



Diseño e implementación de un banco de pruebas y análisis de protocolos de red IEC 60870-5-104 y Modbus TCP dentro de redes de distribución eléctrica utilizando tarjetas de desarrollo

García Catota, Jonathan Marcelo

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica,
Automatización y Control

Ing. Tipán Condolo, Edgar Fernando, MSc.

15 de febrero del 2024



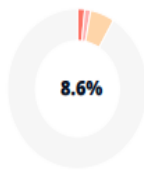
Plagiarism and AI Content Detection Report

TrabajosdeTitulacion_Final5.pdf

Scan details

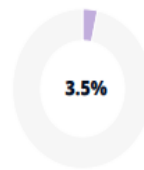
Scan time:	Total Pages:	Total Words:
February 16th, 2024 at 12:28 UTC	118	29355

Plagiarism Detection



Types of plagiarism		Words
Identical	1.7%	501
Minor Changes	1.4%	399
Paraphrased	5.6%	1636
Omitted Words	0%	0

AI Content Detection



Text coverage		Words
AI text	3.5%	1021
Human text	96.5%	28334

[Learn more](#)

🔍 Plagiarism Results: (47)

🌐 Ethernet Shield Arduino parte 1 3%

<https://programarfacil.com/blog/arduino-blog/ethernet-shield-arduino/>

Firma:

.....

Ing. Tipán Condolo, Edgar Fernando, MSc.

Director



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Certificación

Certifico que el trabajo de titulación: **“Diseño e implementación de un banco de pruebas y análisis de protocolos de red IEC 60870-5-104 y Modbus TCP dentro de redes de distribución eléctrica utilizando tarjetas de desarrollo”** fue realizado por el señor **García Catota, Jonathan Marcelo**; el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Sangolquí, 15 de febrero del 2024

Firma:

.....
Ing. Tipán Condolo, Edgar Fernando, MSc.

C. C.: 1711391316



Departamento de Eléctrica, Electrónica y Telecomunicaciones
Carrera de Ingeniería en Electrónica, Automatización y Control

Responsabilidad de Autoría

Yo, **García Catota, Jonathan Marcelo**, con cédula de ciudadanía n° 1718023789, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Diseño e implementación de un banco de pruebas y análisis de protocolos de red IEC 60870-5-104 y Modbus TCP dentro de redes de distribución eléctrica utilizando tarjetas de desarrollo”**, es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 15 de febrero del 2024

Firma

García Catota, Jonathan Marcelo

C.C.: 1718023789



Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización y Control

Autorización de Publicación

Yo **García Catota, Jonathan Marcelo**, con cédula de ciudadanía n° 1718023789, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Diseño e implementación de un banco de pruebas y análisis de protocolos de red IEC 60870-5-104 y Modbus TCP dentro de redes de distribución eléctrica utilizando tarjetas de desarrollo”**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 15 de febrero del 2024

Firma

García Catota, Jonathan Marcelo

C.C.: 1718023789

Dedicatoria

Dedico todo el esfuerzo de este trabajo a mis amados padres que estuvieron en todo momento de mi formación profesional. A mi hijo Emiliano y a mi novia Emily, que son los amores de mi vida y son mi fuente de inspiración para poder salir adelante y superar cualquier adversidad por nuestro bienestar. También quiero dedicar este logro a mis queridos hermanos, a mis apreciados suegros y a las demás personas, docentes, familiares y amigos, que en algún momento me brindaron su apoyo y ayuda para poder concluir mi meta.

Jonathan Marcelo García Catota

Agradecimiento

Agradezco a Dios por todas las bendiciones derramadas en el transcurso de la carrera y por guiarme hasta este momento. Agradezco a mis padres por todo su cariño y apoyo incondicional durante toda mi vida y, en especial a lo largo de mi formación profesional, que fue muy indispensable para poder obtener este logro.

Agradezco a mi hijo Emiliano y a mi novia Emily, por ser el soporte fundamental de mi vida y que con su amor y aliento incansable me impulsaron en todo momento para no rendirme y alcanzar mis objetivos. Agradezco a mis hermanos por toda su ayuda y colaboración que me concedieron durante los momentos más complejos.

Agradezco a mis suegros por su cariño y apoyo que me ofrecieron en todo momento. Finalmente, agradezco al Ing. Edgar Tipán por brindarme toda su colaboración, paciencia y tiempo para desarrollar satisfactoriamente este proyecto, pero principalmente, por ser un excelente guía y amigo a lo largo de este periodo.

Jonathan Marcelo García Catota

Tabla de contenido

Dedicatoria	6
Agradecimiento	7
Resumen	15
Abstract	16
Capítulo I	17
Antecedentes	17
Justificación e importancia	18
Alcance del proyecto	20
Objetivos	21
<i>Objetivo general</i>	21
<i>Objetivos específicos</i>	22
Capítulo II	23
Marco Conceptual	23
<i>Industria 4.0</i>	23
<i>Redes de Distribución Eléctrica</i>	24
<i>Sistemas de control en subestaciones eléctricas</i>	26
<i>Protocolos de comunicación industriales</i>	29
<i>Protocolos de comunicación industriales MODBUS</i>	32
<i>ModbusTCP</i>	46
<i>Protocolos de comunicación industriales de red IEC 60870-5-104</i>	50
<i>Software Wireshark</i>	100
Capítulo III	103
Diseño	103

<i>Diseño de la estación maestro: definición de sus elementos</i>	103
<i>Diseño de las estaciones esclavo: definición de sus elementos</i>	108
<i>Diseño de la red de comunicación: definición de sus elementos</i>	112
<i>Librerías de Comunicación para el Maestro</i>	119
<i>Librerías de Comunicación para los Esclavos</i>	124
Capítulo IV	127
Implementación	127
<i>Implementación del hardware</i>	127
<i>Implementación del software</i>	134
Capítulo V	143
Validación, Pruebas y Resultados	143
<i>Pruebas y evaluación de conexión</i>	143
<i>Pruebas y evaluación de la conectividad del sistema en el estuche</i>	144
<i>Pruebas y evaluación de la comunicación Modbus TCP</i>	145
<i>Pruebas y evaluación de la comunicación IEC60870-5-104</i>	146
<i>Pruebas y evaluación del sistema completo</i>	147
<i>Pruebas y evaluación del monitoreo de red</i>	148
<i>Análisis de los resultados</i>	149
Capítulo VI	150
Conclusiones	150
Recomendaciones	151
Bibliografía	152
Apéndices	157

Índice de Tablas

Tabla 1 <i>Formato del marco de mensaje Modbus</i>	35
Tabla 2 <i>Direcciones Modicon y códigos de función</i>	37
Tabla 3 <i>Lista abreviada de códigos de excepción devueltos</i>	45
Tabla 4 <i>Estructura de Solicitud</i>	47
Tabla 5 <i>Estructura de Respuesta</i>	48
Tabla 6 <i>Significados de los bits del campo de control: transmisión desequilibrada</i>	60
Tabla 7 <i>Códigos de función: transmisión desbalanceada</i>	61
Tabla 8 <i>Códigos de función: transmisión desbalanceada</i>	62
Tabla 9 <i>Significados de los bits del campo de control: transmisión balanceada</i>	68
Tabla 10 <i>Códigos de función: transmisión balanceada</i>	69
Tabla 11 <i>Seguridad de error de capa de enlace</i>	72
Tabla 12 <i>Grupos de códigos de tipo definidos</i>	79
Tabla 13 <i>Estructura del código de información de referencia</i>	80
Tabla 14 <i>Tipos de ASDU: información del proceso en la dirección de monitoreo</i>	81
Tabla 15 <i>Tipos de ASDU: información del proceso en la dirección de monitoreo (next)</i>	82
Tabla 16 <i>Tipos de ASDU: información del proceso en la dirección de control</i>	84
Tabla 17 <i>Tipos de ASDU: parámetro en la dirección de control</i>	84
Tabla 18 <i>Tipos de ASDU: información del sistema en la dirección de control</i>	84
Tabla 19 <i>Tipos de ASDU: parámetro en la dirección de control</i>	85
Tabla 20 <i>Tipos de ASDU: transferencia de archivos</i>	85
Tabla 21 <i>Causa de los códigos de transmisión (COT)</i>	88
Tabla 22 <i>Dirección de ASDU</i>	91
Tabla 23 <i>Elementos de información</i>	97

Índice de Figuras

Figura 1 <i>Esquema general del proyecto</i>	21
Figura 2 <i>Herramientas computacionales de la Industria 4.0</i>	24
Figura 3 <i>Esquema de una red eléctrica inteligente</i>	25
Figura 4 <i>Niveles de control de una subestación eléctrica</i>	28
Figura 5 <i>Arquitectura de control y supervisión</i>	30
Figura 6 <i>Interacción maestro/esclavo Modbus</i>	33
Figura 7 <i>Marco Modbus genérico</i>	33
Figura 8 <i>Transacción Modbus</i>	34
Figura 9 <i>Pila de comunicación Modbus</i>	35
Figura 10 <i>Ejemplo de estado de la bobina de lectura</i>	39
Figura 11 <i>Ejemplo de estado de entrada de lectura</i>	40
Figura 12 <i>Ejemplo de registro de espera de lectura</i>	41
Figura 13 <i>Ejemplo de registro de entrada de lectura</i>	42
Figura 14 <i>Ejemplo de forzado de una sola bobina</i>	43
Figura 15 <i>Ejemplo de preselección de un único registro</i>	44
Figura 16 <i>Ejemplo de forzar varias bobinas</i>	44
Figura 17 <i>Ejemplo de preajuste de varios registros</i>	45
Figura 18 <i>Ejemplo de una solicitud ilegal</i>	46
Figura 19 <i>Estructura de la trama de un maestro Modbus</i>	47
Figura 20 <i>Arquitectura del protocolo IEC 60870-5-104</i>	51
Figura 21 <i>Relación del modelo EPA con el modelo OSI de siete capas</i>	52
Figura 22 <i>Arquitecturas para T101 y T104</i>	53
Figura 23 <i>Representación de secuencia de bits de la trama FT1.2</i>	54
Figura 24 <i>Opciones de trama FT1.2 según IEC6087-5-101</i>	55

	12
Figura 25 <i>Orden de información</i>	57
Figura 26 <i>Servicios de capa de enlace</i>	58
Figura 27 <i>Campo de control-transmisión desequilibrada</i>	60
Figura 28 <i>Campo de control: transmisión balanceada</i>	68
Figura 29 <i>Estructura de mensajes según IEC60870-5-101</i>	74
Figura 30 <i>Estructura del APDU</i>	76
Figura 31 <i>Contenido de la Estructura del APDU, dependiendo del tipo de comando</i>	77
Figura 32 <i>Tipo de campo de identificación</i>	78
Figura 33 <i>Calificador de estructura variable y estructuras</i>	87
Figura 34 <i>Causa del campo de transmisión</i>	88
Figura 35 <i>Dirección común de ASDU</i>	90
Figura 36 <i>Dirección del objeto de información</i>	92
Figura 37 <i>Comando de control emitido desde RTU de modo dual</i>	94
Figura 38 <i>Confirmación de activación ASDU volvió a modo dual RTU</i>	94
Figura 39 <i>Información de monitoreo devuelta al maestro SCADA</i>	95
Figura 40 <i>Objeto de información</i>	100
Figura 41 <i>Ícono del software Wireshark</i>	101
Figura 42 <i>Interfaz del software Wireshark</i>	102
Figura 43 <i>Partes de un Arduino UNO</i>	105
Figura 44 <i>Partes de la Raspberry Pi 4</i>	106
Figura 45 <i>Partes del Arduino Yun</i>	107
Figura 46 <i>Partes del Arduino Nano</i>	109
Figura 47 <i>Partes del Arduino Mega 2560</i>	110
Figura 48 <i>Partes del Arduino Leonardo</i>	111
Figura 49 <i>Partes del Shield Ethernet para Arduino</i>	114
Figura 50 <i>Indicadores del Ethernet shield V1 y V2</i>	115

	13
Figura 51 <i>Conexión Ethernet Shield y Arduino UNO</i>	116
Figura 52 <i>Pines de conexión - comunicación Arduino Uno y shield Ethernet W5100</i>	117
Figura 53 <i>Shield Ethernet ENC28J60</i>	117
Figura 54 <i>Conexión de Ethernet Shield ENC28J60 con Arduino Nano</i>	118
Figura 55 <i>Switch de 4 puertos TP-Link Router Inalámbrico TL-WR740N</i>	119
Figura 56 <i>Obtención de la dirección Ip y la máscara de red en PC</i>	121
Figura 57 <i>Declaración de direcciones IP de las estaciones esclavos</i>	122
Figura 58 <i>Declaración de objetos de comunicación Modbus</i>	122
Figura 59 <i>Inicialización de objetos de comunicación Modbus</i>	123
Figura 60 <i>Envío de datos desde la estación maestro a las estaciones esclavos</i>	123
Figura 61 <i>Diagrama de conexión del Ethernet Shield ENC28J60 a la red</i>	124
Figura 62 <i>Configuración dirección MAC e IP</i>	125
Figura 63 <i>Encabezado para el uso de librería Modbus-Ethernet para ENC28J60</i>	125
Figura 64 <i>Registro Modbus</i>	126
Figura 65 <i>Instancia mb ModbusEthernet</i>	126
Figura 66 <i>Conexión de las alimentaciones de las estaciones maestro y esclavos</i>	128
Figura 67 <i>Diagrama de conexión de estación maestro</i>	129
Figura 68 <i>Diagrama de conexión de estaciones esclavo</i>	131
Figura 69 <i>Adecuación del estuche para colocación de elementos</i>	131
Figura 70 <i>Conexionado de las estaciones maestro y esclavos</i>	132
Figura 71 <i>Colocación de las tarjetas de desarrollo en estuche</i>	132
Figura 72 <i>Banco de pruebas etiquetado</i>	134
Figura 73 <i>Topología en estrella del proyecto</i>	135
Figura 74 <i>Configuración de la IP del esclavo 3 (PC)</i>	137
Figura 75 <i>Declaración de librerías, objetos y variables de la estación esclavo</i>	139
Figura 76 <i>Configuración entradas, MAC e IP, e inicialización de comunicaciones</i>	139

Figura 77 <i>Envío de datos mediante los protocolos Modbus TCP e IEC60870-5-104</i>	140
Figura 78 <i>Declaración de librerías, objetos y variables de la estación esclavo</i>	141
Figura 79 <i>Configuración de entrada, salidas, MAC e IP, e inicialización de comunicaciones</i> . 141	
Figura 80 <i>Envío de datos mediante los protocolos Modbus TCP e IEC60870-5-104</i>	142
Figura 81 <i>Conexión de equipos para comprobación de conectividad del sistema</i>	143
Figura 82 <i>Conexión física para verificación de conectividad en el estuche</i>	144
Figura 83 <i>Comprobación de conectividad de las estaciones en PC</i>	145
Figura 84 <i>Verificación de comunicación Modbus TCP</i>	146
Figura 85 <i>Verificación de comunicación IEC60870-5-104</i>	147
Figura 86 <i>Verificación de la versión final en el sistema completo</i>	148
Figura 87 <i>Verificación de tramas esclavo3 (PC)-esclavo2 con el software Wireshark</i>	149
Figura 88 <i>Verificación de tramas esclavo2-esclavo3(PC) con el software Wireshark</i>	149

Resumen

Los protocolos de red Modbus e IEC60870-5 proporcionan comunicación maestro/esclavo entre dispositivos que se encuentren conectados a través de diferentes tipos de buses de datos. El análisis y estudio a profundidad de estos protocolos de red desde su estructura, tramas y comunicación, además de realizar una buena comprensión de todos estos aspectos en cuanto a datos enviados y recibidos en cada proceso de comunicación entre las estaciones, que son visualizados mediante el software Wireshark, son esenciales para entender, detectar y corregir fallas, de comunicación o ya sea cualquier otra anomalía en relación a controladores, sensores o actuadores, entre estaciones y subestaciones eléctricas tras su posterior estudio. Las tarjetas de desarrollo son conectadas con shields Ethernet que permiten la conexión en red y además estas tarjetas contienen librerías que permiten emplear los protocolos de red Modbus TCP e IEC60870-5-104, mediante el uso de comandos propios de cada librería. El diseño e implementación del banco de pruebas presenta ventajas para el usuario, en donde la simulación de envío y recepción de datos, tanto digitales como analógicos, en el dispositivo están a cargo de interruptores, potenciómetros y diodos led; teniendo en cuenta que posee una sencilla estructura y presentación amigable con el usuario, para así lograr operar el sistema sin muchas complicaciones y captar con facilidad su funcionamiento para cada uno de sus respectivos protocolos.

Palabras clave: Modbus, IEC60870-5, protocolo de red, maestro, esclavo.

Abstract

Modbus and IEC60870-5 network protocols provide master/slave communication between devices that are connected through different types of data buses. The in-depth analysis and study of these network protocols from their structure, frames and communication, in addition to achieving a good understanding of all these aspects in terms of data sent and received in each communication process between the stations, which are visualized through the Wireshark software, are essential to understand, detect and correct communication failures or any other anomaly in relation to controllers, sensors or actuators, between electrical stations and substations after subsequent study. The development cards are connected with Ethernet shields that allow network connection and these cards also contain libraries that allow the use of the Modbus TCP and IEC60870-5-104 network protocols, through the use of commands specific to each library. The design and implementation of the test bench presents advantages for the user, where the simulation of sending and receiving data, both digital and analog, in the device is carried out by switches, potentiometers and LED diodes; taking into account that it has a simple structure and user-friendly presentation, in order to operate the system without many complications and easily grasp its operation for each of its respective protocols.

Keywords: Modbus, IEC60870-5, network protocol, master, slave.

Capítulo I

Antecedentes

La Industria 4.0 impulsa la interconexión de sistemas físicos industriales con el entorno digital, instando a las organizaciones a abandonar su estado actual. Este fenómeno representa un punto de partida para que las entidades comprendan esta nueva revolución y puedan migrar hacia este paradigma, independientemente de su posición actual (Mora y Guerrero, 2020).

En este contexto, las subestaciones eléctricas también han evolucionado para adaptarse a estos cambios. Estas subestaciones, que desempeñan un papel crítico al aislar las fallas y prevenir daños a la infraestructura eléctrica, requieren una supervisión constante y un control eficaz de sus numerosos equipos (Gallardo, 2022). Con el tiempo, surgieron sistemas de supervisión, control y adquisición de datos (SCADA) que, lamentablemente, estaban basados en arquitecturas cerradas y protocolos propietarios de los fabricantes (Gallardo, 2022). Estos sistemas permiten la recopilación y monitorización de datos, así como la generación de informes a distancia para la detección y corrección de errores. Además, la introducción del microprocesador dio lugar a la aparición de los Dispositivos Electrónicos Inteligentes (IED), que cuentan con capacidades de comunicación y protocolos específicos (Gallardo, 2022).

En cuanto a la infraestructura de control en las subestaciones eléctricas, se utiliza un Sistema Integrado de Control Distribuido (SICD), que facilita el control, la supervisión y la protección de la subestación y sus líneas de entrada y salida (Manzano, 2022). Este sistema distribuido se basa en la interconexión de diversos sistemas en red mediante Dispositivos Electrónicos Inteligentes (IED), lo que permite la integración de funciones como el control, la protección y la medición, así como la ampliación de funciones de operación como la autosupervisión, el análisis de señales y fallas, el almacenamiento de datos y eventos, entre otros (Manzano, 2022).

Recientemente, han surgido nuevos protocolos de comunicación con mayor funcionalidad, siendo los más utilizados en sistemas eléctricos de potencia el Modbus, IEC61850, DNP3 e IEC60870-5 (Arias et al., 2020). A pesar de que Modbus es implementado por la mayoría de los fabricantes, presenta limitaciones en cuanto a las funciones que puede soportar, especialmente en operaciones de supervisión y control de instalaciones de gestión de energía eléctrica, debido a su incapacidad para informar automáticamente cambios en el estado de las variables de operación (Feria et al., 2015). En respuesta a esto, se desarrolló el Modbus TCP, que permite la comunicación de equipos industriales y dispositivos físicos a través de una red Ethernet.

Además, la especificación IEC 60870-5-104 se creó utilizando estándares establecidos, clases de datos preestablecidos, procedimientos de transferencia de datos y técnicas de administración de tramas previamente definidas en las normas IEC 60870-10, IEC 60870-5-102 y IEC 60870-5-103 (International Electrotechnical Commission, 2023). Esta norma brinda un conjunto integral de funcionalidades y procedimientos que facultan la notificación de eventos relacionados con protección y control.

Los protocolos mencionados anteriormente están orientados hacia medios de comunicación más versátiles, como TCP/IP sobre Ethernet (Gallardo, 2022). Por lo tanto, es cada vez más esencial que todos los componentes de una red de distribución eléctrica estén interconectados. Es aconsejable, por lo tanto, llevar a cabo un prototipo con el fin de realizar pruebas y análisis de los protocolos de red IEC 60870-5-104 y Modbus TCP dentro de redes de distribución eléctrica, utilizando tarjetas de desarrollo para familiarizarse con estos protocolos.

Justificación e importancia

El presente estudio resulta de gran relevancia, dado que en Ecuador se observa un crecimiento constante en la implementación de sistemas automatizados, en virtud de sus ventajas evidentes. El Centro Nacional de Control de Energía, CENACE, cuya responsabilidad

radica en salvaguardar las condiciones de seguridad y calidad en la operación del Sistema Nacional Interconectado, SIN, ha establecido la obligación de que todas las centrales, estaciones y subestaciones de distribución eléctrica se integren a un sistema de Supervisión, Control y Adquisición de Datos (Scada) a nivel nacional, con el fin de supervisar y monitorear de manera efectiva las redes eléctricas. Esto, a su vez, implica la necesidad imperativa de establecer una comunicación fluida y eficaz entre todos los equipos que conforman dicho sistema.

En la actualidad, las redes de distribución eléctrica en Ecuador están inmersas en el proceso de desarrollo e implementación de esta nueva tecnología, que busca la integración de centrales, estaciones y subestaciones de distribución eléctrica. Para lograr este cometido, se emplean protocolos de comunicación que son estándares para medios de comunicación ampliamente utilizados, tales como TCP/IP sobre Ethernet. Entre estos protocolos se destacan Modbus, IEC61850, DNP3 y IEC 60870-5, y se emplean equipos que facilitan la comunicación a través de dichos protocolos.

En particular, los protocolos de comunicación industrial IEC 60870-5-104 y Modbus TCP pueden considerarse como versiones mejoradas o actualizadas de protocolos que en el pasado requerían conexiones mediante cable serial. Dichos protocolos se han adecuado para ser ejecutados en redes Ethernet y emplear interfaces TCP/IP, por lo que han llegado a ser los más utilizados en cuanto a sistemas de telecontrol se refiere, enlazando estaciones de control con subestaciones mediante IP o TCP, gracias a su capacidad de transmisión de datos de forma confiable y fidedigna.

El conocimiento y la implementación de estos protocolos de comunicación industrial y su aplicación en tecnología moderna abren la puerta para seleccionar el protocolo más adecuado según las necesidades específicas de diseño e implementación de sistemas de energía, con el objetivo de minimizar pérdidas y errores en la transmisión de datos.

En resumen, con la introducción y actualización de esta nueva tecnología en las estaciones de las redes de distribución eléctrica a nivel nacional, es fundamental adquirir un profundo entendimiento de estos protocolos de comunicación, incluyendo sus procedimientos para la transferencia de información y gestión de tramas. Esto permitirá aprovechar plenamente la información recopilada a través de software de monitoreo de la red y, en última instancia, mejorar el rendimiento de los sistemas de energía.

Alcance del proyecto

El presente proyecto, principalmente, trata sobre la implementación de un sistema de comunicación entre dispositivos maestro y esclavos para realizar pruebas de red con las siguientes características:

- La comunicación entre los dispositivos deberá ser a través de red ethernet, utilizando los protocolos de red IEC 60870-5-104 y Modbus TCP con comunicación maestro esclavo.
- La estructura principal del prototipo constará de un dispositivo maestro, un switch o conmutador y dos dispositivos esclavos.
- Cada uno de los dispositivos maestro y esclavo contarán con interruptores, potenciómetros y leds que servirán para representar los datos analógicos y digitales que van ser comunicados entre maestro y esclavos.
- Para el monitoreo y verificación de la comunicación se conectará a la red un computador, el mismo que contará con el software Wireshark que se lo utilizará para el monitoreo de la red.
- Para la implementación de los dispositivos maestro y esclavos se utilizarán tarjetas de desarrollo que posteriormente serán seleccionados dentro del desarrollo del trabajo de titulación.

- Los datos serán transmitidos en tiempo real, pero para su monitoreo y verificación se realizará cada tres segundos, lo cual será configurado y manejado por el maestro de la red.

Figura 1

Esquema general del proyecto.



Nota. En el gráfico se tiene el esquema de un dispositivo maestro que se comunica con dos dispositivos esclavos a través de un switch y que se puede monitorear por intermedio de un computador.

Objetivos

Objetivo general

Diseñar e implementar un banco de pruebas y análisis de protocolos de red IEC 60870-5-104 y Modbus TCP dentro de redes de distribución eléctrica utilizando tarjetas de desarrollo para su estudio y comprensión.

Objetivos específicos

- Establecer la comunicación de los dispositivos desarrollados a partir de tarjetas de desarrollo utilizando los protocolos de red IEC 60870-5-104 o Modbus TCP.
- Investigar los protocolos de red IEC 60870-5-104 y Modbus TCP, desde su estructura, trama y hasta su comunicación a partir de comunicación maestro esclavo.
- Transmitir datos de señales analógicas y digitales entre los dispositivos para simular la comunicación y funcionamiento de una red de distribución eléctrica.

Capítulo II

Marco Conceptual

Industria 4.0

La Industria 4.0 se refiere técnicamente a la cuarta revolución industrial, que su principal característica es el enlace entre los medios de producción industrial con la comunidad digital, para así cumplir con los requerimientos de uso de la cada localidad (Barona López y Velasteguí, 2021). Este cambio de paradigma ha sido posible gracias al crecimiento exponencial de la tecnología y las Tecnologías de la Información y Comunicación (TIC) en las últimas décadas, así como a los esfuerzos continuos de diversas industrias por adoptar y promover la utilización de estas tecnologías (Rozo-García, 2020).

La digitalización y el enlace entre sistemas y procesos industriales mediante el Internet de las cosas (IoT) y el Internet de los servicios (IoS) son parte fundamental de la industria 4.0 con la finalidad de aumentar la flexibilidad y la personalización en los procesos de producción. Esto se traduce en la implementación de tecnologías avanzadas que hacen que las soluciones sean flexibles, inteligentes y completamente autónomas (Rozo-García, 2020).

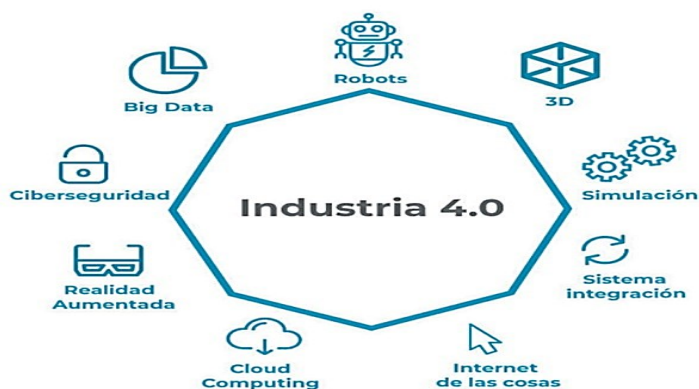
La Figura 2 (EVICITI, 2023), ilustra las herramientas informáticas que son fundamentales en el contexto de la Industria 4.0, tales como el Big Data, la computación en la nube, la ciberseguridad, la robótica, la realidad virtual y aumentada, la visión artificial y el Internet de las Cosas (Luna-López et al., 2021). El Big Data se utiliza para llevar a cabo un análisis masivo de los datos provenientes de cada componente de una fábrica, así como de los datos recopilados de productos, distribuidores y clientes (Luna-López et al., 2021).

Por lo tanto, uno de los usos esenciales del concepto de Industria 4.0 radica en el monitoreo de equipos o procesos, donde la recopilación de datos proporciona información relevante sobre la vida útil de los equipos, los niveles de carga y la detección de posibles

daños, incluyendo datos relacionados con el entorno y la atmósfera. Gracias a esta información, es posible llevar a cabo el monitoreo en tiempo real de equipos o máquinas (Peralta-Abarca et al., 2020). Esto permite la autogestión y la toma de decisiones inmediatas, como por ejemplo, la solicitud de un tanque de combustible cuando los niveles son bajos y se llena automáticamente cuando la capacidad está por debajo del 30% (Stock y Seliger, 2016) (Zawadzki y Żywicki, 2016).

Figura 2

Herramientas computacionales de la Industria 4.0.



Nota. Tomado de ¡Bienvenido a la Industria 4.0, la Cuarta Revolución Industrial!, por EVICITI Technologies, 2023, EVICITI (<https://www.evicit.com.mx/blog-post/bienvenido-a-la-industria-4-0-la-cuarta-revolucion-industrial/>)

Redes de Distribución Eléctrica

La red de distribución eléctrica se compone de un conjunto de elementos cuya función principal es transmitir energía desde los puntos de generación hasta los consumidores. En otras palabras, esta red abarca líneas de distribución primaria, transformadores de distribución, líneas de distribución secundaria, acometidas y medidores (Yebra, 2009).

Según la definición proporcionada por la Asociación Electrotécnica Argentina, una red eléctrica se puede conceptualizar como la fusión entre la infraestructura eléctrica convencional

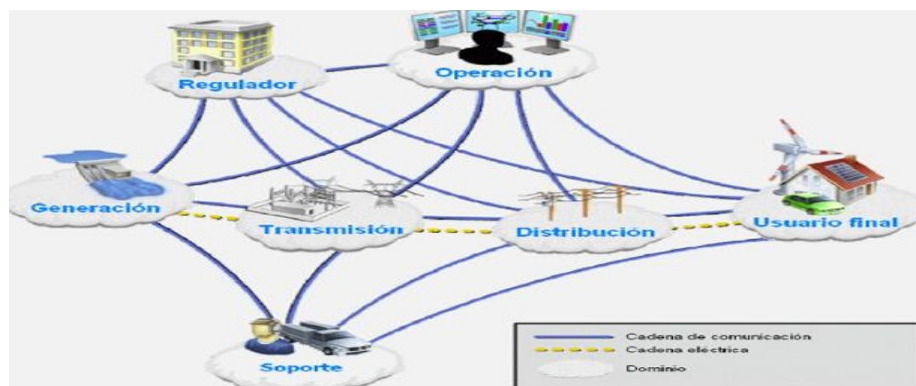
y las tecnologías modernas de información y comunicación. Esta integración permite la recopilación de datos desde diversos puntos de la cadena eléctrica, que abarca desde la generación de energía hasta su consumo por parte del usuario final. Estos datos se transforman en información y acciones destinadas a mejorar la gestión de la red eléctrica en su conjunto.

El propósito fundamental de una red eléctrica radica en elevar la eficiencia, la confiabilidad, la sostenibilidad, la calidad del servicio y del producto, con el fin de enfrentar los desafíos emergentes asociados a la presencia de múltiples generadores y variados patrones de consumo de energía.

Es esencial que la distribución de electricidad se realice mediante una red debidamente diseñada (Medina, 2020), capaz de adaptarse al crecimiento de la carga eléctrica y compuesta por componentes de alta calidad que puedan resistir los efectos de los campos eléctricos y las inclemencias del clima. Durante la prestación del servicio, es crucial que los clientes reciban una alimentación continua con un nivel de voltaje adecuado y una forma de onda sinusoidal limpia (Yebra, 2009).

Figura 3

Esquema de una red eléctrica inteligente.



Nota. Tomado de Ministerio de Energía y Minería, Argentina, Ing. Oscar Medina (www.energia.gov.ar).

Según (Ramírez, 2004), una red de distribución está conformado por:

- Subestaciones receptoras secundarias: La energía recibida de las líneas de subtransmisión se transforma y crea el circuito de distribución primaria.
- Circuitos primarios: el suministro de energía a transformadores de distribución en tensiones de 13 puntos 2 kV, 11 puntos 4 kV y 7620 voltios, cubriendo una serie de departamentos tanto del área urbana como rural.
- Transformadores de distribución: Disponen de conexión al circuito primario y ofrecen servicios a los usuarios o abonados del circuito secundario.
- Circuito secundario: Responsable de la distribución de electricidad a los consumidores en el rango 120/208 - 120/240 V y generalmente hasta 600 V.

Por otro lado, la creación del primer sistema de distribución eléctrica, ha existido la necesidad de medir y registrar la cantidad de energía consumida, lo que ha impulsado la investigación y desarrollo de diversos equipos y herramientas para llevar a cabo esta tarea. En 1882, se introdujo el primer medidor que operaba utilizando una relación de amperios-hora. Luego, en 1885, el científico italiano Galileo Ferraris inventó un dispositivo de conducción magnética que medía la energía en dos fases. Basándose en esta teoría, George Westinghouse diseñó un medidor electromecánico que utilizaba la relación de vatios por hora como principio de medición, y este se convirtió en el sistema de medición eléctrica más utilizado durante un largo período de tiempo (Banguera Gómez, 2020).

Sistemas de control en subestaciones eléctricas

Una subestación eléctrica se configura como un área centralizada dentro de una red de energía eléctrica, compuesta principalmente por terminales destinados al manejo y control de equipos, bloques de líneas de transmisión o distribución, así como transformadores. Además,

puede incluir otros dispositivos necesarios para garantizar la operación y seguridad del sistema (Florián y Gálvez, 2006).

(Gil Lostes y Bernal Agustín, 2011) se refiere a un conjunto de dispositivos con la capacidad de ajustar y controlar diversos parámetros de la potencia eléctrica, tales como tensión, corriente, frecuencia, entre otros. Estos dispositivos funcionan como un medio de interconexión entre las diversas líneas y componentes de un sistema eléctrico, permitiendo la regulación y gestión adecuada de la energía eléctrica dentro de dicho sistema.

La unidad central del sistema de control se encuentra ubicada en la sala de control de la subestación y supervisa las unidades técnicas digitales y numéricas que forman parte del sistema. La instalación y conexión segura de estos equipos en el gabinete es esencial, y el sistema abarca desde relés de señal que se colocan directamente en el equipo eléctrico hasta sistemas más complejos diseñados para el control de redes de nivel superior (Barrantes, 2011).

En las estaciones eléctricas, se emplean sistemas de control basados en computadora, lo que implica que el controlador se configura con el número adecuado de entradas y salidas analógicas y digitales correspondientes a las variables que se deben supervisar y controlar. Además, el controlador se encarga de la adquisición de datos y la ejecución de acciones de control en la planta automatizada (Florián y Gálvez, 2006).

Es importante destacar que una instalación eléctrica puede experimentar diversas fallas que, si no se atienden, pueden causar daños a los equipos eléctricos y electrónicos, inestabilidad en el sistema o incluso poner en peligro al personal que opera la instalación. Por lo tanto, es fundamental aislar la zona donde se produce la falla de manera rápida para prevenir daños en otros componentes de la instalación (Barrantes, 2011).

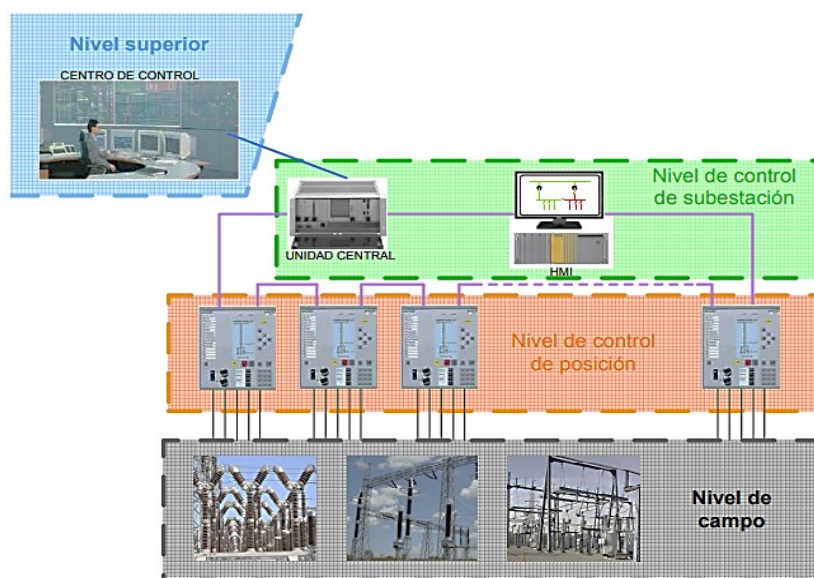
El sistema de control proporciona la capacidad de manipular los dispositivos y equipos de la subestación, pero para hacerlo de manera segura, necesita conocer constantemente el

estado de cada equipo (si está en posición abierta o cerrada) y recibir información antes de manipular cualquier dispositivo eléctrico (Barrantes, 2011).

Según (Barrantes, 2011), desde una perspectiva de control, una subestación eléctrica se divide en tres niveles, cada uno diseñado para satisfacer necesidades específicas de operación, como se ilustra en la Figura 4.

Figura 4

Niveles de control de una subestación eléctrica.



Nota. Tomado de Diseño del sistema de protección y control de subestaciones eléctricas, por Barrantes Lucía, 2011, Escuela Politécnica Superior de la Universidad Carlos III de Madrid.

Para las comunicaciones con el centro de control existen varios protocolos de comunicación, y la elección de uno u otro dependerá de los procedimientos habituales de la compañía eléctrica y sus necesidades de transmisión de información. (Barrantes, 2011). La mayoría de veces se utilizan protocolos no propietarios IEC (International Electro-technical Commission) para incorporar sistemas de varios fabricantes y facilitar la extensión o actualización del sistema de ser necesario (Barrantes, 2011).

Protocolos de comunicación industriales

La comunicación de datos entre sistemas, procesos e instalaciones se erige como uno de los pilares esenciales para que las empresas modernas puedan alcanzar el nivel de competitividad requerido (Valdivia Miranda, 2019). Los protocolos de comunicación en el ámbito industrial han evolucionado a partir de protocolos más antiguos que se basaban en conexiones serie, y también han surgido como resultado de la creación de nuevos estándares destinados a tecnologías innovadoras (Estrada Roque, 2017).

En esencia, los protocolos de comunicación industrial consisten en un conjunto de reglas que posibilitan el intercambio de datos y la interacción entre diversos dispositivos que están conectados en una red (Estrada Roque, 2017). En otras palabras, estos protocolos habilitan la adquisición de datos, la supervisión y el control de las variables presentes en una planta o proceso industrial, generalmente desde un único punto de control, como un panel o sala de control central (Corrales, 2007).

La conversión de protocolo constituye una herramienta empleada con el propósito de facilitar la integración de múltiples dispositivos en una red de comunicaciones. Esta capacidad permite la ejecución de diversas funciones, como la supervisión de variables, la gestión remota y las operaciones de control. Estas operaciones pueden llevarse a cabo tanto desde un centro de control centralizado como desde varias ubicaciones remotas, según los requisitos específicos del sistema y el nivel de complejidad previsto en las fases de diseño. Esto se documenta detalladamente en el trabajo de referencia de (Gómez Tapias y Vargas Torres, 2011), donde se expone con minuciosidad las consideraciones de ingeniería relacionadas con sistemas de control.

Es relevante subrayar que, en el contexto de las instalaciones de subestaciones eléctricas, se encuentran implementados diversos protocolos en los dispositivos de protección y control. Algunos de estos protocolos son creados internamente por los fabricantes, como lo

ejemplifica (Alcantara Fernandez, 2020) en su serie de relés MiCOM, que utiliza los protocolos Courier/K-Bus, diseñados específicamente para esta línea de productos. Por otro lado, los modelos establecidos por algunos grupos de clientes, como DNP V3.0 (Distributed Network Protocol Version 3), son aceptados por algunos fabricantes con el propósito de garantizar una integración fluida de sus productos en redes de comunicación que potencialmente puedan incluir sistemas de protección y control de diversos fabricantes.

En el presente estudio, se plantean dos normas que son muy reconocidas en el sector eléctrico: MODBUS TCP/IP, que fue creada por la organización MODBUS, e IEC 60870-5-104, que en cambio fue creada por la Comisión Electrotécnica Internacional. Estos estándares son ampliamente utilizados en subestaciones eléctricas. La arquitectura propuesta para el desarrollo del algoritmo se basa en un enfoque maestro-esclavo, como se ilustra en la Figura 5 (MODBUS Group, 2006), para las conexiones entre maestro-gateway y gateway-esclavo el medio físico va a ser Ethernet 10/100 a través del uso de cable UTP de cobre.

Figura 5

Arquitectura de control y supervisión.



Nota. Tomado de MODBUS Group, Modbus, por MODBUS Group, 2006

(http://modbus.org/about_us.php).

Es innegable que el cambio climático, la continua y creciente explotación de los recursos naturales del planeta han obligado a las nuevas generaciones a reconsiderar sus acciones. La disponibilidad finita de los combustibles fósiles y sus efectos perjudiciales en términos de contaminación y daño a la capa de ozono hacen imperativo tomar medidas para mitigar o reducir el impacto ambiental.

Según (Ramos Guardarrama et al., 2019), un proyecto de sistema de comunicación debe tener en cuenta los siguientes componentes importantes:

1. La milla final contiene el mayor número de nodos y la mayor dificultad de conexiones.
2. Se debe realizar una evaluación de la capacidad física de los equipos terminales de medición, así como de las barreras arquitectónicas que obstaculizan su despliegue y funcionamiento.
3. La adaptabilidad técnica es esencial para acoplarse a una variedad de condiciones y equipos preexistentes.
4. El sistema debe ser capaz de gestionar un alto volumen de datos y un gran número de terminales de manera eficiente.
5. Se requiere un bajo consumo de energía para mantener la operación sostenible.
6. La distancia de transmisión y recepción debe ser una consideración esencial en la planificación.

En contextos menos desarrollados, los costos operativos pueden ser prohibitivos al utilizar la infraestructura de comunicaciones estándar de Internet para conectar los nodos de la última milla. La implementación de concentradores de red que disminuyan la dificultad de la comunicación bidireccional en lugar de aumentar la cantidad de nodos presentes en este equipamiento es una solución viable.

Por otro lado, el protocolo IEC 61850 es ampliamente reconocido como el estándar líder en la automatización de equipos utilizados en subestaciones eléctricas, independientemente del fabricante de dichos equipos. Este protocolo se destaca por su capacidad para garantizar la interoperabilidad entre una variedad de dispositivos, ofrecer modelos de datos extensos y una amplia gama de servicios, así como ser compatible con las tecnologías de comunicación Ethernet y TCP/IP.

En resumen, el IEC 61850 se ha convertido en una norma esencial en la industria eléctrica, permitiendo una mayor eficiencia y eficacia en la operación, control de subestaciones eléctricas al facilitar la comunicación y la integración de equipos de diferentes fabricantes (Fuentes, 2005)

Protocolos de comunicación industriales MODBUS

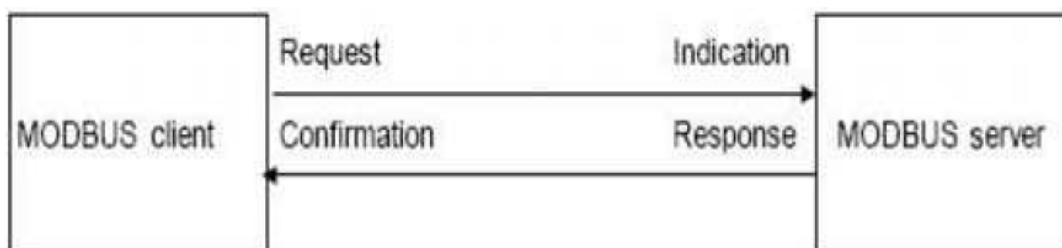
Descripción general. Modbus es un protocolo para la comunicación de capas de aplicación (capa 7 de Open Systems Interconnection OSI) que permite la comunicación cliente-servidor entre dispositivos conectados a varios tipos de buses de red. El protocolo Modbus implementa una arquitectura cliente/servidor y opera esencialmente en un modo de solicitud/respuesta, independientemente del control de acceso a los medios utilizado en la capa 2. Este modelo cliente/servidor se basa en cuatro tipos de mensajes, a saber:

- Solicitudes Modbus, los mensajes enviados en la red por los clientes para iniciar transacciones.
- Confirmaciones Modbus, los mensajes de respuesta recibidos en el lado del cliente.
- Indicaciones Modbus, los mensajes de solicitud recibidos en el lado del servidor.
- Respuestas Modbus, los mensajes de respuesta enviados por los servidores.

Estos servicios de mensajería del modelo cliente/servidor se utilizan para intercambiar información en tiempo real entre dos aplicaciones de dispositivos, entre aplicaciones de dispositivos y dispositivos, o entre dispositivos y aplicaciones HMI/SCADA, como se presenta en la Figura 6.

Figura 6

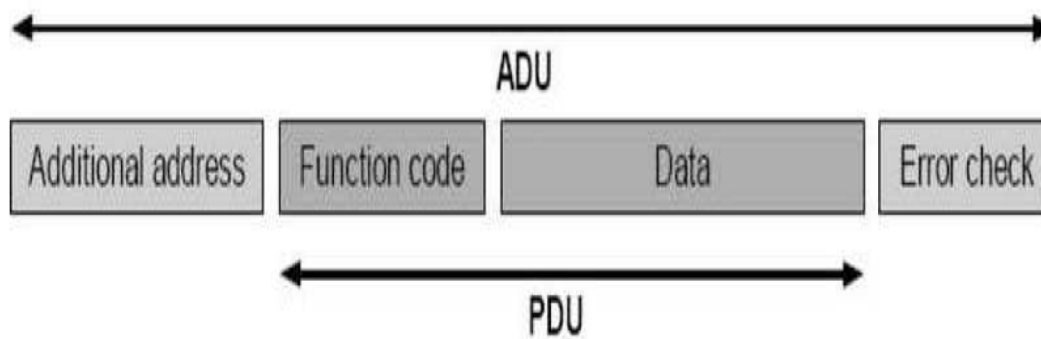
Interacción maestro/esclavo Modbus.



En un escenario sin errores, el intercambio de información entre el cliente y el servidor se puede ilustrar de la siguiente manera. El cliente (en el dispositivo maestro) inicia una solicitud. Después, el protocolo de comunicación MODBUS (capa 7) crea una unidad de datos de protocolo, también conocida como PDU, que incluye un código de función y una solicitud de datos. Esta PDU se transforma en una unidad de datos de aplicación (ADU) en la capa 2 agregando campos que tienen que ver con el bus o la red, como una dirección esclava y una suma de comprobación para encontrar errores. Este proceso se representa en la Figura 7.

Figura 7

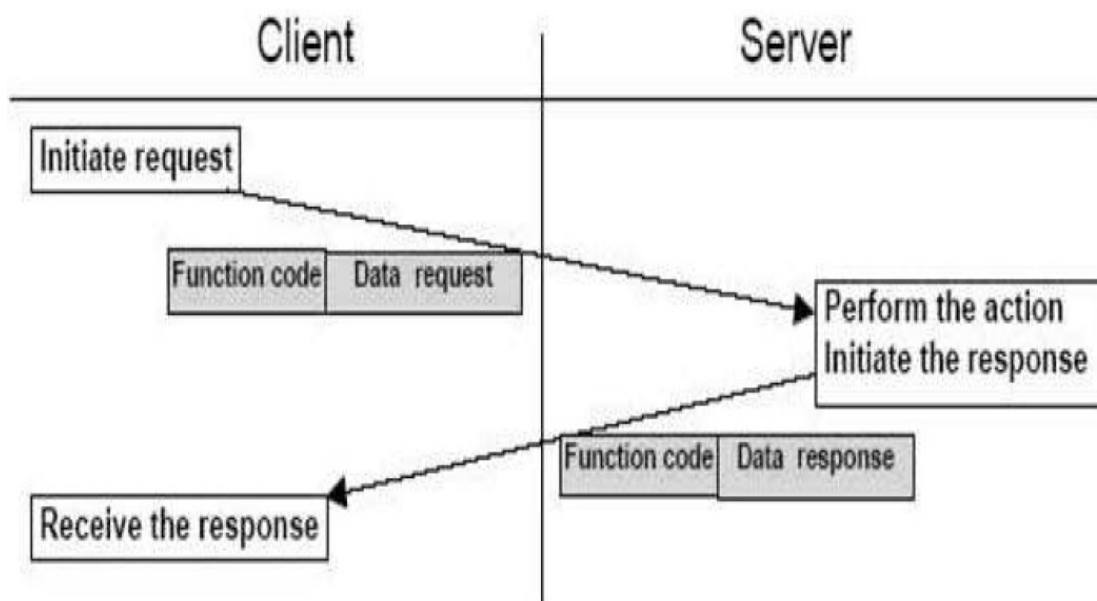
Marco Modbus genérico.



El servidor (en el dispositivo esclavo) luego realiza la acción requerida e inicia una respuesta. La interacción entre el cliente y el servidor se muestra en la Figura 8.

Figura 8

Transacción Modbus.

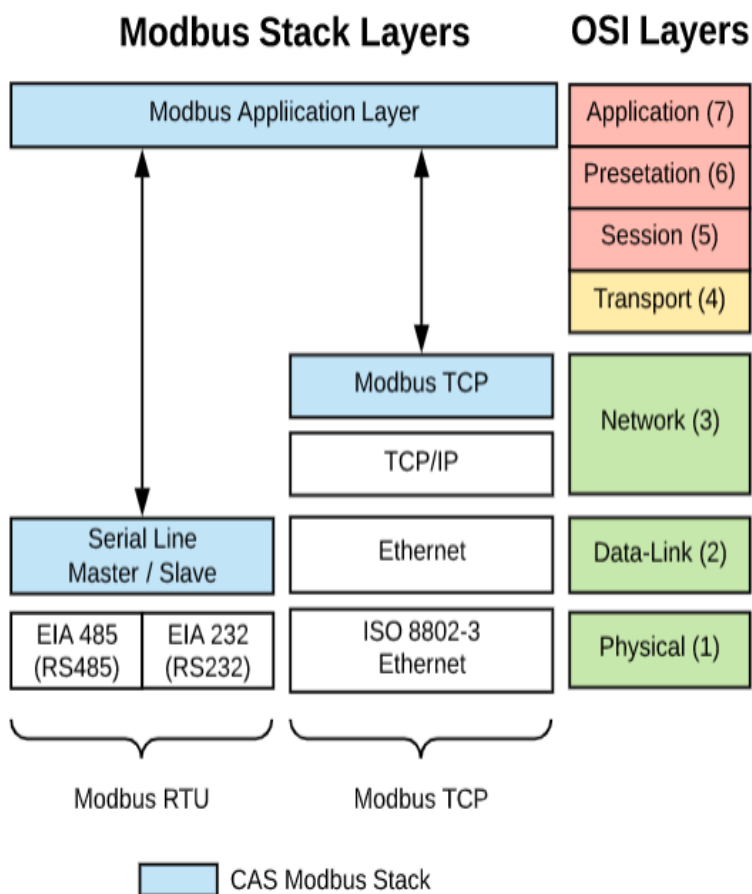


Los diversos tipos de códigos de función, con sus solicitudes y respuestas asociadas, se describen en detalle en este capítulo.

El protocolo de mensajería MODBUS (capa 7) necesita soporte adicional en las capas inferiores para transmitir el mensaje. Un método popular es el uso de un protocolo de capa 2 maestro/esclavo (semidúplex), que transmite los datos en formato serie a través de enlaces de módem tipo RS-232, RS-485 o Bell 202. El enfoque TCP/IP permite la interacción cliente/servidor a través de redes enrutadas, aunque a costa de gastos generales adicionales (tiempo de procesamiento, encabezados, etc.). Se requiere una subcapa adicional para asignar la capa de aplicación MODBUS a TCP. La función de esta subcapa es encapsular la PDU MODBUS para que pueda ser transportada como un paquete de datos por TCP/IP, como se observa en la Figura 9.

Figura 9

Pila de comunicación Modbus.



Estructura del protocolo Modbus. La Tabla 1 ilustra un formato de trama de mensaje Modbus típico.

Tabla 1

Formato del marco de mensaje Modbus.

Campo de dirección	Campo de función	Campo de datos	Campo de verificación de errores
1 byte	1 byte	Variable	2 bytes

El primer campo de cada trama de mensaje es el campo de dirección, que consta de un solo byte de información. En las tramas de solicitud, este byte identifica el controlador al que se dirige la solicitud. La trama de respuesta resultante comienza con la dirección del dispositivo que responde. Cada esclavo puede tener un campo de dirección entre 1 y 247, aunque las limitaciones prácticas limitarán el número máximo de esclavos. Una instalación Modbus típica tendrá un maestro y dos o tres esclavos.

El segundo campo de cada mensaje es el campo de función, que también consta de un solo byte de información. En una solicitud de host, este byte identifica la función que debe realizar el PLC de destino.

Si el PLC de destino puede realizar la función solicitada, el campo de función de su respuesta reflejará el de la solicitud original. De lo contrario, el campo de función de la solicitud se repetirá con su bit más significativo establecido en uno, lo que indica una respuesta de excepción. La Tabla 1 resume las funciones típicas utilizadas.

El tercer campo en un marco de mensaje es el campo de datos, que varía en longitud según la función especificada en el campo de función. En una solicitud de host, este campo contiene información que el PLC puede necesitar para completar la función solicitada. En una respuesta de PLC, este campo contiene cualquier dato solicitado por ese host.

Los últimos dos bytes en un marco de mensaje comprenden el campo de verificación de errores. El valor numérico de este campo se calcula realizando una verificación de redundancia cíclica (CRC-16) en la trama del mensaje. Esta verificación de errores asegura que los dispositivos no reaccionen a los mensajes que pueden haberse dañado durante la transmisión.

La Tabla 2 enumera el rango de direcciones y las compensaciones para estos cuatro tipos de datos, así como los códigos de función que se aplican a cada uno. El diagrama de la Tabla 1 también proporciona una referencia sencilla a los tipos de datos Modbus.

Tabla 2*Direcciones Modicon y códigos de función.*

Tipo de datos	Direcciones absolutas	Direcciones relativas	Códigos de Función	Descripción
Bobinas	00001–09999	0–9998	01	Leer estado de la bobina
Bobinas	00001–09999	0–9998	05	Forzar bobina simple
Bobinas	00001–09999	0–9998	15	Forzar múltiples bobinas
Entradas discretas	10001–19999	0–9998	02	Leer estado de entrada
Registros de entrada	30001–39999	0–9998	04	Leer registros de entrada
Registros de espera	40001–49999	0–9998	03	Leer registro de espera
Registros de retención	40001–49999	0–9998	06	Registro único preestablecido
Registros de retención	40001–49999	0–9998	16	Registros múltiples preestablecidos
-	-	-	07	Estado de excepción de lectura
-	-	-	08	Prueba de diagnóstico de bucle invertido

Códigos de función. Cada trama de solicitud contiene un código de función que define la acción esperada para el controlador de destino. El significado de los campos de datos de solicitud depende del código de función especificado.

Los siguientes párrafos definen e ilustran los códigos de función populares. En estos ejemplos, el contenido de los campos de la trama del mensaje se muestra como bytes hexadecimales.

Leer estado de bobina o salida digital (código de función 01). Esta función permite que el host obtenga el estado ON/OFF de una o más bobinas lógicas en el dispositivo de destino.

El campo de datos de la solicitud consiste en la dirección relativa de la primera bobina seguida por el número de bobinas a leer. El campo de datos de la trama de respuesta consta de un recuento de los bytes de la bobina seguidos de esa cantidad de bytes de datos de la bobina.

Los bytes de datos de la bobina se empaquetan con un bit para el estado de cada bobina consecutiva (1 = ENCENDIDO, 0 = APAGADO). El bit menos significativo del primer byte de datos de la bobina transmite el estado de la lectura de la primera bobina. Si el número de bobinas leídas no es un múltiplo par de ocho, el último byte de datos se completará con ceros en el extremo superior. Si se solicitan varios bytes de datos, el bit de orden inferior del primer byte de datos en la respuesta del esclavo contiene la primera bobina direccionada.

En la Figura 10, el host solicita el estado de las bobinas 000A (00011 decimal) y 000B (00012 decimal). La respuesta del dispositivo objetivo indica que ambas bobinas están encendidas.

Figura 10

Ejemplo de estado de la bobina de lectura.

Request Message						
Address	Function Code	Initial Coil Offset		Number of Points		CRC
		Hi	Lo	Hi	Lo	
01	01	00	0A	00	02	9D C9

Response Frame				
Address	Function Code	Byte Count	Coil Data	CRC
01	01	01	03	11 89

Leer estado de entrada digital (código de función 02). Esta función permite que el host lea una o más entradas discretas en el dispositivo de destino.

El campo de datos de la trama de solicitud consta de la dirección relativa de la primera entrada discreta seguida del número de entradas discretas que se leerán. El campo de datos de la trama de respuesta consta de un recuento de los bytes de datos de entrada discretos seguidos de esa cantidad de bytes de datos de entrada discretos.

Los bytes de datos de entrada discreta se empaquetan con un bit para el estado de cada entrada discreta consecutiva (1 = ENCENDIDO, 0 = APAGADO). El bit menos significativo del primer byte de datos de entrada discreta transmite el estado de la primera lectura de entrada. Si el número de entradas discretas leídas no es un múltiplo par de ocho, el último byte de datos se completará con ceros en el extremo superior.

El bit de orden inferior del primer byte de la respuesta del esclavo contiene la primera entrada digital direccionada. En la Figura 11, el host solicita el estado de las entradas discretas

relativas 0001 y 0002 hexadecimales, es decir, decimal las absolutas 10000 y 10001. La respuesta del dispositivo de destino indica que la entrada discreta 10000 está APAGADA y 10001 están ENCENDIDA.

Figura 11

Ejemplo de estado de entrada de lectura.

Request Message

Address	Function Code	Initial Coil Offset	Number of Points	CRC
		Hi Lo	Hi Lo	
01	02	00 00	00 02	F9 CB

Response Frame

Address	Function Code	Byte Count	Input Data	CRC
01	02	01	02	20 49

Lectura de registros de espera (código de función 03). Esta función permite que el host obtenga el contenido de uno o más registros de retención en el dispositivo de destino.

El campo de datos de la trama de solicitud consta de la dirección relativa del primer registro de retención seguido del número de registros que se leerán. El campo de datos del tiempo de respuesta consta de un recuento de los bytes de datos de registro seguidos de esa cantidad de bytes de datos de registro de retención.

El contenido de cada registro solicitado (16 bits) se devuelve en dos bytes de datos consecutivos (primero el byte más significativo). En la Figura 12, el host solicita el contenido del registro de retención con desplazamiento hexadecimal 0002 o decimal 40003. La respuesta del

controlador indica que el valor numérico del contenido del registro es hexadecimal 07FF o decimal 2047. El primer byte de los datos del registro de respuesta es el byte de orden superior del primer registro direccionado.

Figura 12

Ejemplo de registro de espera de lectura.

Request Message				
Address	Function Code	Starting Register	Register Count	CRC
		Hi Lo	Hi Lo	
01	03	00 02	00 01	25 CA

Response Frame				
Address	Function Code	Byte Count	Register Data	CRC
			Hi Lo	
01	03	02	07 FF	FA 34

Lectura de registros de entrada (código de función 04). Esta función permite que el host obtenga el contenido de uno o más registros de entrada en el dispositivo de destino. El campo de datos de la trama de solicitud consta de la dirección relativa del primer registro de entrada seguido del número de registros que se leerán. El campo de datos de la trama de respuesta consta de un recuento de los bytes de datos de registro seguidos de esa cantidad de bytes de registro de datos de entrada.

El contenido de cada registro solicitado se devuelve en dos bytes de datos de registro consecutivos (primero el byte más significativo). El rango para las variables de registro es 0–4095. En la Figura 13, el host solicita el contenido del registro de entrada con desplazamiento hexadecimal 000 o decimal 30001. La respuesta del PLC indica que el valor numérico del

contenido de ese registro es 03FFH, lo que correspondería a un valor de datos del 25% (si la escala de 0 a 100 %) y se utiliza un convertidor de D a A de 12 bits con una lectura máxima de 0FFFH.

Figura 13

Ejemplo de registro de entrada de lectura.

Request Message						
Address	Function Code	Starting Register		Register Count		CRC
		Hi	Lo	Hi	Lo	
01	04	00	00	00	01	31 CA

Response Frame					
Address	Function Code	Byte Count	Register Data		CRC
			Hi	Lo	
01	04	02	03	FF	F9 80

Forzar bobina simple (código de función 05). Esta función permite que el host altere el estado ON/OFF de una sola bobina lógica en el dispositivo de destino.

La dirección de la bobina correspondiente y el estado deseado de la bobina forman el campo de datos de la trama de solicitud. La bobina se activará con un valor de estado hexadecimal de FF00, mientras que la desactivará con un valor de estado hexadecimal de 0000H. Si no son estos valores de estado serán ilegales.

El marco de respuesta será idéntico al estado solicitado si el controlador puede forzar la bobina especificada al estado deseado. De lo contrario, se devolverá una respuesta de excepción. Si se usa la dirección 00 para indicar el modo de transmisión, todos los esclavos

conectados modificarán la dirección de bobina especificada al estado requerido. La Figura 14 ilustra un intento exitoso de forzar la desactivación de la bobina 11 (decimal).

Figura 14

Ejemplo de forzado de una sola bobina.

Request Message				
Address	Function Code	Coil Offset	New Coil Status	CRC
		Hi Lo	Hi Lo	
01	05	00 0A	00 00	ED C8

Response Frame				
Address	Function Code	Coil Offset	New Coil Status	CRC
		Hi Lo	Hi Lo	
01	05	00 0A	00 00	ED C8

Registro único preestablecido (código de función 06). Esta función permite que el host modifique el contenido de un solo registro de retención en el dispositivo de destino.

El campo de datos de la trama de solicitud consta de la dirección relativa del registro de retención seguido del nuevo valor que se escribirá en ese registro (primero el byte más significativo).

Si el controlador puede escribir el nuevo valor solicitado en el registro especificado, el marco de respuesta será idéntico a la solicitud. De lo contrario, se devolverá una respuesta de excepción. La Figura 15 ilustra un intento exitoso de cambiar el contenido del registro de retención 40003 a 3072 (0C00 Hex). Cuando la dirección del esclavo se establece en 00 (modo de transmisión), todos los esclavos cargarán el registro especificado con el valor especificado.

Figura 15

Ejemplo de preselección de un único registro.

Request Message					
Address	Function Code	Register Offset		Register Value	CRC
		Hi	Lo	Hi Lo	
01	06	00	02	0C 00	2D 0A

Response Frame					
Address	Function Code	Register Offset		Register Value	CRC
		Hi	Lo	Hi Lo	
01	06	00	02	0C 00	2D 0A

Forzar múltiples bobinas o salidas digitales (código de función 0F). Esto obliga a un grupo contiguo (o adyacente) de bobinas a un estado de ENCENDIDO o APAGADO. El siguiente ejemplo establece 10 bobinas que comienzan en la dirección 01 hexadecimal (en la dirección esclava 01) en el estado ON. Si se utiliza la dirección de esclavo 00 en la trama de solicitud, se implementará el modo de transmisión, lo que hará que todos los esclavos cambien sus bobinas en las direcciones definidas, como se observa en la Figura 16.

Figura 16

Ejemplo de forzar varias bobinas.

Request Frame						
Address	Function Code	Address		Byte Count	Data Coil Status	CRC
		Hi	Lo		Hi Lo	
01	0F	00	01	0F	FF 03

Response Frame					
Address	Function Code	Address		Number of Coils	CRC
		Hi	Lo	Hi Lo	
01	0F	00	01	00 0A

Forzar registros múltiples (código de función 10). Esto es similar al preajuste de un solo registro y al forzado de múltiples bobinas. En la Figura 17, una dirección de esclavo 01 tiene 2 registros cambiados comenzando en la dirección 10.

La Tabla 3 enumera los códigos de excepción más importantes que pueden devolverse.

Figura 17

Ejemplo de preajuste de varios registros.

Request Frame							
Address	Function Code	Address		Byte Count	Data Coil Status		CRC
		Hi	Lo		Hi	Lo	
01	0F	00	01	0F	FF	03

Response Frame					
Address	Function Code	Address		Number of Coils	CRC
		Hi	Lo		
01	0F	00	01	00 0A

Tabla 3

Lista abreviada de códigos de excepción devueltos.

Código	Nombre	Descripción
01	Función no válida	La función solicitada no es compatible
02	Dirección de datos ilegal	La dirección de datos solicitada no es compatible
03	Valor de datos ilegal	El valor de datos especificado no es compatible
04	Fallo en el dispositivo asociado	El esclavo no ha podido responder a un mensaje
05	Reconocer	El esclavo está procesando el comando

Código	Nombre	Descripción
06	Ocupado, mensaje rechazado	El esclavo está ocupado

En la Figura 18 se muestra un ejemplo de una solicitud ilegal y la respuesta de excepción correspondiente. La solicitud en este ejemplo es LEER EL ESTADO DE LA BOBINA de los puntos 514 a 521 (ocho bobinas que comienzan con un desplazamiento 0201H). Estos puntos no son soportados en este, por lo que se genera un reporte de excepción indicando el código 02 (dirección ilegal).

Figura 18

Ejemplo de una solicitud ilegal.

Request Message

Address	Function Code	Starting Point	Number of Points	CRC
01	01	02 01	00 08	6D B4

Exception Response Message

Address	Function Code	Exception Code	CRC
01	81	02	C1 91

ModbusTCP

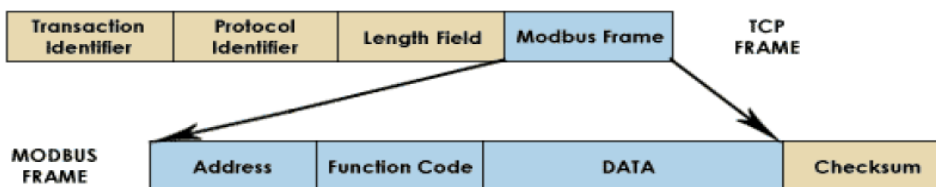
Trama del Protocolo Modbus. La estructura que se forma dentro de un protocolo para enviar o recibir datos, establecer la dirección origen y destino y la cantidad de datos a transmitir, se denomina trama, y está dividida en campos y el tamaño de cada uno. Para comunicarse Modbus TCP utiliza el puerto 502 ethernet. La estructura muestra la trama general del protocolo; y está formada de los siguientes campos:

- ID, es la dirección del esclavo y son números enteros que van del 1 hasta 255, esto identifica que el protocolo puede comunicarse con 255 esclavos diferentes.
- FUNCIÓN, identifica tipo de solicitud que realiza al esclavo y se codifica con números y se identifica por un dato numérico.
- DATO, este campo se utiliza para proporcionar información adicional a la solicitud enviada al esclavo o para responder a la solicitud, por lo que tiene una estructura diferente cuando el mensaje lo envía el maestro o cuando es la respuesta del esclavo.

Trama Maestro. El campo de datos del maestro se encuentra compuesto por dos subcampos: dirección y longitud. La dirección es la encargada de mostrar al esclavo en qué dirección debe buscar lo que se ha requerido mediante la función, y la longitud es la encargada de mostrar cuantos elementos se deben tomar a partir de esa dirección.

Figura 19

Estructura de la trama de un maestro Modbus.



Trama esclavo. Aunque la estructura sigue siendo la misma para el esclavo, se produce un cambio en el campo de Dato, ya que se agregan dos subcampos: el número de bytes suficientes para proporcionar la respuesta y la propia respuesta.

Tabla 4

Estructura de Solicitud.

Nombre	Bytes	Función
Identificador de transacciones	0	Para sincronización

Nombre	Bytes	Función
Identificador de transacciones	1	
Identificador de protocolo	2	Cero para Modbus/TCP
Identificador de protocolo	3	
Longitud	4	Número de bytes restantes en
Longitud	5	esta trama
Estación	6	Dirección de la estación
Función	7	Código de función
Dirección MSB	8	
Dirección LSB	9	Dirección inicial
Cantidad MSB	10	
Cantidad LSB	11	Cantidad

Tabla 5*Estructura de Respuesta.*

Nombre	Bytes	Función
Identificador de transacciones	0	
Identificador de transacciones	1	Para sincronización
Identificador de protocolo	2	Cero para Modbus/TCP
Identificador de protocolo	3	
Longitud	4	Número de bytes restantes
Longitud	5	en esta trama
Estación	6	Dirección de la estación

Nombre	Bytes	Función
Función	7	Código de función
Bytes e datos	8	Bytes de datos
Datos	9...	Datos

Ventajas de uso del protocolo Modbus/TCP. Las siguientes son algunas de las ventajas de uso de este protocolo:

- La expansión y la gestión son sencillas. El agregar Esclavos a la red Modbus, no implica la utilización de recursos de software ni hardware sofisticados.
- No se necesita de softwares o hardware propio de monitoreo, supervisión o configuración. Modbus/TCP se puede emplear en cualquier computadora que utilice o tenga un software compatible con protocolos TCP/IP.
- Dispone en el mercado de diferentes equipos de conversión o puentes para Modbus TCP, los mismos que no necesitan configuración.
- Su simplicidad y tamaño de trama lo hace muy rápido, por lo que puede alcanzar altas tasas de transmisión, garantizando tiempos de respuesta en milisegundos, lo que se considera tiempo real.
- El ingeniero de mantenimiento puede acceder, de manera inmediata, al sistema de control de la planta desde el confort de su hogar.
- Permite administrar sistemas distribuidos geográficamente utilizando las tecnologías de Internet e Intranet actuales.

Por ejemplo, si la trama Modbus TCP del esclavo es: 0001 0000 0006 11 03 006B 0003

- 01, identificador de operación: Leer estado de una bobina.

- 00, identificador de protocolo: Modbus TCP.
- 0006, longitud del mensaje: 6 bytes desde el siguiente byte.
- 11, identificador de unidad (17 = 11hex).
- 03, código de Función: Leer registros de retención de salida analógica.
- 006B, la dirección de datos del primer registro solicitado: 40108-40110 = 107 =6Bhex.
- 03, el número total de registros solicitados: Leer 3 registros 40108 a 40110.

Protocolos de comunicación industriales de red IEC 60870-5-104

La normativa IEC 60870-5-104 fue desarrollada tomando como base estándares previamente establecidos, tipos de datos, métodos de transferencia de información y técnicas de administración de tramas que habían sido previamente delineados en las normas IEC 60870-101, IEC 60870-5-102 e IEC 60870-5-103. Como resultado, esta norma incluye una amplia gama de funcionalidades y procedimientos que permiten la comunicación de sucesos vinculados con la protección, el restablecimiento y las irregularidades en los valores de las variables que son registradas por los sistemas de protección y control del sistema eléctrico.

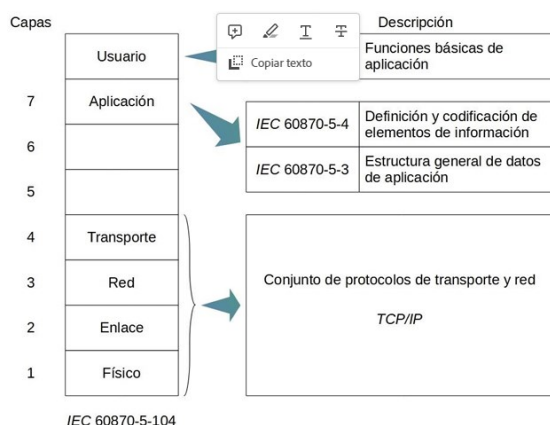
Una de las tareas que llevan a cabo los dispositivos de traducción de protocolo es la elección y consolidación de las variables de interés, ajustándose a las especificidades del sistema en cuestión. Esto se debe a que, con frecuencia, los dispositivos de protección proporcionan conjuntos de datos que no siempre son pertinentes para las operaciones en curso. Si se incluyeran todas las variables suministradas por el fabricante, existe la posibilidad de que se agoten las licencias de los sistemas SCADA de manera rápida, ya que la integración de solo unos pocos dispositivos podría ser suficiente para alcanzar dicho límite.

La representación precisa de la arquitectura del protocolo IEC 101 se ajusta a un modelo compuesto por cuatro capas. Sin embargo, en el caso del protocolo IEC 104, se hace necesario incorporar capas adicionales del modelo OSI con el propósito de facilitar la transmisión de mensajes a través de la red. Estas capas adicionales incluyen la capa de red, que hace posible la transmisión de información a través de múltiples redes por intermedio del protocolo IP (Internet Protocol), y la capa de transporte, que permite la transmisión de información en ambas direcciones mediante el protocolo TCP (Transmission Control Protocol).

La relación entre las secciones que conforman la parte 5 de la norma y la arquitectura del protocolo IEC 104 se representa de manera gráfica en la Figura 20.

Figura 20

Arquitectura del protocolo IEC 60870-5-104.



Arquitectura del protocolo. En cuanto a DNP3, IEC 60870-5 está fundamentada en el modelo EPA para las comunicaciones de datos, también conocido como arquitectura de rendimiento mejorado de tres capas, que exhibe cómo el modelo EPA es una forma sintetizada del OSI de siete capas.

En la Figura 21 se muestra cómo se relacionan el modelo EPA y el modelo OSI. Esto muestra que el modelo EPA básicamente omite las capas de presentación, sesión y transporte del modelo OSI.

La estructura del modelo EPA es apropiada para un sistema operativo continuo que opera en una sola red. Una capa que normalmente se agrega a la parte superior de la representación básica del modelo EPA se identifica como la capa de usuario. Esto se incluye para representar los diversas funciones o procesos que deben definirse para proporcionar operaciones del sistema de telecontrol. Estos deben definirse para permitir la interoperabilidad entre equipos, lo que resultará en un sistema de telecontrol totalmente operativo en lugar de solo un sistema de comunicación de datos.

Un modelo de cuatro capas, como se muestra en el lado derecho de la Figura 22, proporciona una representación precisa de la arquitectura del protocolo para el primer perfil compañero estándar IEC 60870-5-101 o T101 definido. Es necesario agregar capas adicionales al modelo OSI en el caso de la versión en red IEC 60870-5-104, o el perfil T104.

Envía mensajes a través de redes mediante protocolos de red comunes. Estas son las capas de transporte y red que se utilizan para los protocolos TCP e IP. La Figura 22 muestra las dos arquitecturas

Figura 21

Relación del modelo EPA con el modelo OSI de siete capas.

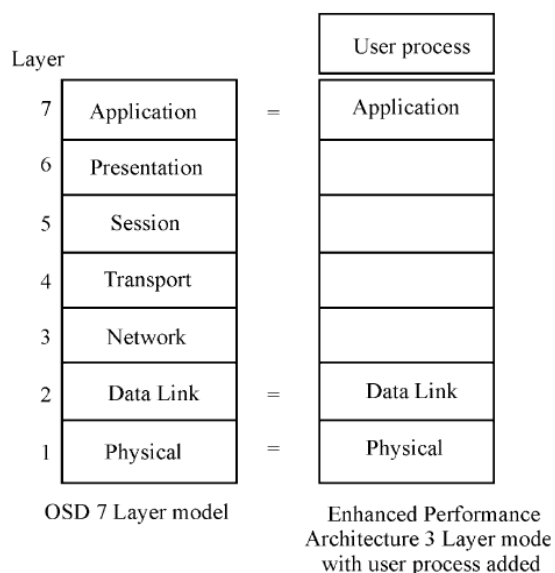
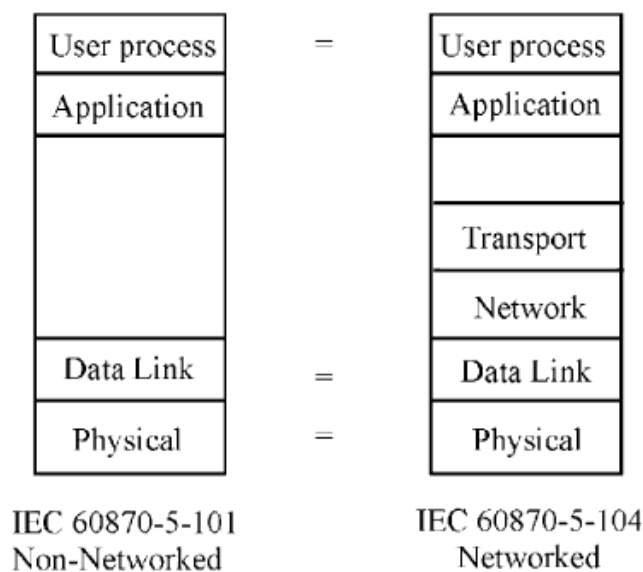


Figura 22

Arquitecturas para T101 y T104.



Como se desprende claramente de la Figura 22, el funcionamiento de las capas inferiores de la versión en red, IEC 60870-5-104, es completamente diferente al de la versión sin red, IEC 60870-5-101. Estas capas corresponden a todas las capas por debajo de la capa de aplicación, que para estas arquitecturas son las capas relacionadas con el transporte de mensajes.

Selecciones de estándares. El beneficio del modelo de referencia OSI y los modelos EPA derivados de él y que se muestran en la Figura 22 es que proporcionan una trama para la descripción del funcionamiento del protocolo.

Describir el funcionamiento del protocolo es cuestión de especificar las funciones de cada capa y especificar la estructura de la información que pasa entre las capas.

Según el estándar IEC, es el estándar complementario IEC 60870-5-101 el que especifica completamente el protocolo. Hace esto refiriéndose a las secciones principales del estándar IEC 60870-5 y haciendo selecciones particulares de las opciones que pueden estar disponibles dentro de esas secciones.

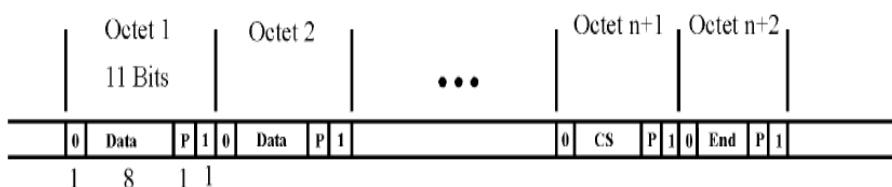
Formato de trama. El formato de trama utilizado por T101 se denomina formato FT1.2. Hay dos formas de esto, una de longitud fija y otra de longitud variable. La trama de longitud fija está restringida al uso de tramas que no transportan datos de usuario y, por lo tanto, se usa solo para tramas de acuse de recibo y comando de control de enlace de datos. Además de las tramas de longitud fija y variable, hay una trama de "carácter de control único", que consta de un solo byte. Esto puede utilizarse únicamente para el reconocimiento.

Estas tramas se muestran en las Figura 23 y Figura 24. La Figura 23 muestra el bit real patrón que se vería en el canal físico, interpretado de izquierda a derecha. Esto la representación incluye los bits de inicio y parada que se transmiten con cada byte u octeto de trama. Por lo tanto, el primer octeto se muestra en la parte superior y los siguientes octetos se muestran a continuación.

La aclaración del orden de bits y octetos se incluye en la siguiente sección.

Figura 23

Representación de secuencia de bits de la trama FT1.2.



Per 8 bit octet

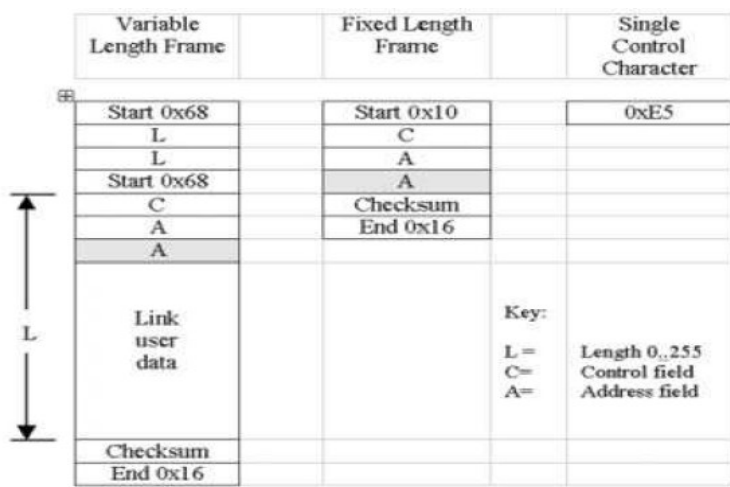
- 1 Start bit = 0
- 8 Data bits
- 1 Parity bits even parity
- 1 Stop bit = 1

Per - Frame

- Checksum modulo 256
- End character

Figura 24

Opciones de trama FT1.2 según IEC6087-5-101.



A partir de la representación de la secuencia de bits en la Figura 23, es posible ver que la velocidad máxima de datos de la trama es aproximadamente ocho undécimos de la velocidad de transmisión de bits. Esto se reduce aún más cuando se tienen en cuenta los gastos generales de la trama, como el direccionamiento, los caracteres de inicio y fin, la suma de comprobación y la información de control. Estos gastos generales se pueden ver en la Figura 24, que muestra la estructura general de la trama.

Se observan los siguientes puntos sobre las tramas de enlace de datos:

- Solo la trama de longitud variable puede transportar datos de usuario.
- La cantidad de octetos de datos de usuario de enlace que puede transmitir la trama de longitud variable es de hasta 253.
- Para que la trama sea válida, es necesario que la longitud L se repita dos veces y los dos valores de L sean iguales.

La longitud máxima de trama es de 261 octetos. Sin embargo, el fabricante o el usuario del sistema pueden especificar una longitud de trama máxima inferior como parámetro del sistema.

- La trama de longitud fija tiene una longitud de 5 o 6 octetos.
- El campo de dirección A puede tener 1 o 2 octetos, determinados por un parámetro fijo del sistema.
- Una dirección de difusión se define como 0xFF o 0xFFFF para direcciones de 1 y 2 octetos, respectivamente.
- La suma de comprobación es la suma módulo 256 de los datos de usuario de la trama (no los datos de usuario del enlace). Estos son los datos entre el último carácter de inicio y la suma de verificación, L octetos para la trama variable.
- Las reglas establecen que no se permite más de un intervalo de inactividad de tiempo de bit entre caracteres dentro de la trama, y que se debe permitir un intervalo de inactividad de 33 tiempos de bit después de que el receptor detecte un error de trama.

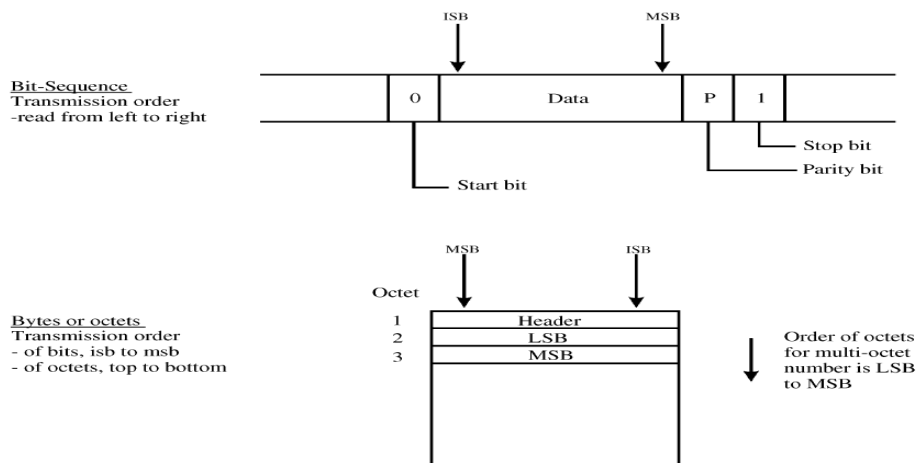
Orden de la información. Un detalle técnico que puede ser difícil de encontrar en los estándares es el orden de bits y bytes. Según IEC 60870-5, como para DNP3, el siguiente orden es estándar.

Los bits se transmiten comenzando con el byte menos significativo (LSB) y terminando con el byte más significativo (MSB). Cuando se da una representación de secuencia de bits como la de la Figura 23, los bits se muestran en este orden. Sin embargo, cuando la estructura de un mensaje en términos de bytes u octetos, el MSB está a la izquierda y el LSB está a la derecha, lo que es coherente con la ponderación numérica de los bits. Una buena manera de resolver esto mentalmente es imaginar que el octeto se desplaza hacia la derecha desde un

registro UART hacia el canal de comunicación. De manera similar, para los bytes, el byte menos significativo (LSB) se transmite primero y el byte más significativo (MSB) se transmite en último lugar. El orden de bits y bytes se ilustra en la Figura 25.

Figura 25

Orden de información.



El formato de trama utilizado por T101 se denomina formato FT1.2. Hay dos formas de esto, una de longitud fija y otra de longitud variable. La trama de longitud fija solo se puede utilizar para tramas de acuse de recibo y comando de control de enlace de datos debido a que solo puede usarse para tramas que no transportan datos de usuario. Además de los de longitud fija y variable.

En el caso de envío de un mensaje o comando para el que no sea indispensable una respuesta de la estación a la que se comunica, se utiliza el servicio Send/No Reply. Se emplea para enviar mensajes de difusión y mensajes que no requieren confirmación de recepción. Para el envío de un comando o datos que requieran la transmisión de forma confiable, se emplea el servicio Enviar/Confirmar. Es necesario una respuesta de confirmación para este servicio. Para obtener información de la estación controlada, se utiliza el servicio Solicitar/Responder. Para este caso, la respuesta de la estación controlada es con los datos pedidos en lugar de una confirmación.

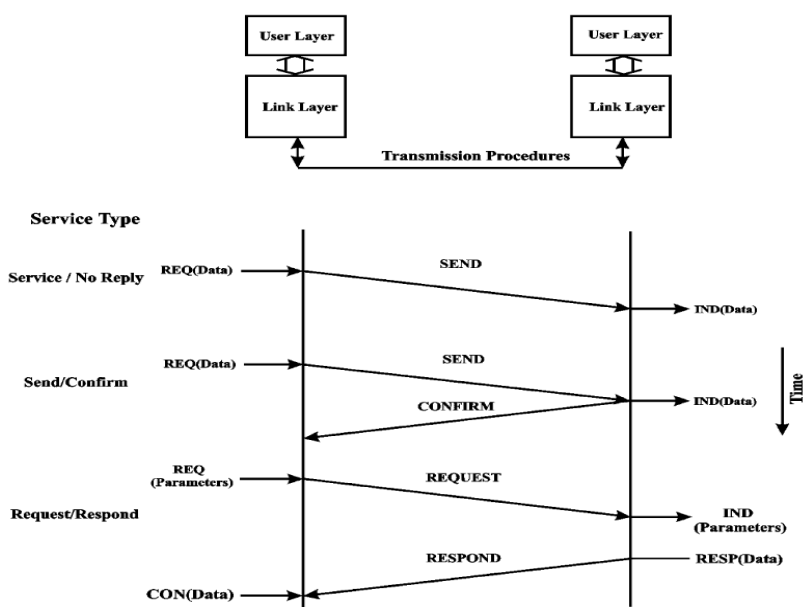
Estos servicios se ilustran en la Figura 26. Esto muestra la capa de enlace de dos estaciones en un canal de comunicación y un diagrama de secuencia de tiempo de las interacciones del servicio.

A cada lado se muestran los comandos, datos y respuestas que se pasan entre la capa de enlace y el usuario del servicio, que es la capa de aplicación. Estos se denominan "primitivas de servicio". En el centro, la transmisión del mensaje está etiquetada con el nombre del procedimiento de transmisión.

Los procedimientos de transmisión son un conjunto de reglas que aseguran que las transmisiones se lleven a cabo con éxito en respuesta a las solicitudes de los usuarios del enlace. Deben ser capaces de hacer frente a los errores en el canal de transmisión que pueden introducir errores o provocar la pérdida de información. Los procedimientos de transmisión son diferentes para enlaces no balanceados y balanceados, y se describen para cada uno en las siguientes secciones.

Figura 26

Servicios de capa de enlace.



Inicialización del enlace. Un servicio de enlace de datos conocido como inicialización de enlace se realiza luego de que una estación se haya encontrado fuera de línea y vuelve a encontrarse disponible por primera vez. En tanto la estación esclava se encuentre fuera de línea, la estación maestra envía funciones de solicitud de estado de enlace en intervalos regulares hasta que se recibe un estado de respuesta de enlace. Las secuencias se muestran a continuación para los modos no balanceado y balanceado.

Inicialización de estación/enlace, modo no balanceado:

- El maestro envía una solicitud de estado del enlace hasta que se recibe el estado del enlace.
- Maestro envía restablecimiento de enlace.
- El enlace está activo al recibir ACK.
- El esclavo genera un evento de finalización de la inicialización de la estación.

Inicialización de estación/enlace, modo balanceado:

- Cada estación envía una solicitud de estado del enlace hasta que se recibe el estado del enlace.
- Cada estación envía reinicio de enlace.
- El enlace está activo al recibir ACK en cada estación.
- Uno o ambos pueden generar un evento de finalización de inicialización de la estación.

Procedimientos de transmisión no balanceada. Se requieren procedimientos de transmisión no balanceados para enlaces que no sean punto a punto, es decir, enlaces multipunto. Para estos enlaces, la estación de control debe controlar el tráfico de datos consultando las estaciones remotas en busca de datos. Solo cuando la estación principal o de

control en un enlace sondea una estación secundaria en particular, esa estación puede responder.

El campo de control. El campo de control de la trama de datos es fundamental para el funcionamiento de los procedimientos de transmisión. Este campo es casi idéntico al utilizado por el protocolo DNP3 porque se derivó del mismo documento fuente, IEC 60870-5-2 1992. En cuanto a cómo se debe interpretar el campo de control precisa de que la comunicación sea un mensaje primario o sea un mensaje secundario. La Figura 27 muestra el campo de control para los procedimientos de transmisión no balanceada, incluidas las breves descripciones de los significados de los códigos de función para los mensajes primarios y secundarios.

Figura 27

Campo de control-transmisión desequilibrada.

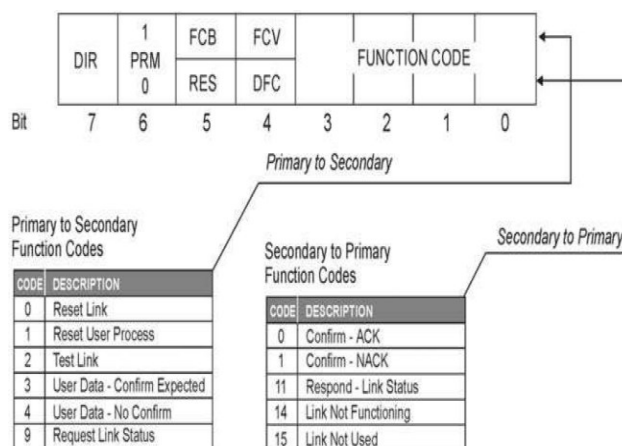


Tabla 6

Significados de los bits del campo de control: transmisión desequilibrada.

Código	Sentido	Descripción
PRM	Mensaje principal	1 => Trama de la estación principal o iniciadora
PCB	Bit de recuento de trama	Alterna entre 0 y 1 para tramas secuenciales

Código	Sentido	Descripción
FCV	Recuento de tramas válido	1 => FCB es válido
		0 => ignorar FCB
RES	Reservado	= 0
DFC	Bit de control de flujo de datos	Establecido en 1 por la estación secundaria si el envío adicional de datos de usuario provocará un desbordamiento del búfer
ACD	Bit de demanda de acceso	Establecido en 1 si hay datos de Clase 1 disponibles

La Tabla 7 y la Tabla 8 muestran los significados detallados de los códigos de función en el campo de control (Tabla 6). Los significados son diferentes dependiendo de si el mensaje es una transmisión primaria o secundaria. El bit de conteo de cuadros se usa solo en la dirección primaria y solo es válido para ciertas funciones. Esto se indica mediante el estado del bit válido de recuento de tramas.

Códigos de función de la estación principal. Los códigos 5–7, 12–15 están reservados.

Tabla 7

Códigos de función: transmisión desbalanceada.

Códigos de Función	Tipo de trama	Función de Servicio	Bit FCY
0	SEND–CONFIRM esperado	Restablecimiento de enlace remoto	0
1	SEND–CONFIRM esperado	Restablecimiento del proceso de usuario	0

Códigos de Función	Tipo de trama	Función de Servicio	Bit FCY
2	ENVIAR-CONFIRMAR esperado	Reservado	-
3	ENVIAR-CONFIRMAR esperado	Datos del usuario	1
4	ENVIAR-SIN RESPUESTA esperado	Datos de usuario (sin confirmar)	0
8	SOLICITUD-RESPUESTA esperada	Solicitud de demanda de acceso	0
9	SOLICITUD-RESPUESTA esperada	Solicitar estado de enlace	0
10	SOLICITUD-RESPUESTA esperada	Solicitar datos de usuario clase 1	1
11	SOLICITUD-RESPUESTA esperada	Solicitar datos de usuario clase 2	1

Códigos de función de la estación secundaria. Los códigos 2-7, 10, 12-13 están reservados.

Tabla 8

Códigos de función: transmisión desbalanceada.

Códigos de Función *	Tipo de trama	Función de Servicio
0	CONFIRMAR	ACK – acuse de recibo positivo**
1	CONFIRMAR	NACK – Mensaje no aceptado, enlace ocupado

Códigos de Función *	Tipo de trama	Función de Servicio
8	RESPONDER	Datos del usuario
9	RESPONDER	NACK – Datos solicitados no disponibles
11	RESPONDER	Estado del enlace (DFC = 0 o 1) o demanda de acceso
14		El servicio de enlace no funciona
15		Servicio de enlace no utilizado o implementado

Se debe tener en cuenta que se puede usar el código de control 0 x E5 en lugar de las tramas FC0 o FC9. Las funciones de los bits de control se explican con más detalle en los siguientes párrafos.

Bit primario. El bit PRM se establece si la trama es primaria (de inicio) o secundaria (de respuesta). Esto es utilizado directamente por la capa de enlace para interpretar el código de función.

Bits de recuento de tramas. Estos son el bit de recuento de tramas (FCB) y el bit válido de recuento de tramas (FCV). Estos bits solo se utilizan para mensajes primarios. El bit de conteo de tramas se usa para detectar pérdidas o duplicación de tramas en una estación secundaria. El bit válido de recuento de tramas permite el uso del FCB. Cuando el bit FCV es verdadero, el FCB se alterna para cada transacción SEND–CONFIRM exitosa entre las mismas estaciones primaria y secundaria. Cómo funcionan es así:

- Después del inicio del enlace de datos o de una transacción fallida, una estación secundaria no aceptará ninguna trama de ENVÍO-CONFIRMACIÓN principal con

FCV = 1 hasta que se complete una transacción de reinicio. Esto significa que solo aceptará un enlace RESET o un comando de proceso de usuario RESET.

- Después de que una estación secundaria recibe una trama de enlace RESET de una primaria y responde con CONFIRM, ese enlace estará operativo hasta que ocurra un error de transmisión de trama.
- La estación secundaria esperará que la siguiente trama contenga FCV = 1 y FCB = 1.
- El siguiente mensaje principal ENVIAR-CONFIRMAR tendrá FCV = 1 y FCB = 1. La estación secundaria aceptará esto ya que el FCB es válido y está configurado, como se esperaba.
- Cada mensaje primario ENVIAR-CONFIRMAR subsiguiente tendrá el FCB borrado o configurado a su vez.

Bit de control de flujo de datos. El bit de control de flujo de datos (DFC) se incluye en las tramas secundarias. La estación secundaria establecerá DFC=1 si un ENVÍO adicional de datos de usuario hará que su búfer se desborde. Al recibir una trama con DFC=1, una estación primaria dejará de enviar datos, pero solicitará el estado del enlace hasta que DFC=0.

Bit de demanda de acceso. Hay dos clases de datos definidos, clase 1 y clase 2. Los datos de clase 1 tienen mayor prioridad que los datos de clase 2. El bit ACD se lo define como un medio para que la estación secundaria muestre a la estación primaria que existen datos de clase 1.

Campo de dirección. La longitud del campo de dirección de la trama de capa de enlace es de uno o dos octetos y se establece como un parámetro fijo del sistema. La dirección de enlace de la estación secundaria se encuentra en este campo. La dirección de enlace de la estación secundaria a la que se orienta el mensaje se incluye en una trama transmitida por la

estación primaria en un enlace. Una trama que se transmite de la estación secundaria a la estación principal tiene su propia dirección de enlace. De esta manera, la estación primaria puede determinar la estación secundaria de cual proviene el mensaje.

Procedimientos de transmisión. En esta sección se discuten brevemente los procedimientos. Los procedimientos son muy similares a los utilizados por el protocolo DNP3. Se puede hacer referencia a esto para una mayor ilustración de los conceptos presentados. Se debe tener en cuenta que, en los siguientes procedimientos, no se inicia ningún procedimiento nuevo hasta que finaliza el procedimiento anterior. Por esta razón, un bit de cómputo de tramas de un solo bit es suficiente para la protección contra errores de número de secuencia de tramas; hay una ventana de marco de exactamente uno.

Procedimientos ENVIAR/NO RESPUESTA. La trama se transmite y se requiere un tiempo de inactividad de línea mínimo de 33 tiempos de bit antes de cualquier otra transmisión por parte del primario. Al recibir el mensaje en el secundario, se verifica si hay errores comparándolo con el octeto de suma de verificación y, si es válido, se notifica al usuario del servicio.

Procedimientos ENVIAR/CONFIRMAR. La estación primaria transmitirá el mensaje. Si no se recibe una confirmación de la estación secundaria dentro de un período de tiempo de espera configurado, retransmitirá el mensaje hasta un número configurado de reintentos.

Si la estación secundaria recibe el mensaje, responderá con una confirmación positiva o negativa, código de función 0 o 1. FC = 0 significa que el mensaje se recibió y aceptó correctamente, y en este caso finaliza el procedimiento. FC = 1 significa que el secundario no puede aceptar el mensaje porque su búfer está lleno o por alguna otra razón. En este caso, el principal volverá a intentarlo, hasta un número configurado de reintentos.

Este procedimiento utiliza el bit de recuento de tramas (FCB) para garantizar que la secuencia del mensaje no se vea perturbada. El FCB se alterna con cada transmisión ENVIAR/CONFIRMAR desde la estación principal, y la estación secundaria mantiene un indicador de FCB esperado. Si un mensaje enviado por el primario no es confirmado por el secundario, se retransmite con el FCB sin cambios. Por lo tanto, un mensaje perdido o corrompido en la dirección de transmisión primaria a secundaria, cuando se retransmita, todavía tendrá el valor FCB esperado. Alternativamente, si el problema fuera que la confirmación de la estación secundaria se perdió o corrompió, la estación secundaria podría reconocer la recepción del mensaje transmitido desde la primaria por el bit FCB sin cambios. En este caso retransmite el mensaje de confirmación original.

Como alternativa al envío de una trama de confirmación, que tiene una longitud mínima de seis octetos (es decir, si se utiliza una dirección de un solo octeto), se permite la respuesta de un solo carácter de control (hexadecimal $0 \times E5$). Esta opción se puede utilizar para mejorar la eficiencia de la transmisión cuando no es necesario transmitir ninguna otra información a la estación principal.

Procedimientos de SOLICITUD/RESPUESTA. Este procedimiento es similar a ENVIAR/CONFIRMAR, excepto que, en lugar de recibir una confirmación de la estación secundaria, se devuelve una trama que contiene datos o se devuelve una respuesta negativa que indica que no hay datos disponibles. En el caso de una respuesta negativa, se puede devolver una trama con $FC = 9$ o el carácter de control único ($0 \times E5$).

En cuanto a los procedimientos ENVIAR/CONFIRMAR, el bit de conteo de tramas se alterna en cada extremo para cada transmisión de mensaje, y esto se usa para detectar errores en la transmisión en cualquier dirección. Básicamente, cuando tanto el primario como el secundario han incrementado sus bits de recuento de tramas y están de acuerdo, el procedimiento está completo.

Filosofía de la transmisión. Para la transmisión no balanceada, solo la estación principal o de control en un enlace puede iniciar transmisiones. Debido a esto, es necesario que la estación primaria implemente un sistema de sondeo para determinar si hay cambios de datos disponibles en cada estación secundaria.

Para lograr esto, la estación de control sondeará cada estación secundaria de forma cíclica para obtener datos. Por lo general, sondeará los datos de usuario de clase 2 utilizando el código de función de solicitud-respuesta 10. La estación secundaria luego devolverá cualquier dato de clase 2 que tenga disponible y, al mismo tiempo, indicará si hay algún dato de clase 1 disponible. establecer el bit de demanda de acceso (ACD). Por lo general, los valores analógicos se asignarán a la clase 2 y se actualizarán durante la exploración cíclica, y todos los demás datos, como los eventos, se asignarán a la clase 1.

Se debe tener en cuenta que, aunque la operación de sondeo la lleva a cabo la capa de enlace, la secuencia de sondeo en sí es implementada por un nivel superior del protocolo, el nivel de proceso de usuario. El nivel superior genera solicitudes de servicios utilizando primitivas de servicio para especificar las acciones de sondeo que llevará a cabo la capa de enlace.

Procedimientos de transmisión balanceada. Los procedimientos de transmisión balanceada sólo pueden utilizarse para enlaces punto a punto equipados con comunicación de canal dúplex. Bajo transmisión balanceada, cada estación puede actuar como estación primaria y secundaria simultáneamente. El campo de control y los procedimientos de transmisión se modifican ligeramente del caso desequilibrado para adaptarse a esta operación.

Campo de control. Hay dos modificaciones al campo de control del enlace de datos para la transmisión balanceada. La primera es que no se requiere el bit de demanda de acceso (ACD), porque cualquiera de las estaciones puede iniciar transmisiones. El segundo es la inclusión de un bit de dirección (DIR). Esto indica la dirección de transmisión de un mensaje

entre las dos estaciones. El campo de control para la transmisión balanceada se muestra en la Figura 28 (ver también la Tabla 9).

Figura 28

Campo de control: transmisión balanceada.

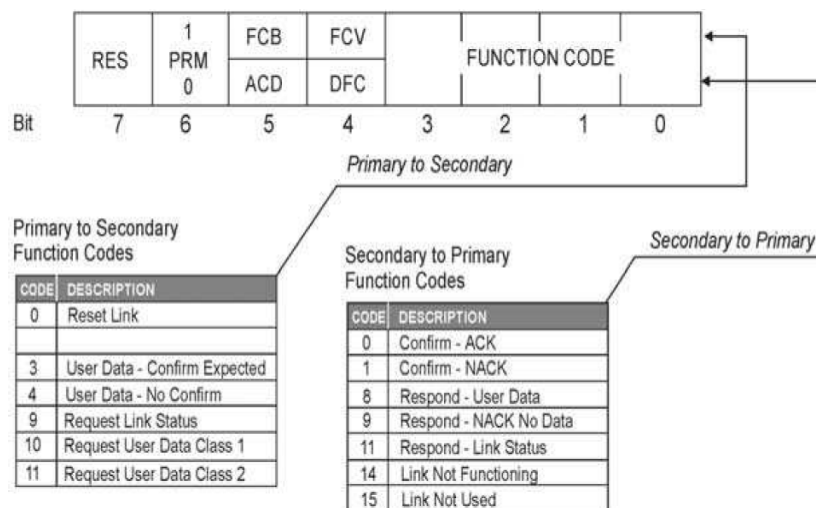


Tabla 9

Significados de los bits del campo de control: transmisión balanceada.

Código	Sentido	Descripción
DIR	Dirección del mensaje	1 => A a B 0 => B a A
PRM	Mensaje principal	1 => Trama de la estación principal o iniciadora
FCB	Bit de recuento de tramas	Alterna entre 0 y 1 para tramas secuenciales
FCV	Recuento de tramas válidas	1 => FCB es válido 0 => Ignorar FCB
RES	Reservado	= 0

Código	Sentido	Descripción
DFC	Bit de control de flujo de datos	Establecido en 1 por la estación secundaria si el envío adicional de datos de usuario provocará un desbordamiento del búfer

La Tabla 10 muestra los significados detallados de los códigos de función en el campo de control. Los cambios de los códigos para transmisión no balanceada se muestran en cursiva.

Códigos de función de la estación principal. Los códigos 5–7, 12–15 están reservados.

Tabla 10

Códigos de función: transmisión balanceada.

Código de función	Tipo de trama	Función de servicio	Bit FCV
0	ENVIAR–CONFIRMAR esperado	Restablecimiento de enlace remoto	0
1	ENVIAR–CONFIRMAR esperado	Restablecimiento del proceso de usuario	0
2	ENVIAR–CONFIRMAR esperado	Prueba de función del enlace	1
3	ENVIAR–CONFIRMAR esperado	Datos de usuario esperados	1
4	ENVIAR–SIN RESPUESTA esperado	Datos de usuario (sin confirmar)	0
8		Usuario por solo desequilibrado	-

Código de función	Tipo de trama	Función de servicio	Bit FCV
9	SOLICITUD-RESPUESTA esperada	Estado de solicitud del enlace	0
10		Utilizado solo por desequilibrados	1
11		Utilizado solo por desequilibrados	1

Los cambios en las funciones de la transmisión desequilibrada son:

- Se ha agregado un código de función principal de enlace de prueba FC = 2.
- Se han eliminado las solicitudes de demanda de acceso y de datos de usuario de clase 1 o 2 (códigos de función principal FC = 8, 10, 11).
- Se han eliminado los códigos de función de respuesta secundaria FC = 8, 9.

Estos cambios reflejan los cambios en la forma en que se transmiten los datos. En la transmisión balanceada, el servicio de solicitud/respuesta no se usa para la transmisión de datos de usuario, sino solo para verificar el estado del enlace. Los datos del usuario son transmitidos directamente por la capa de enlace utilizando el servicio de envío.

Campo de dirección. El campo de dirección para comunicaciones equilibradas puede ser cero, uno o dos octetos. Por lo tanto, existe la opción de no tener un campo de dirección en las comunicaciones balanceadas. Este es el caso porque como solo hay una estación en cada extremo del enlace, no es necesario que el proceso secundario en ninguno de los extremos

verifique la dirección. En efecto, para un enlace balanceado, el bit de dirección (DIR) sustituye a la dirección, haciendo que la inclusión de la dirección del enlace sea redundante.

Procedimientos de transmisión. Los procedimientos de transmisión para la transmisión balanceada son los mismos que para la transmisión no balanceada. La única diferencia es que hay procesos primarios y secundarios operando simultáneamente en cada estación. Esto es similar a dos enlaces no balanceados operando en paralelo. Estos procesos mantienen una secuencia de bits de conteo de tramas separada para cada enlace primario-secundario y los utilizan para detectar errores y recuperarse de ellos de la misma manera que funciona un solo enlace primario-secundario desequilibrado.

Filosofía de la transmisión. Para una transmisión equilibrada, cualquiera de las estaciones del enlace puede iniciar la transmisión. Por lo tanto, no es necesario sondear una estación para obtener datos, ya que puede enviarlos directamente cuando están disponibles. Esto cambia los servicios usados en la capa de enlace del caso de transmisión no balanceada. Cuando la transmisión no balanceada utiliza el servicio de solicitud/respuesta para obtener datos de usuario de la estación secundaria, para la transmisión balanceada, la estación con los datos solo usa el servicio de envío/confirmación directamente. La comparación de las Figuras 10.13 y 10.14 muestra que los códigos secundarios para responder con datos de usuario FC8 y FC9 no están disponibles para transmisión balanceada. En su lugar, los códigos de función principal FC3 o FC4 se utilizan para enviar los datos directamente.

Seguridad del enlace de datos. IEC 60870-5-101 y DNP3 utilizan los formatos de trama FT1.2 y FT3 especificados por IEC 60780-5-1, respectivamente. Estos formatos difieren en sus disposiciones de seguridad como se muestra en la Tabla 11. IEC 60870-5-101 utiliza una suma de comprobación de 8 bits y un tamaño de trama máximo de 255 bytes. DNP utiliza un código de redundancia cíclica (CRC) de 16 bits por cada 16 bytes de datos de usuario contenidos en el cuerpo de su trama.

Tabla 11

Seguridad de error de capa de enlace.

Protocolo	IEC 60870-5	Método de seguridad	Bits de error de distancia de Hamming	Longitud máxima asegurada en bytes
IEC 60870-5-101	FT1.2	8 bits checksum	4	255
DNP 3.0	FT3	16 bits CRC	6	16

Comparación de funciones de seguridad. El efecto de estas diferencias en la seguridad a menudo se cita en términos de "distancia de Hamming". Esto es igual a la cantidad mínima de errores de un solo bit que se requieren para permitir que un mensaje incorrecto se acepte erróneamente como un mensaje bueno que falla para que el sistema de seguridad falle. Estos son 4 y 6 para los métodos utilizados por FT1.2 y FT3. Sin embargo, estas cifras ignoran el efecto de la proporción de bits de código de seguridad a bits de mensaje, que en el caso de DNP es mayor debido a la inclusión de códigos CRC dentro de la sección del cuerpo del formato de trama FT3.

Enlace vs datos de la aplicación. Existe una distinción entre datos de enlace y datos de nivel de aplicación que es sutil y puede ser difícil de comprender a primera vista. Se deriva del hecho de que los mensajes del nivel de la aplicación se tratan simplemente como datos en el nivel de la capa de enlace y puede ilustrarse mejor con un ejemplo.

Considere el caso en el que funciona un enlace desequilibrado entre la estación primaria A y la estación secundaria D. Suponga que el nivel de aplicación de la estación D requiere algunos datos particulares de A. Esto es en la dirección inversa a la mayoría del tráfico

de datos, que generalmente proviene de la estación secundaria. estación a la primaria. En este caso, la solicitud de nivel de aplicación de D generará una demanda de acceso de clase 1 en el nivel de enlace. Esto tendrá que esperar hasta que la estación primaria A sondee datos de esa estación. En este punto, si se han sondeado datos de clase 2, responderá con cualquier dato de clase 2, o con una trama FC9 'NACK - sin datos', pero en cualquier caso con el bit ACD (demanda de acceso) establecido. La estación primaria A luego sondeará los datos de clase 1 de la estación D, y en este punto D podrá transmitir sus datos de clase 1. Esto contendrá la solicitud de nivel de aplicación de D, para que la estación primaria envíe los datos de nivel de aplicación que requiere.

La trama de la capa de enlace que se solicitó mediante el sondeo del primario A, llevó una solicitud de nivel de aplicación de la estación D. Esto dio como resultado que la capa de aplicación de A usara el servicio de envío para devolver los datos solicitados a la estación D. Tenga en cuenta que una solicitud de capa de enlace en dirección A a D (la encuesta) ha resultado en una solicitud de nivel de aplicación de D a A.

En las siguientes secciones que tratan sobre la capa de aplicación y superiores, la operación de la capa de enlace puede verse generalmente como un mecanismo de transporte de mensajes que se puede suponer que solo transporta los mensajes de nivel de aplicación según sea necesario. Sin embargo, el conocimiento del mecanismo de transporte es importante para comprender los tiempos de respuesta, ya que estos dependerán del modo de transmisión y la frecuencia de sondeo para el modo no balanceado.

Capa de aplicación. Las secciones restantes describen el funcionamiento de la capa de aplicación y superior del protocolo IEC 60870-5. La mayor parte de esta sección se aplica a las versiones del protocolo, IEC 60870-5-101 e IEC 60870-5-104, tanto en red como sin conexión a red. Las áreas que son diferentes se tratan en el siguiente capítulo bajo

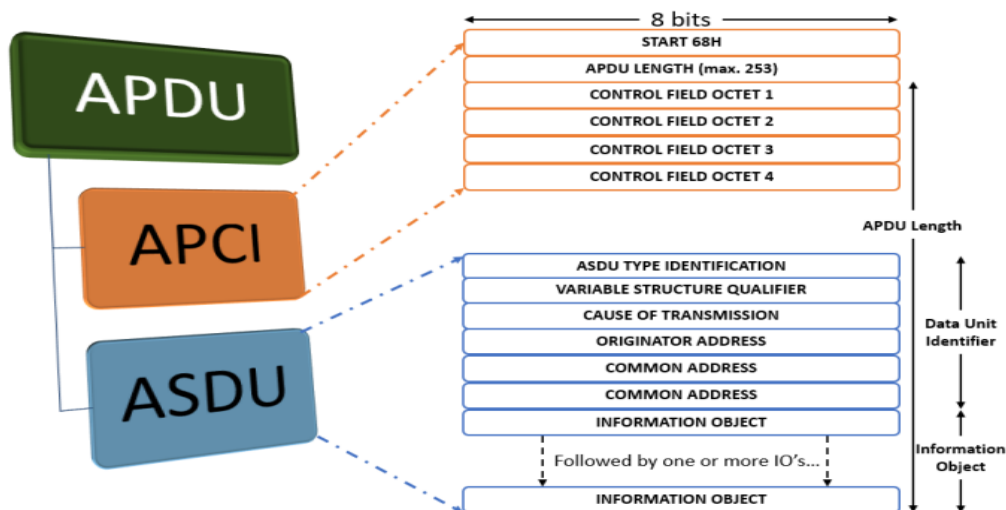
Consideraciones avanzadas de IEC 60870-5. Esta información se presenta en la siguiente secuencia:

- La estructura general del mensaje
- La estructura ASDU
- El direccionamiento y el enrutamiento de mensajes
- Los elementos de información
- El conjunto de ASDU. 10.5.1

Estructura general del mensaje. En la vista previa, se presentó la estructura general del mensaje según IEC 60870-5. Esto se representa en la Figura 29. Esto muestra la ASDU de la capa de aplicación y muestra cómo la capa de enlace de datos la transporta como datos de usuario de enlace según IEC 60870-5-101. Para la versión en red IEC 60870-5-104, la ASDU es transportada por los protocolos TCP/IP en lugar de la capa de enlace T101, por lo que la trama de enlace que se muestra en la Figura 29 no se aplica a este caso.

Figura 29

Estructura de mensajes según IEC60870-5-101.



El único punto importante a tener en cuenta sobre la relación entre la ASDU y la trama de la capa de enlace es que se permite un máximo de una ASDU por trama. Esto establece un límite superior para el tamaño de la ASDU en 255 octetos, menos 2 o 3 octetos para los campos de dirección y control de enlace.

Estructura del APCI. Las APDUs son transmisión de tramas realizadas por la capa de aplicación. Sus siglas significan Unidad de Datos del Protocolo de Aplicación y es la unidad de datos que se proporciona desde o hacia la capa de aplicación. El APDU tiene una cabecera que abarca información de control llamada APCI (Información de Control del Protocolo de Aplicación), así como información de elementos de campo (variables analógicas y digitales).

APCI es un encabezado de 6 octetos que incluye un octeto de inicio (68H), seguido de una longitud de 1 octeto y campos de control de 4 octetos. Los dos bits menos significativos del primer octeto del campo de control determinan el tipo de mensaje de telecontrol, que puede ser formato I (00b), formato S (01b) o formato U (11b). Un IFormat APCI transporta información de sensores y control, y siempre va seguido de una ASDU. Se utiliza una APCI de formato S para enviar acuses de recibo de número de secuencia, y la APCI de formato U se utiliza para enviar mensajes de mantenimiento de conexión (Figura 30).

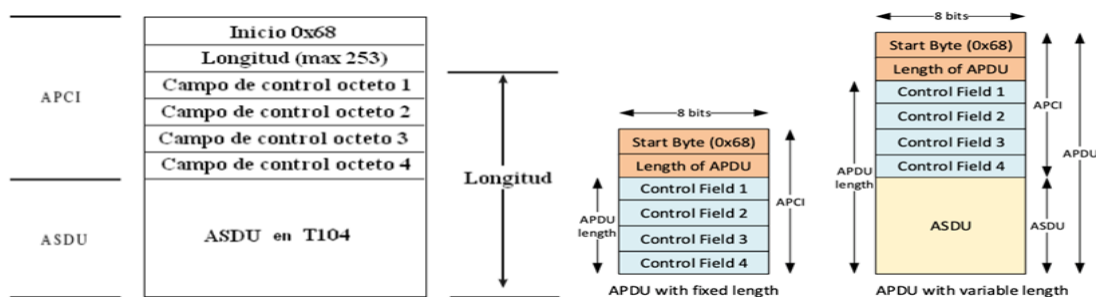
ASDU es una estructura de datos de longitud variable que consta de un identificador de unidad de datos (DUI) de longitud fija y un objeto de información (IO) de longitud variable. Finalmente, IEC 104 también define cuatro temporizadores, sus descripciones funcionales son las siguientes:

- T0: el tiempo de espera de este tiempo activará una solicitud SYN.
- T1: el tiempo de espera de este temporizador activará una solicitud de cierre activa en una conexión establecida. Además, el evento de tiempo de espera T1 en una estación de control puede activar automáticamente una nueva conexión redundante y un cambio de conexión automático.

- T2: la expiración de este temporizador hará que el receptor envíe un formato S con un número de secuencia recibido (mensaje ACK) que es un reconocimiento a nivel de aplicación por haber recibido un número específico de APDU.
- T3: la expiración de este temporizador indica que no se han recibido mensajes IEC 104 del otro lado para una conexión TCP abierta determinada, es decir, inactividad. Una recepción de cualquier tipo de mensaje, formato I, formato S o formato U restablecerá este temporizador.

Figura 30

Estructura del APDU.



Estructura del campo de control. Los tres formatos del campo de control son I, S y U; y cada uno tiene una longitud de cuatro octetos e identifica la función: S (Supervisión), I (Información) y U (Sin número).

Se utiliza el campo de control de formato I para APDUs que contienen ASDUs (información). La primera posición del bit contiene un cero que representa este campo (Figura 31).

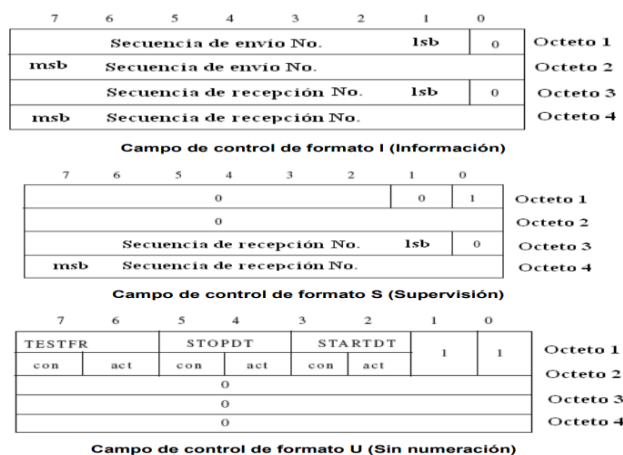
El campo de control de formato S (supervisión) señala con un 1 en la primera posición del bit, consecutivo por un cero en la posición 2 del bit. Existe tan sólo un número de secuencia de recepción en el campo. Dicho formato es empleado para los APDUs que abarcan tan sólo la cabecera APCI. Ninguna información adjunta es contenida en este formato, debido a que la función de supervisión sólo se restringe a controlar el transporte de los APDUs (Figura 31).

El campo de control del formato U (sin numeración) también son empleados en los APDUs que abarcan sólo el APCI. Se lo emplea como un mecanismo para el arranque y la parada del flujo de información. En este campo de control no existen números de secuencia. Además, se la puede utilizar donde se encuentre disponible más de una conexión entre estaciones, y un cambio entre las conexiones se la tiene que aplicar sin pérdida de datos (Figura 31).

Un número de secuencia es otorgado a cada uno de los APDUs que son enviados. Estos números de secuencia se los emplean en dichos campos de control para el control del paso de los APDUs por entre la red, lo cual sucede en bidireccionalmente. El mayor número de secuencia captado por el receptor de APDUs es reportado hacia el remitente y este último usa esto como un soporte para que los APDUs, que esporádicamente llegasen a perderse, puedan ser reenviados. El receptor tiene la opción de emplear una transmisión de información de campo de control formato I (I format) o utilizar una función de supervisión de campo de control formato S (S format) para que el remitente sea informado del número de secuencia. Todo esto será posible en cuanto el receptor se encuentre simultáneamente enviando información en dirección contraria.

Figura 31

Contenido de la Estructura del APDU, dependiendo del tipo de comando.



Estructura de la ASDU. Esta estructura se conforma por dos secciones principales, los mismos que son el identificador de la unidad de datos y los mismos datos, que se encuentran conformados por uno o varios objetos de información. Dentro de esta estructura el tipo específico de datos es determinado por el identificador de unidad, el mismo que suministra direccionamiento con la finalidad de determinar la identidad exacta de los datos y abarca información extra en el campo de causa de transmisión. Por lo cual, ahora se discuten a su vez los campos de la ASDU.

Identificación de tipo. Este es un campo entero que no contiene ni un solo signo de un octeto. En la Figura 32 se presenta como su contenido se lo puede considerar como un código en los rangos.

Figura 32

Tipo de campo de identificación.

BIT	7						0
Identificación de Tipo							

Rango de Códigos	Propósito
<1 127>	Definiciones tipo estándar
<128 135>	Reservado para mensajes de enrutamiento – privado
<136 255>	Para uso especial – privado

Las siguientes notas se aplican a estos códigos:

- El valor <0> no se utiliza.
- El rango <128..255> no está definido por el estándar y puede ser utilizado por proveedores particulares para funciones específicas del sistema. Sin embargo, esto tiene implicaciones para la interoperabilidad.

Dentro del rango de las definiciones de tipo estándar, se puede determinar que en la actualidad existen 58 tipos específicos. En la Tabla 12 se presenta la asociación de dichos

tipos, en donde se observan los grupos generales y el número de códigos de identificación de tipo que se definen.

Tabla 12

Grupos de códigos de tipo definidos.

Códigos de tipo definidos	Grupo
<1..40>	Procesar información en la dirección del monitor
<45..51>	Información de proceso en dirección de control
<70>	Información del sistema en la dirección del monitor
<100..106>	Información del sistema en la dirección de control
<110..113>	Parámetro en dirección de control
<120..126>	Transferencia de archivos

Es importante tener en cuenta que la identificación de tipo se aplica a toda la ASDU, por lo tanto, si hay varios objetos de información contenidos en la ASDU, todos son del mismo tipo.

La Tabla 12 proporciona una lista completa de los tipos de ASDU. La tabla se divide en los grupos de códigos que se muestran en la Tabla 12. Los detalles completos de cada tipo de ASDU individual se proporcionan en "Conjunto de ASDU" más adelante.

Las tablas de la Tabla 14 a la Tabla 20 también incluye referencias de códigos de información que pueden encontrarse. Estas referencias están definidas por IEC 60870-5-5. Proporcionan un sistema de referencia jerárquico usando la siguiente estructura (Tabla 13).

Tabla 13*Estructura del código de información de referencia.*

Nivel	Símbolo	Descripción
1	M_	Información monitoreada
	C_	Información de control
	P_	Parámetro
	F_	Transferencia de archivos
2	Varios	Información Monitor:
		Single Point Information M_SP
		Double Point Information M_DP
		Bitstring M_BO
		Valor Medido M_ME
		Información Control:
		Double_Command C_DC
		Single_Command C_SC
		Parámetros:
		Parámetros de Medidas P_ME
Transferencia de archivos:		
Directorio F_DR		
3	_Nx	Sin etiqueta de tiempo
	_Tx	Tiempo etiquetado
	_xA	Tipo A: estado y normalizado, con calidad
	_xB	Tipo B: escalado, con calidad
	_xC	Tipo C: punto flotante corto, con calidad
_xD	Tipo D: normalizado sin calidad	

Por ejemplo, M_ME_TA_1 es información monitoreada, un valor medido, con etiqueta de tiempo, y tipo A, que es un valor normalizado con calidad.

Tabla 14

Tipos de ASDU: información del proceso en la dirección de monitoreo.

Tipo No.	Descripción	Referencia
<0>	No definido	
<1>	Contenido de punto único	M_SP_NA_1
<2>	Contenido de punto único con etiqueta de tiempo	M_SP_TA_1
<3>	Contenido de doble punto	M_DP_NA_1
<4>	Contenido de punto doble con etiqueta de tiempo	M_DP_TA_1
<5>	Contenido de posición de paso	M_ST_NA_1
<6>	Contenido de posición de paso con etiqueta de tiempo	M_ST_TA_1
<7>	Secuencia de bits de 32 bits	M_BO_NA_1
<8>	Secuencia de bits de 32 bits con etiqueta de tiempo	M_BO_TA_1
<9>	Valor normalizado	M_ME_NA_1
<10>	Valor normalizado con etiqueta de tiempo	M_ME_TA_1
<11>	Valor escalado	M_ME_NB_1
<12>	Valor escalado con etiqueta de tiempo	M_ME_TB_1
<13>	Número de punto flotante corto	M_ME_NC_1

Tipo No.	Descripción	Referencia
<14>	Número de punto flotante corto con etiqueta de tiempo	M_ME_TC_1
<15>	Totales conformados	M_IT_NA_1
<16>	Totales conformados con etiqueta de tiempo	M_IT_TA_1
<17>	Suceso de equipo de protección con etiqueta de tiempo	M_EP_TA_1
<18>	Suceso de inicio empaquetados de equipos de protección con etiqueta de tiempo	M_EP_TB_1
<19>	Contenido empaquetado del circuito de salida del equipo de protección con etiqueta de tiempo	M_EP_TC_1
<20>	Contenido empaquetado de un solo punto con detección de cambio de estado	M_PS_NA_1
<21>	Valor normalizado sin descriptor de calidad	M_ME_ND_1
<22..29>	Reservado para más definiciones compatibles	

La Tabla 15 muestra los tipos que se agregaron a esta categoría con la enmienda 2 de IEC60870-5-101. Estos proporcionan un formato de etiqueta de tiempo más largo.

Tabla 15

Tipos de ASDU: información del proceso en la dirección de monitoreo (next).

N° de tipo	Descripción	Referencia
<30>	Contenido de un solo punto con etiqueta de tiempo CP56Tiempo2a	M_SP_TB_1

N° de tipo	Descripción	Referencia
<31>	Contenido de doble punto con etiqueta de tiempo CP56Tiempo2a	M_DP_TB_1
<32>	Contenido de posición de paso con etiqueta de tiempo CP56Tiempo2a	M_ST_TB_1
<33>	Secuencia de bits de 32 bits con etiqueta de tiempo CP56Tiempo2a	M_BO_TB_1
<34>	Valor normalizado con etiqueta de tiempo CP56Time2a	M_ME_TD_1
<35>	Valor escalado con el tiempo etiqueta CP56Time2a	M_ME_TE_1
<36>	Punto flotante corto número con etiqueta de tiempo CP56Time2a	M_ME_TF_1
<37>	Totales conformados con etiqueta de tiempo CP56Tiempo2a	M_IT_TB_1
<38>	Suceso de equipos de protección con etiqueta de tiempo CP56Time2a	M_EP_TD_1
<39>	Suceso de inicio empaquetados de protección equipo con etiqueta de tiempo CP56Time2a	M_EP_TE_1
<40>	Contenido empaquetado del circuito de salida de equipo de protección con etiqueta de tiempo CP56Tiempo2a	M_EP_TF_1
<41..44>	Reservado para más definiciones compatibles	

Tabla 16

Tipos de ASDU: información del proceso en la dirección de control.

N.º de tipo	Descripción	Referencia
<45>	Comando único	C_SC_NA_1
<46>	Comando doble	C_DC_NA_1
<47>	Mando paso de regulación	C_RC_NA_1
<48>	Comando de consigna, valor normalizado	C_SE_NA_1
<49>	Comando de consigna, valor escalado	C_SE_NB_1
<50>	Comando de punto de ajuste, número de punto flotante corto	C_SE_NC_1
<51>	Secuencia de bits de 32 bits	C_BO_NA_1

Tabla 17

Tipos de ASDU: parámetro en la dirección de control.

Nº de tipo	Descripción	Referencia
<70>	Fin de la inicialización	M_EI_NA_1
<70..99>	Reservado para más definiciones compatibles	

Tabla 18

Tipos de ASDU: información del sistema en la dirección de control.

Nº de tipo	Descripción	Referencia
<100>	Instrucción de interrogación	C_IC_NA_1
<101>	Instrucción de interrogación de contador	C_CI_NA_1

N° de tipo	Descripción	Referencia
<102>	Instrucción de lectura	C_RD_NA_1
<103>	Instrucción de sincronización de reloj	C_CS_NA_1
<104>	Instrucción de prueba	C_TS_NA_1
<105>	Restablecer instrucción de proceso	C_RP_NA_1
<106>	Instrucción de adquisición de retardo	C_CD_NA_1
<107..109>	Reservado para más definiciones compatibles	

Tabla 19

Tipos de ASDU: parámetro en la dirección de control.

N° de tipo	Descripción	Referencia
<110>	Parámetro del valor medido, valor normalizado	P_ME_NA_1
<111>	Parámetro del valor medido, valor escalado	P_ME_NB_1
<112>	Parámetro del valor medido, número de punto flotante corto	P_ME_NC_1
<113>	Activación de parámetros	P_AC_NA_1
<114.. 119>	Reservado para más definiciones compatibles	

Tabla 20

Tipos de ASDU: transferencia de archivos.

N° de tipo	Descripción	Referencia
<120>	Archivo listo	F_FR_NA_1
<121>	Trama lista	F_SR_NA_1

Nº de tipo	Descripción	Referencia
<122>	Llamar directorio, seleccionar Archivo, llamar Archivo, llamar sección	F_SC_NA_1
<123>	Última sección, último segmento	F_LS_NA_1
<124>	Archivo de confirmación, sección de confirmación	F_AF_NA_1
<125>	Segmento	F_SG_NA_1
<126>	Directorio	F_DR_TA_1
<127>	Reservado para más definiciones compatibles	

Calificador de estructura variable. Este es un octeto único el mismo que determina el número de elementos de información y la forma en que se direccionan. Este calificador incluye un número binario de siete bits y un campo de un bit el cual determina la estructura de información que se emplea. En la Figura 33 se presenta el campo calificador de estructura variable, a continuación de una especificación de las dos estructuras de información.

Además, en la Figura 33 se puede observar que existen dos estructuras diferentes, las cuales necesitan del estado del bit más significativo del calificador de estructura variable, el mismo que se lo denomina bit SQ, que puede contemplarse como el bit calificador de estructura.

En el momento en que $SQ = 0$, la estructura se convierte en una cadena de objetos de información. Donde cada objeto de información porta su propia dirección, por lo que los elementos de información contenidos no requieren tener direcciones sucesivas. Además, el valor N de siete bits se encuentra determinado por el número de objetos de información, por tal razón, en esta ASDU, puede existir incluso hasta 127 objetos de información.

En el momento en que $SQ = 1$, la estructura abarca solo un objeto de información, sin embargo, el mismo puede abarcar diversos elementos de información, los cuales absolutamente todos son del mismo formato, como un valor medido. En esta situación, existe solo una dirección de objeto de información y tan solo una etiqueta de tiempo, si es que se usa.

El efecto del bit SQ es que para cada número de identificación de tipo puede haber efectivamente dos ASDU. Esto se ve en el subapartado que presenta el conjunto de ASDU, donde se verá que algunos tipos tienen ambas variaciones ($SQ = 0$ y $SQ = 1$), y otros tienen solo una de estas.

Causa de transmisión. El campo de causa de transmisión, o denominado COT, se emplea para el control del enrutamiento de mensajes en la red de comunicaciones y en una estación, conduciendo la ASDU al programa o tarea adecuados para su posterior procesamiento. Esta subsección analizará inicialmente la estructura y el significado de los subcampos dentro del COT, y luego analizará cómo se utilizan (Figura 34).

Figura 33

Calificador de estructura variable y estructuras.

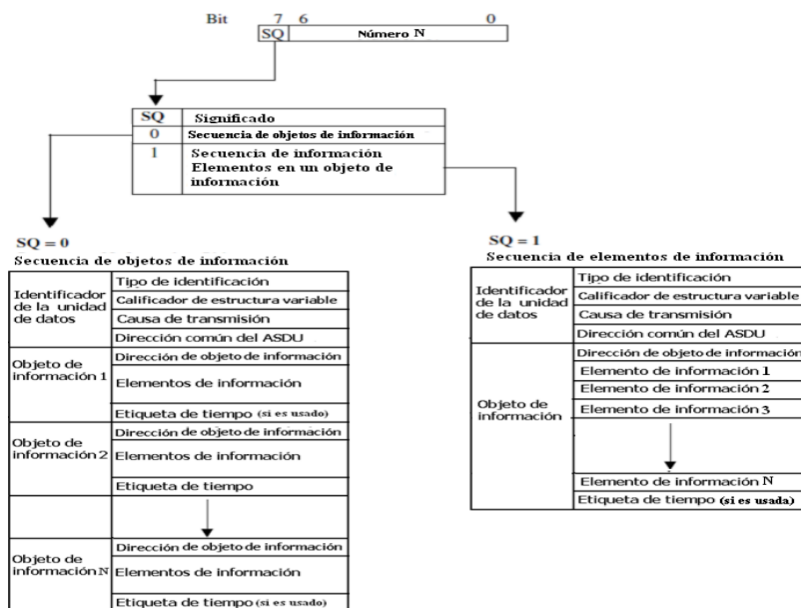
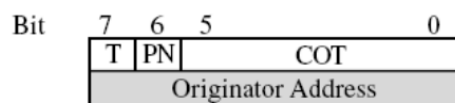


Figura 34

Causa del campo de transmisión.

**Key:**

COT Cause of Transmission <0..63>

PN Positive / Negative Confirm bit

T Test bit

OA Originator Address <0..255>

COT es un código de seis bits que es empleado para analizar la información en la estación de destino y los códigos se muestran en la Tabla 21. Para cada tipo de ASDU definido existe un subconjunto definido de códigos que son significativos para estos y que se proporcionan en la sección que muestra el conjunto de ASDU.

El bit de confirmación positivo/negativo es denominado bit PN. Cuando se lo utiliza con comandos de control esto resulta significativo y se lo emplea en el momento en que el comando de control se refleja en la dirección del monitor, produciendo una señal de su ejecución o cuando no se ejecuta, y en el instante en que el bit PN ya no es relevante, este se pone a cero.

El bit T o también llamado de prueba se lo utiliza en el momento en que las ASDU se producen sólo para pruebas y se lo emplea para comprobar el equipo y la transmisión.

Tabla 21

Causa de los códigos de transmisión (COT).

Código COT	Causa de transmisión
0	No utilizado
1	Periódico, cíclico
2	Análisis de fondo

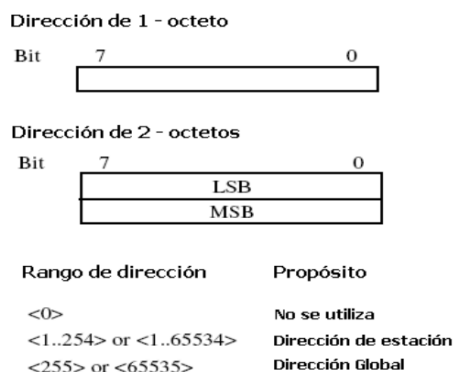
Código COT	Causa de transmisión
3	Espontáneo
4	Inicializado
5	Solicitud o pedido
6	Activación
7	Confirmación de activación
8	Desactivación
9	Confirmación de desactivación
10	Terminación de activación
11	Información de retorno provocada por un comando remoto
12	Información de retorno provocada por un comando local
13	Transferencia de archivos
14–19	Reservado para futuras definiciones
20	Interrogado por el interrogatorio de la estación
21–36	Interrogado por grupo (1–16) interrogatorio
37	Solicitado por el contrario solicitud general
38–41	Solicitado por grupo (1–4) solicitud de contador
42–43	Reservado para futuras definiciones
44	Identificación de tipo sin conocer
45	Motivo de transmisión sin conocer
46	Dirección común sin conocer de ASDU
47	Dirección de objeto de contenido desconocido

La dirección del originador es opcional según el sistema. Proporciona un medio para que una estación de control se identifique explícitamente. Si existe tan sólo una estación de control en un sistema, la dirección del originador no es necesaria, sin embargo, si es necesaria en el momento que existan algunas estaciones ya son estaciones de modo dual que pueden actuar como estaciones controladas o como controladoras. En estas circunstancias, la dirección del originador se puede utilizar para enviar confirmaciones de comandos a la estación de control particular que emitió el comando, en lugar de a todo el sistema. La única excepción a esto es la dirección de origen de cero <0>. Esto tiene el mismo efecto que si no hubiera una dirección de origen. Se denomina dirección por defecto y su efecto es que el mensaje se transmite a todas las estaciones. El uso de la dirección del remitente se analiza con mayor detalle en el mensaje de direccionamiento y enrutamiento.

Dirección común de ASDU. Tiene una longitud de uno o dos octetos como se observa en la Figura 35, fijada para cada sistema. Al ser común a todos los datos contenidos en la ASDU se lo denomina dirección común y se lo interpreta como una dirección de estación, pero se lo puede estructurar con el fin de formar direcciones de estación/sector donde las estaciones individuales se dividen en varias unidades lógicas y una de las direcciones que no se utiliza es la <0>.

Figura 35

Dirección común de ASDU.



La dirección más alta 0xFF o 0xFFFF es global, lo cual significa que todas las estaciones analizarán una ASDU con esta dirección. En la Tabla 22, se enumeran las ASDU que se encuentran restringidas para la utilización de la dirección global. En el momento que se inicia la misma función de aplicación de forma simultánea es cuando se la utiliza.

Tabla 22

Dirección de ASDU.

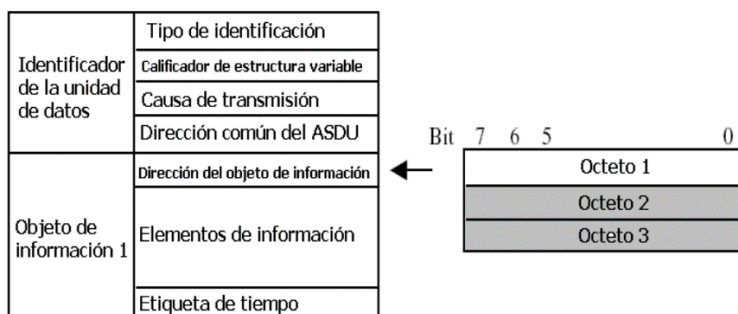
Tipo	Descripción	Referencia
100	Comando de interrogación	Responder con una instantánea de datos del sistema en particular a la hora común
101	Comando de interrogación de contador	Congelar totales en tiempo común
103	Comando de sincronización de relojes	Sincroniza los relojes con la hora común
105	Comando de proceso de reinicio	Reinicio simultáneo

Dirección del objeto de información. Es el primer campo del objeto de información que se encarga de distinguir datos específicos dentro de una estación concreta y su longitud puede ser uno, dos o tres octetos, como se puede observar en la Figura 36. Para el caso de tres octetos, solo se permiten sistemas de direcciones estructuradas y en una estación se le permiten 65 536 direcciones de objetos de información distintos, como el direccionamiento de dos octetos. La dirección del objeto de información de cero se encuentra reservada para la ocasión en que la dirección no sea importante.

En un sistema, los datos específicos se identifican de forma única mediante la combinación de la dirección común y la dirección del objeto de información.

Figura 36

Dirección del objeto de información.



Un ejemplo de cómo podría funcionar esto en la práctica es cuando hay varios Dispositivos electrónicos inteligentes IED conectados a una RTU submaestra, que a su vez está conectada a una estación maestra. Estos podrían ser reconectores en un sistema de distribución. Cada IED tendrá una estructura de datos idéntica internamente, según lo determine el fabricante del dispositivo. Lo más probable es que se trate de algunas decenas de elementos de información o puntos de datos, que podrían ser direccionados por una dirección de objeto de información de un octeto. Sin embargo, como el sistema incluye RTU que tienen muchos más puntos de datos, se utiliza el direccionamiento de dos octetos en todo el sistema.

A nivel de configuración del sistema, habrá un solo modelo para ese tipo de reconector, y la configuración de cada dispositivo en la base de datos del sistema será una cuestión de usar el mapeo de puntos estándar para ese tipo de dispositivo y agregar el número de estación a formar referencias puntuales únicas.

Direccionamiento y encaminamiento de mensajes.

Controlar y monitorear direcciones. Un concepto importante para comprender el direccionamiento según IEC 60870-5 es la diferencia entre las direcciones de control y supervisión. Supongamos que la jerarquía de todo el sistema implica un control centralizado. Cada estación se identifica como estación de control o estación controlada según el protocolo. La estructura de las redes de comunicación suele ser acorde a esto. En un enlace de

comunicación desequilibrado, la estación de control será la estación primaria en el nivel del enlace y la estación controlada será la estación secundaria.

Esto se debe naturalmente a que la estructura jerárquica se compone de varias estaciones controladas, que están controladas por una o al menos varias estaciones de control.

En tal sistema, la estación de control envía mensajes de control, tales como comandos o preguntas, y las acciones y comentarios que recibe la estación de control son el resultado de estos mensajes.

Direccionamiento de ASDU. Este campo de uno o dos octetos se encarga de enviar los mensajes a la dirección de control y definir la estación a la que se apunta la ASDU. No obstante, el campo de dirección común que está en la dirección del monitor tiene la dirección de la estación que devuelve los datos, lo cual es fundamental para que los datos puedan determinarse de forma original y otorgarse a los puntos correctos en las imágenes de datos del sistema.

Para varios casos, una estación, puede actuar como una estación de control, solo para preguntar a la estación maestra buscando datos o para empezar alguna acción en otra estación controlada. Dicho procedimiento se lo conoce como operación de dirección inversa. En cuanto que para una estación de modo dual puede trabajar en la dirección de avance y en la de retroceso. En el momento en que una estación de modo dual transmite una ASDU de control en otra estación, esta debe configurar la dirección de la estación controlada como la dirección de la ASDU común, es fundamental para que la estación requerida pueda reconocer el mensaje como dirigido hacia la misma (Figura 37).

Si se utiliza una operación en dos fases, se requiere una comunicación extra con la estación de control para transmitir un mensaje de confirmación de la acción y un mensaje de ejecución. Sin embargo, los mensajes de dirección del monitor llevan la dirección de la estación

controlada, por lo que no se pueden enrutar las comunicaciones de vuelta a la estación controlada. En cambio, se emplea el octeto de dirección del originador del campo de causa de transmisión para esta intención. Su funcionamiento se describe en los siguientes párrafos y en la Figura 38 y Figura 39.

Figura 37

Comando de control emitido desde RTU de modo dual.

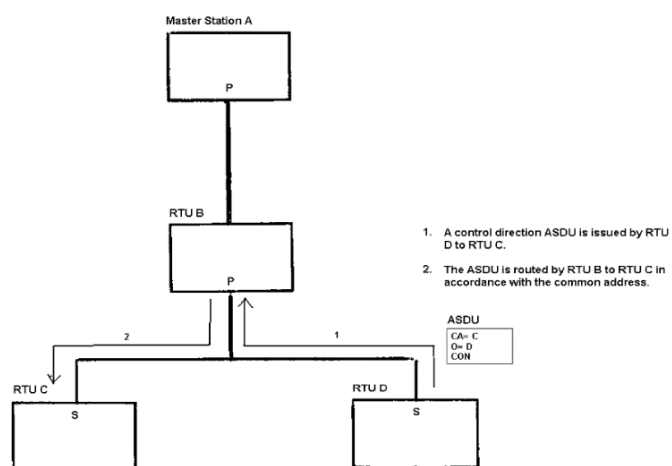
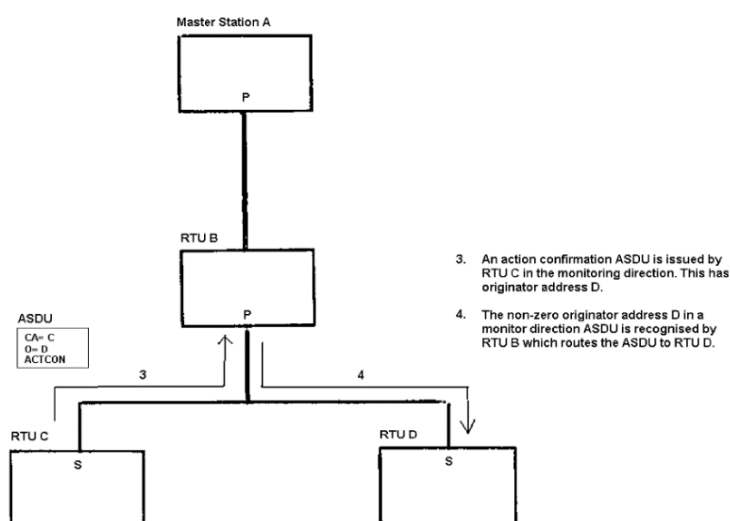


Figura 38

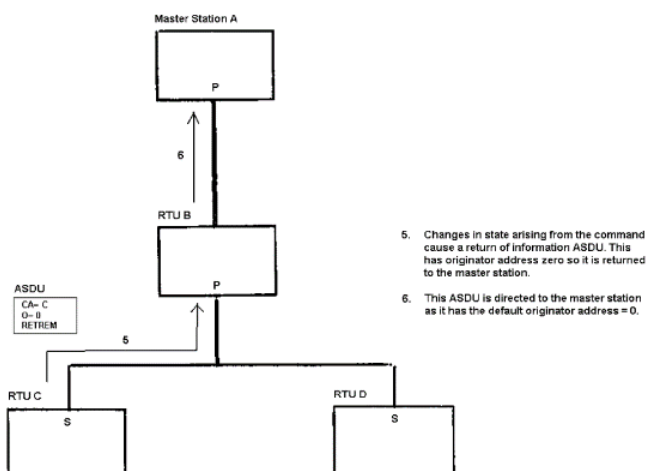
Confirmación de activación ASDU volvió a modo dual RTU.



Al enviar direcciones de control ASDU, la estación de modo dual debe agregar un valor distinto de cero al octeto de dirección de la dirección de transmisión. Aunque ASDU no afecta la dirección de control, el monitor de dirección lo emplea para enviar mensajes de confirmación y finalización al remitente. Cuando la estación controlada retorna una confirmación de acción o cualquier otro mensaje que surge de esta ASDU de control, incluye la dirección de origen de la ASDU de dirección de control en la respuesta de dirección de supervisión. Cada dispositivo de enrutamiento intermedio tiene la responsabilidad de encontrar una dirección diferente de cero en la dirección de seguimiento de ASDU y retornarla de regreso al originador.

Figura 39

Información de monitoreo devuelta al maestro SCADA.



Es obvio que una estación de modo dual en el sistema debe estar en el rango <1..256> debido a que el subcampo de dirección base contiene solo un octeto de longitud. Si no se encuentra dentro del rango, el mapeo entre la dirección inicial y la dirección pública debe emplearse.

Cabe señalar que también pueden surgir algunos cambios dentro de la estación controlada a partir de la acción de control que deben transmitirse a la estación maestra en lugar de a la estación de modo dual que inició la acción de control. Por lo general, estos serían para

transmitir el estado del sistema modificado y también podrían incluir eventos con etiqueta de tiempo. Las ASDU de dirección del monitor que transportan esta información deben dirigirse a la estación maestra y, posiblemente, a otras áreas de la red, si es necesario. Estas ASDU reciben un valor de dirección de origen de cero. Por lo tanto, en un sistema que contiene estaciones de modo dual, es necesario utilizar el subcampo de dirección del originador, y todos los mensajes de dirección del monitor que regresan a la estación maestra tendrán este campo configurado en cero.

Estos conceptos se ilustran en la Figura 37. Esto muestra un sistema con RTU intermedias que actúan como controladores y concentradores de datos, y múltiples estaciones controladas vinculadas a estos. Uno de estos emite un comando a una estación par. Se generan tanto mensajes de confirmación de acción como de cambio de datos, y se enrutan a sus destinos correctos mediante el uso del campo de dirección del originador.

Elementos de información. Los datos de la solicitud se envían en uno o más objetos de datos de la ASDU, como se vio en secciones anteriores del modelo ASDU. Un solo objeto de datos puede contener varios tipos de datos, o un solo objeto puede contener varios tipos de datos, dependiendo del modelo de indicador variable. Sin embargo, el contenido es fundamental para comunicar información jurídica. La traducción del texto de las ASDU a la ley se basa en el sistema de información.

En esta sección se presenta este conjunto de bloques de construcción de información. Estos se mencionan en la siguiente sección cuando se define el conjunto de ASDU. En las siguientes definiciones, deben tenerse en cuenta estas reglas de interpretación:

- Las descripciones de las teclas dan el estado lógico para un bit establecido, es decir, bit = 1.
- Las posiciones de bits en blanco están reservadas y deben borrarse, es decir, bit = 0.

- Las posiciones de bit se han numerado <0..7> para mantener la coherencia dentro de este texto y las potencias de 2 que representan las posiciones. Tenga en cuenta que los documentos IEC en realidad usan <1..8>. Esto es solo una cuestión de definición y no cambia el significado.

El conjunto de elementos de información se enumera en la Tabla 23. A esto le siguen representaciones de cada uno de los elementos de datos, agrupados por tipo general.

Tabla 23

Elementos de información.

Símbolo de tipo general	Descripción
<i>Proceso</i>	
SIQ	Información de punto único con descriptor de calidad
DIQ	Información de doble punto con descriptor de calidad
BSI	Información de estado binario
SCD	Detección de estados y cambios
QDS	Descriptor de calidad
VTI	Valor con indicación de estado transitorio
NVA	Valor normalizado
SVA	Valor escalado
R32-IEEE STD 754	Número de punto flotante corto
BCR	Lectura de contador binario
<i>Protección</i>	
SEP	Evento único de equipos de protección
SPE	Eventos de inicio de equipos de protección

Símbolo de tipo general	Descripción
OCI	Información del circuito de salida del equipo de protección
QDP	Descriptor de calidad para eventos de equipos de protección
<i>Comandos</i>	
SCO	Comando único
DCO	Doble mando
RCO	Comando paso de regulación
<i>Tiempo</i>	
CP56Time2a	Tiempo binario de siete octetos
CP24Time 2 ^a	Tiempo binario de tres octetos
CP16Time2a	Tiempo binario de dos octetos
<i>Calificadores</i>	
QOI	Calificador de interrogación
QCC	Calificador de comando de contrainterrogación
QPM	Calificador de parámetro de valores medidos
QPA	Calificador de activación de parámetros
QRP	Calificador del comando de proceso de descanso
QOC	Calificador de comando
QOS	Calificador del comando de punto de ajuste
<i>Transferencia de archivos</i>	
FRQ	Calificador de archivo listo
SRQ	Calificador de sección lista
SCQ	Calificador de selección y llamada

Símbolo de tipo general	Descripción
LSQ	Calificador de última sección o segmento
AFQ	Acuse de recibo calificador de archivo o sección
NOF	Nombre del archivo
NOS	Nombre de la sección
LOF	Longitud del archivo o sección
LOS	Longitud del segmento
CHS	Suma de verificación
SOF	Estado del expediente
<i>Misceláneas</i>	
COI	Causa de inicialización
FBP	Patrón de bits de prueba fijo, dos octetos

Bits de calidad. No se trata de elementos de información en sí mismos, en cambio se presentan como bits individuales dentro de los elementos de información, que se determinan en esta sección.

Tenga en cuenta que los bits de calidad se configuran o borran independientemente unos de otros. El examen de estas muestras que pueden usarse para diferenciar entre diferentes tipos de situaciones o problemas que pueden estar afectando los datos. Que se utilicen todos dependerá del sistema. Un enfoque simple sería interpretar cualquiera de los bits de calidad que se configuran como "valor incorrecto", mientras que los enfoques más sofisticados pueden diferenciar en función de los bits reales establecidos. Sus significados individuales se explican más adelante.

Bloqueado (BL). Esto significa que el valor del punto es el mismo que tenía antes de ser bloqueado. El bloqueo impide la actualización del valor del punto.

Sustituido (SB). Aquí es donde un valor ha sido sustituido o forzado por entrada manual o de otra manera. Significa que el valor no se deriva de la medición normal.

No tónico (NT). Esto significa que el valor no se actualizó correctamente la última vez que debía actualizarse.

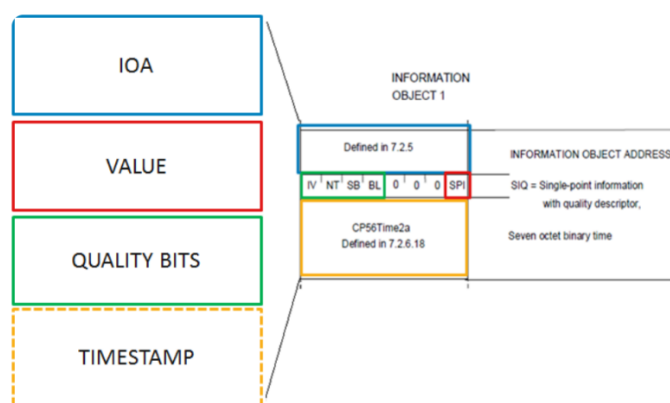
Inválido (IV). Esto indica que el valor no se puede usar porque puede ser incorrecto debido a una falla u otra condición anormal.

Bit de desbordamiento (OV)

Tiempo inválido transcurrido (EI)

Figura 40

Objeto de información.



Software Wireshark

Wireshark es un software de código abierto y gratuito que permite analizar los protocolos de comunicación y a partir de esto, realizar un análisis con el mayor detalle posible. Según (Sharpe et al., 2011) algunas de las razones más relevantes por lo que la gente usa esta aplicación son:

- Lo utilizan los administradores de red para solventar problemas de red..
- Lo utilizan los ingenieros de seguridad de redes para evaluar problemas de seguridad.

- Es empleado por ingenieros de control de calidad para comprobar aplicaciones de red.
- Utilizado por los desarrolladores para eliminar implementaciones de protocolos.
- Los usuarios lo emplean para comprender los componentes internos del protocolo de red.

Figura 41

Ícono del software Wireshark.



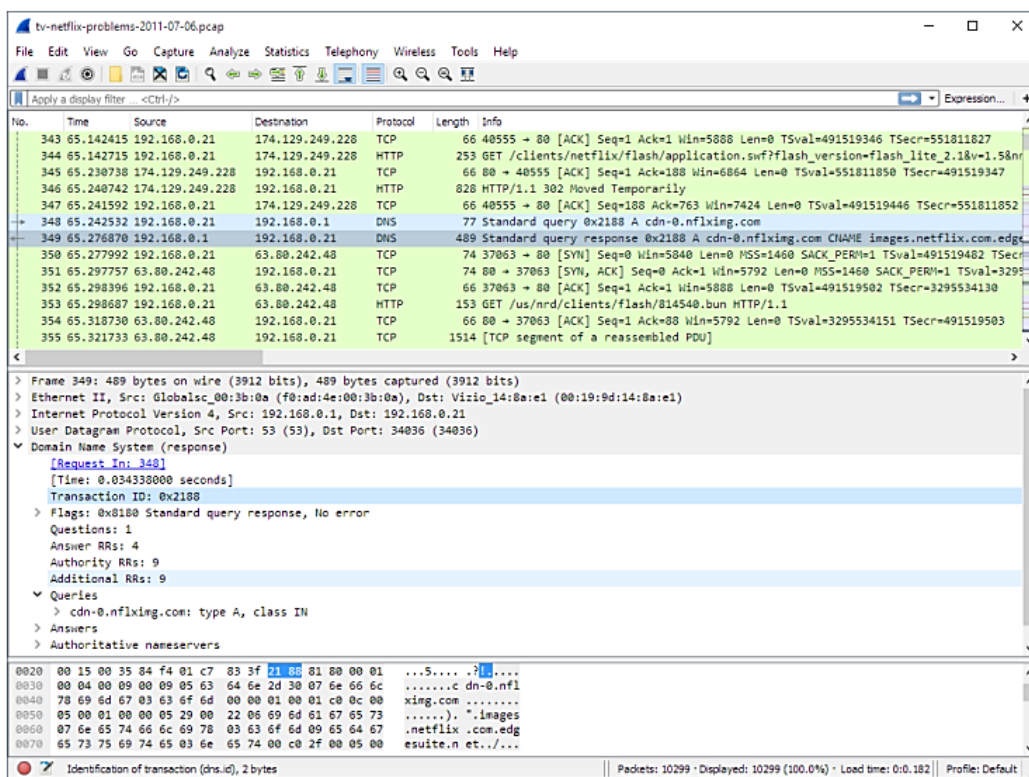
Varios de los principales recursos de Wireshark, según (Sharpe et al., 2011), se listan a continuación:

- Se encuentra disponible para sistemas operativos UNIX y Windows.
- Utiliza una interfaz de red para capturar datos de paquetes en vivo.
- Permite abrir archivos que tengan datos de paquetes capturados con tcpdump/WinDump, Wireshark, entre otros programas.
- Esta aplicación permite importar paquetes desde archivos de texto que contengan volcados hexadecimales de datos de paquetes.
- Permite mostrar paquetes que contienen datos de protocolo muy detallados.
- Permite guardar los datos del paquete que se capturaron.
- Permite la exportación de paquetes en varios formatos de archivos de captura.
- Permite filtrar paquetes.

- Permite la búsqueda de paquetes.
- Permite colorear la visualización de paquetes según los filtros

Figura 42

Interfaz del software Wireshark.



Nota. Tomado de Wireshark User's Guide: Version 4.3.0, por Sharpe, Warnicke y Lamping, 2011.

Capítulo III

Diseño

Diseño de la estación maestro: definición de sus elementos

El principal propósito del banco de pruebas es comprobar la comunicación entre equipos, lo que permite la identificación de las tramas, los datos enviados y recibidos entre el maestro y los esclavos y la comunicación con los dos protocolos seleccionados, por lo que se establecen las siguientes consideraciones de diseño para el maestro:

- Conectividad a través de puerto ethernet.
- Posibilidad de selección del protocolo a utilizarse.
- Conexión con dos esclavos físicos y una computadora como tercer esclavo.
- Envío de solicitudes de datos a los esclavos, mismos cuya recepción se verificarán a través de indicadores luminosos en el maestro.
- Envío de datos a los esclavos a través de interruptores y potenciómetro, mismos que se reflejaran en los esclavos a partir de indicadores luminosos.
- Verificación de la tramas enviadas y recibidas en el computados a través del software de monitoreo de redes Wireshark.

Para la selección de la tarjeta de desarrollo del maestro a utilizarse se consideraron las siguientes consideraciones:

- Posibilidad de comunicación ethernet integrada o a través de un shield de comunicación Ethernet.
- Disponibilidad de librerías para comunicación Modbus TCP.
- Disponibilidad de librerías para comunicación IEC 60870-5-104.

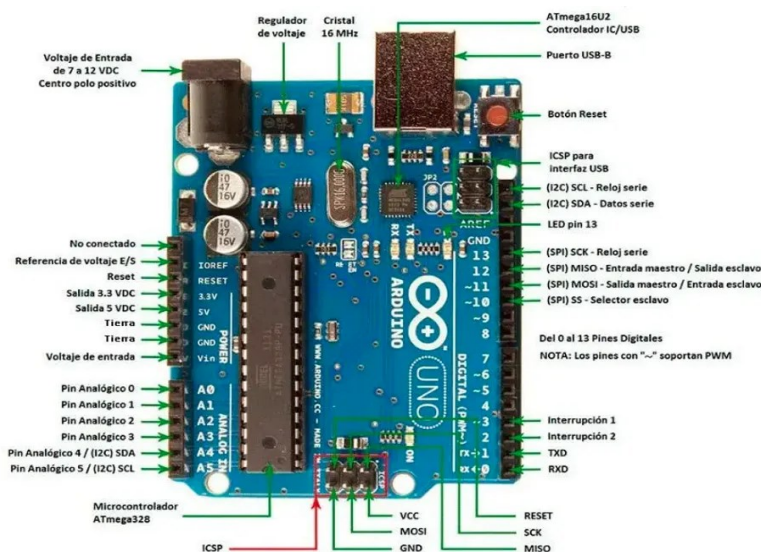
- Disponibilidad en el mercado nacional.
- Costo accesible.

Opciones de tarjetas de desarrollo para la estación maestro. Las tarjetas de desarrollo son dispositivos electrónicos digitales que cuentan con un microcontrolador programable, con todos los elementos necesarios para su funcionamiento y programación, que puede ejecutar diferentes instrucciones y, por lo tanto, son adecuadas para crear todo tipo de dispositivos 'inteligentes', se comunican con softwares de programación las cuales se basan en librerías y recursos que permiten la programación del microcontrolador insertado dentro de la tarjeta, reduciendo el trabajo de armado de la arquitectura necesaria para su funcionamiento al conexionado a través de pines y terminales de entrada o salida de los puertos que dispone el mismo, se han difundido a nivel mundial y son de uso frecuente en el desarrollo de prototipos, por su facilidad de uso, configuración y programación.

Arduino UNO. Es una placa de desarrollo muy popular, está construida en base el microcontrolador ATmega328P, cuenta con 14 pines de entrada/salida digital (6 pueden usarse con PWM), 6 entradas analógicas, un cristal de 16Mhz, un conector jack de alimentación, terminales para conexiones ICSP y un botón de reseteo, como se muestra en la Figura 43 (Lukianova, 2023). Se puede energizar a través de una fuente DC o el puerto USB.

Figura 43

Partes de un Arduino UNO.



Nota. Tomado de Arduino.cl, Principales partes de un Arduino, MCI Electronics, por Lukianova, 2023 (<https://arduino.cl/principales-partes-de-un-arduino/>).

Se programa en Arduino IDE, que es una plataforma muy amigable y que cuenta con una serie de librerías, recursos integrados para la comunicación y programación de la tarjeta de desarrollo.

Dispone de una amplia variedad de shields para diferentes funciones que integran sensores, actuadores y puertos de comunicación, dentro del que se incluye algunos shields para comunicación ethernet, cuenta con diferentes plataformas de usuarios que comparten su experiencia y diferentes librerías que facilitan la comunicación con diferentes equipos y protocolos, dentro de los cuales se incluye librerías para protocolos IEC 60870-5-104 y Modbus TCP.

Raspberry Pi. Básicamente es una computadora de bajo costo con un tamaño compacto por sus múltiples funcionalidades. Básicamente es una computadora de tamaño compacto con un procesador ARM Cortex-172 con cuatro núcleos a 1.5 GHz. Como se puede

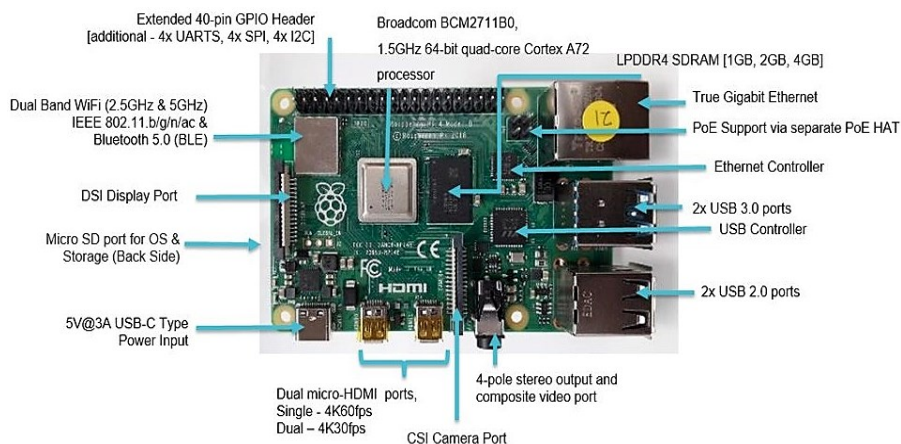
observar en la Figura 44 (OMNIRETRO, 2019), la Raspberry Pi 4 cuenta con conexiones inalámbricas Bluetooth 5.0 y Wi-fi 802.11ac y una conexión de red más rápida mediante Ethernet Gigabit (1000Mb).

Para su alimentación tiene un conector USB-C que permite una mejora de energía a 1.2 A con respecto a sus modelos anteriores, además tiene conectores micro HDMI, USB 2.0 y USB 3.0 (2 por cada uno). También cuenta con GPIO de 40 pines a un nivel de 3.3V que permite la conexión de sensores y actuadores que operen al mismo nivel de voltaje, por lo que si se requiere trabajar con otros voltajes se necesita de un convertor de niveles lógicos.

Una de las dos imágenes que se pueden cargar en la memoria de la Raspberry Pi es Raspbian, el sistema operativo oficial de la placa Raspberry Pi. La otra imagen es NOOBS (New Out Of the Box Software), que permite descargar e instalar varios sistemas operativos en la placa Raspberry Pi de manera fácil empleando lenguajes de programación como Scratch y Python.

Figura 44

Partes de la Raspberry Pi 4.

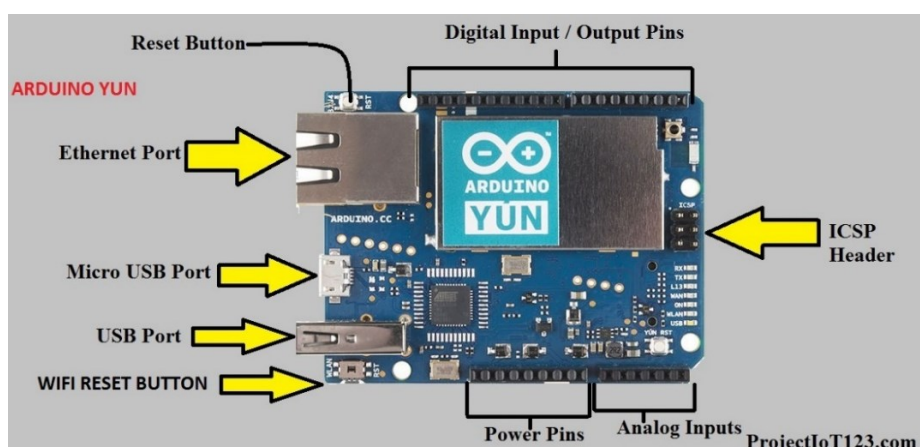


Nota. Tomado de OMNIRETRO, Conoce la nueva Raspberry Pi 4: Especificaciones y Características, por OMNIRETRO, 2019 (<https://omniretro.com/blog/tecnologia/conoce-la-nueva-raspberry-pi-4-especificaciones-y-caracteristicas/>).

Arduino Yun. Es una tarjeta de desarrollo basada en un sistema Linux, el mismo que permite la conexión avanzada a redes y aplicaciones. Como se muestra en la Figura 45 (Kashif, 2019), se basa en el microcontrolador ATmega32u4 y cuenta con Ethernet y soporte WiFi incorporado. También contiene un puerto USB-A, una conexión micro USB, una ranura para tarjeta micro-SD, un cristal oscilador de 16 MHz, un header ICSP y con 3 botones de reinicio.

Figura 45

Partes del Arduino Yun.



Nota. Tomado de ProjectIoT, Arduino Yun for Beginners, por Kashif, 2019 (<https://projectiot123.com/2019/05/14/arduino-yun-for-beginners/>).

Se programa en Arduino IDE que contiene varias librerías y recursos integrados para la comunicación y programación de la tarjeta de desarrollo.

Esta tarjeta de desarrollo también dispone de una amplia variedad de shields para diferentes funciones que integran sensores, actuadores y puertos de comunicación, dentro del que se tiene algunos shields para comunicación ethernet. Además, cuenta con diferentes plataformas de usuarios que comparten su experiencia y diferentes librerías que facilitan la comunicación con diferentes equipos y protocolos, dentro de los cuales se incluye librerías para protocolos IEC 60870-5-104 y Modbus TCP.

Basados en la disponibilidad en el mercado, costo accesible, entorno de programación amigable entre otros se opta por el Arduino Uno, como tarjeta de desarrollo para la estación maestro.

Diseño de las estaciones esclavo: definición de sus elementos

El principal propósito del banco de pruebas es la comunicación entre equipos, lo que permite la identificación de las tramas y los datos enviados y recibidos entre el maestro y los esclavos y los dos protocolos seleccionados, por lo que se establecen las siguientes consideraciones de diseño para los esclavos:

- Conectividad a través de puerto ethernet.
- Posibilidad de trabajar con los protocolos a utilizarse.
- Conexión con el maestro físico, otro esclavo y una computadora como tercer esclavo.
- Recepción de datos del maestro, mismos cuya recepción se verificarán a través de indicadores luminosos en el esclavo.
- Envío de datos a los maestros a través de interruptores, mismos que se reflejaran en el maestro a partir de indicadores luminosos.
- Verificación de la tramas enviadas y recibidas en el computados a través del software de monitoreo de redes Wireshark.

Para la selección de la tarjeta de desarrollo de los esclavos a utilizarse se consideraron las siguientes consideraciones:

- Posibilidad de comunicación Ethernet integrada o a través del shield correspondiente.
- Disponibilidad de librerías para comunicación Modbus TCP.

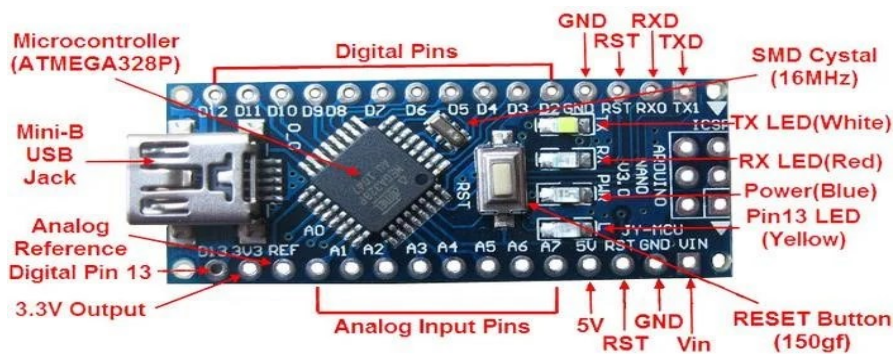
- Tamaño reducido.
- Disponibilidad de librerías para comunicación IEC 60870-5-104.
- Disponibilidad en el mercado nacional.
- Costo accesible.
- Compatibilidad con tarjeta Arduino Uno del maestro.

Opciones de tarjetas de desarrollo para las estaciones esclavos. Basado en los requerimientos planteados y su compatibilidad con la tarjeta Arduino Uno, seleccionada para el maestro, se establecen las siguientes opciones de tarjetas de desarrollo para los esclavos.

Arduino Nano. Es una tarjeta de desarrollo de tamaño compacto basada en el microcontrolador ATmega328P. Tal como se observa en la Figura 46 (Solectroshop, 2020) Además, contiene 14 pines de entrada/salida digital, 6 entradas analógicas, un cristal de 16Mhz, una conexión Mini-USB, terminales para conexiones ICSP y un botón de reseteo.

Figura 46

Partes del Arduino Nano.



Nota. Tomado de Solectroshop. Nueva serie Arduino Nano - qué Arduino es el mejor para comprar en 2020, por Solectroshop, 2020 (<https://solectroshop.com/es/blog/arduino-nano-every-guia-a-la-instalacion-y-proyectos-basicos--n42>)

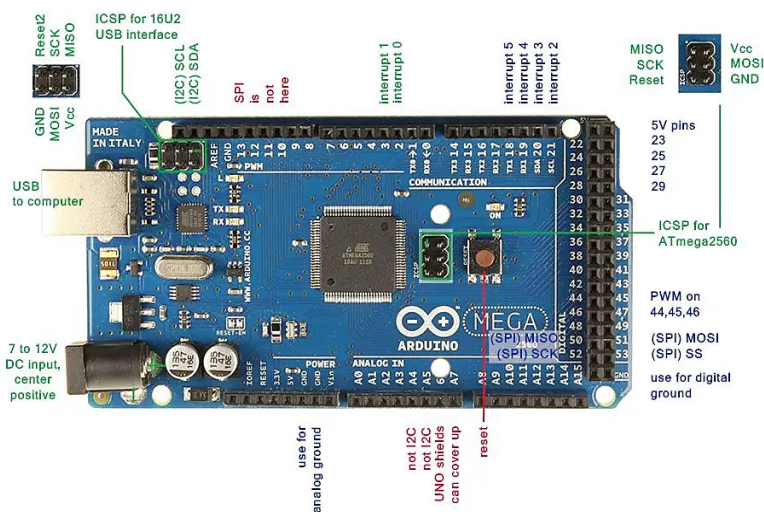
Su programación se la realiza en Arduino IDE, la misma que es una plataforma muy amigable y que cuenta con una serie de librerías y recursos integrados para la comunicación y programación de la tarjeta de desarrollo.

Esta tarjeta de desarrollo cuenta con una amplia gama de shields para diversas funciones que integran sensores, actuadores y puertos de comunicación, incluyendo shields para comunicación ethernet. También cuenta con una variedad de plataformas de usuarios para intercambiar experiencias y librerías para la comunicación con varios equipos y protocolos, incluyendo librerías para protocolos IEC 60870-5-104 y IEC 60870-5-104.

Arduino Mega 2560. Se trata de una placa de desarrollo que utiliza el microcontrolador ATmega2560. La Figura 47 (García, 2023) muestra 54 entradas/salidas digitales, 16 entradas analógicas, un cristal de 16Mhz, una conexión USB, un jack para alimentación DC, un conector ICSP y un botón de reseteo. 15 de las entradas digitales pueden usarse como salidas PWM.

Figura 47

Partes del Arduino Mega 2560.



Nota. Tomado de Panamahitek, Arduino Mega: Características y Capacidades, por García, 2013 (<https://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>)

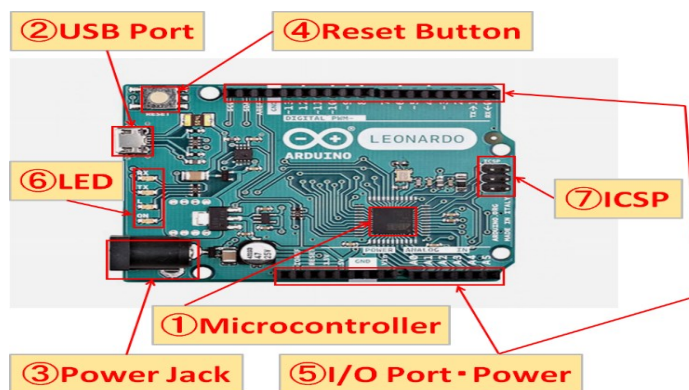
La programación se lleva a cabo a través de Arduino IDE, una plataforma relativamente sencilla que ofrece una amplia gama de librerías y recursos integrados que permiten la comunicación y programación de la tarjeta de desarrollo.

Cuenta con una amplia variedad de shields para cumplir diversas funciones, y entre una de ellas es la comunicación ethernet. También cuenta con una variedad de plataformas de usuarios para intercambiar experiencias y librerías para la comunicación con varios equipos y protocolos, incluyendo librerías para protocolos Modbus TCP e IEC 60870-5-104.

Arduino Leonardo. Tarjeta basada en el microcontrolador ATmega32u4. La Figura 48 (Spiceman, 2022) muestra sus componentes: un oscilador de cristal de 16 MHz, una conexión micro USB, un conector de alimentación, un conector ICSP, un botón de reinicio y 20 pines de entrada/salida digital. De los 20 pines, 7 se pueden usar como salidas PWM y 12 como entradas analógicas.

Figura 48

Partes del Arduino Leonardo.



Nota. Tomado de Spiceman, Arduino Leonardo Specifications/Functions, por Spiceman, 2022 (<https://spiceman.net/arduino-leonardo/>).

La programación la realiza a través de Arduino IDE, una plataforma amigable con el usuario y que ofrece una amplia variedad de librerías y recursos integrados que permiten la comunicación y programación de la tarjeta de desarrollo.

Esta tarjeta de desarrollo cuenta con una amplia gama de shields para diversas funciones que integran sensores, actuadores y puertos de comunicación, incluyendo shields para comunicación ethernet. Además, cuenta con una variedad de plataformas de usuarios para intercambiar experiencias y librerías para la comunicación con varios equipos y protocolos, incluyendo librerías para protocolos IEC 60870-5-104 y IEC 60870-5-104.

Basados en la disponibilidad en el mercado, tamaño compacto, costo accesible, entorno de programación amigable entre otros se opta por el Arduino Nano, como tarjeta de desarrollo para las estaciones esclavos.

Diseño de la red de comunicación: definición de sus elementos

La principal ventaja de contar con una red LAN (red de área local) es que los dispositivos conectados a la red pueden interactuar y comunicarse entre sí sin la necesidad de una conexión a internet. Por tal motivo, montar la red a partir de una conexión Ethernet es una gran opción.

Para que el correcto funcionamiento de una red LAN los dispositivos que lo conforman deben:

- Contar con una dirección MAC propia única.
- Contar una dirección IP asignada y propia del equipo.
- Contar la máscara de red común.
- Contar con la puerta de enlace (gateway) o la dirección IP del router.

Todo dispositivo de red tiene una dirección MAC de 6 bytes, también conocida como dirección física. Esta es asignada de fábrica y es una dirección única, muy similar a un número de serie, todo equipo que posea la posibilidad de conexión en red dispone de una.

La dirección IP es un código de cuatro bytes que identifica o asigna un nombre a un dispositivo en una red, es asignada por el administrador de red, pero lo cual no implica que sea exclusivo de cada dispositivo. Un dispositivo puede recibir una dirección IP diferente según la configuración de la red.

Generalmente una dirección IP tiene una estructura de cuatro bytes separados por un punto, por ejemplo: 192.168.10.6

La máscara de red está compuesta por cuatro bytes, lo que permite entender la forma en que se debe asignar las direcciones IP en la red. Si una red usa como máscara 255.255.0.0, los dos primeros bytes de la dirección IP deben ser iguales para todos los dispositivos en la red. Por ejemplo, las direcciones 10.2.2.3, 10.2.7.6 y 10.2.7.5, en donde los dos primeros bytes, se mantienen iguales para todos los dispositivos.

La puerta de enlace es la dirección IP del router. Por lo general, es que la dirección IP sea la primera en la red, pero puede tener cualquier dirección. Para el caso anterior, la puerta de enlace debería ser 10.2.0.1, por ejemplo.

Es posible asignar la dirección IP, la máscara de subred y la puerta de enlace en un dispositivo de dos maneras:

- Empleando una dirección IP estática: para lo cual se debe configurar manualmente los parámetros de red al dispositivo. Esta opción ofrece la ventaja de que el dispositivo siempre tenga la misma dirección IP.
- Empleando dirección IP dinámica (DHCP): para lo cual el router es el encargado de rellenar todos los parámetros al dispositivo de forma automática, pero no hay la opción de que los dispositivos cuenten con la misma dirección IP.

Para el caso del proyecto se utilizará el primer método, ya que se requiere conocer la dirección IP del Maestro y los Esclavos para poder acceder a ellos a través de la red. Si se

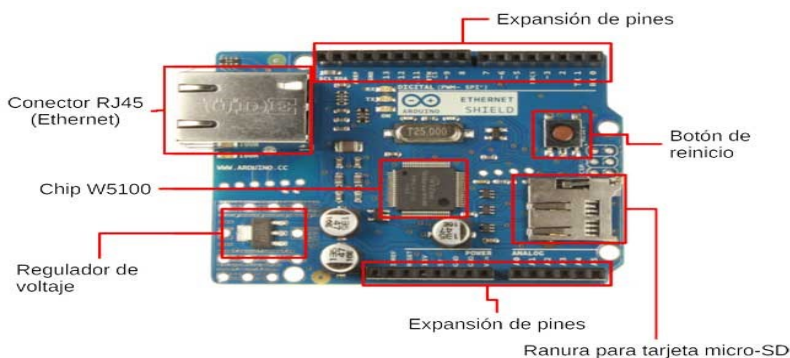
toma la opción de la asignación dinámica cada vez que el router se enciende, el Arduino puede tomar una IP diferente, lo que dificultaría el funcionamiento.

Selección de shields Ethernet para estaciones maestro y esclavos. A partir de todo lo mencionado anteriormente y una vez seleccionadas las tarjetas de desarrollo, tanto para el maestro como para los esclavos, se establece la necesidad de seleccionar los shields ethernet para cada estación, considerando su compatibilidad y que dispongan con las librerías adecuadas para el desarrollo de los protocolos establecidos.

Para el maestro se establece el shield Ethernet W5100, exclusiva para Arduino, tal como se muestra en la Figura 49 (Guerra, 2021). Este módulo puede ser utilizado en las tarjetas desarrollo Arduino Uno, Arduino Mega (1280 o 2560), Arduino Duemilanove (168 o 328) y Arduino Leonardo, permitiendo su conexión a una red gracias a su puerto Ethernet. Además, es compatible con las librerías Ethernet que tiene la IDE de Arduino, por lo que no es necesario de descargar librerías adicionales.

Figura 49

Partes del Shield Ethernet para Arduino.



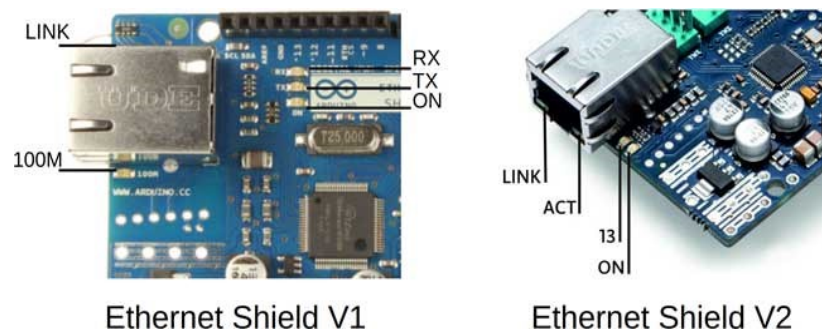
Nota. Tomado de Programarfacil.com, Ethernet Shield Arduino tutorial paso a paso, por Guerra, 2021 (<https://programarfacil.com/blog/arduino-blog/ethernet-shield-arduino/>).

Este shield está disponible en dos versiones, Ethernet Shield V1 y V2, que se basa en el chip Ethernet Wiznet W5100. Además, los módulos del Ethernet shield incluyen una serie de luces indicadoras, como se muestra en la Figura 50 (Guerra, 2021):

- El indicador ON o PWR; indica que la placa Arduino y el Shield están alimentados.
- El indicador 13; conectado al pin 13 del Arduino (solo en la versión 2).
- El indicador LINK; indica la presencia de un enlace de red y pestañea cuando se transmiten o se reciben datos.
- El indicador 100M; indica que la conexión es de 100 Mb/s. Cuando está apagado significa que la red es de 10 Mb/s (este LED solo está presente en la versión 1).
- El indicador ACT; pestañea cuando hay actividad de transmisión o recepción (solo en la versión 2).
- El indicador RX; pestañea cuando se reciben datos (sólo en la versión 1).
- El indicador TX; pestañea cuando se transmiten datos (sólo en la versión 1).

Figura 50

Indicadores del Ethernet shield V1 y V2.



Nota. Tomado de Programafacil.com, Ethernet Shield Arduino tutorial paso a paso, por Guerra, 2021 (<https://programafacil.com/blog/arduino-blog/ethernet-shield-arduino/>).

La conexión del módulo Ethernet con el Arduino Uno es simple ya que este este módulo posee una distribución de pines exacta para únicamente introducir el Arduino directamente como se muestra en la Figura 51 (Guerra, 2021).

Figura 51

Conexión Ethernet Shield y Arduino UNO.

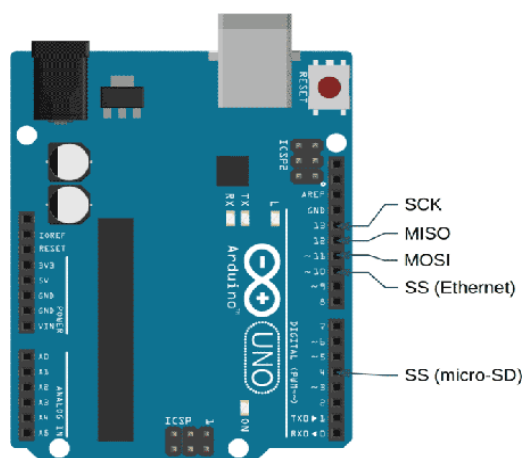


Nota. Tomado de Programarfacil.com, Ethernet Shield Arduino tutorial paso a paso, por Guerra, 2021 (<https://programarfacil.com/blog/arduino-blog/ethernet-shield-arduino/>).

Este shield cuenta con una interfaz serial SPI por lo que al conectarlo al Arduino Uno se debe tener en cuenta los pines del bus SPI de este último, ya que estos pines son exclusivamente para la comunicación con el shield, por lo que ya no se las podrá utilizar para otras funciones. En la Figura 52 se presentan los pines del Arduino Uno para la comunicación con el shield, los mismos que son los pines digitales 11, 12 y 13; el pin 10 es reservado para SS para el chip Ethernet y pin 4 como SS para la tarjeta SD.

Figura 52

Pines de conexión - comunicación Arduino Uno y shield Ethernet W5100.



Para el esclavo se establece el shield Ethernet ENC28J60, como se presenta en la Figura 53 (IHC, 2023), si bien esta desarrollado por Microchip, existen librerías para hacerlo trabajar con las tarjetas desarrollo Arduino Nano, Arduino Uno y Arduino Mega (1280 o 2560). Consume más memoria que el Ethernet Shield, pero para aplicaciones básicas es muy útil. Existe gran información y ejemplos de este módulo en la red, así como diferentes librerías, una de las librerías más usadas es la librería Ethercard, tiene una versión de shield que es compatible con Arduino Nano, por lo que se utiliza esta versión de shield.

Figura 53

Shield Ethernet ENC28J60.

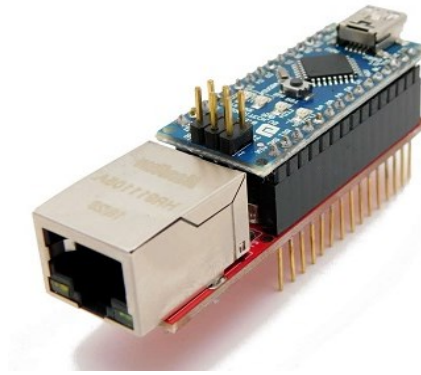


Nota. Tomado de Indian Hobby Center, Ethernet Shield For Nano (Without Nano), por IHC, 2023 (<https://www.indianhobbycenter.com/products/ethernet-shield-for-nano-enc28j60>).

Este shield cuenta con una interfaz serial SPI por lo que al conectarlo al Arduino Nano se debe tener en cuenta los pines SPI de este último, los mismos que son los pines digitales 11, 12 y 13; y el pin 10 es reservado para seleccionar el controlador ENC28J60. Estos pines son exclusivamente para SPI, por lo que ya no se las podrá utilizar para otras funciones. La conexión del módulo ethernet con el Arduino Nano es muy simple ya que este este módulo posee una distribución de pines exacta para únicamente introducir el Arduino directamente como se muestra en la Figura 54 (Nettigo, 2023).

Figura 54

Conexión de Ethernet Shield ENC28J60 con Arduino Nano.



Nota. Tomado de Nettigo Odkryj tajmiki elektroniki. Ethernet Shield for Nano, ENC28J60 (<https://nettigo.eu/products/ethernet-shield-for-nano-enc28j60>).

Para el enlace en red de las estaciones maestro y esclavos se requiere de un equipo como es el switch, de los cuales se optó por el TP-Link Router Inalámbrico TL-WR740N, debido a la disponibilidad inmediata en el mercado. Este equipo es un dispositivo que permite la conexión de red de cable/inalámbrico en el que se incluye un router y un switch de 4 puertos, como se visualiza en la Figura 55 (TP-Link, 2018), y cuenta con una velocidad de transmisión de datos de hasta 150Mbps.

Figura 55

Switch de 4 puertos TP-Link Router Inalámbrico TL-WR740N.



Nota. Tomado de TP-Link, TL-WRN740N Router Inalámbrico N 150 Mbps, por TP-Link, 2018 (<https://www.tp-link.com/ar/home-networking/wifi-router/tl-wr740n/>)

Para la comunicación entre los equipos una vez seleccionados los equipos de la red, se procede a la selección de las librerías de comunicación tanto para el maestro como para el esclavo, considerando que al ser shields diferentes cada uno requerirá de sus propias librerías:

Librerías de Comunicación para el Maestro

Librería Ethernet. Está librería está específicamente diseñada para funcionar con las dos versiones del shield Ethernet o cualquier otro módulo que esté basado en los chips W5100, W5200 o W5500.

Dicha librería cuenta con una serie de clases para operar con el módulo Ethernet, incluidas las siguientes:

- Ethernet; permite inicializar la librería Ethernet y configura la red.
- EthernetServer; permite recibir y enviar datos a un cliente conectado y este puede ser un navegador web o una aplicación móvil.

- EthernetClient; es empleada cuando la tarjeta de desarrollo funciona como cliente de otro servidor o cuando un servidor recibe.

Principales funciones de la clase Ethernet. Ethernet es una clase de tipo estática, lo que significa que no es necesario crear una instancia ni un objeto de clase. La clase Ethernet se escribe directamente y luego se llama a la función correspondiente. Algunas de estas características son:

- Ethernet.init(); permite la configuración del pin digital que se usa como SS para controlar el módulo Ethernet. Dado que se está utilizando un shield Ethernet, para este caso, no es requerido usar esta función porque la clase Ethernet emplea de forma predeterminada el pin 10. Su sintaxis sería:

```
Ethernet.init(SSpin);
```

Siendo: SSpin el pin utilizado para controlar el chip del Ethernet Shield.

- Ethernet.begin(); esta función inicia la librería Ethernet y establece los parámetros de red del módulo. Dependiendo del número de parámetros que admite, se le puede llamar de cinco maneras diferentes. Cinco parámetros están disponibles para la primera sobrecarga de la función:

```
Ethernet.begin(mac, ip, dns, gateway, subnet);
```

Siendo:

- mac: dirección MAC del Ethernet Shield.
- ip: dirección IP que se asignará al Ethernet Shield.
- dns: dirección IP del servidor DNS (usualmente es la dirección IP del router).
- gateway: puerta de enlace.

- subnet: máscara de subred.

La Figura 56 muestra cómo un usuario de Windows puede abrir la consola y ejecutar el comando ipconfig para obtener la dirección IP y la máscara de red.

Figura 56

Obtención de la dirección Ip y la máscara de red en PC.

```

C:\Users\Jose>ipconfig
Configuración IP de Windows

Adaptador de LAN inalámbrica Wi-Fi:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 2:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de Ethernet Ethernet:
    Sufijo DNS específico para la conexión. . :
    Vinculo; dirección IPv6 local. . . : fe80::6873:38aa:8663:8a3e%9
    Dirección IPv4. . . . . : 192.168.1.103
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . : fe80::c671:54ff:feae:d200%9
                                           192.168.1.1

Adaptador de Ethernet Ethernet 2:
    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :
  
```

Librería ModbusTCP. La librería Modbus TCP, para el Shield Ethernet, es muy similar a la librería Modbus compatible con el Shield ENC29J60, en cuanto a su contenido e instrucciones en su comunicación como esclavo en el protocolo, por lo que en ese aspecto la descripción es similar a la presentan más adelante.

Sin embargo, donde difiere de la librería mencionada está en el hecho de que esta librería admite la comunicación en modo Maestro, por lo que se establece la utilización de esta librería para el Maestro de la comunicación, considerando que la librería es limitada en cuanto a las comunicaciones que se pueden realizar como maestro respecto a los comandos que se pueden realizar en el modo Esclavo, lo que a su vez limita las opciones de comunicación del sistema, aspecto que se describe a continuación.

Para la comunicación Modbus, como maestro, la librería tiene la instancia ModbusMaster, misma que contiene las siguientes instancias:

- void frameHreg(word offset, word value), para envío de datos en registros.
- void frameCoil(word offset, bool value); para envío de datos en bobinas.

Por esta situación o limitación de la librería es que la comunicación Modbus del sistema se realiza de forma unidireccional, en apariencia, ya que para validar la recepción del dato el esclavo responde al maestro una vez recibida una trama.

En el encabezado, se requiere la declaración de las direcciones IP de los esclavos a conectarse con el Maestro, como se presenta en la Figura 57.

Figura 57

Declaración de direcciones IP de las estaciones esclavos.

```
// Direcciones IP de los clientes
byte dst1[] = { 192, 168, 1, 150 }; //ESCLAV01
byte dst2[] = { 192, 168, 1, 160 }; //ESCLAV02
byte dst3[] = { 192, 168, 1, 100 }; //ESCLAV03 (PC)
```

En la Figura 58 se muestra que, para la comunicación del maestro con diferentes esclavos, deben declararse cuantos objetos de tipo ModbusMasterIP, como esclavos se vayan a conectar al maestro.

Figura 58

Declaración de objetos de comunicación Modbus.

```
// declaración de objetos de comunicación modbus, uno para cada cliente
ModbusMasterIP mbm1;
ModbusMasterIP mbm2;
ModbusMasterIP mbm3;
```

Para inicializar los objetos, se utiliza el comando `config()`, uno para cada objeto declarado, para una mejor referencia, se visualiza la inicialización en la Figura 59.

Figura 59

Inicialización de objetos de comunicación Modbus.

```
// Inicialización de las comunicaciones Modbus con los clientes
mbm1.config();
mbm2.config();
mbm3.config();
```

Finalmente, en la Figura 60, se muestra que, para realizar el envío de datos de la estación maestro a las estaciones esclavos, deben utilizarse los comandos `sendCoil` para el envío de bobinas y `sendHreg` para el envío de datos analógicos, en este caso tipo registro para el protocolo.

Figura 60

Envío de datos desde la estación maestro a las estaciones esclavos.

```
//Envío de datos al Cliente 1
mbm1.sendCoil(dst1, SWITCH1_ISTS, digitalRead(switchPin1));
mbm1.sendCoil(dst1, SWITCH2_ISTS, digitalRead(switchPin2));
mbm1.sendHreg(dst1, POT1_IREG, analogRead(PotPin1));
```

Librería IEC60870-5-104. La librería implementa la comunicación a través del protocolo IEC60870-5-104 utilizando tarjetas de desarrollo Arduino, se puede utilizar múltiples instancias de objetos maestro y esclavo para conectar Arduino con PLC, RTU u otros ARDUINOS que utilicen este protocolo. Esta librería está limitada a los siguientes comandos determinados en la estructura de comunicación del protocolo.

- `M_SP_NA_1`; Información de punto único.
- `M_DP_NA_1`; Información de punto doble.
- `M_ME_NA_1`; Valor medido, valor normalizado.

- C_SC_NA_1; Comando único.
- C_DC_NA_1; Comando doble.
- C_IC_NA_1; Comando de interrogación.
- M_EI_NA_1; Fin de inicialización.

Se realizaron algunas modificaciones a la librería para que sea compatible con el Shield Ethernet (estación maestro) y Ethernet ENC28J60 (estaciones esclavos), únicamente para los comandos equivalentes en protocolo MODBUS.

Librerías de Comunicación para los Esclavos

Librería EtherCard. Es un controlador para el chip Microchip ENC28J60, compatible con Arduino IDE. En la misma se proporcionan rutinas de alto nivel para permitir una variedad de propósitos, incluida la transferencia simple de datos hasta el manejo HTTP. Esta librería sólo es compatible con el hardware basado en el chip ENC28J60. Además, utiliza la interfaz SPI del microcontrolador y necesita de un pin dedicado para CS, además de los pines SO, SI y SCK de la interfaz SPI.

Figura 61

Diagrama de conexión del Ethernet Shield ENC28J60 a la red.



Se utiliza a la instancia ether de la librería EtherCard definida globalmente y puede usarse para acceder a la librería. Para iniciar la llamada a la librería se utiliza el comando ether.begin(). En este caso su uso se limita como una librería que es requerida por las librerías Modbus TCP de este shield, que es utilizada dentro de la librería ModbusIP_ENC28J60.h.

En principio se configura la dirección Mac y la IP, se puede asignar cualquier valor, evitando duplicar con algún otro equipo conectado a la red, acorde a la distribución de direcciones que se realice, en void setup(), como se puede observar en la Figura 62.

Figura 62

Configuración dirección MAC e IP.

```
ModbusIP mb;
void setup() {
  //dirección MAC
  byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF,0xFE, 0xED};
  //dirección IP
  byte ip [] = {192,168,1,160};
  mb.config (mac,ip); // se confirma las direcciones
```

En void loop() se dará el uso del módulo, en este caso no es relevante profundizar en este aspecto dado que la librería Modbus se encarga de la conexión con la librería Ethercard.

Librería Modbus-Ethernet para ENC28J60. Esta librería que permite que Arduino se comunique mediante protocolo Modbus, actuando como esclavo utilizando una comunicación TCP/IP para el shield Ethernet ENC28J60. En este caso el encabezado para la utilización de la librería, necesita de la librería Ethercard.h y Modbus.h, tal como se presenta en la Figura 63.

Figura 63

Encabezado para el uso de librería Modbus-Ethernet para ENC28J60.

```
1  #include <EtherCard.h>
2  #include "Modbus.h"
3  #include "ModbusIP_ENC28J60.h"
```

En la Figura 64 se observa cómo se establece el registro Modbus para representar el interruptor a partir de la bobina de salida digital y la salida del registro donde se recibirá el registro de una variable analógica.

Figura 64

Registro Modbus.

```
4  const int LAMP1_COIL = 1;
5  const int LAMP2_COIL = 2;
6  const int POT_HREG = 1;
```

Además, en la Figura 65, se presenta la creación de la instancia de mb (ModbusEthernet) que se utilizará para el llamado a los comandos y rutinas establecidos en la librería para la comunicación Modbus TCP. El desarrollo del uso e instrucciones de la librería se realiza más adelante.

Figura 65

Instancia mb ModbusEthernet.

```
// ModbusIP object

ModbusIP mb;
void setup() {
//dirección MAC
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF,0xFE, 0xED};
//dirección IP
byte ip [] = {192,168,1,160};
mb.config (mac,ip);// se confirma las direcciones
```

Capítulo IV

Implementación

Una vez seleccionados los elementos principales (tarjetas de desarrollo y shields para las estaciones maestro y esclavos) que se utilizarán para el desarrollo del proyecto, se inicia con la implementación del banco de pruebas. En el mismo se añaden elementos complementarios que servirán para alimentar a cada estación, simular el envío y recepción de señales, a través de sensores y actuadores.

Implementación del hardware

En principio se escoge la manera en que se van a alimentar cada una de las estaciones, para lo cual, por facilidad se utiliza la alimentación de cada una de las tarjetas de desarrollo antes seleccionadas y en base a esto se conoce que estas tarjetas se pueden alimentar de forma similar a través de las siguientes opciones:

- Alimentación USB, que admite 5V de entrada que se pueden obtener al conectar a la PC o a otro dispositivo compatible con USB.
- Alimentación por medio de los pines GND y Vin, que admite un rango de 6V a 12V de entrada a través de una fuente externa.
- Alimentación por medio de los pines GND y 5V, que admite 5V de entrada a través de una fuente externa.

Por facilidad de conexión e integración de las fuentes de todas las estaciones, además de tener en cuenta que se utiliza una computadora como otra estación, se opta por utilizar la alimentación USB de las tarjetas de desarrollo seleccionadas. Para lo cual se añade un HUB USB para concentrar todas las alimentaciones de las estaciones, tal como se observa en la Figura 66.

Figura 66

Conexión de las alimentaciones de las estaciones maestro y esclavos.



Estación maestro. Dado que la estación maestro se encarga de monitorear y controlar señales tanto digitales como analógicas, para el banco de pruebas se simulan dichas señales mediante la implementación de interruptores y potenciómetros, en donde el dato digital se representa en el accionamiento de los interruptores de dos pines y el dato analógico se representa con la variación de la intensidad de los potenciómetros. Además de la conexión del interruptor selector con llave para el cambio de protocolo de comunicación.

Listado de elementos para la implementación de la estación maestro. Los elementos que componen a la estación maestro son los siguientes:

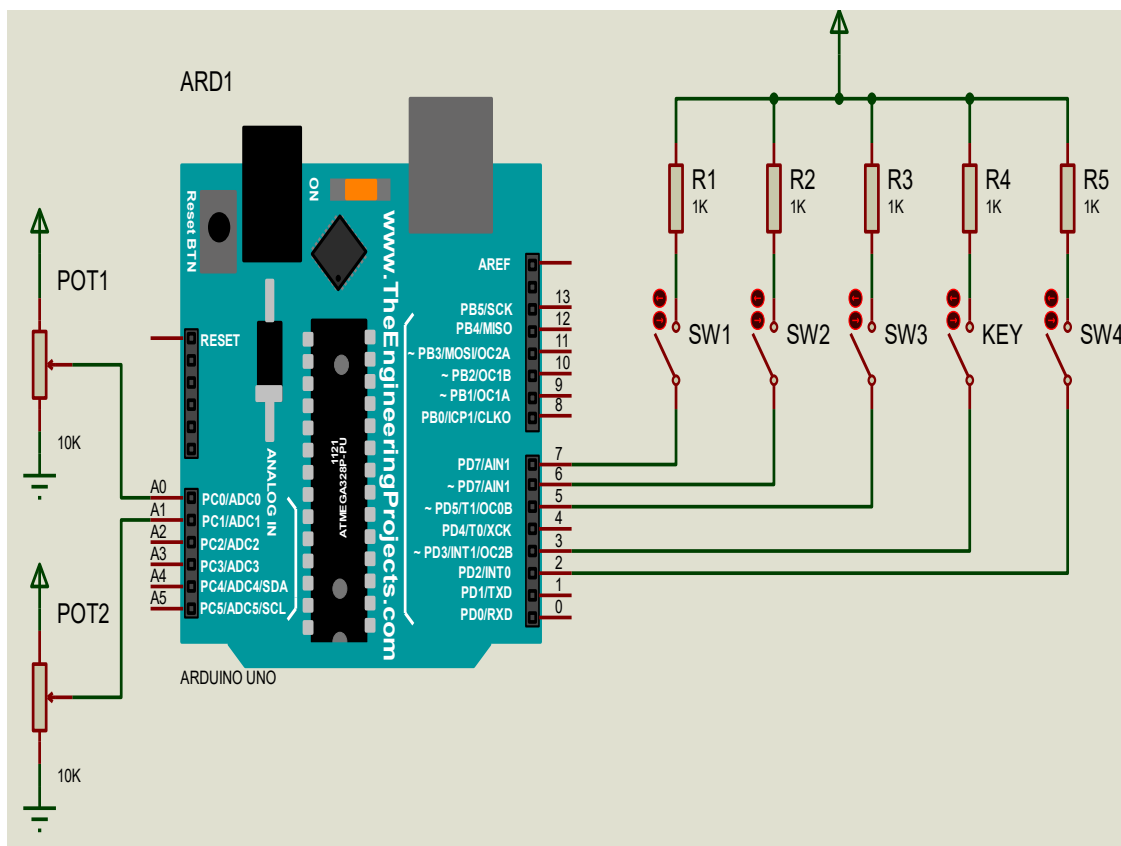
- Arduino Uno.
- Ethernet Shield W5100 V1.
- Interruptor selector con llave (1 unidad, conectada a las estaciones maestro y esclavos).
- Resistencias 1k Ω (5 unidades).

- Interruptor 2 pines NC (4 unidades, dos para cada esclavo).
- Potenciómetros 10k Ω (2 unidades, uno para cada esclavo).
- Cables para protoboard tipo hembra-hembra.

En la Figura 67 se presenta la conexión de cada uno de los elementos antes mencionados para conformar la estación maestro. Donde el Arduino Uno y el Ethernet Shield W5100 V1 se conectan entre sí teniendo en cuenta la similar distribución de sus pines. Los interruptores de dos pines y el interruptor selector con llave son asignados a los pines digitales del módulo (2, 3, 5, 6 y 7). Los potenciómetros que van a simular las señales analógicas son asignados a los pines analógicos del módulo (A0 y A1).

Figura 67

Diagrama de conexión de estación maestro.



Estaciones esclavo. En vista de que las estaciones esclavo se encargan de la recepción de datos digitales y analógicos del maestro y el accionamiento de sus indicadores en función de los datos recibidos, para el banco de pruebas se simulan dichas acciones mediante la implementación de diodos led, en donde el dato digital se representa en el encendido y apagado de los leds y el dato analógico se representa en la variación de la intensidad luminosa del led correspondiente. También está conectado al interruptor selector con llave para el cambio de protocolo de comunicación.

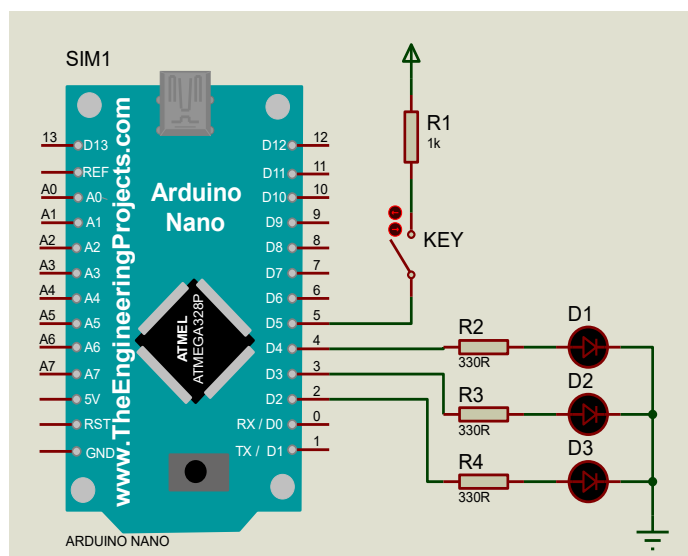
Listado de elementos para la implementación de las estaciones esclavo. Los elementos que componen a las estaciones esclavos son los siguientes:

- Arduino Nano.
- Shield Ethernet ENC28J60.
- Interruptor llave SELECTOR (1 unidad, conectada a las estaciones maestro y esclavos).
- Resistencias 330Ω (3 para cada estación esclavo).
- Diodos Led (3 para cada estación esclavo).
- Cables para protoboard tipo hembra-hembra.

En la Figura 68 se puede observar la conexión de los elementos enlistados que conforman las estaciones esclavo. Donde el Arduino Nano y el shield Ethernet ENC28J60 tienen un conexionado sencillo sólo hay que tener en cuenta la distribución de sus pines. El interruptor selector con llave es conectado al pin digital del módulo D5, recordar que es el mismo interruptor de la estación maestro. Los diodos Led son asignados a los pines digitales D2, D3 y D4 conectadas con sus respectivas resistencias de 330Ω .

Figura 68

Diagrama de conexión de estaciones esclavo.



Elaboración de caja o estuche. Se tiene en cuenta que el banco de pruebas debe ser portátil, por lo que el dispositivo debe ser liviano optando así por un estuche de material plástico, en el cual se realizan las adecuaciones necesarias para colocar los elementos correspondientes a cada una de las estaciones que se enlistaron anteriormente. En primer lugar, se colocan interruptores, diodos led, resistencias y potenciómetros que van posicionados en la parte frontal, tal como se observa en la Figura 69.

Figura 69

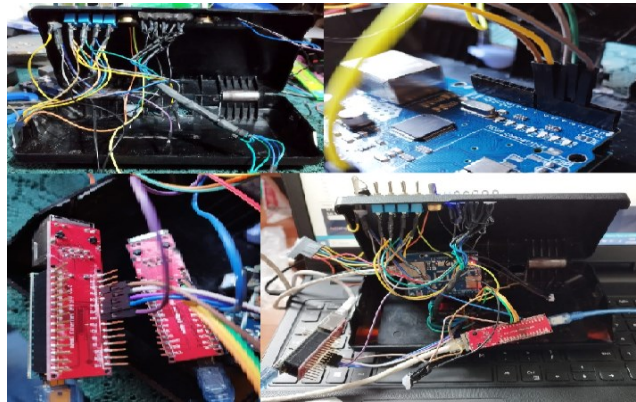
Adecuación del estuche para colocación de elementos.



Se empieza a realizar el cableado desde los elementos ya colocados para el respectivo conexionado en cada uno de los módulos, tomando en consideración las alimentaciones de cada elemento y las tierras comunes para la interconexión de los módulos, tal como se visualiza en la Figura 70.

Figura 70

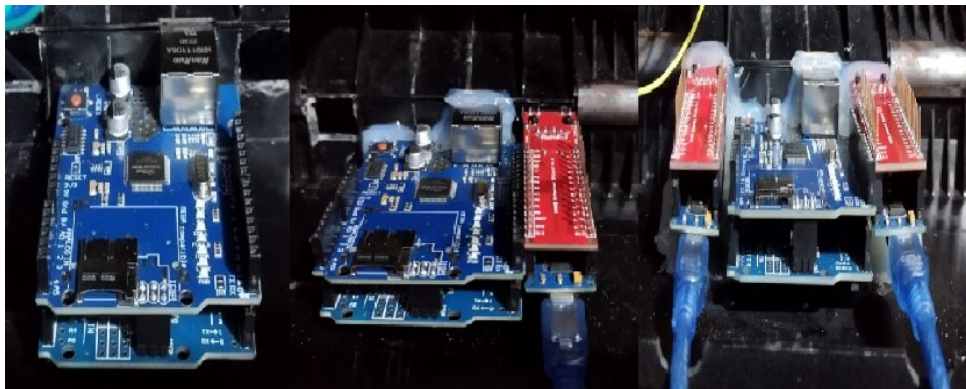
Conexionado de las estaciones maestro y esclavos.



En la Figura 71 se observa la colocación de las tarjetas de desarrollo de cada una de las estaciones en el estuche, fijándolas en sus espacios antes designados para posteriormente realizar su conexión respectiva y finalmente situar los cables en el estuche de modo que queden firmes y acomodados.

Figura 71

Colocación de las tarjetas de desarrollo en estuche.



El banco de pruebas se le dan los últimos detalles, como se puede observar en la Figura 72, para lo cual se coloca un papel adhesivo para recubrir todo el estuche, donde además se ubican las etiquetas de cada uno de los interruptores e indicadores de cada una de las estaciones de la siguiente manera:

- M1.1 y M1.2; interruptores que envían los datos digitales desde la estación maestro a la estación esclavo 1.
- M2.1 y M2.2; interruptores que envían los datos digitales desde la estación maestro a la estación esclavo 2.
- MP1, potenciómetro que envía los datos analógicos desde la estación maestro a la estación esclavo 1.
- MP2, potenciómetro que envía los datos analógicos desde la estación maestro a la estación esclavo 2.
- S1.1 y S1.2; indicadores que reciben los datos digitales enviados desde la estación maestro hacia la estación esclavo 1.
- S2.1 y S2.2; indicadores que reciben los datos digitales enviados desde la estación maestro hacia la estación esclavo 2.
- SP1, indicador que recibe los datos analógicos enviados desde la estación maestro hacia la estación esclavo 1.
- SP2, indicador que recibe los datos analógicos enviados desde la estación maestro hacia la estación esclavo 2.
- SEL, interruptor para el cambio de protocolo a usarse (Modbus TCP o IEC 60870-5-104).

Figura 72

Banco de pruebas etiquetado.



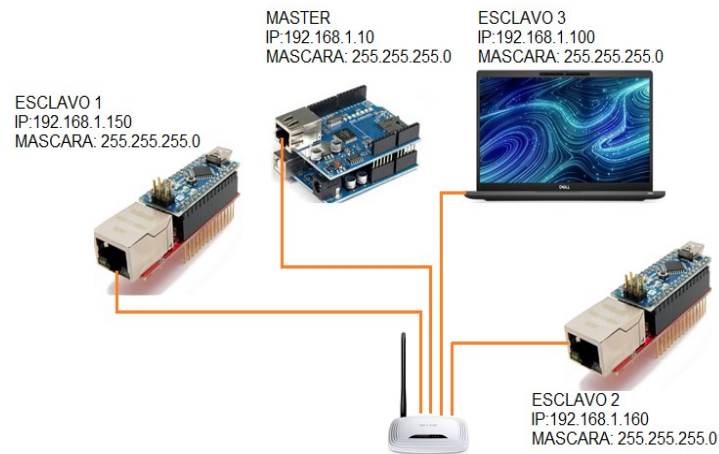
Implementación del software

Para realizar la programación de cada una de las estaciones, en primer lugar, se desarrolla la topología de red física a implementarse. El tipo de topología a utilizarse es el tipo estrella ya que el switch (TP Link) se encarga de administrar las conexiones entre las estaciones, tal como se observa en la Figura 73. A su vez, para cada una de las estaciones, se establecen las diferentes direcciones IP como se enlista a continuación:

- IP del Maestro: 192.168.1.10
- IP del esclavo 1: 192.168.150
- IP del esclavo 2: 192.168.160
- IP del esclavo 3 (PC): 192.168.100

Figura 73

Topología en estrella del proyecto.



Comandos principales para librería Modbus. A su vez es necesario conocer la estructura y funciones de algunos de los comandos principales y primordiales que son empleados dentro de las librerías, que serán necesarias para la implementación del software de cada estación:

- `mb.addCoil (Lamp1Coil)`; agrega el registro tipo bobina (salida digital), el mismo que se encarga de activar el LED o lámpara y verificar su estado. La librería permite establecer un valor inicial para el registro, por lo que, al utilizar este comando de esta forma, el valor predeterminado es falso.
- `mb.addCoil (Lamp1Coil, true)`; para este caso, el registro se agrega y se establece en verdadero.
- `mb.task ()`; este comando se encarga de responder solicitudes y cambiar los registros si es necesario; se la llama solo una vez, al principio del ciclo.
- `digitalWrite (LedPin, mb.coil (Lamp1Coil))`; el valor del registro `Lamp1Coil` se utiliza para controlar la lámpara o el LED.

Del mismo modo, existen otros comandos disponibles en la librería que serán útiles para la implementación y que por ende es necesario conocer su estructura y funcionamiento:

- Comandos que agregan registros y configuran el valor inicial si se especifica.
 - void addCoil (offset word, bool value)
 - void addHreg (offset word, word value)
 - void addIsts (offset word, bool value)
 - void addIreg (offset word, word value)

- Comandos que establecen un valor para el registro.
 - bool setCoil (offset word, bool value)
 - bool setHreg (offset word, word value)
 - bool setIsts (offset word, bool value)
 - bool setIReg (offset word, word value)

- Comandos que devuelven el valor de un registro.
 - bool coil (offset word)
 - word hreg (word offset)
 - bool ists (offset word)
 - word ireg (word offset)

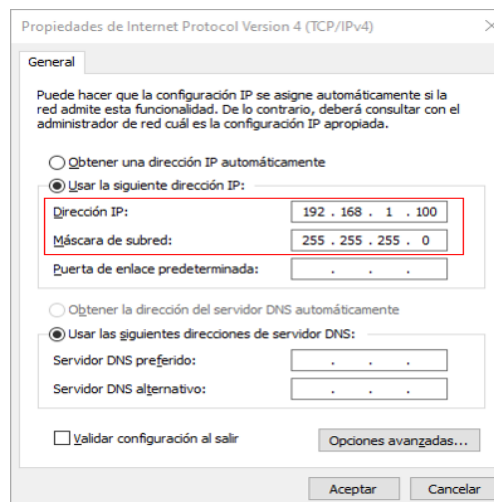
Comandos principales para la librería IEC60870-5-104. En cuanto a la estructura y funciones de algunos de los comandos principales y primordiales empleados dentro de las librerías del protocolo IEC60870-5-104, se tiene:

- `iec.addCoil (Lamp1Coil)`; agrega el registro tipo bobina (salida digital), el mismo que se encarga de activar el LED o lámpara y verificar su estado. La librería permite establecer un valor inicial para el registro, por lo que, al utilizar este comando de esta forma, el valor predeterminado es falso.
- `iec.addCoil (Lamp1Coil, true)`; para este caso, el registro se agrega y se establece en verdadero.
- `iec.task ()`; este comando se encarga de responder solicitudes y cambiar los registros si es necesario; se la llama solo una vez, al principio del ciclo.

Configuración de direcciones IP de las estaciones. Para cada una de las estaciones se debe configurar una dirección IP que debe ser única para no tener inconvenientes con la comunicación y que fueron asignadas anteriormente tal como se presenta en la Figura 73. A excepción de la estación esclavo 3 (PC), que su configuración se la realiza desde las propiedades de red e Internet, tal como se muestra en la Figura 74, la configuración de la IP de cada una de las estaciones se las realiza a través del IDE de Arduino, en el área de configuración (`void setup()`).

Figura 74

Configuración de la IP del esclavo 3 (PC).



Para una mejor comprensión del funcionamiento del banco pruebas, la visualización del funcionamiento de los protocolos dentro del mismo será con lógica inversa, es decir, lo contrario la una con la otra. Tal es que cuando se trabaje con el protocolo de comunicación Modbus, los interruptores que envían los datos digitales desde la estación maestro hacia las estaciones esclavos, al estar cerrados (posición hacia arriba) los leds correspondientes se encienden. En cambio, para el protocolo de comunicación IEC60870-5-104, los interruptores que envían los datos digitales, al estar abiertos (posición hacia abajo) los leds correspondientes se encenderán.

Programación de la estación maestro. En la Figura 75 se observa la declaración inicial necesaria para cada uno de los protocolos, que son descritos a continuación:

- Se declaran las librerías a utilizarse para la comunicación Ethernet y para la comunicación de cada uno de los protocolos (Modbus.h, ModbusIP2.h, IEC104 e IEC104IP.h).
- Se declaran las direcciones IP de todas las estaciones esclavos con las que se van a comunicar la estación maestro.
- Se declaran las direcciones de los interruptores y potenciómetros que se encargan de transmitir los datos.
- Se asignan los pines de entradas digitales y analógicas correspondientes a cada uno de los interruptores y potenciómetros.
- Se declaran los objetos para la comunicación Modbus TCP e IEC60870-5-104.

Figura 75

Declaración de librerías, objetos y variables de la estación esclavo.

```

1  #include <SPI.h>
2  #include <Ethernet.h>
3  #include "Modbus.h"
4  #include "ModbusIP2.h"
5  #include "IEC104.h"
6  #include "IEC104IP.h"
7
8
9  // Direcciones IP de los clientes
10 byte dst1[] = { 192, 168, 1, 150 }; //ESCLAVO1
11 byte dst2[] = { 192, 168, 1, 160 }; //ESCLAVO2
12 byte dst3[] = { 192, 168, 1, 100 }; //ESCLAVO3 (PC)
13
14
15 // direcciones de los interruptores que seran transmitidas
16 const int SWITCH1_ISTS = 1;
17 const int SWITCH2_ISTS = 2;
18
19 // direcciones de los potenciómetros que seran transmitidos
20 const int POT1_IREG = 1;
21 const int POT2_IREG = 2;
22
23 // asignación de pines de entrada del arduino
24 const int switchPin1 = 2;
25 const int switchPin2 = 5;
26 const int switchPin3 = 6;
27 const int switchPin4 = 7;
28 const int switchPin5 = 3;
29
30 // asignación de entradas analógicas del arduino
31 const int PotPin1 = A0;
32 const int PotPin2 = A1;
33
34 // declaración de objetos de comunicación modbus, uno para cada cliente
35 ModbusMasterIP mbm1;
36 ModbusMasterIP mbm2;
37 ModbusMasterIP mbm3;
38
39 IEC104MasterIP iec1;
40 IEC104MasterIP iec2;
41 IEC104MasterIP iec3;

```

En la sección de configuración del programa, como se presenta en la Figura 76, se configuran los interruptores como entradas, se configuran las direcciones MAC e IP de la estación maestro, en base a las direcciones asignadas anteriormente, y se inicializan las comunicaciones Ethernet y las comunicaciones Modbus TCP e IEC60870-5-104 con las estaciones esclavos.

Figura 76

Configuración entradas, MAC e IP, e inicialización de comunicaciones.

```

44 void setup() {
45     pinMode(switchPin1, INPUT);
46     pinMode(switchPin2, INPUT);
47     pinMode(switchPin3, INPUT);
48     pinMode(switchPin4, INPUT);
49     pinMode(switchPin5, INPUT);
50     // Dirección MAC Ethernet del shield
51     byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xEE };
52     // Dirección IP del shield
53     byte ip[] = { 192, 168, 1, 10 };
54     //Inicialización de la comunicación Ethernet
55     Ethernet.begin(mac, ip);
56
57     // Inicialización de las comunicaciones Modbus con los clientes
58     mbm1.config();
59     mbm2.config();
60     mbm3.config();
61
62     iec1.config();
63     iec2.config();
64     iec3.config();
65 }

```

En la Figura 77, se observa el desarrollo del programa principal donde se realiza el envío de datos digitales y analógicos a cada una de las estaciones esclavos a través de los protocolos de comunicación Modbus TCP e IEC60870-5-104.

Figura 77

Envío de datos mediante los protocolos Modbus TCP e IEC60870-5-104.

```

67 void loop() {
68   if(switchPin5){ //MODBUS
69     //Envío de datos al Cliente 1
70     mbm1.sendCoil(dst1, SWITCH1_ISTS, digitalRead(switchPin1));
71     mbm1.sendCoil(dst1, SWITCH2_ISTS, digitalRead(switchPin2));
72     mbm1.sendHreg(dst1, POT1_IREG, analogRead(PotPin1));
73     //Envío de datos al Cliente 2
74     mbm2.sendCoil(dst2, SWITCH1_ISTS, digitalRead(switchPin3));
75     mbm2.sendCoil(dst2, SWITCH2_ISTS, digitalRead(switchPin4));
76     mbm2.sendHreg(dst2, POT1_IREG, analogRead(PotPin2));
77     delay(100);
78     //Envío de datos al Cliente 3
79     mbm3.sendCoil(dst3, SWITCH1_ISTS, digitalRead(switchPin1));
80     mbm3.sendCoil(dst3, SWITCH2_ISTS, digitalRead(switchPin2));
81     mbm3.sendHreg(dst3, POT1_IREG, analogRead(PotPin2));
82     delay(100);
83   }
84   else{//IEC
85     //Envío de datos al Cliente 1
86     iec1.sendCoil(dst1, SWITCH1_ISTS, digitalRead(switchPin1));
87     iec1.sendCoil(dst1, SWITCH2_ISTS, digitalRead(switchPin2));
88     iec1.sendHreg(dst1, POT1_IREG, analogRead(PotPin1));
89     //Envío de datos al Cliente 2
90     iec2.sendCoil(dst2, SWITCH1_ISTS, digitalRead(switchPin3));
91     iec2.sendCoil(dst2, SWITCH2_ISTS, digitalRead(switchPin4));
92     iec2.sendHreg(dst2, POT1_IREG, analogRead(PotPin2));
93     delay(100);
94     //Envío de datos al Cliente 3
95     iec3.sendCoil(dst3, SWITCH1_ISTS, digitalRead(switchPin1));
96     iec3.sendCoil(dst3, SWITCH2_ISTS, digitalRead(switchPin2));
97     iec3.sendHreg(dst3, POT1_IREG, analogRead(PotPin2));
98     delay(100);
99   }
100  }

```

Programación de las estaciones esclavos. En este caso las estaciones esclavo 1 y esclavo 2 contienen la misma lógica de programación, a excepción de la sección de configuración de la dirección IP, ya que cada una cuenta con una propia. En la Figura 78 se observa la declaración inicial necesaria para cada uno de los protocolos de comunicación y son descritos a continuación:

- Se declaran las librerías a utilizarse para la comunicación Ethernet y para la comunicación de cada uno de los protocolos (Modbus.h, ModbusIP_ENC28J60.h, IEC104 e IEC104IP.h).
- Se declaran las direcciones de los diodos led que se encargan de transmitir los datos.
- Se asignan los pines de entrada y salidas digitales correspondientes a los diodos led.
- Se declaran los objetos para la comunicación Modbus TCP e IEC60870-5-104.

Figura 78

Declaración de librerías, objetos y variables de la estación esclavo.

```

1  #include <EtherCard.h>
2  #include "Modbus.h"
3  #include "ModbusIP_ENC28J60.h"
4  #include "IEC104.h"
5  #include "IEC104IP.h"
6
7  const int LAMP1_COIL = 1;
8  const int LAMP2_COIL = 2;
9  const int POT_HREG = 1;
10
11 // // asignación de entrada y salidas del arduino
12 const int ledPin1 = 2;
13 const int ledPin2 = 3;
14 const int ledPin3 = 4;
15 const int switchPin1 = 5;
16 int ledState = LOW; // ledState used to set the LED
17 unsigned long currentMillis = 0;
18 unsigned long previousMillis = 0; // will store last time LED was updated
19
20 // Objeto ModbusIP
21 ModbusIP mb;
22
23 // Objeto IEC104IP
24 IEC104IP iec;

```

En la sección de configuración del programa, como se presenta en la Figura 79, se configuran el interruptor llave SELECTOR como entrada digital y los diodos led como salidas digitales. Además, se configuran las direcciones MAC e IP de la estación esclavo, en base a las direcciones asignadas anteriormente, y se inicializan las comunicaciones Ethernet y las comunicaciones Modbus TCP e IEC60870-5-104 con las estaciones esclavos.

Figura 79

Configuración de entrada, salidas, MAC e IP, e inicialización de comunicaciones.

```

26 void setup() {
27 //dirección MAC
28 byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF,0xFE, 0xED};
29 //dirección IP
30 byte ip [] = {192,168,1,160};
31 mb.config (mac,ip);// se confirma las direcciones
32 iec.config (mac,ip);
33 pinMode(ledPin1, OUTPUT);
34 pinMode(ledPin2, OUTPUT);
35 pinMode(ledPin3, OUTPUT);
36 pinMode(switchPin1, INPUT);
37 mb.addCoil(LAMP1_COIL);
38 mb.addCoil(LAMP2_COIL);
39 mb.addHreg(POT_HREG);
40 iec.addCoil(LAMP1_COIL);
41 iec.addCoil(LAMP2_COIL);
42 iec.addHreg(POT_HREG);
43 }

```

En la Figura 80, se observa el desarrollo del programa principal para la estación esclavo donde se realiza la recepción de datos digitales y analógicos a través de los protocolos de comunicación Modbus TCP e IEC60870-5-104.

Figura 80

Envío de datos mediante los protocolos Modbus TCP e IEC60870-5-104.

```

44 void loop() {
45
46     if(digitalRead(switchPin1)//MODBUS
47     {
48         mb.task();// actualiza para que el trabajo se realice con normalidad
49         // envía los valores del potenciómetro
50         digitalWrite(ledPin1, mb.Coil(LAMP1_COIL));
51         digitalWrite(ledPin2, mb.Coil(LAMP2_COIL));
52         currentMillis = millis();
53         if (currentMillis - previousMillis >= mb.Hreg(POT_HREG)) {
54             previousMillis = currentMillis;
55             if (ledState == LOW) {
56                 ledState = HIGH;
57             } else {
58                 ledState = LOW;
59             }
60             digitalWrite(ledPin3, ledState);
61         }
62     }
63     else{IEC
64
65         iec.task();// actualiza para el trabajo se realice con normalidad
66         if(iec.Coil(LAMP1_COIL)==1)
67         {
68             digitalWrite(ledPin1, LOW);
69         }
70         else
71         {
72             digitalWrite(ledPin1, HIGH);
73         }
74         if(iec.Coil(LAMP2_COIL)==1)
75         {
76             digitalWrite(ledPin2, LOW);
77         }
78         else
79         {
80             digitalWrite(ledPin2, HIGH);
81         }
82         currentMillis = millis();
83         if (currentMillis - previousMillis >= iec.Hreg(POT_HREG)) {
84             previousMillis = currentMillis;
85             if (ledState == HIGH) {
86                 ledState = LOW;
87             } else {
88                 ledState = HIGH;
89             }
90             digitalWrite(ledPin3, ledState);
91         }
92     }
93 }

```

Capítulo V

Validación, Pruebas y Resultados

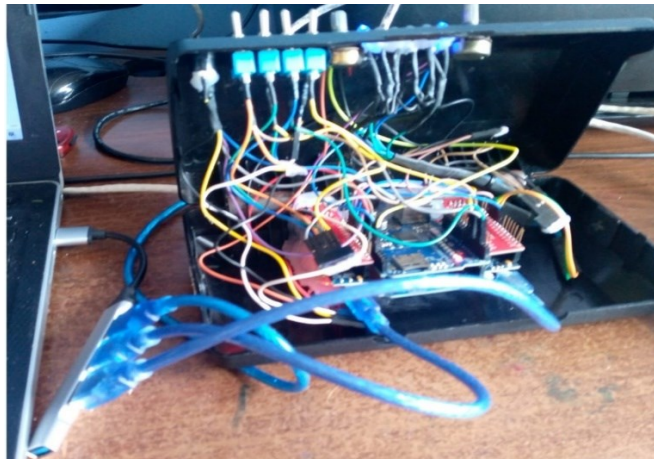
Para validar el funcionamiento del sistema, se realizaron diferentes pruebas, que permiten comprobar inicialmente de forma individual y posteriormente de forma colectiva el funcionamiento del sistema, por lo que se dividieron las pruebas acordes a los requerimientos del sistema.

Pruebas y evaluación de conexión

Para verificar la conectividad del sistema se hizo la prueba de conectividad de los equipos a través del CMD de Windows, previamente, como se observa en la Figura 81, se hicieron cables UPT punto punto para conectar los equipos al Switch TP-LINK, y comprobar su conectividad con la prueba de ping a cada equipo.

Figura 81

Conexión de equipos para comprobación de conectividad del sistema.



En la prueba se evidenció que todos los interruptores, potenciómetros y diodos Leds están conectados de forma adecuada y funcionan correctamente, que el HUB USB alimenta de forma adecuada a todos los módulos, estando desde esta perspectiva el sistema listo para comprobarse.

Pruebas y evaluación de la conectividad del sistema en el estuche

Para verificar la conectividad del sistema en el estuche, se verificó de forma individual con programas individuales el funcionamiento de las entradas y salidas de los terminales de cada dispositivo maestro y esclavo, de igual manera se verificó el funcionamiento del HUB USB para la alimentación de los 3 dispositivos, tal como se muestra en la Figura 82.

Figura 82

Conexión física para verificación de conectividad en el estuche.



Al realizar la prueba respectiva, como se presenta en la Figura 83, se comprobó la conectividad ethernet de todos los equipos, evidenciándose la conexión de los equipos en red.

Figura 83

Comprobación de conectividad de las estaciones en PC.

```

C:\Users>ping 192.168.1.100

Haciendo ping a 192.168.1.100 con 32 bytes de datos:
Respuesta desde 192.168.1.100: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.100: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.100: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.100: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.1.100:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users>ping 192.168.1.10

Haciendo ping a 192.168.1.10 con 32 bytes de datos:
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.1.10: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.1.10:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users>ping 192.168.1.150

Haciendo ping a 192.168.1.150 con 32 bytes de datos:
Respuesta desde 192.168.1.150: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.150: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.150: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.150: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.1.150:
    Paquetes: enviados = 4, recibidos = 2, perdidos = 2
    (50% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 1ms, Media = 1ms

C:\Users>ping 192.168.1.160

Haciendo ping a 192.168.1.160 con 32 bytes de datos:
Respuesta desde 192.168.1.160: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.160: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.160: bytes=32 tiempo=1ms TTL=64
Respuesta desde 192.168.1.160: bytes=32 tiempo=1ms TTL=64

Estadísticas de ping para 192.168.1.160:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 1ms, Máximo = 1ms, Media = 1ms
  
```

Pruebas y evaluación de la comunicación Modbus TCP

Para verificar la comunicación Modbus TCP, se carga una versión inicial del programa realizado para comunicación Modbus, comprobando la comunicación de forma individual la comunicación de la estación maestro con cada estación esclavo, cambiando el estado de los interruptores del maestro y comprobando el cambio de estados en cada esclavo. De igual manera, se comprobó el cambio de frecuencia en los esclavos en función de los potenciómetros del maestro, tal como se observa en la Figura 84.

Figura 84

Verificación de comunicación Modbus TCP.



Se verifica el correcto funcionamiento de la comunicación Modbus entre las estaciones maestro y esclavos, cumpliendo con el requerimiento planteado, en base al funcionamiento establecido en la implementación de software.

Pruebas y evaluación de la comunicación IEC60870-5-104

Para verificar la comunicación IEC60870-5-104, se carga una versión inicial del programa realizado para comunicación IEC104, comprobando la comunicación de forma individual la comunicación de la estación maestro con cada estación esclavo, cambiando el estado de los interruptores del maestro y comprobando el cambio de estados en cada esclavo. De igual manera se comprobó el cambio de frecuencia en los esclavos en función de los potenciómetros del maestro, como se presenta en la Figura 85.

Figura 85

Verificación de comunicación IEC60870-5-104.



Se verifica el correcto funcionamiento de la comunicación IEC60870-5-104 entre el maestro y los esclavos, cumpliendo con el requerimiento planteado, en base al funcionamiento establecido en la implementación de software, funcionando en lógica inversa respecto al funcionamiento en Modbus.

Pruebas y evaluación del sistema completo

Para verificar la comunicación completa del sistema, se carga una versión final del programa realizado para comunicación Modbus TCP e IEC60870-5-104, seleccionando el protocolo a partir del interruptor llave SELECTOR, comprobando la comunicación de forma individual la comunicación de la estación maestro con cada estación esclavo, acorde al protocolo seleccionado, cambiando el estado de los interruptores del maestro y comprobando el cambio de estados en cada esclavo, de igual manera se comprobó el cambio de frecuencia en los esclavos en función de los potenciómetros del maestro.

Figura 86

Verificación de la versión final en el sistema completo.



Nota. Las dos imágenes de la parte superior utilizan comunicación Modbus TCP y las dos imágenes de la parte inferior utilizan comunicación IEC60870-5-104.

Se verifica el correcto funcionamiento del sistema, funcionando de forma adecuada con cada protocolo de comunicación entre el maestro y los esclavos, cumpliendo con el requerimiento planteado, en base al funcionamiento establecido en la implementación de software.

Pruebas y evaluación del monitoreo de red

Para verificar la comunicación en lo que a las tramas y comunicación respecta, se utiliza al PC como esclavo, conectado con el software ModbusPoll y capturando las tramas con el software Wireshark, verificando de esta forma las tramas de comunicación enviadas y recibidas por las estaciones maestro y esclavos en este protocolo, tal como se presenta en la Figura 87 y la Figura 88. Lamentablemente para el protocolo IEC60870-5-104 no se pudo desarrollar las

mismas pruebas, debido a que no se encontró un software que simule un esclavo en este protocolo para que se genere la respuesta desde la PC a la estación maestro.

Figura 87

Verificación de tramas esclavo3 (PC)-esclavo2 con el software Wireshark.

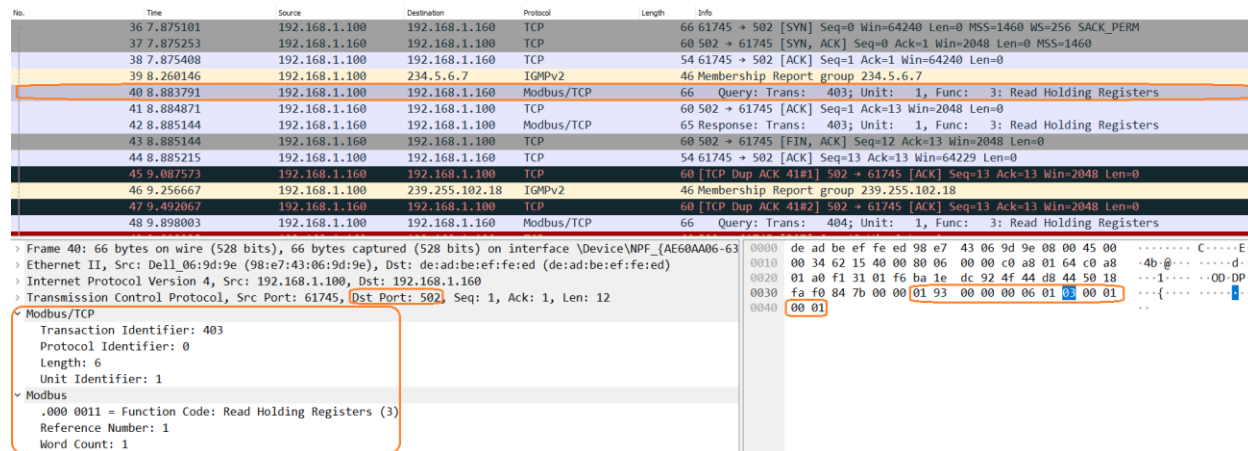
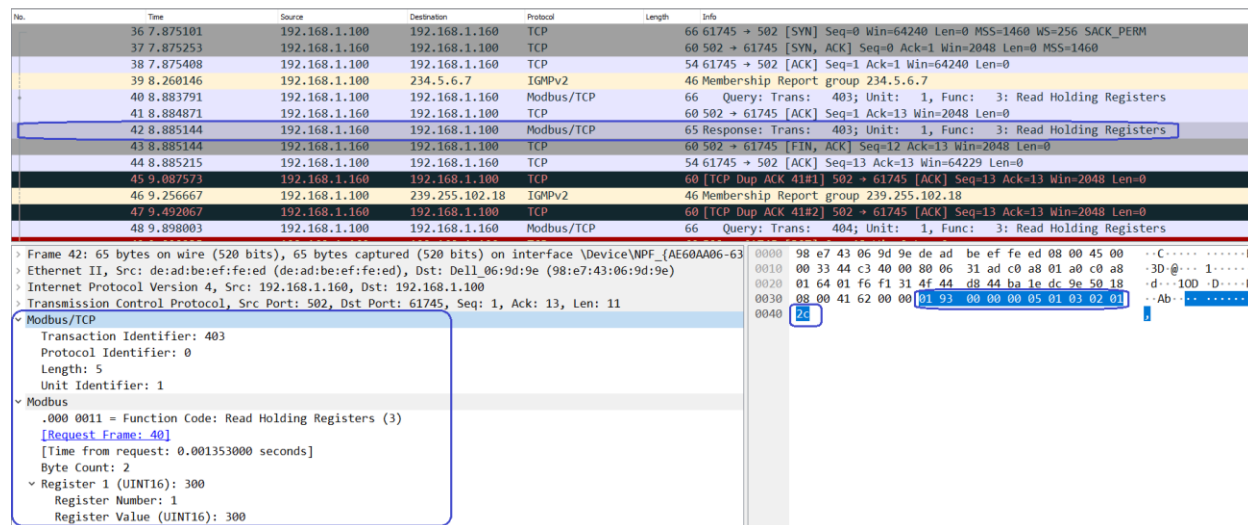


Figura 88

Verificación de tramas esclavo2-esclavo3(PC) con el software Wireshark.



Análisis de los resultados

De todas las pruebas realizadas, se comprueba que todo el sistema funciona de forma adecuada, por lo que se consideran aprobadas todas las pruebas realizadas al sistema, por lo que se considera finalizado el trabajo.

Capítulo VI

Conclusiones

- En base a lo desarrollado en el presente proyecto se concluye que el banco de pruebas es 100% funcional ya que cumple con los requisitos propuestos inicialmente permitiendo la comunicación entre las tarjetas de desarrollo por intermedio de los protocolos de red Modbus TCP e IEC60870-5-104.
- El diseño del banco de pruebas permite al usuario realizar con facilidad las pruebas de comunicación al manipular los interruptores y potenciómetros que se encargan de enviar los datos para encender, apagar o variar la frecuencia de los diodos led.
- En el análisis de las tramas a través del software Wireshark se pudo comprobar de mejor manera la estructura del protocolo de red Modbus TCP a partir de la comunicación entre la estación maestro y la estación esclavo 3 (PC).
- Se comprobó la comunicación entre las estaciones maestro y esclavos desconectando los cables de red de cada uno de las estaciones y verificando si la transmisión y recepción de los datos sufrían algún cambio al manipular las entradas del sistema.
- Se requiere la utilización de softwares de comunicación para el protocolo Modbus TCP e IEC60870-5-104 para el funcionamiento y verificación de tramas en el PC, utilizando el software Wireshark, en el caso de Modbus TCP se utilizó Modbus Poll, pero para el IEC60870-5-104, no se consiguió un software que permita realizar esta comprobación, por lo que no se pudo verificar la trama, pese a que si se comprobó su funcionamiento en el banco de pruebas.

Recomendaciones

- Revisar las conexiones de los cables de red del banco de pruebas antes de comenzar con el funcionamiento, debido a que si uno de estos no está correctamente conectado la comunicación entre sus correspondientes estaciones no se podrá realizar.
- Revisar e identificar las direcciones IP de cada una de las estaciones, principalmente de la estación esclavo 3 (PC), para que no existan falencias en la comunicación y poder realizar un mejor análisis de tramas entre las estaciones deseadas.
- Para trabajos futuros, realizar la investigación sobre otros protocolos de red que sean compatibles con los shields utilizados en este proyecto y sus respectivas tarjetas de desarrollo, y a su vez consultar la compatibilidad y usabilidad de varios protocolos de red en otras tarjetas de desarrollo.

Bibliografía

- Alcantara Fernandez, A. (2020). *Mejora del sistema de protección de los generadores síncronos mediante IED's MiCOM P643 en la central hidroeléctrica El Machu, distrito de Chongos Alto–Electrocentro. Huancayo. 2019.* [Tesis de Ingeniería], Universidad Continental, Huancayo. Obtenido de <https://hdl.handle.net/20.500.12394/8222>
- Arias, A., Castillo, A., Roa, Á., Bidó, E., García, J., Hernández, D., & Aybar, M. (2020). Protocolos y topologías utilizadas en los sistemas de comunicación de las microrredes eléctricas. *Ciencia, Ingenierías y Aplicaciones*, 81-95.
- Banguera Gómez, L. (2020). *Estudio, análisis y modelamiento de los sistemas eléctricos de distribución en el contexto de redes eléctricas inteligentes industria 4.0 y automatización dentro de convenio marco de cooperación interinstitucional 080 de 2019 entre la Universidad Distrita.* Bogotá: RAP-E. Obtenido de <https://repository.udistrital.edu.co/bitstream/handle/11349/24730/BangueraGomezLuisMiguel2020.pdf?sequence=1&isAllowed=y>
- Barona López, G., & Velasteguí, L. E. (5 de Agosto de 2021). Automatización de procesos industriales mediante Industria 4.0. *AlfaPublicaciones*, 3(3.1), 98–115.
doi:<https://doi.org/10.33262/ap.v3i3.1.80>
- Barrantes, L. (2011). *Diseño del sistema de protección y control de subestaciones eléctricas.* Leganés: Escuela Politécnica Superior de la Universidad Carlos III de Madrid.
- Corrales, L. (2007). *Interfaces de comunicación industrial.* Quito: Area de Automatización e Instrumentación Industriales.
- De la Barrera Gonzalez, A., Jimenez-Ramirez, O., & Sánchez García, J. C. (Mayo de 2013). Programación de la trama de PROFIBUS DP utilizando RS-485, en un micro controlador

Microchip. *Congreso Internacional de Informática Aplicada, CIIA 2013*.

doi:10.13140/2.1.4587.4883

Espinoza Castro, M. L., & Baño Soto, J. P. (2018). *Integración de redes industriales mediante un software de monitoreo y adquisición de datos en el laboratorio de control y manipulación automática de la carrera de Ingeniería de Mantenimiento*. Riobamba: Escuela Superior Politécnica de Chimborazo. Obtenido de <http://dspace.esPOCH.edu.ec/handle/123456789/9867>

Estrada Roque, J. A. (2017). *Protocolos de comunicaciones industriales*. Logicbus SA de CV. Obtenido de <https://www.logicbus.com.mx/pdf/articulos/Protocolos-de-Comunicaci%C3%B3n-Industrial.pdf>

EVICITI. (2023). *Industria 4.0: la 4ta Revolución Industrial*. Obtenido de ¡Bienvenido a la Industria 4.0, la Cuarta Revolución Industrial!: <https://www.eviciti.com.mx/blog-post/bienvenido-a-la-industria-4-0-la-cuarta-revolucion-industrial/>

Feria, J., Camargo, J., & Muskus, E. (2015). Diseño e implementación de un algoritmo para translación de protocolo entre las normas IEC60870-5-104 y MODBUS TCP/IP. Ingenium.

Florián, D., & Gálvez, L. (2006). *Sistema de monitoreo y control de subestaciones eléctricas orientados a la gestión de la demanda y basado en sistemas de control inteligentes*. Pontificia Universidad Católica del Perú.

Fuentes, G. (2005). IEC 61850, El Nuevo Estándar en Automatización de Subestaciones. ABB Power Technologies. Obtenido de ABB Power Technologies.

Gallardo, E. (2022). Protocolo de comunicación para la integración de dispositivos en los sistemas Scada de las subestaciones eléctricas. *Revista Electrónica de Estudios Telemáticos*, 59-82.

- García, A. (23 de Enero de 2023). *Arduino Mega: Características y Capacidades*. Obtenido de Panamahitek: <https://panamahitek.com/arduino-mega-caracteristicas-capacidades-y-donde-conseguirlo-en-panama/>
- Gil Lostes, S., & Bernal Agustín, J. (2011). *Entorno para el entrenamiento de operadores de Subestaciones Eléctricas*. [Tesis de Ingeniería], Zaragoza. Obtenido de <https://zaguan.unizar.es/record/5615>
- Gómez Tapias, F., & Vargas Torres, H. (2011). Planteamiento del diseño de subestaciones eléctricas. *Épsilon*, 16, 79-112.
- Guerra, J. (2021). *Ethernet Shield Arduino tutorial paso a paso*. Obtenido de Programarfacil.com: <https://programarfacil.com/blog/arduino-blog/ethernet-shield-arduino/>
- IHC. (2023). *Ethernet Shield For Nano (Without Nano)*. Obtenido de Indian Hobby Center: <https://www.indianhobbycenter.com/products/ethernet-shield-for-nano-enc28j60>
- International Electrotechnical Commission*. (2023). Obtenido de IEC: <http://www.iec.ch/>
- Kashif. (14 de Mayo de 2019). *Arduino YUN for Beginners*. Obtenido de <https://projectiot123.com/2019/05/14/arduino-yun-for-beginners/>
- Lukianova, N. (2023). *Principales partes de un Arduino*. Obtenido de Arduino.cl: <https://arduino.cl/principales-partes-de-un-arduino/>
- Luna-López, M., Hernández-Lozano, M., Aldana-Franco, R., Álvarez-Sánchez, E., Leyva-Retureta, J., Ricaño-Herrera, F., & Aldana-Franco, F. (2021). Sistema inteligente de monitoreo para condiciones ambientales en Industria 4.0. *Científica*, 25(2), 01-10. doi.: <https://doi.org/10.46842/ipn.cien.v25n2a07>

- Manzano, B. (Julio de 2022). Estudio del funcionamiento del protocolo IEC 60870-5-104 en el entorno ferroviario y de las herramientas existentes para su estudio y desarrollo.
- Medina, O. (2020). *Ministerio de Energía y Minería*. Obtenido de Argentina: www.energia.gov.ar
- MODBUS Group. (2006). Obtenido de Modbus: http://modbus.org/about_us.php
- Mora, D., & Guerrero, L. (Abril de 2020). *Industria 4.0: el reto en la ruta hacia las organizaciones digitales*. Obtenido de <http://hdl.handle.net/10644/7910>
- Nettigo. (2023). *Ethernet Shield for Nano, ENC28J60*. Obtenido de Nettigo Odkryj tajmiki elektroniki: <https://nettigo.eu/products/ethernet-shield-for-nano-enc28j60>
- OMNIRETRO. (24 de Junio de 2019). *Conoce la nueva Raspberry Pi 4: Especificaciones y características*. Obtenido de <https://omniretro.com/blog/tecnologia/conoce-la-nueva-raspberry-pi-4-especificaciones-y-caracteristicas/>
- Peralta-Abarca, J. d., Martínez-Bahena, B., & Enríquez-Urbano, J. (2020). *Industria 4.0. Inventio, la génesis de la cultura universitaria en Morelos*, 16(39).
doi:10.30973/inventio/2020.16.39/4
- Ramírez, S. (2004). *Redes de Distribución de Energía*. Manizales: Centro de Publicaciones Universidad Nacional de Colombia.
- Ramos Guardarrama, J., Hernández Areu, O., & Silverio Freire, R. (2019). Sistema de supervisión para el monitoreo de redes eléctricas inteligentes. *Ingeniería Energética*, XL(3), 264-272.
- Rozo-García, F. (1 de Abril de 2020). Revisión de las tecnologías presentes en la industria 4.0. *Revista UIS Ingenierías*, 19(2), 177-191. doi:<https://doi.org/10.18273/revuin.v19n2-2020019>
- Sharpe, R., Warnicke, E., & Lamping, U. (2011). *Wireshark User's Guide*.

- Solectroshop. (13 de Septiembre de 2020). *Nueva serie Arduino Nano - qué Arduino es el mejor para comprar en 2020*. Obtenido de Solectroshop:
<https://solectroshop.com/es/blog/arduino-nano-every-guia-a-la-instalacion-y-proyectos-basicos--n42>
- Spiceman. (11 de Diciembre de 2022). *Arduino Leonardo Specifications/Functions*. Obtenido de Spiceman: <https://spiceman.net/arduino-leonardo/>
- Stock, T., & Seliger, G. (2016). Opportunities Of Sustainable Manufacturing In Industry 4.0. *Procedia CIRP*, 40, 536-541. doi: <https://doi.org/10.1016/j.procir.2016.01.129>
- TP-Link. (2018). *TL-WRN740N Router Inalámbrico N 150 Mbps*. Obtenido de TP-Link:
<https://www.tp-link.com/ar/home-networking/wifi-router/tl-wr740n/>
- Valdivia Miranda, C. (2019). *Comunicaciones Industriales* (1 ed.). Madrid: Ediciones Paraninfo, SA.
- Yebra, J. (2009). *Sistemas Eléctricos de Distribución*. Barcelona: Reverté.
- Zawadzki, P., & Żywicki, K. (Septiembre de 2016). Smart Product Design And Production Control For Effective Mass Customization In The Industry 4.0 Concept. *Management and Production Engineering*, 7(3), 105–112. doi: <https://doi.org/10.1515/mper-2016-0030>

Apéndices