el acabado de la estructura final de la maleta didáctica, con perfiles y ángulos de aluminio que dan mayor robustez y estética, adicionalmente cuenta con un mango para su fácil manejo.

Unas de las ventajas de esta sencilla y robusta maleta didáctica es su portabilidad y su fácil operación, ya que en su interior cuenta con los dispositivos cableados u dispuesto de tal forma que simplifica el aprendizaje de las practicas.

4 CAPÍTULO 4: PRÁCTICAS.

4.1 ESCRITURA DE UN BIT CONFIGURANDO UN TWIDO MAESTRO Y UN TWIDO ESCLAVO.

En esta práctica se realiza la escritura de un bit desde un “Maestro Modbus” para el cual se eligió un PLC Twido de base compacta el mismo que cuenta con las siguientes características:

- Alimentación 110VAC - 220VAC.
- Fuente de alimentación 24VDC integrada.
- 14 Entradas digitales y 10 Salidas digitales.
- 1 Puerto serie Modbus integrado (Utilizado para programación).

Adicionalmente se le incorporó el puerto serial 2 de comunicación Modbus, por medio del cual se transmitirán las tramas y por ende las peticiones hacia el esclavo.

Para el equipo esclavo se eligió otro PCL Twido con las mismas características del “Maestro Modbus” al cual también se le incorporó el puerto serial 2 de comunicación para la recepción de las tramas y peticiones que provengan desde el “Maestro Modbus”.

4.1.1 PROGRAMA PARA EL EQUIPO MAESTRO.

Par efectos prácticos se ha obviado el proceso de creación de un nuevo proyecto en el software TwidoSuite, el cual ya se lo detalló anteriormente en el capítulo 3.

Una vez situado en la pantalla de configuración de hardware se agrega al segundo puerto serie de comunicación, el cual se lo consigue en la ventana llamada “Catalogo” a lado derecho de la pantalla.

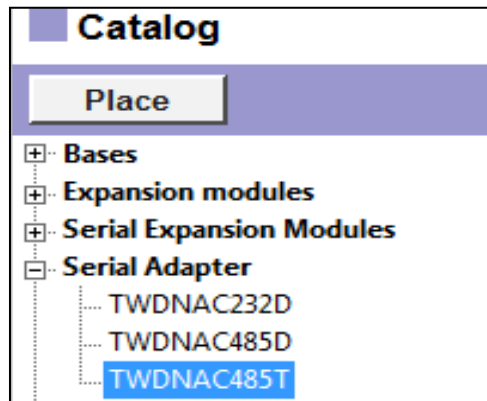


Figura 53. Ventana Catálogo Maestro. Despliega los elementos a agregar para configuración de hardware.

Luego se procede a configurar el puerto serie (antes agregado) para indicar el protocolo y dirección de red que se va a utilizar, se hace clic en el botón “OK” para aceptar los cambios.

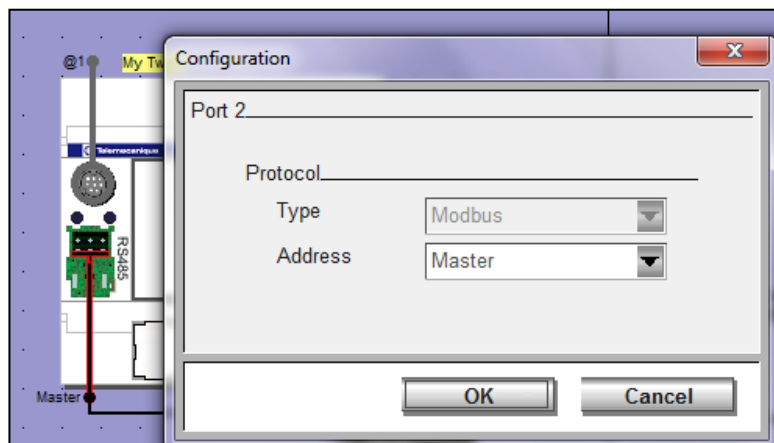


Figura 54. Configuración red Maestro. Se configura el PLC en la red Modbus como equipo Maestro.

Después se procede a agregar en la configuración el equipo esclavo, para esto se lo selecciona desde la pantalla “Catálogo”, “Elementos de red” y seleccionamos el equipo Modbus Twido.

Cabe indicar que en esta lista no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

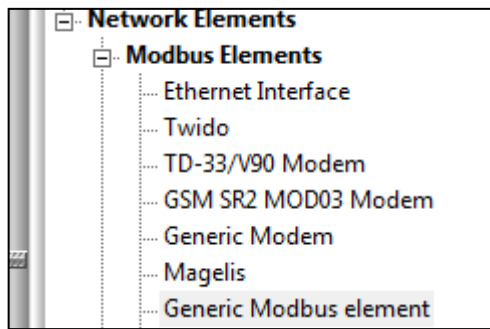


Figura 55. Elemento Modbus.
Desde la ventana “Catálogo” se puede agregar un elemento a la red.

Una vez agregado el elemento esclavo Modbus a igual que en el equipo configurado como Maestro, se procede a configurar el tipo de la red y la dirección que se va a asignar al esclavo.

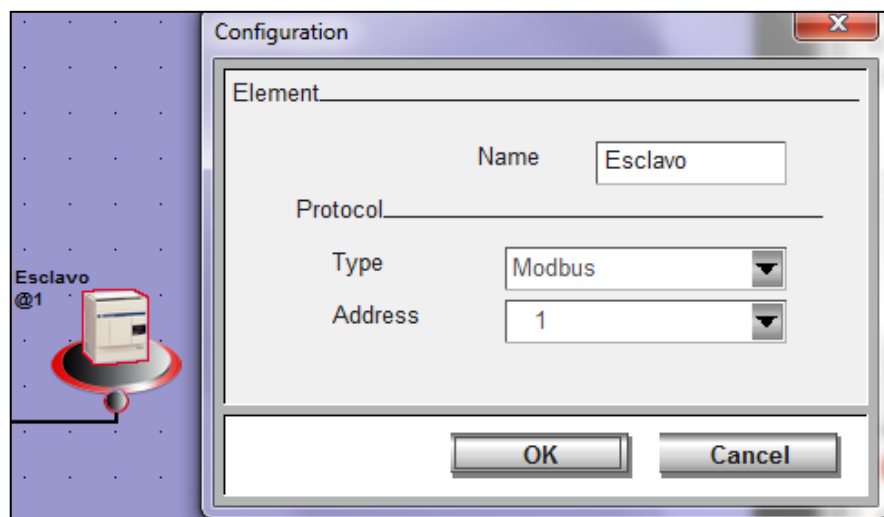


Figura 56. Configuración de red en esclavo.
Se finaliza dando clic en el botón OK.

Para finalizar la configuración del hardware se procede a enlazar los equipos en la red y verificar los parámetros de transmisión.

- Baudrate: según el fabricante es recomendable trabajar a una velocidad de transmisión de 19200 baudios.
- DataBits: Como se utiliza el formato RTU, se selecciona 8(RTU).
- Parity: La paridad será par.
- Bit Stop: Se utiliza 1 bit de parada.

- Response timeout: tiempo de espera de respuesta en 10s.

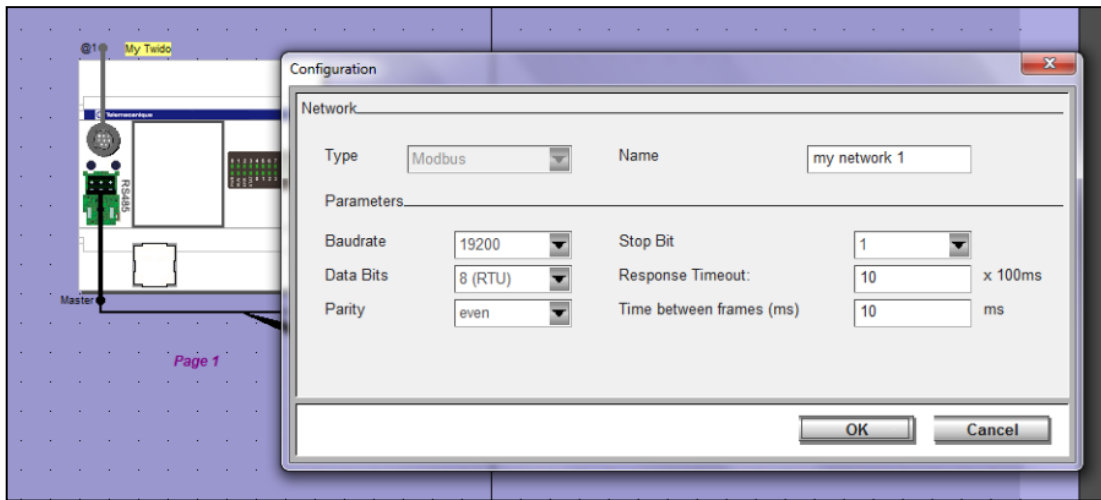


Figura 57. Parámetros de transmisión.

Ventana con campos para configurar cada parámetro de transmisión.

Una vez terminada la configuración del hardware se procede a la realización del programa en el cual se desarrollan las tablas de petición con lo indica la siguiente figura:

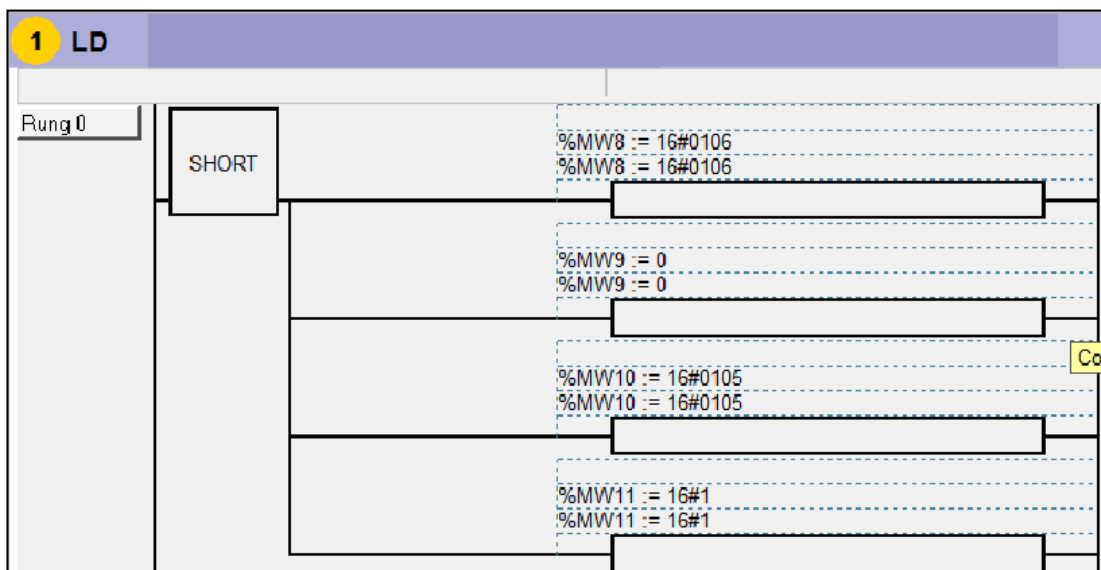


Figura 58. Rung 0 escritura de bit.

Desarrollo de la tabla para la escritura del bit en TwidoSuite.

Para comenzar a elaborar la tabla de petición en la realización de la escritura de un bit se debe estudiar el funcionamiento de la función 05.

La función 05 permite modificar el estado de una DO del esclavo, es decir mediante este comando podemos modificar algún bit de alguna de las variables internas del esclavo.

Se inicia la tabla con la palabra de memoria %MW8 la cual corresponde a la subtabla de control utilizada para controlar la tabla de envío y recepción dentro de la tabla de escritura, está conformada por dos bytes el más significativo contiene el código de emisión y recepción (01) y el menos significativo corresponde la longitud de la tabla de envío (06).

En la palabra de memoria %MW9 contiene dos bytes y corresponde al offset o corrimiento de envío y recepción.

Para la siguiente palabra %MW10 contiene la dirección del esclavo y el código de función de la tabla de escritura respectivamente.

La palabra de memoria %M11 lleva la dirección de bit que va a ser escrito, este caso la %M1 del esclavo.

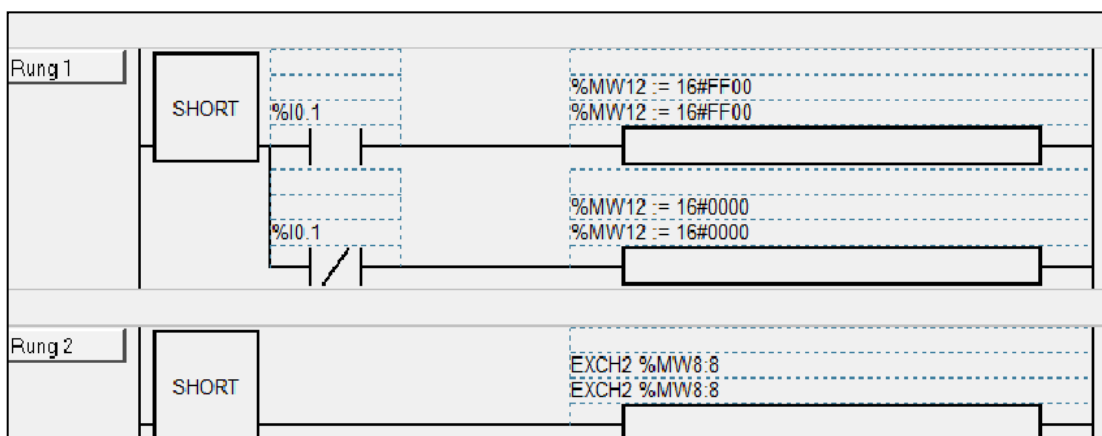


Figura 59. Rung 1 y 2 para la escritura de un bit.
Se realiza la escritura del bit enviando la palabra %MW12.

Luego en la %MW12 se coloca el valor del bit que se va a escribir, en este caso esta escritura podrá tener dos valores posibles los cuales vas a depender del estado de la entrada digital %I0.1. Para escribir un “1” se coloca el numero hexadecimal “16#FF00” y en el otro caso para escribir un “0” se sitúa el numero hexadecimal “16#0000”.

Para finalizar utilizaremos la función “EXCHx” por medio de esta podremos realizar el envío de las tabla desarrollada, siendo “x” el número del puerto Modbus y luego

se asigna la dirección de la primera memoria utilizada para la tabla y el número de memorias que ocupa la tabla.

4.1.2 PROGRAMA PARA EL EQUIPO ESCLAVO.

Para el programa del esclavo se realizan las mismas configuraciones en el hardware como se realizó para el equipo maestro, con la diferencia que se intercambian las posiciones, en este caso ahora el maestro será el elemento Modbus agregado desde la ventana “Catálogo” como se muestra en la figura:

Configuración Hardware esclavo. Ahora el maestro sería el elemento Modbus.

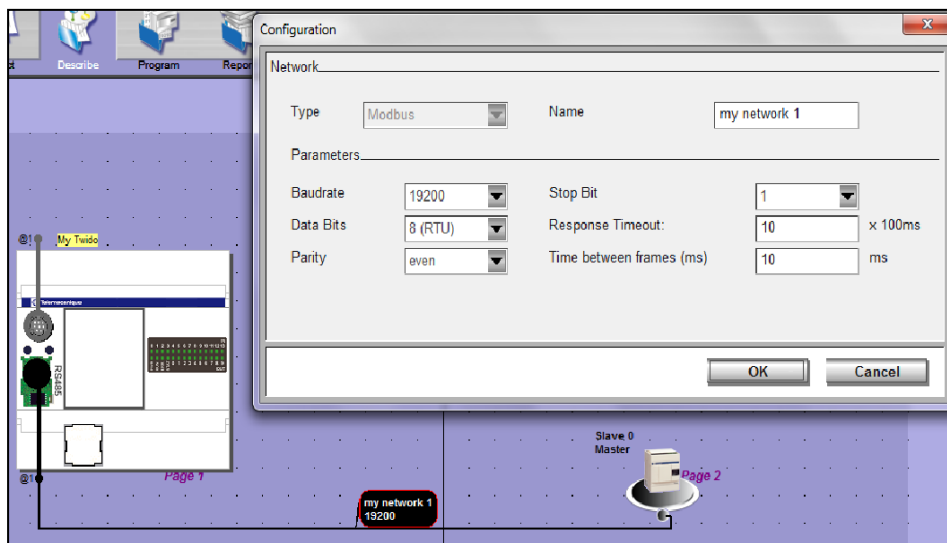


Figura 60. Configuración de red del esclavo para escritura. Se detalla la configuración del esclavo y sus parámetros.

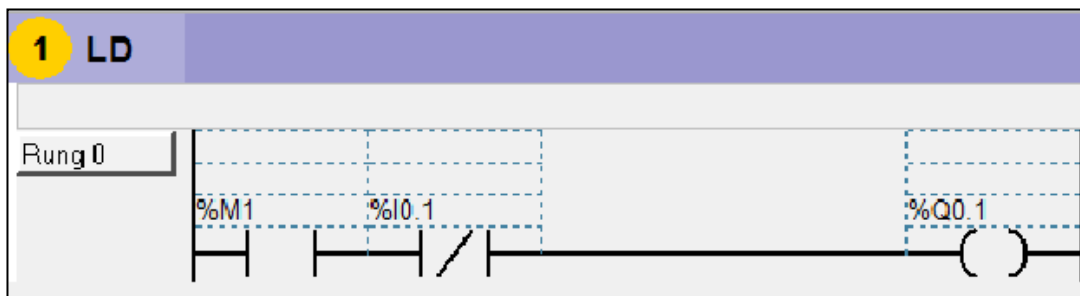


Figura 61. Escribir Bit esclavo. Primera línea de programación en el esclavo.

En la línea de programación del equipo esclavo se asigna a %M1 (bit a escribir desde el Maestro) para que active la salida %Q0.1 del mismo, la cual puede ser desactivada mediante la entrada digital %I0.1 del mismo esclavo.

4.2 LECTURA DE UN BIT CONFIGURANDO UN TWIDO MAESTRO Y UN TWIDO ESCLAVO.

En la presente práctica se realiza la lectura de un bit desde un equipo configurado como “Maestro Modbus” para el cual se eligió un PLC Twido con base compacta el cual cuenta con las siguientes características:

- Alimentación 110VAC - 220VAC.
- Fuente de alimentación 24VDC incorporada.
- 14 Entradas y 10 Salidas digitales.
- 1 Puerto serie Modbus integrado (Utilizado para programación).

Adicionalmente se le incorporó el puerto serial 2 de comunicación Modbus, el cual servirá para la transmisión de las tramas y por ende las peticiones hacia el esclavo.

Para el equipo esclavo se eligió otro PCL Twido con las mismas características del equipo “Maestro Modbus” al cual también se le incorporó el puerto serial 2 de comunicación, para la recepción de las tramas y peticiones del Maestro.

4.2.1 PROGRAMA PARA EL EQUIPO MAESTRO.

Par efectos prácticos se ha obviado el proceso de creación de un nuevo proyecto en el software TwidoSuite, el cual ya se lo detalló anteriormente en el capítulo 3.

Una vez situado en la pantalla de configuración de hardware se agrega al segundo puerto serie de comunicación, el cual se lo consigue en la ventana llamada “Catalogo” a lado derecho de la pantalla.

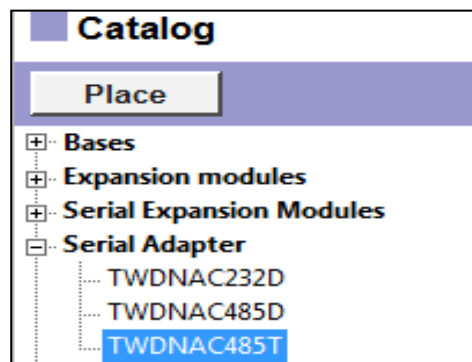


Figura 62. Ventana Catálogo Esclavo. Despliega elementos a agregar para configuración de hardware.

Luego se procede a configurar el puerto serie (antes agregado) para indicar el protocolo y dirección de red que se va a utilizar, se hace clic en el botón “OK” para aceptar los cambios.

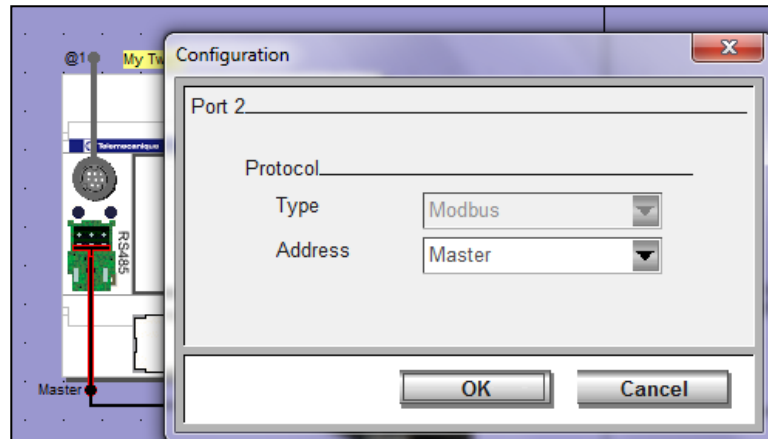


Figura 63. Configuración red Maestro.
Se configura el PLC en la red.

Después se procede a agregar en la configuración el equipo esclavo, para esto se lo selecciona desde la pantalla “Catálogo”, luego ingresamos a “Elementos de red” y seleccionamos el equipo Modbus Twido.

Cabe indicar que en esta lista no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

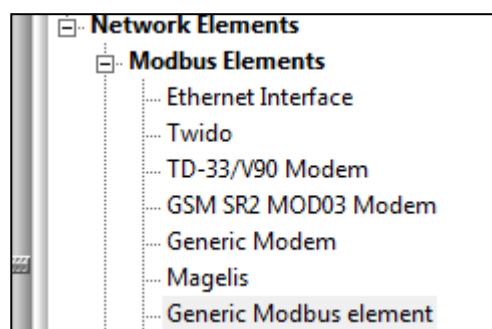


Figura 64. Elemento Modbus.
Desde la ventana “Catálogo” se puede agregar un elemento a la red.

Una vez agregado el elemento esclavo Modbus a igual que en el equipo configurado como Maestro, se procede a configurar el tipo de la red y la dirección que se va a asignar al esclavo.

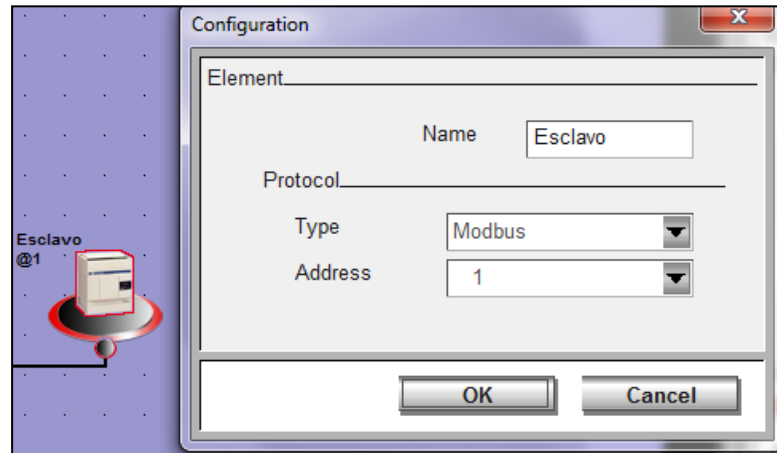


Figura 65. Configuración de red en esclavo.
Se finaliza dando clic en el botón OK.

Para finalizar la configuración del hardware se procede a enlazar los equipos en la red y verificar los parámetros de transmisión.

- Baudrate: según el fabricante es recomendable trabajar a una velocidad de transmisión de 19200 baudios.
- DataBits: Como se utiliza el formato RTU, se selecciona 8(RTU).
- Parity: La paridad será par.
- Bit Stop: Se utiliza 1 bit de parada.

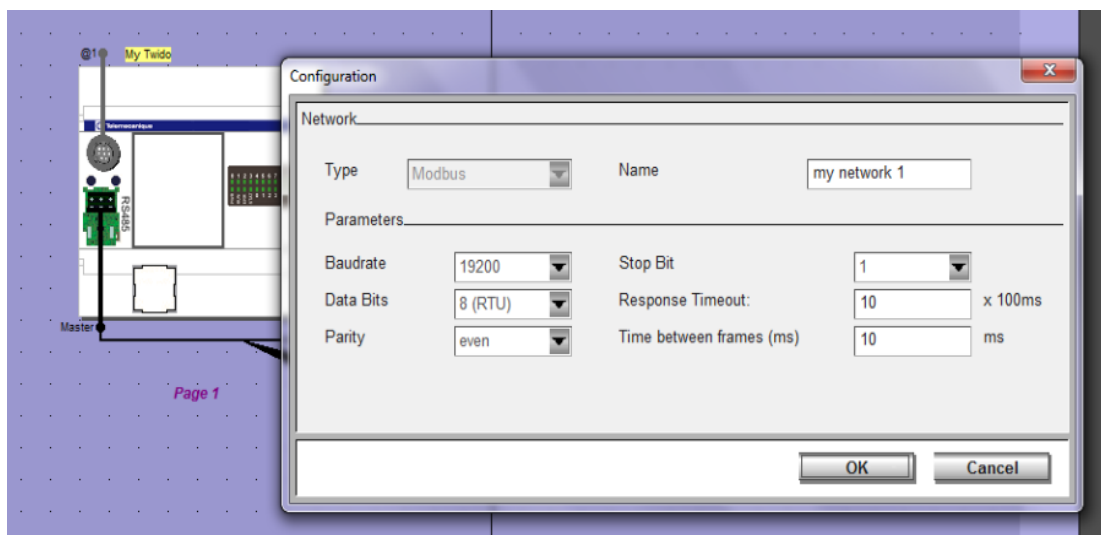


Figura 66. Parámetros de transmisión.
Ventana con campos para configurar cada parámetro de transmisión.

Una vez terminada la configuración del hardware se procede a la realización del programa en el cual se desarrollan las tablas de petición con lo indica la siguiente figura:

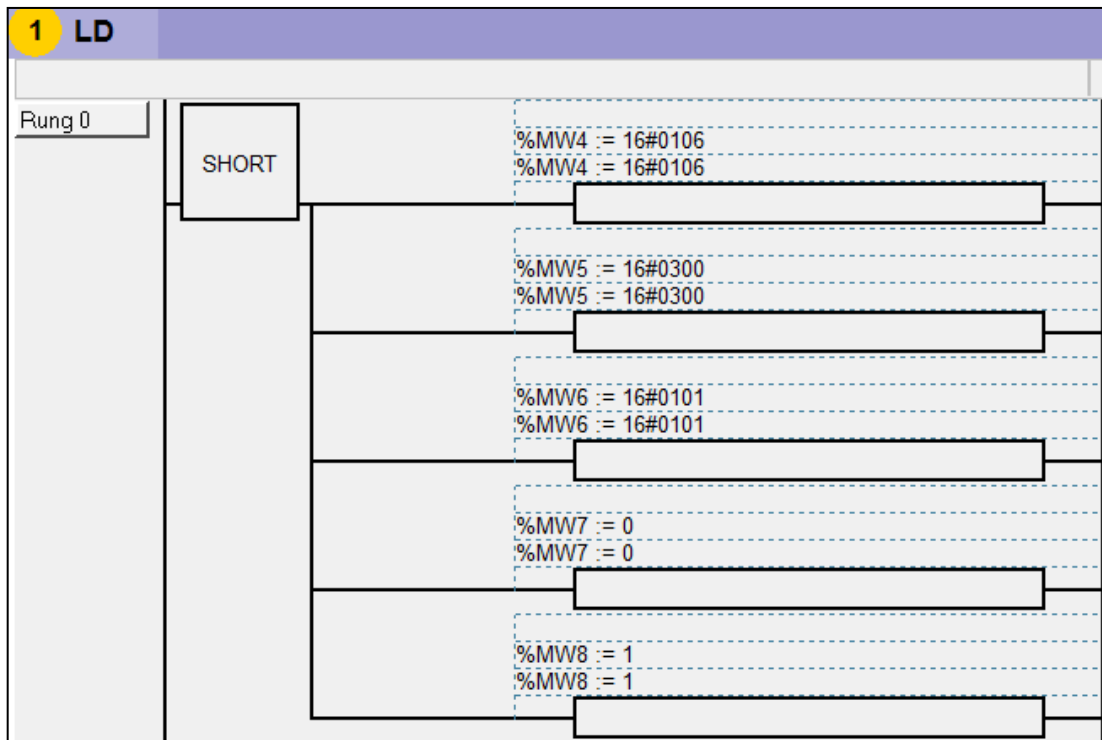


Figura 67. Rung 0 lectura de bit.

Se detalla el desarrollo de la tabla para la lectura del bit.

Para comenzar a elaborar la tabla de petición en la realización de la escritura de un bit se debe entender el funcionamiento de la función 05.

La función 01 permite leer el estado de un bit del esclavo, es decir mediante este comando podemos leer algún bit de alguna de las variables internas del esclavo.

Se inicia la tabla con la palabra de memoria %MW4 la cual corresponde a la sub-tabla de control utilizada para controlar la tabla de envío y recepción dentro de la tabla de escritura, está conformada por dos bytes el más significativo contiene el código de emisión y recepción (01) y el menos significativo corresponde la longitud de la tabla de envío (06).

En la palabra de memoria %MW5 contiene dos bytes y corresponde al offset o corrimiento de envío y recepción.

Para la siguiente palabra %MW6 contiene la dirección del esclavo y el código de función de la tabla de escritura respectivamente.

La palabra de memoria %MW7 lleva la dirección del primer bit que va a ser leído, este caso la %M0 del esclavo. Luego en la %MW8 colocamos el número de bits a leer, en este caso solo se va a leer un bit.

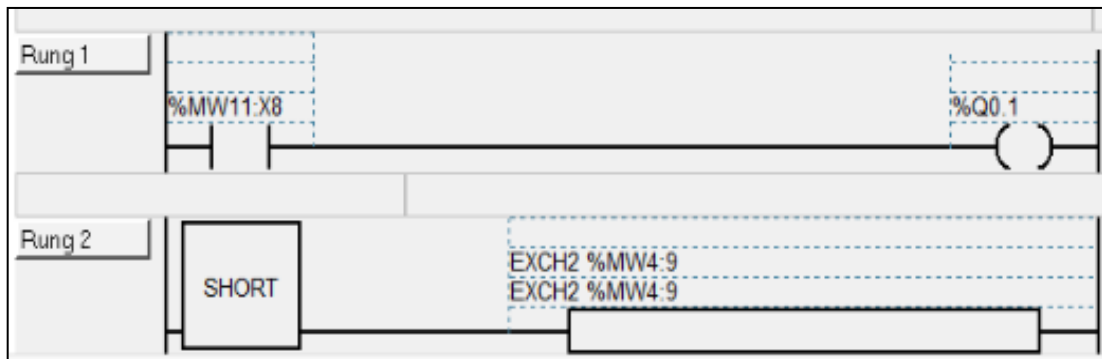


Figura 68. Rung 2 y 3, lectura de bit. Se muestra el direccionamiento del bit leído y la ejecución de la función EXCHx.

Siguiendo el orden de la tabla de la función 01, el valor leído del bit se aloja en la palabra %MW11. Para finalizar se ejecuta la función EXCHx, donde “x” es el puerto Modbus a utilizar, luego se coloca el rango de palabras de memoria a enviar en la trama.

4.2.2 PROGRAMA PARA EL EQUIPO ESCLAVO

Para el programa del esclavo se realizan las mismas configuraciones en el hardware como se realizó para el equipo maestro, con la diferencia que se intercambian las posiciones, en este caso ahora el maestro será el elemento Modbus agregado desde la ventana “Catálogo” como se muestra en la figura:

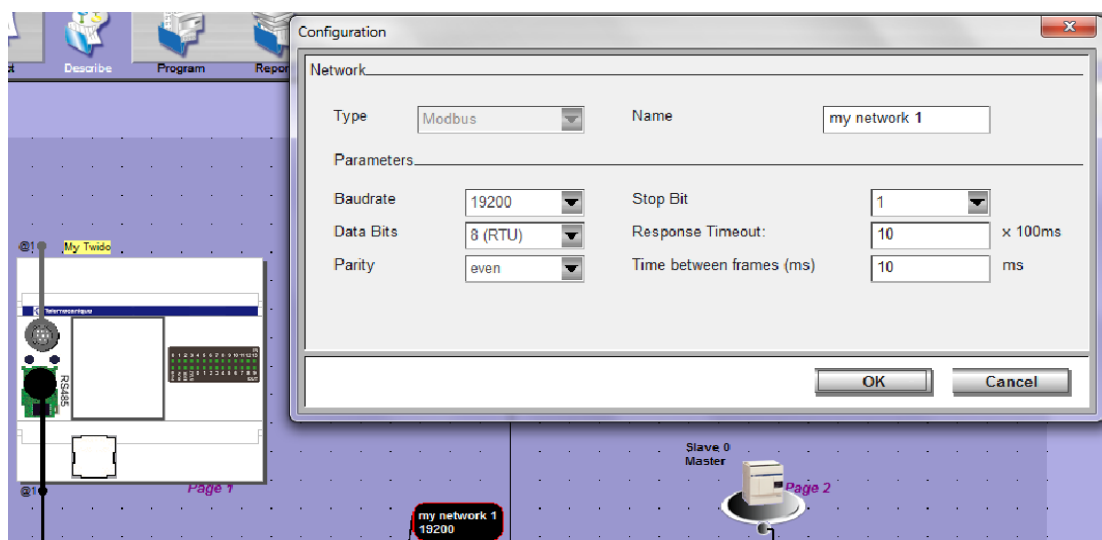


Figura 69. Configuración esclavo. Se detalla la configuración del esclavo y sus parámetros.

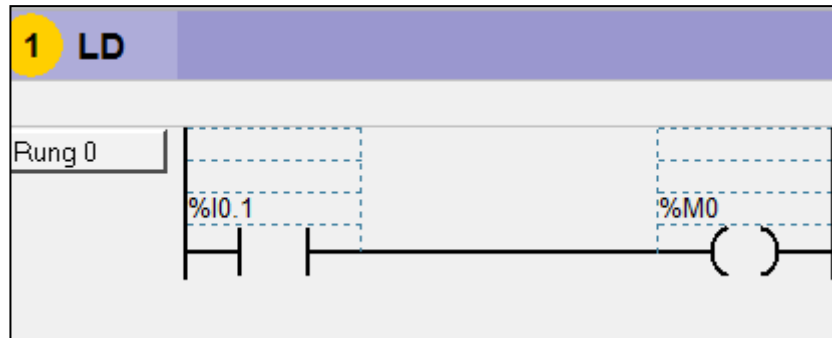


Figura 70. Leer Bit esclavo.
Primera línea de programación del equipo esclavo.

En la línea de programación del equipo esclavo se asigna a %IO.1 para que active al bit de memoria %M0, este a su vez será leído por el Maestro, el cual activará su salida digital %Q0.1 dependiendo del estado de %M0.

4.3 ESCRITURA DE UNA PALABRA DE MEMORIA CONFIGURANDO UN TWIDO MAESTRO Y UN ZELIO LOGIC ESCLAVO.

En esta práctica se utilizará el relé inteligente Zelio Logic de Schneider Electric, el cual cuenta con las siguientes características:

- Alimentación de 24 VDC.
- 2 Entradas digitales, 4 entradas análogas y 4 salidas de relé.

Adicionalmente se agregó una interfaz de comunicación Modbus RTU, para la transmisión y recepción de las tramas.

Es importante indicar que este relé inteligente solo soporta las siguientes funciones Modbus:

- Función 03 (Leer múltiples palabras o registros).
- Función 06 (Escribir una palabra o registro).
- Función 16 (Escribir múltiples palabras o registros).
- Función 43 (Leer identificación del dispositivo).

Intercambio de datos posibles para palabras de Entradas / Salidas entre la aplicación Zelio Logic y el Maestro.

- Palabras de entrada: J1XT1 a J4XT1
- Palabras de salidas: O1XT1 a O4XT1

Tabla 11. Palabras de memoria disponibles en Zelio Logic.

Código de Función	Registro Modbus	Entradas/Salidas
03 (Leer multiples palabras)		
06 (Escribir una palabra)	Word 16	J1XT1
	Word 17	J2XT1
16 (Escribir múltiples palabras)	Word 18	J3XT1
	Word 19	J4XT1
	Word 20	O1XT1
03 (Leer multiples palabras)	Word 21	O2XT1
	Word 22	O3XT1
	Word 23	O4XT1

Nota: esta son las palabras de memoria que dispone el relé inteligente para ser escritas o leídas.

A continuación se realizará la programación en el PLC Twido configurado como maestro Modbus, el cual escribirá en la palabra de memoria 17 (J2XT1) del Zelio configurado como esclavo.

4.3.1 PROGRAMA PARA EL EQUIPO MAESTRO.

Se procede a configurar el puerto serie (antes agregado) para indicar el protocolo y dirección de red que se va a utilizar, se hace clic en el botón “OK” para aceptar los cambios.

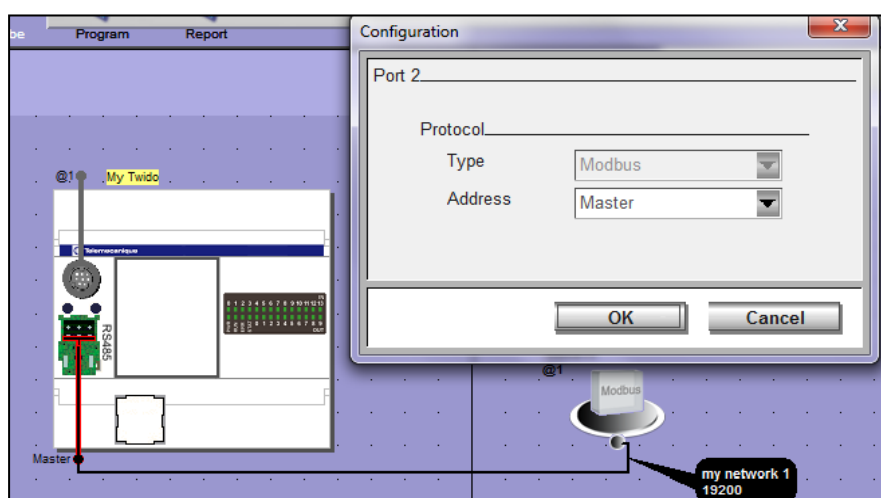


Figura 71. Configuración red Maestro. Se configura el PLC en la red.

Se procede a colocar al equipo Twido como maestro y el tipo de comunicación Modbus. Después se procede a agregar en la configuración el equipo esclavo, para esto se lo selecciona desde la pantalla “Catálogo”, luego ingresamos a “Elementos de red” y seleccionamos el equipo Modbus Twido.

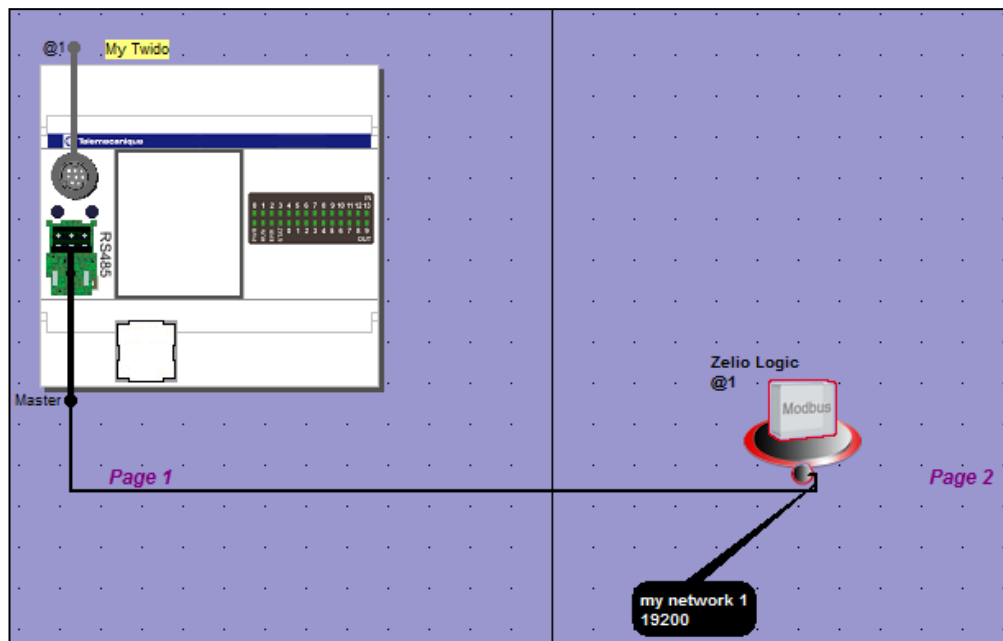


Figura 72. Configuración red Maestro con el Esclavo Zelio Logic. Se configura el PLC en la red.

Cabe indicar que en esta lista no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

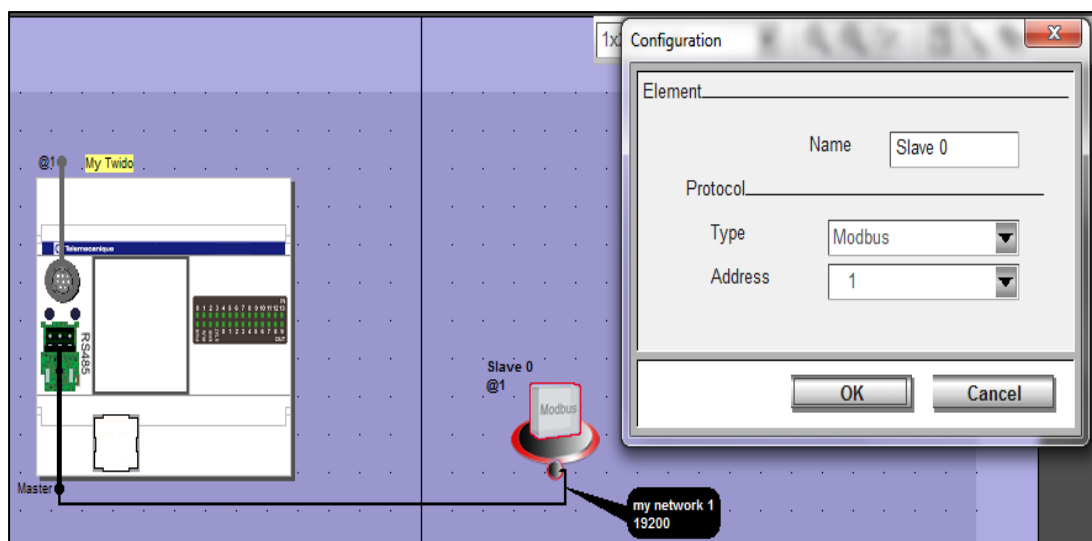


Figura 73. Configuración red esclavo. Se configura el PLC en la red.

Para finalizar la configuración del hardware se procede a enlazar los equipos en la red y verificar los parámetros de transmisión.

- Baudrate: según el fabricante es recomendable trabajar a una velocidad de transmisión de 19200 baudios.
- DataBits: Como se utiliza el formato RTU, se selecciona 8(RTU).
- Parity: La paridad será par.
- Bit Stop: Se utiliza 1 bit de parada.

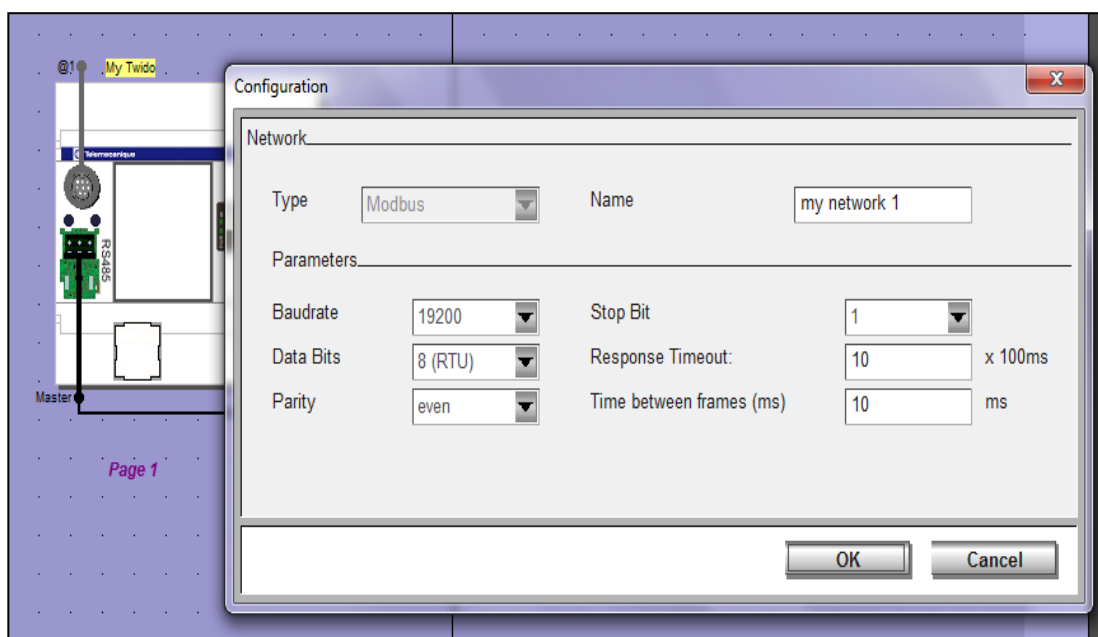


Figura 74. Parámetros de transmisión.

Se detalla la configuración del esclavo y sus parámetros de transmisión.

Una vez terminada la configuración del hardware, se procede a la realización del programa en el cual se desarrollan las tablas de petición utilizando el código de petición 16 (10 hexadecimales) como lo indica la siguiente figura:

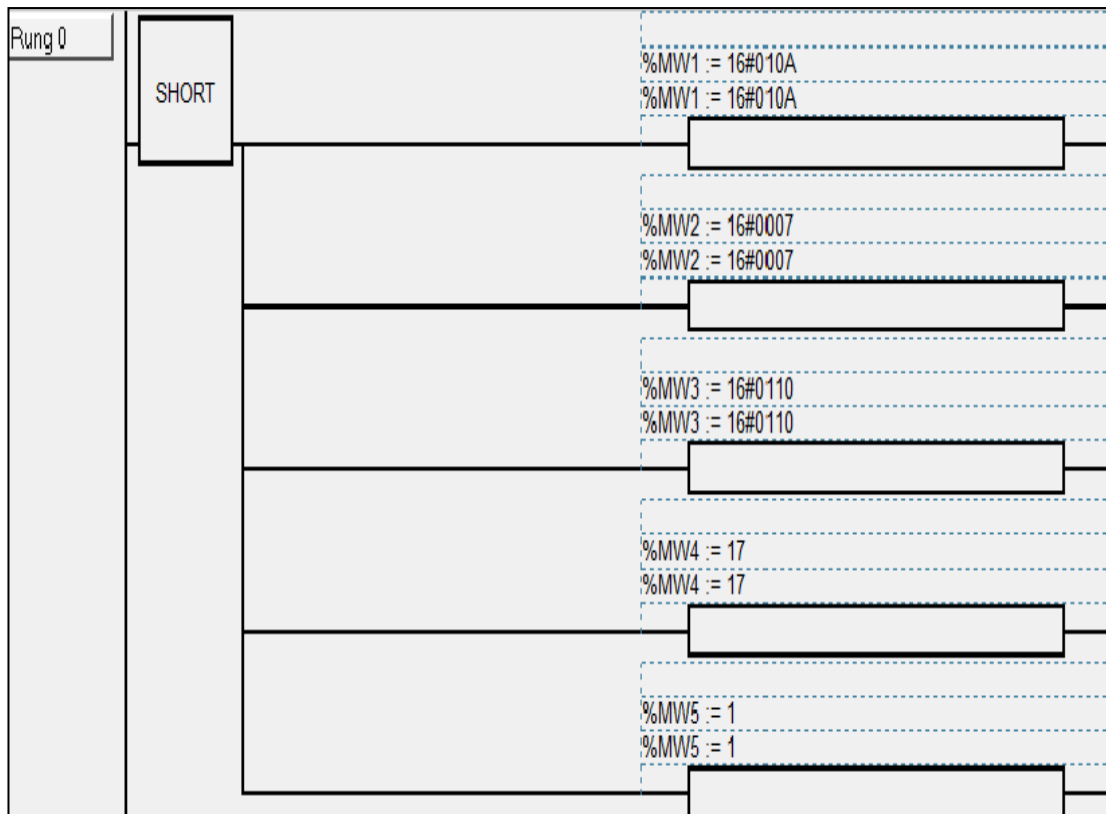


Figura 75. Rung 0 Escritura de una palabra sobre Zelio.
Se detalla el desarrollo de la tabla para la escritura de una palabra de memoria.

La función 16 permite escribir el estado de múltiples palabras del esclavo, es decir mediante este comando podemos escribir algunas palabras de memoria de alguna de las variables internas del esclavo.

Se inicia la tabla con la palabra de memoria %MW1 la cual corresponde a la sub-tabla de control utilizada para controlar la tabla de envío y recepción dentro de la tabla de escritura, está conformada por dos bytes el más significativo contiene el código de emisión y recepción (01) y el menos significativo (0A) corresponde a la longitud de la tabla.

En la palabra de memoria %MW2 contiene dos bytes y corresponde al offset o corrimiento de envío y recepción.

En la palabra %MW3 contiene la dirección del esclavo (01) y el código de función de la tabla de escritura respectivamente (10h).

En la palabra %MW4 es la dirección donde se va a escribir los valores de la palabra del maestro escrita.

En la palabra %MW5 representa el número de palabras que se van a escribir.

En la palabra %MW6 es la dirección de la escritura de N palabras para este caso dos bit, ya que puede ser un (uno lógico o cero lógico).

Para la siguiente línea de programación del equipo se asigna a %IO.1 para que active los valores que se van a escribir en la palabra de memoria %MW17.

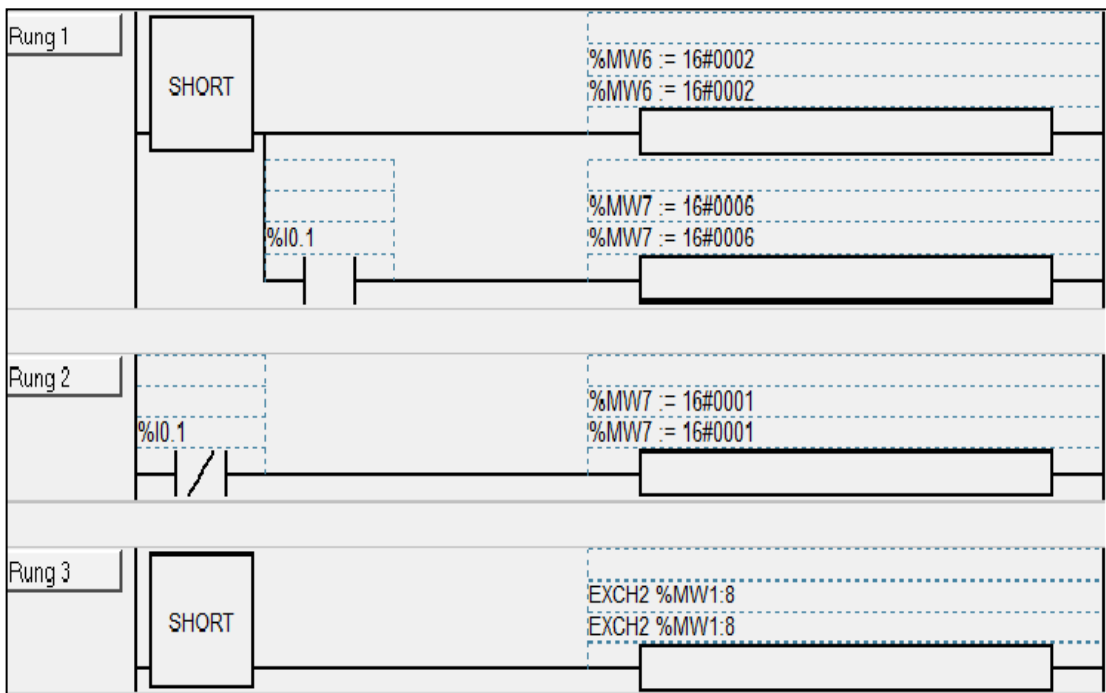


Figura 76. Escritura de palabra.

Valores que se van a escribir en la palabra de memoria %MW17.

Para finalizar utilizaremos la función “EXCHx” por medio de esta podremos realizar el envío de la tabla desarrollada, siendo “x” el número del puerto Modbus y luego se asigna la dirección de la primera memoria utilizada para la tabla y el número de memorias que ocupa la tabla.

4.4 COMUNICACIÓN ENTRE MAESTRO PLC TWIDO CON EL ESCLAVO ALTIVAR 312 UTILIZANDO MACROS DRIVE.

La configuración del maestro y la configuración del esclavo se hacen en el programa del maestro, en el cual se configura el puerto como maestro y la dirección del variador como esclavo, el mismo que cuenta con las siguientes características:

- Capacidad de 1 HP (0.75 KW).
- Tensión nominal de alimentación de 200 V a 240 V.

- Frecuencia de alimentación de 50 Hz a 60 Hz.
- Monofásico.
- Corriente de línea 7.5 A - 8.9 A.
- Disipación de potencia de 60W a carga nominal.
- Puerto de comunicación Modbus RTU

Este variador cuenta con una interfaz Modbus RTU integrada, la cual permitirá configurarlo para establecer la comunicación y por medio de registros o espacios de memorias Modbus asignados, poder realizar las peticiones desde algún maestro dentro de la red.

Esto nos permitirá más adelante realizar prácticas de “Marcha” y “Paro” del motor el cual esta controlador por el variador. También podremos controlar la frecuencia del equipo mediante el acceso al registro asignado por el fabricante el cual lo podremos encontrar en la hoja de técnica.

4.4.1 PROGRAMA PARA EL EQUIPO MAESTRO.

Se procede a configurar el puerto serie (antes agregado) para indicar el protocolo y dirección de red que se va a utilizar, se hace clic en el botón “OK” para aceptar los cambios.

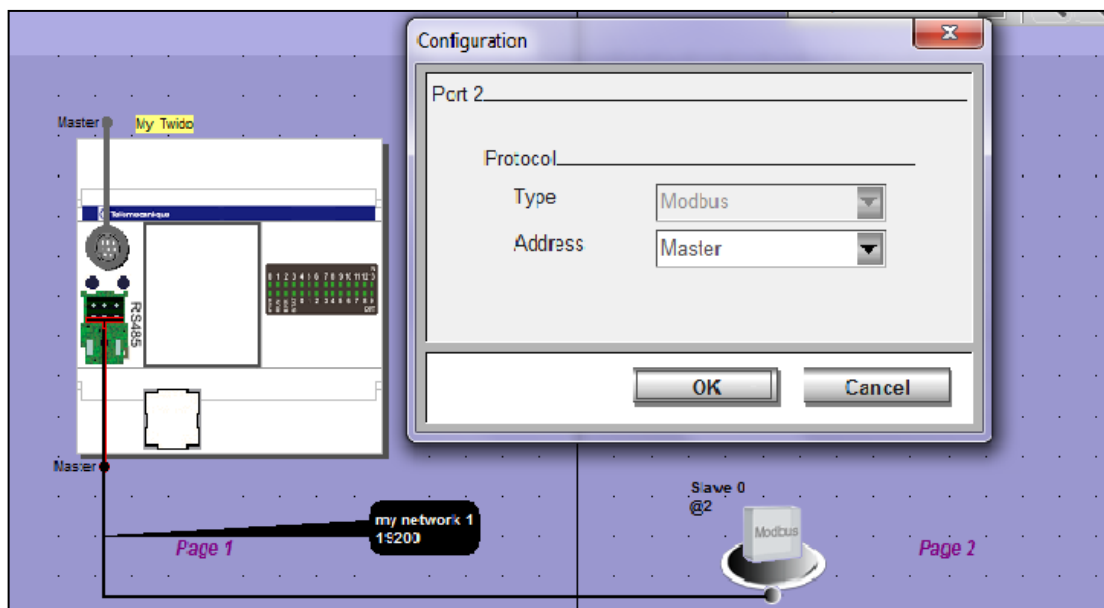


Figura 77. Configuración red Maestro.
Se configura el PLC en la red.

Cabe indicar que en esta lista no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

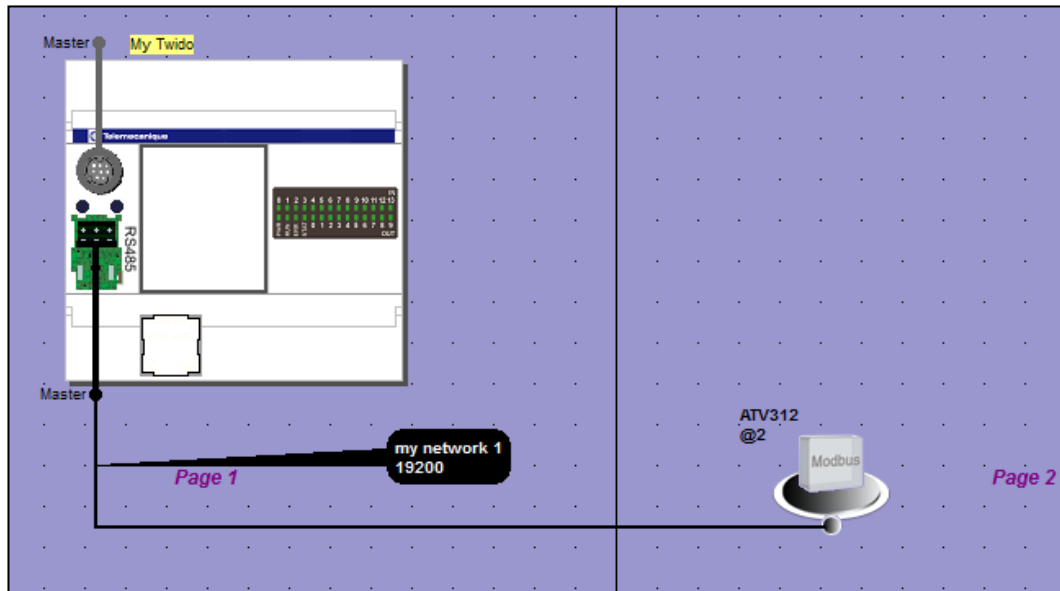


Figura 78. Configuración red Maestro con el ATV 312.
Se configura el PLC en la red.

Cabe indicar que en esta lista no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

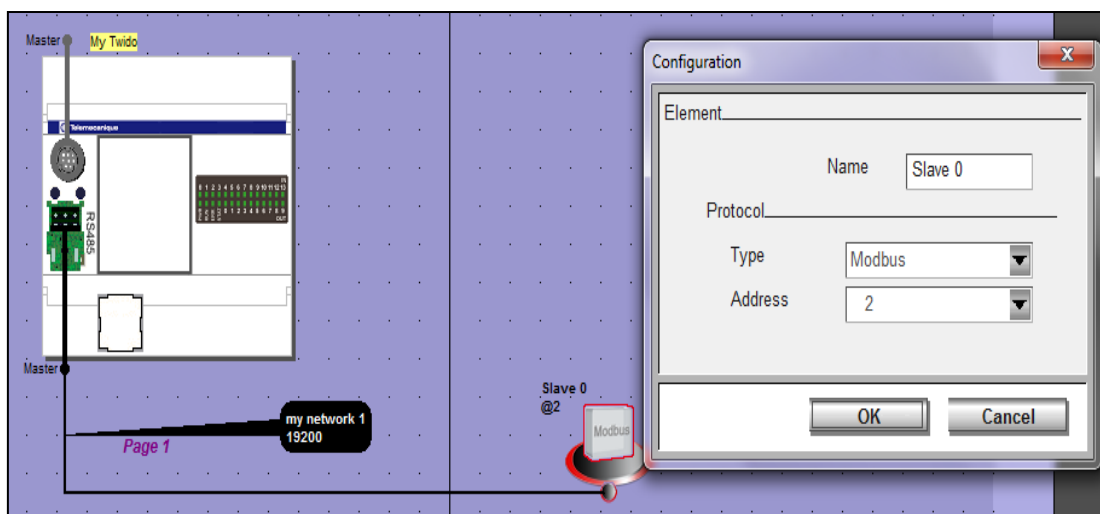


Figura 79. Configuración red esclavo ATV 312.
Se configura el PLC en la red.

Para finalizar la configuración del hardware se procede a enlazar los equipos en la red y verificar los parámetros de transmisión.

- Baudrate: es recomendable trabajar a una velocidad de transmisión de 19200 baudios.
- DataBits: Como se utiliza el formato RTU, se selecciona 8(RTU).
- Parity: La paridad será par.
- Bit Stop: Se utiliza 1 bit de parada

Una vez terminada la configuración del hardware se procede a la realización del programa en el cual se desarrollan las tablas de petición con lo indica la siguiente figura:

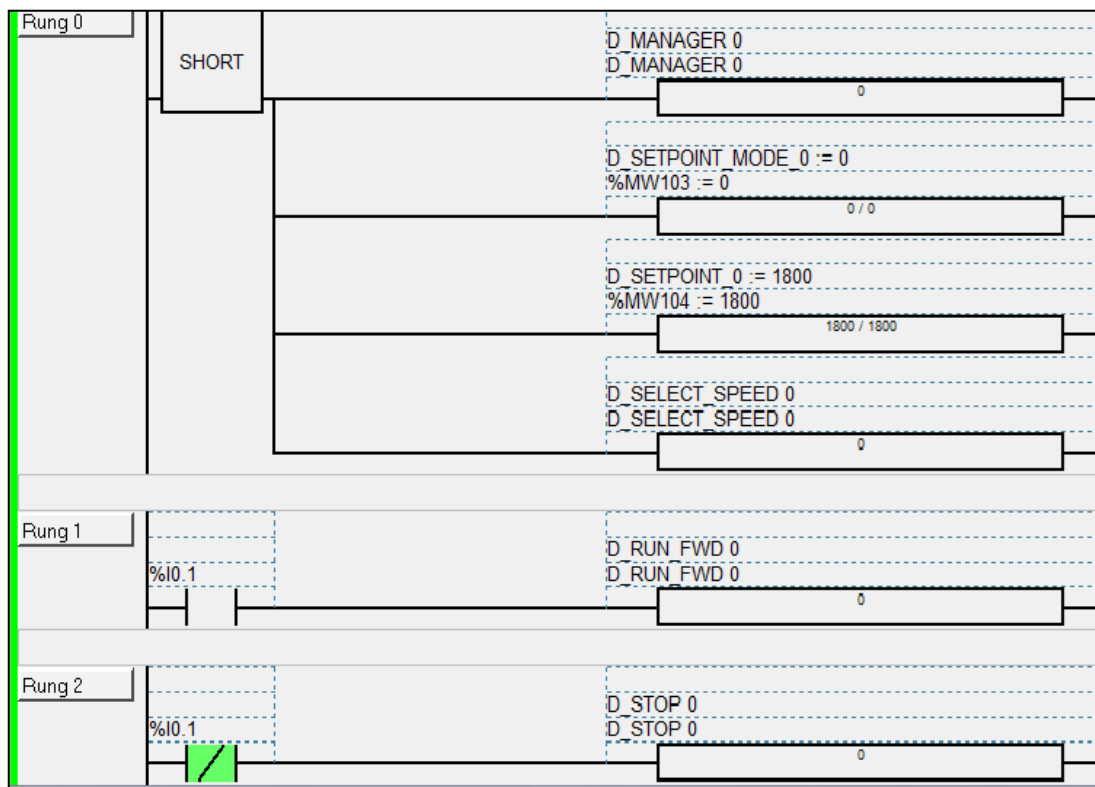


Figura 80. Rung 0, 1 y 2 funciones de Macro Drive.
Se detalla el desarrollo de cada función de la macro drive.

Función D_MANAGER:

Esta función controla a la unidad de velocidad del ATV 312 y además se encarga de que el dispositivo siga funcionando. Esta función debe activarse al comenzar el ciclo de funcionamiento del PLC Twido y antes de cativar cualquier otra función de la macro drive.

Para esta función al colocar el cero en este parámetro indica la instancia de esclavo a la que debe aplicarse a la macro.

D_SETPOINT_MODE 0:

Esta función guía al ATV 312 para que funcione en modo de consigna.

D_SETPOINT:

Esta función debe introducirse con el valor correspondiente a la velocidad tras haber activado el modo consigna, para este caso tendrá seteado una velocidad de 1800 RPM.

D_SELECT_SPEED:

La macro drive introduce este símbolo con el valor del bit leído tras la realización.

D_RUN_FWD:

Guía a la unidad de velocidad del ATV312 en la ejecución del proceso Ejecutar siguiente (giro a la derecha).

D_STOP:

Guía a la unidad de velocidad del ATV312 en la ejecución de la secuencia de parar el giro del motor.

4.5 COMUNICACIÓN ENTRE UN EQUIPO PLC MAESTRO Y DOS EQUIPOS ESCLAVOS.

En esta práctica se realiza la lectura de la palabra de memoria (20) del Zelio Logic, dependiendo del valor de esta palabra se va activar mi variador de frecuencia a una velocidad ya determinada por el PLC maestro.

Una vez situado en la pantalla de configuración de hardware se agrega al segundo puerto serie de comunicación, el cual se lo consigue en la ventana llamada “Catalogo” a lado derecho de la pantalla.

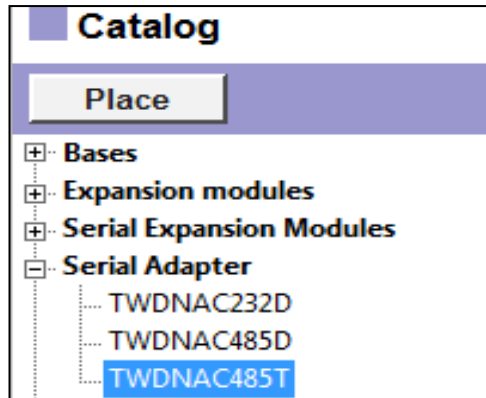


Figura 81. Ventana Catálogo Maestro. Despliega los elementos a agregar para configuración de hardware.

Luego se procede a configurar el puerto serie (antes agregado) para indicar el protocolo y dirección de red que se va a utilizar, se hace clic en el botón “OK” para aceptar los cambios.

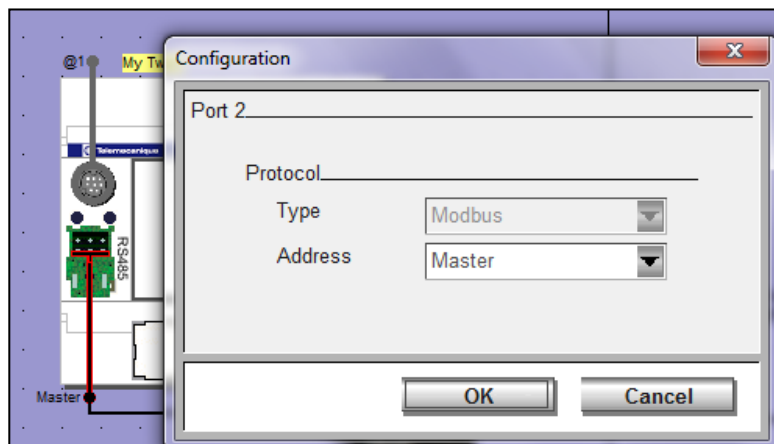


Figura 82. Configuración red Maestro. Se configura el PLC en la red Modbus como equipo Maestro.

Después se procede a agregar en la configuración el equipo esclavo, para esto se lo selecciona desde la pantalla “Catálogo”, “Elementos de red” y seleccionamos el equipo Modbus Twido.

Cabe indicar que en esta lista no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

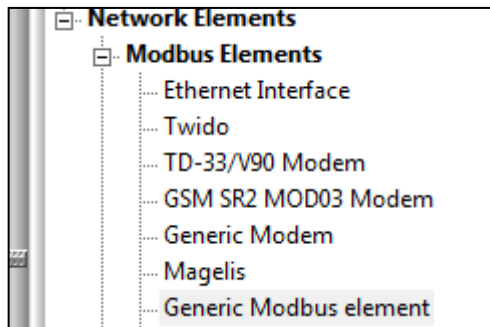


Figura 83. Elemento Modbus.
Desde la ventana “Catálogo” se puede agregar un elemento a la red.

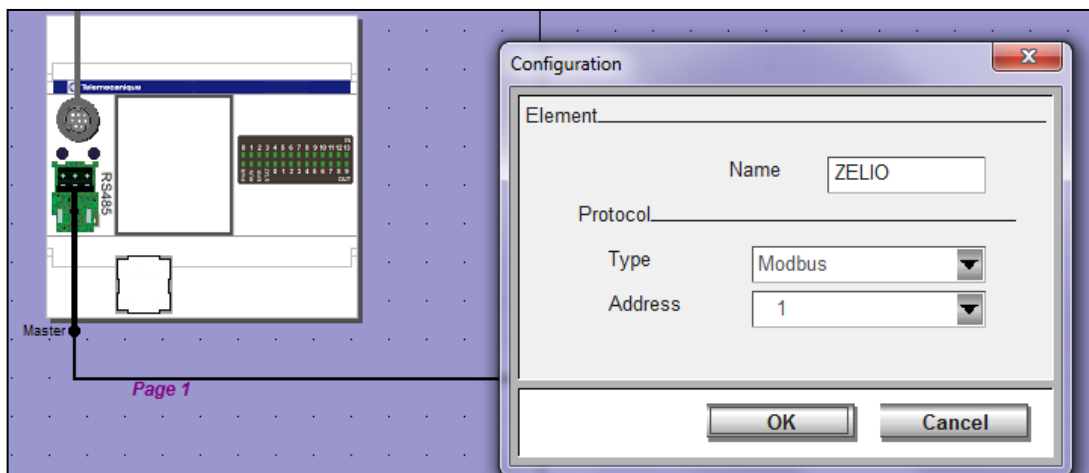


Figura 84. Configuración red esclavo Zelio.
Se configura el Zelio con dirección 1.

Una vez agregado el elemento esclavo Modbus a igual que en el equipo configurado como Maestro, se procede a configurar el tipo de la red y la dirección que se va a asignar al esclavo Zelio.

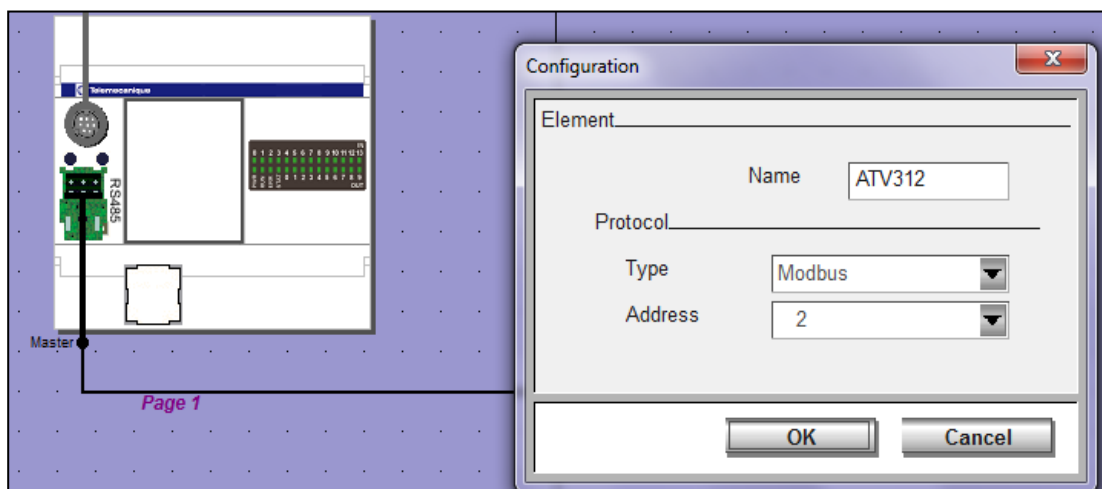


Figura 85. Configuración red esclavo ATV 312.
Se configura el variador de frecuencia con dirección 2.

Agregado el elemento esclavo Modbus a igual que en el equipo configurado como Maestro y esclavo zelio se procede a configurar el tipo de la red y la dirección que se va a asignar al esclavo ATV 312.

En la lista de elementos no se encuentran disponibles todos los dispositivos de una red Modbus, debido a la innumerable cantidad de equipos que pueden integrarse. Para dar solución a esto existe un “Elemento genérico” el cual cumple con la función de cualquier otro dispositivo que se quiera agregar.

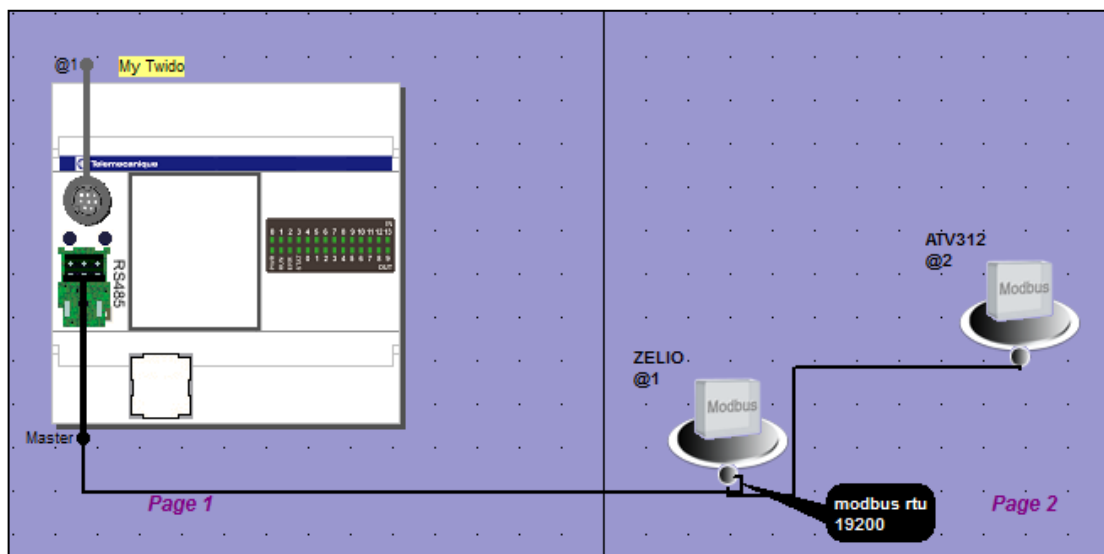


Figura 86. Configuración red Maestro con el Zelio y el ATV 312.

Para finalizar la configuración del hardware se procede a enlazar los equipos en la red y verificar los parámetros de transmisión.

- Baudrate: según el fabricante es recomendable trabajar a una velocidad de transmisión de 19200 baudios.
- DataBits: Como se utiliza el formato RTU, se selecciona 8(RTU).
- Parity: La paridad será par.
- Bit Stop: Se utiliza 1 bit de parada.

Una vez terminada la configuración del hardware se procede a la realización del programa en el cual se desarrollan las tablas de petición con lo indica la siguiente figura:

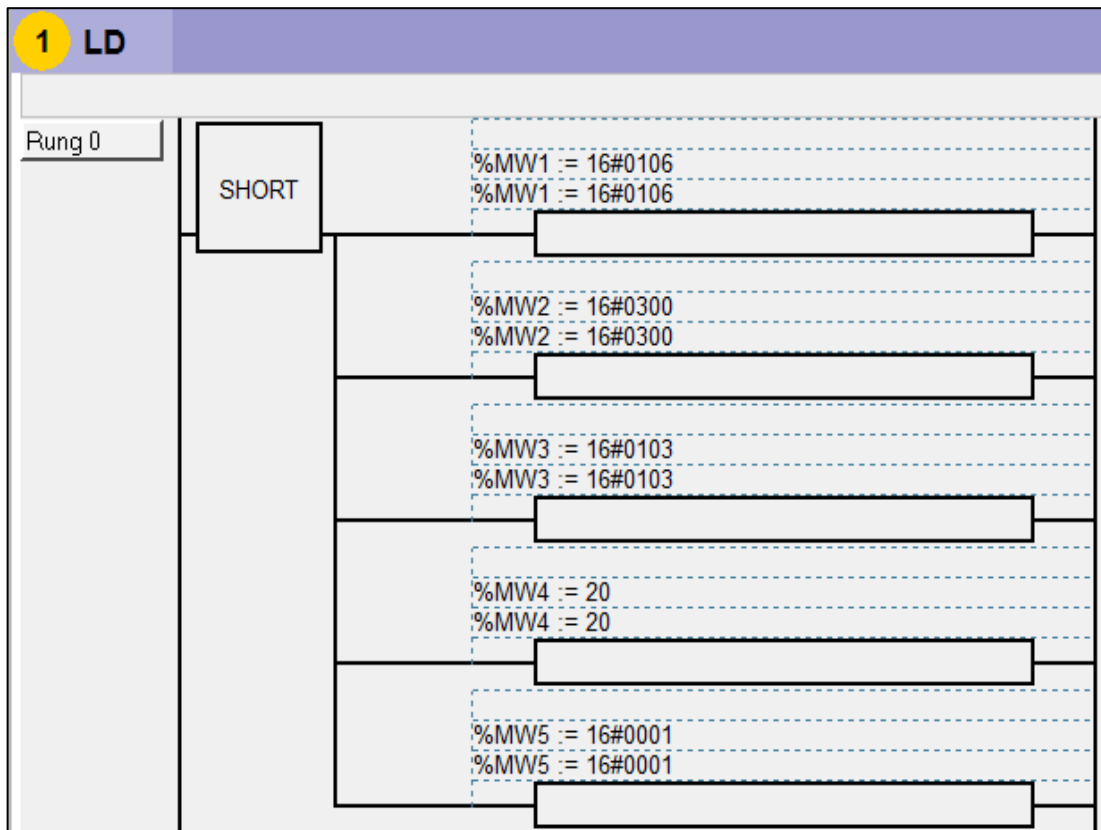


Figura 87. Rung 0 lectura de una palabra.

Se detalla el desarrollo de la tabla para la lectura de una palabra de memoria del Zelio.

Se inicia la tabla con la palabra de memoria %MW1 la cual corresponde a la sub-tabla de control utilizada para controlar la tabla de envío y recepción dentro de la tabla de escritura, está conformada por dos bytes el más significativo contiene el código de emisión y recepción (01) y el menos significativo corresponde la longitud de la tabla de envío (06).

En la palabra de memoria %MW2 contiene dos bytes y corresponde al offset o corrimiento de envío y recepción.

En la palabra de memoria %MW3 contiene la dirección del esclavo (01) y lee el registro de salida.

En la palabra %MW4 es la dirección donde se va a escribir los valores de la palabra del maestro escrita.

En la palabra %MW5 representa el número de palabras que se van a escribir.

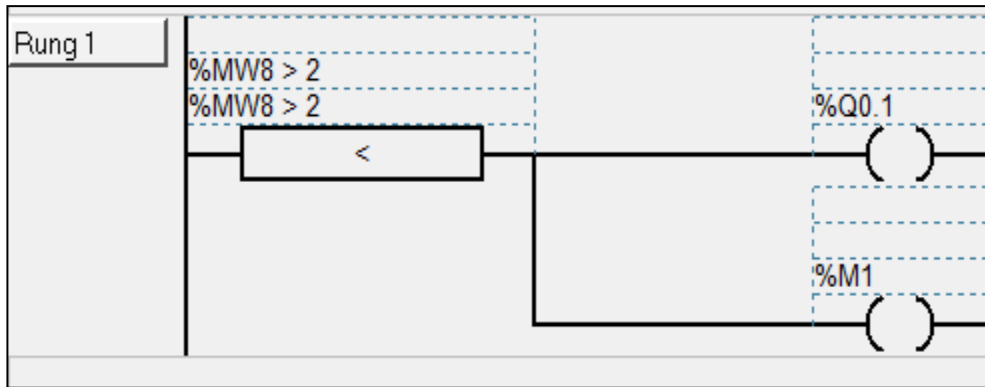


Figura 88. Escritura de palabra.
 Valor que se va a escribir en la palabra de memoria %MW4.

En la palabra de memoria %MW8 es donde va a quedar guardada la respuesta de mi esclavo Zelio, %Q0.1 es la salida del maestro, esta salida se activa dependiendo de lo que se lea en el esclavo Zelio, además al activarse la salida %Q0.1 se activa la marca %M1 del maestro.

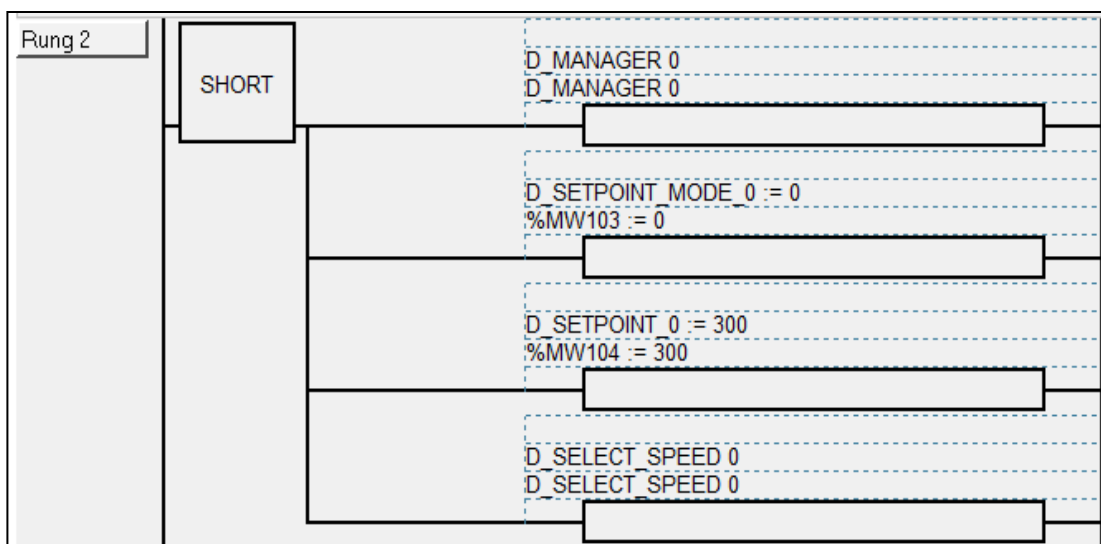


Figura 89. Rung 2 funciones de Macro Drive.
 Se detalla el desarrollo de cada función de la macro drive.

Función D_MANAGER:

Esta función controla a la unidad de velocidad del ATV 312 y además se encarga de que el dispositivo siga funcionando. Esta función debe activarse al comenzar el ciclo de funcionamiento del PLC Twido y antes de cativar cualquier otra función de la macro drive.

Para esta función al colocar el cero en este parámetro indica la instancia de esclavo a la que debe aplicarse a la macro.

D_SETPOINT_MODE 0:

Esta función guía al ATV 312 para que funcione en modo de consigna.

D_SETPOINT:

Esta función debe introducirse con el valor correspondiente a la velocidad tras haber activado el modo consigna, para este caso tendrá asignado una velocidad de 300 RPM.

D_SELECT_SPEED:

La macro drive introduce este símbolo con el valor del bit leído tras la realización.

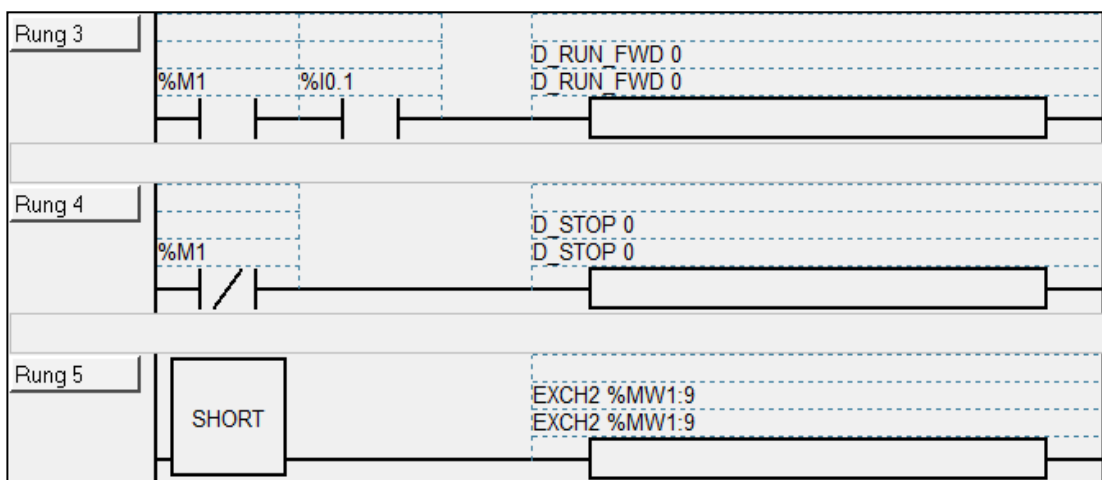


Figura 90. Rung 3,4 y 5 funciones de Macro Drive.

Se detalla el desarrollo de cada función de la macro drive.

Activada la marca %M1 se cerrara el contacto normalmente abierto esperando la línea de programación del equipo %I0.1 para que active el arranque del ATV 312.

Se coloca el contacto normalmente cerrado de la marca %M1 para que permita la ejecución del proceso del variador.

D_RUN_FWD:

Guía a la unidad de velocidad del ATV312 en la ejecución del proceso Ejecutar siguiente (giro a la derecha).

D_STOP:

Guía a la unidad de velocidad del ATV312 en la ejecución de la secuencia de parar el giro del motor.

Para finalizar utilizaremos la función “EXCHx” por medio de esta podremos realizar el envío de las tabla desarrollada, siendo “x” el número del puerto Modbus y luego se asigna la dirección de la primera memoria utilizada para la tabla y el número de memorias que ocupa la tabla.

CONCLUSIONES.

Buscar información de los protocolos de comunicación industrial es complicado debido a que la información que se encuentra es contradictoria, de acuerdo a las traducciones de cada protocolo ya que estas son adaptadas a la terminología propia de cada país, esto complica la comprensión de los mismos, ya que al realizar una traducción esta no siempre se obtienen una buena interpretación debido a que se dan varios nombres a un elemento, por esta razón para esta tesis de grado se trató al máximo de obtener los estándares del protocolo de comunicación industria Modbus RTU de fuentes oficiales como lo es Schneider Electric.

Para poder realizar las diversas practicas los autores han diseñado un cable el cual en un extremo tiene un conector RJ45 y en el otro extremo dos conectores de tres pines hembra , a los cuales se les coloco termo incogible para darle mayor protección, este tipo de cable fue diseñado para ser colocado en el variador y en el zelio , para el Twido se diseñó un cable que en su extremo tiene dos conectores de tres pines hembra y en el otro extremo se colocó estaño para darle rigidez a las terminaciones de las puntas de los cables , además se diseñó dos pares de cables que en sus terminaciones poseen conector de tres pines macho para que al momento de realizar cualquier práctica ya sea con cualquier equipo solo se conecte el cable hacia cada equipo ,cabe indicar que este diseño de conectores se realiza para que al realizar las diversas practicas el estudiante no este manipulando en la conexión y desconexión al momento de realizar la comunicación.

RECOMENDACIONES.

La automatización de los procesos industriales y más aun específicamente la implementación de redes de comunicación industriales es de gran utilidad en la industria ya que hay una optimización de cables al momento de realizar una programación, es por ello que a los estudiantes salesianos se les debe inculcar el aprendizaje de los diversos protocolos industriales de comunicación, para que cuando salgan egresados tengan mayor desarrollo y un mejor desenvolvimiento en el ámbito de la automatización y control de máquinas, equipos, procesos, etc.

Para lograr un mayor entendimiento de cómo se realiza la programación se debe tratar al máximo de obtener los estándares del protocolo de comunicación industrial Modbus de fuentes oficiales y en idioma inglés, debido a que cuando uno realiza la traducción al idioma español hay palabras en las cuales se les da varias interpretaciones a un mismo elemento esto puede ocasionar confusión al momento de entender las tablas de comunicación al momento de realizar la programación.

Al momento de hacer prácticas se recomienda que en lo mínimo el estudiante realice la desconexión y conexión en lo que tiene que ver con los cables de comunicación debido a si uno de los hilos de los cables se interrumpe no habla comunicación por lo tanto habrá un fallo al momento de leer o escribir un byte.

CRONOGRAMA DE ACTIVIDADES.

	MES 1				MES 2				MES 3				MES 4				MES 5				MES 6			
ACTIVIDAD	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Introducción al protocolo Modbus																								
Compra y adquisición de equipos.																								
Diseño y construcción de maleta didáctica.																								
Acoplamiento de equipos en maleta.																								
Cableado y montaje de equipos en maleta didáctica																								
Configuración de equipos																								
Pruebas de funcionamiento																								
Realización de correcciones y modificaciones																								
Presentación del funcionamiento general de cada módulo																								
Desarrollo teórico.																								

PRESUPUESTO.

A continuación se detalla en una tabla el presupuesto para la realización de este proyecto:

CANTIDAD	DESCRIPCIÓN	VALOR
2	Diseño y ensamblaje de maletas didácticas	\$ 1.100
2	Autómata maestro y equipos esclavo	\$ 1.200
4	Módulo de comunicación Modbus	\$ 500
2	Equipo de protección eléctrica	\$ 120
2	Cable comunicación Modbus	\$ 120
2	Variador de frecuencia	\$ 640
2	Motor asíncrono	\$ 400
	Materiales y herrajes para implementación	\$ 200
Global	Transportes y Varios	\$ 150
VALOR TOTAL:		\$ 4.430

REFERENCIAS.

- Schneider Electric. (06 de 2012). *ATV12 Modbus Communication Manual*. Recuperado el 15 de 08 de 2014, de ATV12 Modbus Communication Manual : http://download.schneider-electric.com/files?p_File_Id=214601873&p_File_Name=ATV12_Modbus_EN_BBV28590_02.pdf
- Alicante, F. A.-U. (11 de 10 de 2010). *Comunicación con RS-485 y MODBUS*. Recuperado el 15 de 08 de 2014, de Comunicación con RS-485 y MODBUS: <http://rua.ua.es/dspace/bitstream/10045/18990/1/AA-p3.pdf>
- Antonio Creus Solé. (2005). *Instrumentación industrial*. Mexico: Marcombo.
- Automática, U. d. (05 de 2010). *Redes locales en entornos industriales. Buses de Campo*. Recuperado el 15 de 10 de 2014, de Redes locales en entornos industriales. Buses de Campo: <http://www.isa.uniovi.es/docencia/redes/Apuntes/tema9.pdf>
- Automática, U. d.-I. (05 de 2010). *Autómatas Programables*. Recuperado el 15 de 11 de 2014, de Autómatas Programables: http://isa.uniovi.es/docencia/iea/teoria/plc_resumen.pdf
- Barragan, J. (10 de 06 de 2013). *Tutorial Virtual*. Recuperado el 05 de 10 de 2014, de Tutorial Virtual: <http://uhu.es/antonio.barragan/content/modbus-tcp>
- Bartolomé, J. (20 de 1 de 2011). *Tolaemon*. Recuperado el 03 de 11 de 2014, de Tolaemon: <http://www.tolaemon.com/docs/modbus.htm>
- Carolina Lagos, C. d. (15 de 09 de 2006). *Protocolos de Comunicación Industrial*. Recuperado el 12 de 10 de 2014, de Protocolos de Comunicación Industrial: <http://www.aie.cl/files/file/comites/ca/articulos/diciembre2-06.pdf>
- Distefano, I. M. (08 de 2008). *ELECTRÓNICA GENERAL Y APLICADA, COMUNICACIONES INDUSTRIALES*. Recuperado el 20 de 11 de 2014, de ELECTRÓNICA GENERAL Y APLICADA, COMUNICACIONES INDUSTRIALES: http://www.infopl.net/files/documentacion/comunicaciones/infoPLC_net_ComunicacionEntornos_Industriales.pdf
- Electric, S. (2010). Redes Industriales. En S. Electric, *Electricidad* (págs. 199-231).
- ETI, D. E.-C. (06 de 2009). *Comunicaciones Industriales*. Recuperado el 06 de 12 de 2014, de Comunicaciones Industriales: http://www.infopl.net/files/documentacion/comunicaciones/infopl_net_00presentacioncursocomindbabel.pdf

- IDA, M. (24 de 10 de 2006). *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE*. Recuperado el 6 de 12 de 2014, de MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE:
http://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf
- IDA, M. (20 de 12 de 2006). *MODBUS OVER SERIAL LINE*. Recuperado el 25 de 10 de 2014, de MODBUS OVER SERIAL LINE:
http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf
- IDA, M. (26 de 04 de 2012). *MODBUS APPLICATION PROTOCOL SPECIFICATION*. Recuperado el 12 de 11 de 2014, de MODBUS APPLICATION PROTOCOL SPECIFICATION:
http://www.modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
- Ingeniería, U. d.-F. (2011). Buses de campo y protocolos en redes industriales. *SUM*, 85-96.
- Instrumentación, A. C. (05 de 2014). *Herramientas para supervisión y adquisición de datos*. Recuperado el 12 de 11 de 2014, de Herramientas para supervisión y adquisición de datos: <http://www.arian.cl/espanol/rs485com.htm>
- Instruments, N. (16 de 10 de 2014). *Informacion detallada del Protocolo Modbus*. Recuperado el 25 de 10 de 2014, de Informacion detallada del Protocolo Modbus: www.ni.com/white-paper/52134/es/pdf
- International, M. I. (04 de 2006). *Introduction to Fieldbus*. Recuperado el 02 de 10 de 2014, de Introduction to Fieldbus:
http://www.automation.com/pdf_articles/fieldbus.pdf
- Kryon_Ingeniería. (08 de 05 de 2014). *Protocolo Modbus*. Recuperado el 12 de 01 de 2015, de Protocolo Modbus.
- Manuel Jiménez Buendía. (2010). *Protocolo Modbus*. Cartajena.
- MEX, A. (05 de 2008). *Controladores Logicos programables*. Recuperado el 3 de 10 de 2014, de Controladores Logicos programables:
<http://automatica.mex.tl/imagesnew/5/0/1/4/2/Presentaci%C3%B3n%20P.L.C..pdf>
- Mista, R. (06 de 07 de 2012). *Guía básica para el uso del Modbus*. Recuperado el 15 de 12 de 2014, de Guía básica para el uso del Modbus:
<file:///C:/Users/jmacias/Downloads/102076164-Presentacion-de-Modbus.pdf>
- OMRON. (09 de 05 de 2010). *Introduccion a los Automatas Programables*. Recuperado el 15 de 12 de 2014, de Introduccion a los Automatas Programables:

<ftp://ftp.unicauca.edu.co/Facultades/FIET/DEIC/Materias/SW%20para%20aplicaciones%20Industriales%20II/Sw%20II/Conferencias/Capitulo%201.pdf>

- ORG, A. (03 de 2008). *Automatas.ORG*. Recuperado el 10 de 10 de 2014, de Automatas.ORG: <http://www.automatas.org/abb/comunicaciones.htm>
- Sanchez, J. A. (2006). *Instrumentación y control avanzado de Procesos*. Diaz de Santos.
- Schneider Electric. (06 de 2011). *TwidoSuite V2.3 Programming Guide*. Recuperado el 15 de 08 de 2014, de TwidoSuite V2.3 Programming Guide: http://download.schneider-electric.com/files?p_File_Id=27603565&p_File_Name=35011386_K01_000_05.pdf
- Schneider-Electric. (05 de 2008). *Manual Twido*. Recuperado el 20 de 01 de 2015, de Manual Twido: http://www.equiposdidacticos.com/pdf/catalogos/Manual_Twido.pdf
- Schneider-Electric. (06 de 2011). *TwidoSuite V2.3 Programming Guide*. Recuperado el 10 de 11 de 2014, de TwidoSuite V2.3 Programming Guide: http://download.schneider-electric.com/files?L=en&p=533&p_docId=&p_docId=&p_Reference=35011386K01000&p_EnDocType=User%20guide&p_File_Id=27603565&p_File_Name=35011386_K01_000_05.pdf
- SIEMENS. (05 de 2015). *Siemens Automation Cooperates with Education*. Recuperado el 05 de 12 de 2015, de Siemens Automation Cooperates with Education: https://www.swe.siemens.com/spain/web/es/industry/automatizacion/sce_educacion/e_education/pages/default.aspx
- SIEMENS_AG. (06 de 2010). *Módulos_Modbus RTU RS485*. Recuperado el 01 de 2015, de Módulos_Modbus RTU RS485: https://cache.automation.siemens.com/dnl_iis/TE/TE2MjQwNwAA_44880906_HB/USMII_MODBUS_AOM_OI_ES_es-ES.pdf
- Suyama, M. (30 de 08 de 2004). *Protocolos de comunicaciones*. Recuperado el 12 de 11 de 2014, de Protocolos de comunicaciones: <http://www.desarrolloweb.com/articulos/1617.php>
- WEG. (1 de 05 de 2010). *Manual de la Comunicación Modbus-RTU*. Recuperado el 05 de 01 de 2014, de Manual de la Comunicación Modbus-RTU: <http://ecatalog.weg.net/files/wegnet/WEG-srw01-manual-de-la-comunicacion-modbus-rtu-10000521680-4.0x-manual-espanol.pdf>