Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Electrónica

Módulo de transporte de MMS sobre IP, desde dispositivos de protección o control, hacia una red de área local

Informe de Proyecto de Graduación para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura

Leonidas Arburola Briceño

Cartago, 2008

INSTITUTO TECNOLOGICO DE COSTA RICA

ESCUELA DE INGENIERIA ELECTRONICA

PROYECTO DE GRADUACIÓN

TRIBUNAL EVALUADOR

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Electrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

Ing. Faustino Montes de Oca M.

Profesor asesor

Ing. Luis P. Méndez Badilla

Profesor lector

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica

Cartago, 4 agosto 2008

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 7 agosto 2008

Leonidas Arburola Briceño

Cédula: 503490832

RESUMEN

En una subestación eléctrica, los Dispositivos Electrónicos Inteligentes (IED), son comprados para realizar el transporte de cierta cantidad de datos, desde los equipos de protección y control, hacia una red de área local (LAN). En caso de que se quiera transportar datos adicionales, es necesaria la compra de otro IED. El problema radica en que estos dispositivos tienen un costo elevado, aproximadamente $3000 y son sistemas cerrados que no permiten mejoras.

Como solución, para reducir el costo que implica comprar otro IED, se diseñó un método alternativo para el transporte de datos, mediante el desarrollo del software para un sistema empotrado, que implementa una de las funciones del IED (transportar mensajes MMS sobre IP, desde un dispositivo de protección o control hacia una LAN). El costo del microcontrolador utilizado es de $275, esto lo justifica como una alternativa de menor costo a la compra de un IED.

Palabras claves: IEC 61850, IED, MMS, red de área local (LAN), microcontrolador.

ABSTRACT

In an electrical substation, intelligent electronic devices (IED) are purchased to make the transportation of certain amount of data, from protective and control equipment to a local area network (LAN). To carry additional data is needed buying another IED. The problem is that these devices have a high cost, about $ 3000 and are closed systems that do not allow for improvements.

As a solution, to reduce the cost of buying another IED, was designed an alternative method for transporting data, using an algorithm programmed in an embedded system, which implements one of the functions of the IED (Transport MMS messages over IP, from a protective or control device to a LAN). The cost of the microcontroller used is $275; this is justified as a cheaper alternative to buying an IED.

Keywords: IEC 61850, IED, MMS, local area network (LAN), microcontroller.

DEDICATORIA

A mis papás, Juan Rafael Arburola Rojas y Romy Briceño Tijerino, porque este Proyecto de Graduación es el resultado de toda una vida de esfuerzo y dedicación.

A mis hermanos, porque este logro es también de ellos.

AGRADECIMIENTO

A los Profesores de la Escuela de Ingeniería Electrónica, que durante mi carrera ayudaron no solo en mi formación académica sino también personal.

Al Ing. Adalberto Sánchez Tercero, por brindarme la oportunidad de realizar este proyecto, en el Departamento de Ingeniería de Control y Automatización.

Al Ing. Manrique Murillo Calvo, por su aporte invaluable para el buen término de este proyecto.

INDICE GENERAL

INDICE DE FIGURAS

INDICE DE TABLAS

Glosario

Manufacturing Message Specification (MMS): Mensaje de Especificación del Fabricante.
Virtual Manufacturing Device **(VMD):** Un modelo de software que representa la funcionalidad de un dispositivo real.

Usuario MMS: Cualquier dispositivo que envía o recibe servicios de MMS

Servicio MMS: Es un medio de transmitir información entre dos usuarios de MMS. Se utilizan para manipular el VMD y por tanto al dispositivo real.

Cliente: Entidad que ha enviado una solicitud de un servicio MMS a otro usuario.

Servidor: Entidad que responde a una solicitud de servicio MMS.

Objeto MMS: Es la representación de una entidad física, es caracterizado por datos y operaciones que permiten el acceso a los atributos del objeto.

NamedVariable**:** Nombres de variable, corresponde al nombre de una variable MMS. Objeto contenido en el VMD.

NamedVariableList**:** Listas de Nombres de variables, corresponde a una lista de variables o objetos MMS. Objeto contenido en el *VMD*.

Domain**:** Nombre en inglés para Dominios, Representan áreas de la memoria, que agrupan variables específicas, por ejemplo variables de configuración. Objeto contenido en el *VMD*.

Interfaz Humano Máquina (HMI): representa un software con funciones de control, supervisión y adquisición de datos (SCADA).

Capítulo 1: Introducción

El proyecto se desarrolló en el ICE, en el departamento de Ingeniería de Control y Automatización de la Unidad Estratégica de Negocios de Proyectos y Servicios Asociados del Sector Energía. En esta sección se describe el problema existente, su entorno, importancia y solución seleccionada, así como algunos de los beneficios que podrán derivarse al darle solución al problema.

1.1 Administración de información en la red de área local de una subestación eléctrica

Una de las partes de mayor importancia, durante el proceso de distribución energética, tiene lugar en una subestación eléctrica. En estas subestaciones, típicamente se pueden encontrar:

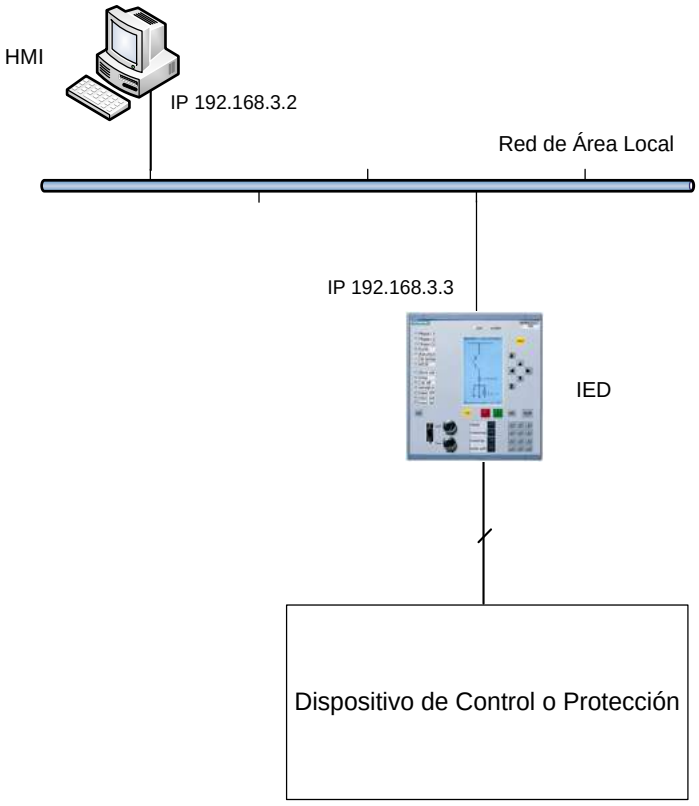
- Interfaz Humano Máquina (HMI).
- Dispositivos de protección o control.
- Dispositivos Electrónicos Inteligentes (IED).
- Red de Área Local Ethernet (LAN).

En la figura 1.1 se muestra un ejemplo de la configuración típica, de una subestación eléctrica. En ésta figura, la HMI, corresponde a un programa con funciones de control, supervisión y adquisición de datos (SCADA).

Los dispositivos de protección o control, corresponden a equipos tales como: transformadores, disyuntores, bancos de capacitores o transformadores de instrumentación, entre otros.

Los IED, corresponden a equipo con dos funciones principales:

- Permitirle a los dispositivos de protección y control, operar de manera autónoma o manual.
- Transporte de mensajes, desde los dispositivos de protección o control, hacia una red de área local, esto por medio de Mensajes de Especificación del fabricante (MMS) sobre IP. Estos mensajes son transportados solo a petición remota de la HMI.



Microsoft Paint

Figura 1.1 Configuración de una subestación eléctrica funcionando bajo la norma IEC 61850 [19]

De las dos funciones mencionadas para los IED, la segunda es de interés para este proyecto. Un dispositivo de control o protección, contiene una gran variedad de información, no toda esta información es necesaria, por lo que el ICE compra IED fabricados a la medida, para realizar el transporte de solo una parte de ésta información. En caso de que surja la necesidad de transportar información adicional, desde los equipos de protección y control, hacia una red de área local, se debe comprar otro IED. El problema que se presenta es que estos dispositivos tienen un costo elevado, alrededor de los $3000 y son sistemas cerrados que no permiten mejoras.

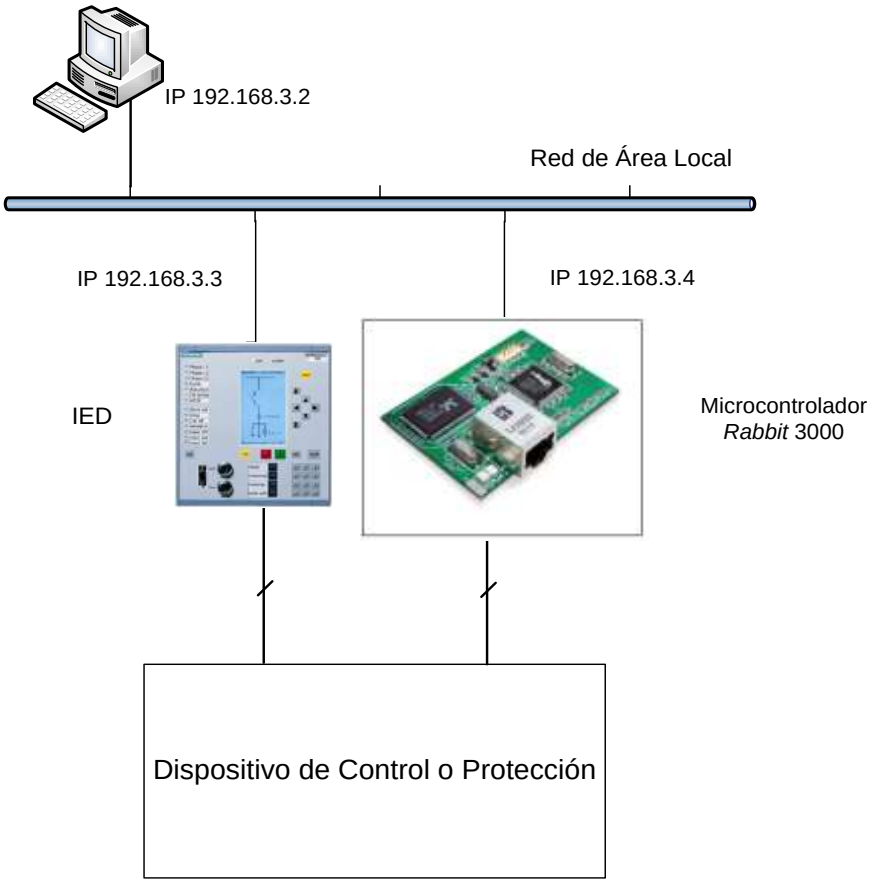
Con el desarrollo de este proyecto, el ICE tendría una base, para crear sus propios módulos de transporte de MMS sobre IP, siendo esta una alternativa menos costosa a la compra de un IED para ésta función.

1.2 Solución alternativa para el transporte de MMS sobre IP, desde un equipo de protección o control, hacia una LAN

Para solucionar el problema existente, se desarrolló una aplicación de software para un sistema empotrado. El costo del hardware utilizado fue de $275, y en éste, se programaron las rutinas para realizar el transporte de MMS sobre IP, desde un dispositivo de protección o control hacia una LAN.

El sistema empotrado que se utilizó, fue el módulo de desarrollo RCM 3300, éste cuenta con un microprocesador *Rabbit* 3000 y fue seleccionado por ser orientado a aplicaciones de red y tener un costo relativamente bajo, si se compara con otros microcontroladores con características similares.

A nivel de bloques, en la figura 1.2 se muestra un esquema de la solución implementada.



Microsoft Visio 2007

Figura 1.2 Diagrama de bloques de la solución seleccionada [19] y [17]

Capítulo 2: Meta y objetivos

2.1 Meta

Lograr un óptimo funcionamiento del software desarrollado para un dispositivo electrónico, de modo que el ICE pueda utilizarlo como solución alternativa de menor costo a la compra de un IED.

2.2 Objetivo General

Desarrollar el software para un sistema electrónico, que permita el transporte de MMS sobre IP, desde un equipo de protección o control, hacia una red de área local Ethernet.

2.3 Objetivos Específicos

1. Escribir las rutinas para que el sistema electrónico, pueda ser conectado a una red de área local, según la secuencia de tramas y formato requerido por el estándar IEC 61850.

2. Escribir las rutinas para que el sistema electrónico, pueda realizar el transporte de MMS sobre IP hacia una LAN Ethernet.

Capítulo 3: Marco teórico

Este capítulo presenta información, sobre los principales conceptos relacionados, con la solución dada al problema. Como se mencionó, se desarrolló el software para un sistema electrónico, con funciones para realizar el transporte de MMS sobre IP, desde un dispositivo de protección o control, hacia una red de área local Ethernet, según lo establece el estándar IEC 61850.

3.1 Perfil de la norma IEC 61850

El estándar IEC 61850, utiliza el modelo de referencia de Interconexión de Sistemas Abiertos (OSI), éste define una pila de protocolos, en total siete capas de red. Las capas son: Aplicación, Presentación, Sesión, Transporte, Red, Enlace de datos y Física. En la figura 3.1 se puede observar el modelo de capas OSI.

Microsoft Office Visio

Figura 3.1 Modelo de referencia OSI

Los datos en la capa de aplicación, corresponden a Mensajes de Especificación del Fabricante (MMS). Conforme este mensaje se desplaza hacia abajo en la pila de protocolos, cada capa agrega información, esto es conocido como el proceso de encapsulación.

La forma que cada dato toma en cualquiera de las capas, es llamada Unidad de Datos de Protocolo (PDU). Durante la encapsulación, cada capa encapsula el PDU que recibe de la capa superior, de acuerdo con el protocolo que está siendo usado.

En cada una de estas capas, un PDU tiene diferentes nombres, para reflejar su nueva apariencia.

En la figura 3.2 se muestra el proceso de encapsulación. En la parte izquierda de ésta figura, se observa el mensaje MMS pasando por cada una de las capas de red y la información que es agregada en cada capa. A la derecha, se observa el nombre que recibe el PDU en cada capa de red.

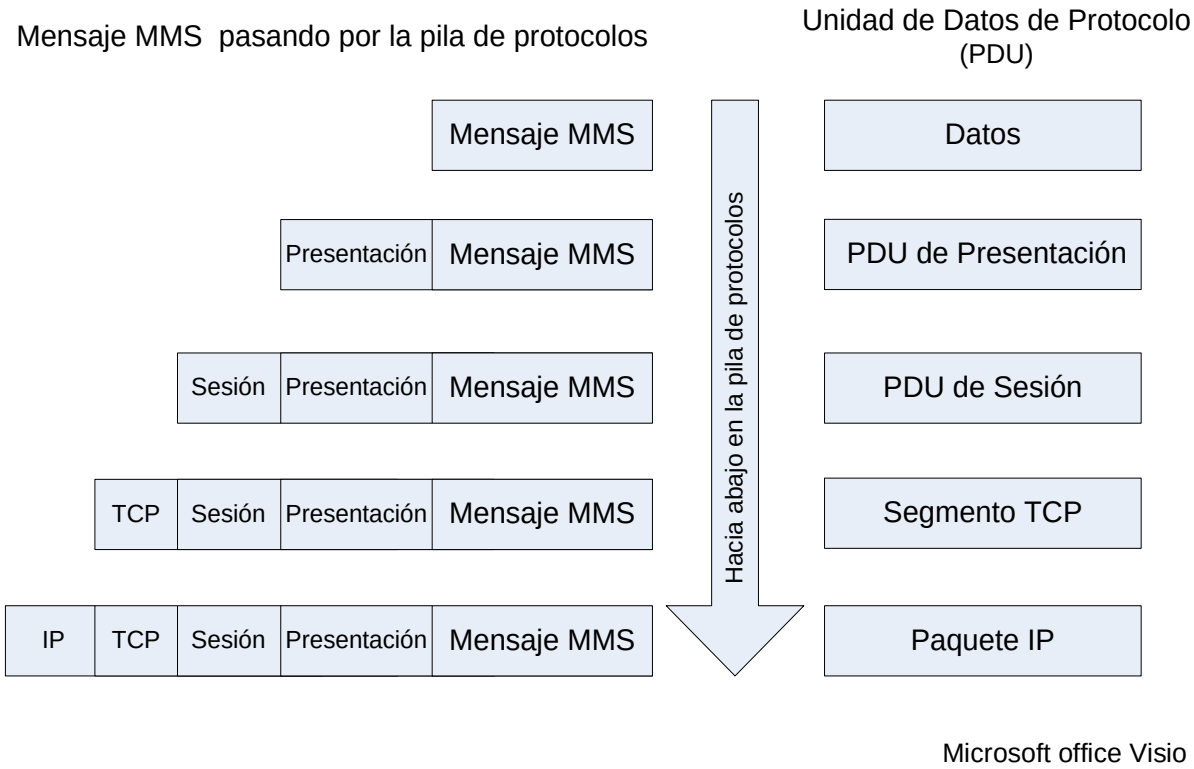


Figura 3.2 Proceso de encapsulación [23]

Luego del proceso de encapsulación, se tiene un mensaje MMS en un paquete IP, listo para ser transportado por una red de área local Ethernet.

3.2 Capa física y de enlace de datos

En la capa física y de enlace de datos, el estándar utilizado es Ethernet. A nivel de capa física, éste define el cableado sobre cable trenzado sin blindaje (UTP) y la interfaz física RJ45. A nivel de capa de enlace de datos, Ethernet define el

direccionamiento local, detección de errores, y control sobre el acceso a la capa física.

3.3 Capa de red

La capa de red describe el método de transferir tramas entre dispositivos en redes diferentes. El protocolo involucrado es el IPv4. Específicamente, se utiliza el servicio de red orientado a conexión [17]. Esto significa que sólo el primer paquete de cada mensaje tiene que llevar la dirección destino. Con este paquete se establece la ruta que deberán seguir todos los paquetes pertenecientes a esta conexión. Cuando llega un paquete que no es el primero, se identifica a que conexión pertenece y se envía por el enlace de salida adecuado, según la información que se generó con el primer paquete y que permanece almacenada en cada conmutador o nodo.

3.4 Capa de transporte

En la capa de transporte se utiliza el protocolo COTP [1], éste tiene un esquema de empaquetamiento para delimitar los PDU en forma de segmentos TCP.

Cada segmento TCP está compuesto por dos partes, un encabezado de paquete y un TPDU. El formato del encabezado del paquete es constante sin importar el tipo de paquete y corresponde a 4 bytes. El primer byte contiene la versión (que siempre será 3), el segundo byte es un campo reservado (*reserved)* y tiene un valor de cero. Los dos últimos bytes del encabezado contienen la longitud de todo el paquete incluyendo el encabezado.

Versión	Reservado	Longitud		TPDU
3	0	2	0	04 30 2A 6F .....

Figura 3.3 Contenido de un segmento TCP

3.5 Capa de sesión

El protocolo utilizado en esta capa es el ISO 8327: ISO *Session Protocol* [3] y [4].
En esta capa se deben definir los procedimientos para la transferencia de datos e información de control de una sesión a otra, entre dos entidades.
Las PDU de sesión son transferidas utilizando el servicio de transferencia para el transporte de datos (TSDU), este está formado por un número de unidades de datos de protocolo de sesión (SPDU). Puede haber hasta 4 SPDU.
La información necesaria para la definición de los parámetros de la sesión, puede ser consultada en la sección de anexos.

3.6 Capa de presentación

El mensaje MMS de la capa de aplicación, es codificado o decodificado en la capa de presentación, mediante el *Abstract Syntax Notation One Basic Encoding Rules* (ASN.1 BER) [5]. ASN.1 BER es un conjunto de reglas para convertir los datos definidos en ASN.1, en una representación particular para la transmisión hacia otro sistema.
En general una codificación en ASN.1 BER codifica los datos mediante etiqueta, longitud, contenido, a su vez en contenido puede haber información anidada. La siguiente figura muestra un ejemplo de contenido anidado.

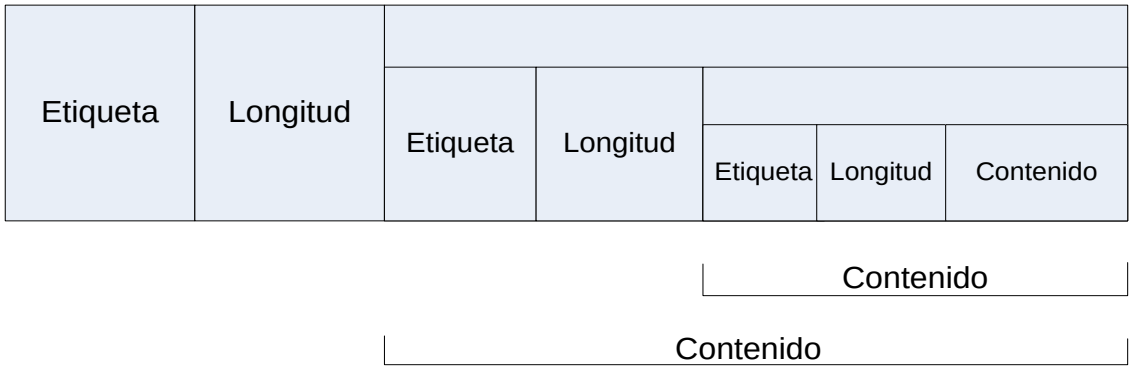


Figura 3.4 Paquete codificado en ASN.1 BER con contenido anidado

Adicionalmente a la codificación y decodificación ASN.1 BER, se debe agregar información, para la definición de los parámetros de la capa de presentación, éstos pueden ser consultados en la sección de anexos.

3.7 Capa de Aplicación

3.7.1 Agrupación de la información en una subestación eléctrica

IEC 61850 define un total de 13 grupos diferentes para agrupar datos. La idea, es que todos los datos que se pudieran originar en una subestación eléctrica, puedan ser clasificados en uno de estos grupos. La tabla 3.1 muestra los 13 grupos. Toda esta información que se genera en una subestación eléctrica, debe ser representada en Mensajes de Especificación del Fabricante.

Tabla 3.1 Nodos Lógicos

Grupos de Nodos Lógicos	Designación del grupo	Número
Nodos Lógicos del Sistema	L	2
Funciones de Protección	P	27
Funciones relacionadas con Protección	R	10
Control Supervisor	C	4
Referencias Genéricas	G	3
Interconexión y Archivo	I	4
Control Automático	A	4
Medición	M	7
Conmutación	X	2
Transformadores de instrumentación	T	2
Transformadores de Potencia	Y	4
Otros equipos y sistemas de potencia	Z	14
Sensores	S	3
Total		**86**

Cada uno de los grupos de la tabla anterior, está dividido en Nodos Lógicos. Hay 86 tipos diferentes, y cada uno de ellos, está compuesto por datos que representan algunas aplicaciones de significado específico. Así por ejemplo, el grupo de Funciones de Protección, posee 27 diferentes Nodos Lógicos.

Tabla 3.2 Clases de datos

Clase de Datos	Número
Información del Sistema	13
Información de Dispositivos Físicos	11
Medición	66
Valores medidos	14
Datos controlables	36
Información de Estado	85
Configuración	130
Total	**355**

Además, hay 355 clases de datos, divididas en 7 categorías. Estos son utilizados para construir los nodos lógicos. La tabla anterior muestra las 7 categorías.

Como se puede observar, la cantidad de información que se puede adquirir de un dispositivo de control o protección, puede ser muy grande. Es por eso que el ICE compra los IED a la medida, para manejar solo la cantidad de datos que necesitan.

3.7.2 Objetos y servicios MMS

A nivel de capa de aplicación, se debe agregar la señalización correspondiente, para definir una solicitud de asociación entre dos aplicaciones y el contexto de esta asociación, para esto se utiliza ISO 8650 parte 1 [6], la notación puede ser consultada en la sección de anexos.

Como se mencionó en la sección 3.7.1, toda la información que se genera en una subestación eléctrica, es representada en MMS, ésta información recibe el nombre de objetos MMS.

IEC 61850 hace uso de objetos y servicios MMS, cada mensaje que se recibe corresponde a un PDU MMS y aunque hay una gran cantidad de MMS PDU, para este proyecto se consideraron solo los que están involucrados en la inicialización y lectura:

confirmed-RequestPDU

- GetNameList

- GetNamedVariable
- GetNamedVariableList
- GetVariableAccessAtributtes

confirmed-ResponsePDU

- GetNameList
- GetNamedVariable
- GetNamedVariableList
- GetVariableAccessAtributtes

cancel-RequestPDU

initiate-RequestPDU

initiate-ResponsePDU

conclude-RequestPDU

conclude-ResponsePDU

Capítulo 4: Procedimiento Metodológico

4.1 IED para el transporte de información, desde los dispositivos de protección y control, hacia una LAN Ethernet

Como ya se mencionó, una de las funciones de los IED, es el transporte de información, desde los dispositivos de protección y control, hacia una LAN Ethernet.

Los IED, son comprados para transportar una cantidad de información específica, en caso de que se quiera transportar información adicional, se debe comprar otro IED. El problema que se presenta, es que el precio de estos dispositivos es muy elevado, y la compra implica un desperdicio de recursos, ya que aunque el IED tiene otras funciones, solo se utiliza una de ellas.

El estándar IEC 61850, define cómo los IED, realizan el transporte de información, en la forma ya mencionada. Para conocer cuál es la pila de protocolos utilizada por la norma IEC 61850, se procedió a estudiar los documentos que forman éste estándar [7]. Está documentación fue proporcionada por el ICE. En esos documentos se definen dos opciones para implementar la pila de protocolos, ambas opciones se muestran en la tabla 4.1.

Tabla 4.1 Opciones de implementación para la pila de protocolos de la norma IEC 61850

Capa de red	**Protocolo**	**Protocolo**
Aplicación	Servicios web	MMS
Presentación	Lenguaje de Marcas Extensible (XML)	ASN.1 BER
Sesión	HTTP	ISO de Sesión
Transporte	TCP	TCP
Red	IP	IP
Enlace de datos	Ethernet	Ethernet
Física		

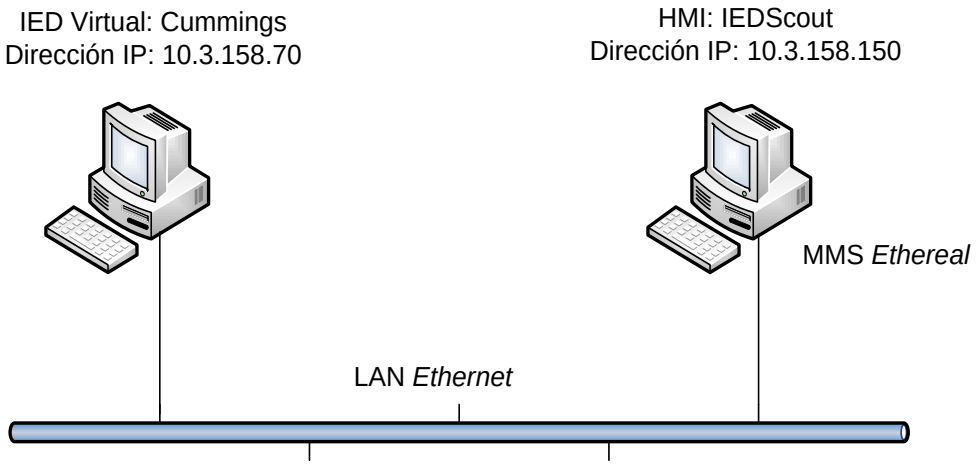
La opción con los protocolos de servicio web, XML y HTTP, fue descartada luego de realizar una investigación en Internet, y consulta de varios artículos escritos en referencia a esta norma [19] al [23]. En ellos se mostraba que esta pila de

protocolos no es una opción común en subestaciones eléctricas, es por esto que los programas HMI existentes en el mercado, utilizan la opción con los protocolos MMS, ASN.1 BER, ISO de sesión, TCP, IP y Ethernet.

Ya identificados los protocolos involucrado, se realizó una búsqueda bibliográfica. Los documentos con información relevante que fueron utilizados para realizar el marco teórico, y en general para consultas a lo largo del proyecto, fueron del [1] al [13].

Además, durante la investigación realizada, se determinó, que para conectar un dispositivo electrónico a una LAN en una subestación eléctrica, es necesaria cierta secuencia, sin embargo, en los documentos consultados, no se encontró cual debía ser ésta. Para determinar la secuencia, se hicieron pruebas simulando el tráfico de red, que se genera en una subestación eléctrica, cuando un programa HMI y un IED se comunican mediante una LAN Ethernet.

Para las pruebas, en una computadora se instaló el programa HMI *IEDScout* [14] y el analizador de paquetes de red MMS *Ethereal* [18]. En otra computadora se ejecutó un programa que simula un IED virtual, el nombre del programa es *Cummings* [14]. La figura 4.1 muestra el escenario en que se realizaron las pruebas, así como la asignación de las direcciones IP.



Microsoft Visio 2007

Figura 4.1 Configuración para realizar una captura del tráfico de red

Con el analizador de red, se capturó el tráfico en la interfaz con la dirección IP 10.3.158.150, ésta prueba permitió determinar, cuál es la secuencia para establecer la comunicación entre un IED y una HMI.

4.2 Solución alternativa para el transporte de MMS sobre IP, en una LAN Ethernet

De acuerdo al problema ya planteado, la solución debe ser tal, que el costo de la misma, lo justifique como una alternativa más conveniente que otras opciones, como adquirir un IED, únicamente para el transporte de mensajes sobre IP, en una LAN Ethernet.

Además, tanto para la generación de los paquetes IP, como para la interpretación, se debe hacer un intercambio de información, con los parámetros establecidos en las normas internacionales de la ISO e IEC.

También, para la solución, se debe considerar que el dispositivo seleccionado, debe funcionar como un elemento de red, y por tanto debe tener posibilidades de configurarle una dirección IP, máscara de red, así como el enrutador por defecto.

Como solución, se desarrolló el software para un sistema electrónico, que realice el transporte de MMS sobre IP, desde un dispositivo de protección o control, hacia una LAN Ethernet. Para hacer referencia a este sistema, en adelante, se utilizará el nombre de: *Módulo de transporte MMS sobre IP*.

Para el sistema electrónico, se eligió un microcontrolador orientado a funciones de red y con una interfaz física RJ 45, para permitir la conexión a una LAN Ethernet. Evaluando información acerca de microcontroladores, se encuentra el *Rabbit* 3000, éste es orientado a funciones de red y tiene un costo relativamente bajo, si se compara con otros microcontroladores de características similares. Además, mediante consulta del material que proporciona el fabricante del microcontrolador: hoja de datos, manual de usuario TCP/IP Vol. 1 y 2 [17], se determinó, que éste microcontrolador tiene una implementación de las capas de red de la 1 a la 4, con los protocolos Ethernet, IP y TCP.

Bajo ese criterio, se elije el microcontrolador *Rabbit* 3000, con su módulo de desarrollo RCM 3300. El precio del microcontrolador y del módulo de desarrollo fue de $275, este precio resulta razonable, si se compara con los $3000 que puede llegar a costar un IED.

El proceso de desarrollo del software para el *Módulo de transporte MMS sobre IP*, se dividió en cinco etapas:

Escritura de los requerimientos del software: Una vez definido el hardware, se procedió a plantear lo requerimientos del software a desarrollar, éstos se detallan en el capítulo 5. Los requerimientos definen, qué es lo que el sistema debe hacer, escrito de la manera más específica posible.

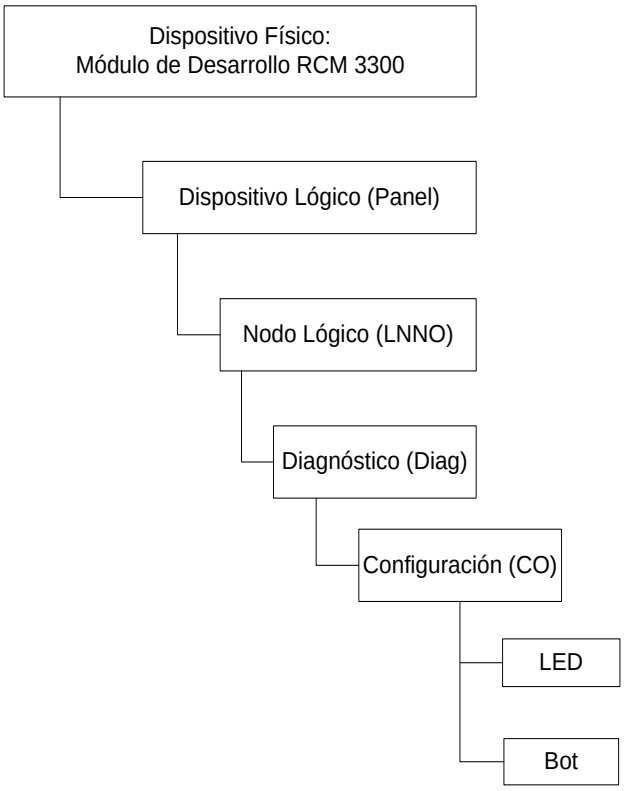
Diseño del software: A partir de los requerimientos del software, se realizó el diseño del mismo. Como producto de esta etapa, se generaron los diagramas de flujo mostrados en la sección 5.1, estos diagramas muestran, cómo el software realizará lo planteado en los requerimientos.

Programación: Con base en la información de la etapa anterior, se desarrolló el código fuente. Debido al microcontrolador seleccionado, se utilizó el programa Dynamic C para la programación.

Integración del software en el hardware: Una vez que se desarrolló el código fuente, el software se integró con el hardware. Para este proceso, se utilizó también el programa Dynamic C, este realiza la compilación del código fuente y el enlace (*linking*), para generar el código objeto, que fue descargado en el microcontrolador *Rabbit* 3000.

Pruebas para verificar del funcionamiento del sistema: Luego del proceso de integración, se verificó el óptimo funcionamiento del *Módulo de transporte MMS sobre IP*. Para ésta etapa, se eligió un programa HMI, que permitiera la adquisición de datos de manera remota, mediante una LAN Ethernet, bajo la norma IEC 61850. El programa seleccionado fue el *IEDScout* V1.5 [15], los criterios para su selección se detallan la sección de apéndices.

Además, para poder realizar ésta verificación, se desarrolló una rutina, que emula el funcionamiento de un equipo de control, cuya información quiere ser transportada a una LAN mediante el *Módulo de transporte MMS sobre IP*. Ésta rutina controla un LED y un interruptor, que se encuentran en el módulo de desarrollo, donde está montado el microcontrolador. El LED y el interruptor representan dos variables de configuración (CO), dentro de un grupo de variables de diagnóstico (Diag). Todas estas variables están agrupadas en un nodo lógico, el LLNO. A su vez, todo lo anterior esta contenido en un dispositivo lógico llamado Panel. En la figura 4.2 se muestra un diagrama con la agrupación de los datos dentro del módulo de desarrollo RCM 3300.



Microsoft Office Visio 2007

Figura 4.2 Estructura de datos dentro del módulo de desarrollo.

Capítulo 5: Desarrollo del software para un microcontrolador *Rabbit* 3000, con la función de transportar MMS sobre IP

La función del sistema electrónico utilizado como solución, es la de transportar MMS sobre IP, desde un dispositivo de protección o control, hacia una LAN Ethernet. Como primer paso, para el desarrollo del software para el microcontrolador *Rabbit* 3000, se procedió a plantear los requerimientos del software a desarrollar, éstos se presentan a continuación:

1. Durante la inicialización del sistema, se debe establecer una dirección IP y un puerto, éstos forman el socket de comunicación.

2. Durante la inicialización del sistema, se debe establecer una máscara de red y enrutador por defecto. También, se debe habilitar la interfaz física de red.

3. Luego de la inicialización del sistema, se debe abrir una conexión pasiva sobre el socket de comunicación.

4. Al recibir una petición de conexión TCP, el sistema debe responder según lo establece el mecanismo de acuerdo de tres vías (*three way handshaking*).

5. Cuando se reciba una petición de conexión MMS de una HMI, se debe iniciar el proceso de conexión del dispositivo electrónico, a una LAN, según lo establece el estándar IEC 61850.

6. Cuando se reciba una petición de lectura (*Read*) de una HMI, se debe responder la solicitud, haciendo el transporte de MMS sobre IP, desde el dispositivo de control o protección hacia la LAN, según lo establece el estándar IEC 61850.

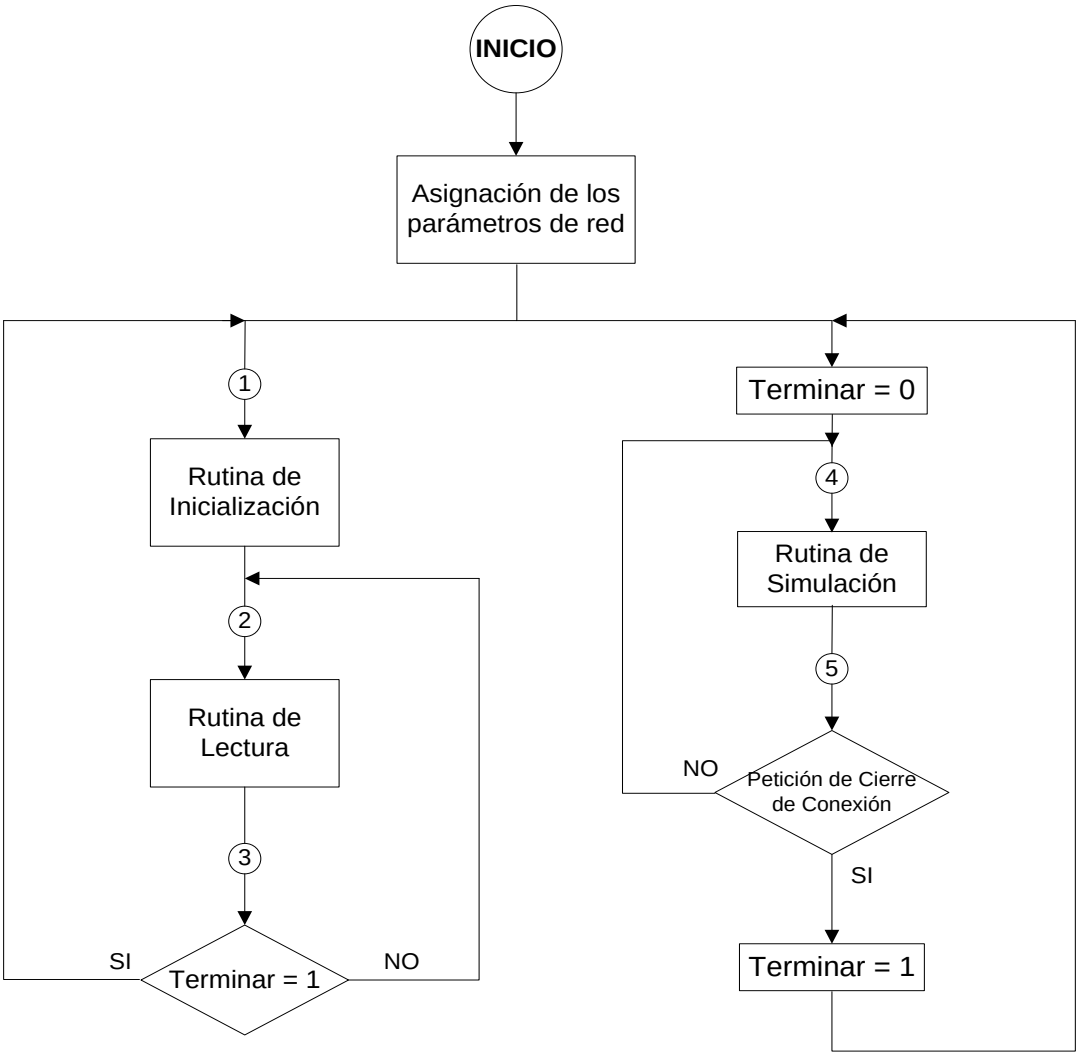
A partir de estos requerimientos, se procedió a escribir el código fuente. En las rutinas escritas se distinguen dos tareas que se están realizando en paralelo.

- Tarea 1: Realiza la secuencia de inicialización, para conectar el *Módulo de transporte MMS sobre* IP a una LAN. También, responder a las peticiones de Lectura, hecha por una HMI.

- Tarea 2: Emula el comportamiento de un dispositivo de control. cierre de la conexión, en caso de que el HMI lo pida.

En la figura 5.1, se muestra un diagrama de flujo con la lógica del programa principal.

Inicialmente, al *Módulo de transporte MMS sobre IP*, se le asignó una dirección IP estática, un número de puerto, y una máscara de red definida. La dirección IP fue 192.168.1.4, el puerto elegido es el 102, y la mascara de red 255.255.255.0.



Microsoft Office Visio 2007

Figura 5.1 Diagrama de flujo del programa principal

5.1 Rutina de Inicialización

Como primer paso, se debe establecer una conexión TCP, mediante el procedimiento llamado negociación en tres pasos (*three way handshake*). Durante el establecimiento, algunos parámetros son configurados para asegurar la entrega ordenada de los datos y la robustez de la comunicación. En este caso el programa HMI es el que realiza la solicitud de conexión, mientras que el *Módulo de transporte MMS sobre IP* espera mediante una conexión pasiva.

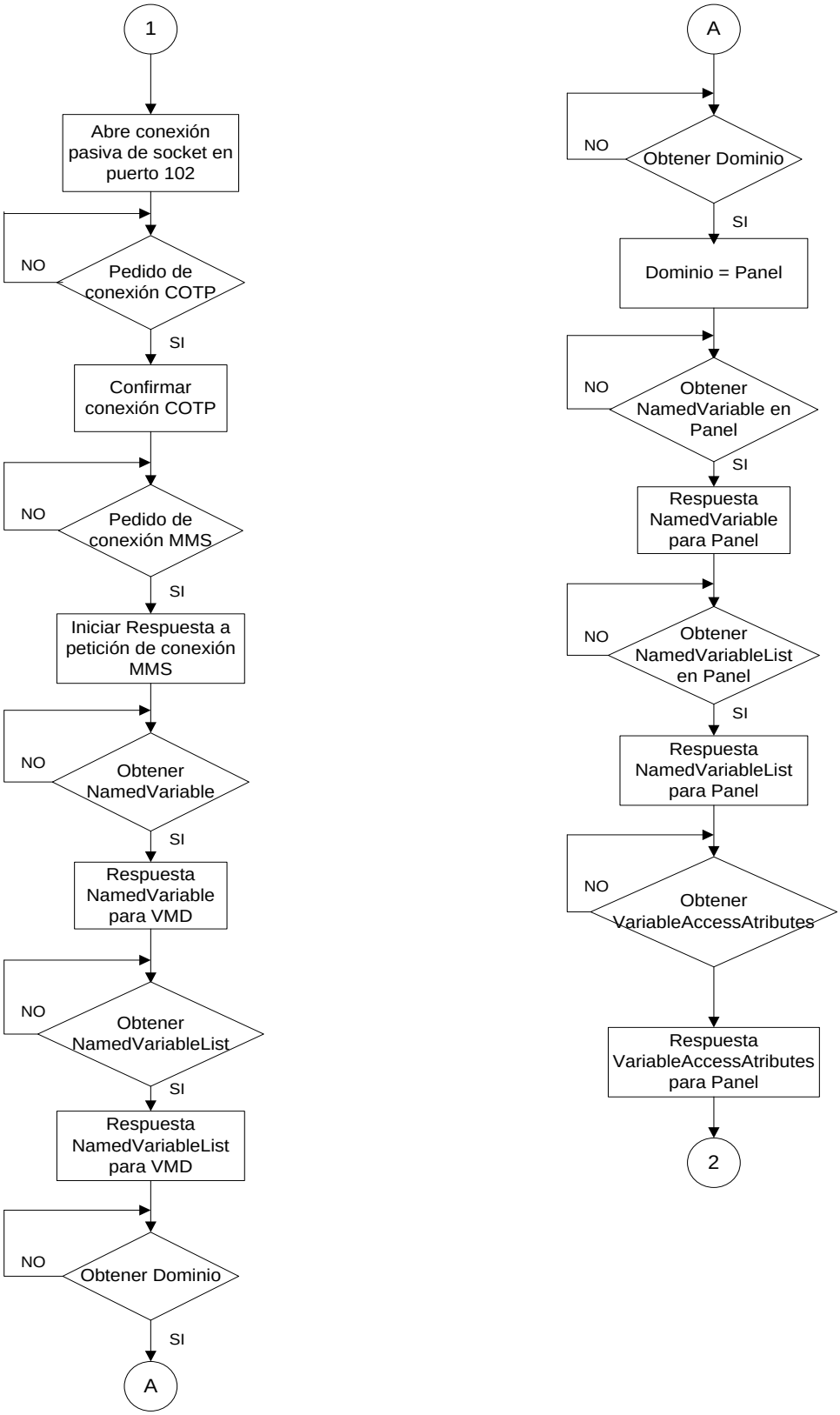
Por conexión pasiva, se debe entender una conexión, que se activará hasta que alguien se conecte a la dirección IP y puerto específico, estos dos datos forman el

socket de comunicación. Por defecto se establece que el microcontrolador recibe y envía por el puerto 102, mientras que el HMI puede elegir un valor cualquiera para el puerto.

Inicialmente, el *IEDScout* no tiene información acerca del *Módulo de transporte MMS sobre IP*, se tiene que dar un intercambio de paquetes, para establecer el nodo lógico y el tipo de datos que contiene el Módulo, así como los valores de estos datos. La figura 5.3 muestra un diagrama de flujo, con la secuencia necesaria para conectar el microcontrolador a la LAN y así conocer la información que contiene, todo esto según lo define el estándar IEC 61850. Esta figura es un detalle de la rutina de inicialización mostrada en la figura 5.1.

Cuando el cliente se conecta al socket, la conexión se activa y se queda en espera de que se establezcan los servicios de la capa de transporte, esto mediante un paquete de pedido de conexión de COTP (CR COTP PDU). A esto, se le responde con un paquete de confirmación de conexión COTP (CC COTP TPDU). Durante este intercambio, se negocia toda la información y parámetros necesarios para el funcionamiento de las entidades de transporte. La información necesaria para el CR TPDU, CC TPDU y los TPDU que se agregan en cada intercambio pueden ser consultados en [2].

Luego de establecer la conexión de la capa de transporte, se intercambian los mensajes necesarios, para que el programa cliente, pueda conocer todos los objetos y servicios disponibles dentro del *Módulo*. Cada vez que se haga un intercambio de información, mediante un paquete, habrá un número asociado a este. Los paquetes que el HMI y el Modulo de transporte MMS sobre IP intercambian, deben corresponder al mismo número, este número se llama INVOKE ID y puede ser de hasta 32 bits, el objetivo de este número identificar una petición con su respectiva respuesta.



Microsoft Office Visio 2007

Figura 5.2 Secuencia de inicialización del servidor

A continuación, el programa *IEDScout* hace un pedido de los nombres de variable, si los hay, el módulo responde con los mismos. Luego, el cliente hace un pedido de los nombres de las listas de variables, en caso de que existan, el servidor responderá.

Después el cliente pregunta si hay dominios, dentro el *VMD*, a esto el servidor contesta con el nombre del dispositivo físico (Panel). A partir de ese momento el cliente sabrá, que por el momento hay un objeto MMS y su nombre es Panel.

A continuación, el cliente empieza a investigar cuáles son los objetos dentro de Panel, preguntando primero por los nombres de variables y luego por los nombres de listas de variables.

Para este proyecto los nombres de variable corresponden a:
LNNO
LLNO$CO
LNNOCODiag
LNNOCODiag$LED
LNNOCODiag$BOT

Y los nombres de lista a:

LLNO$CO

Cuando el cliente pida los nombres de lista en CO, el servidor debe responder con el tipo de dato y valor de las variables contenidas en LNNOCODiag, que para este caso son LNNOCODiag$LED y LNNO$CO$Diag$BOT.

Esto significa que LED y BOT, son atributos, que a su vez están contenidos en Diag, que es un dato de diagnóstico. Los datos y atributos anteriores están dentro de un nodo lógico del sistema llamado LLNO. El dispositivo físico y quien en última instancia es visible en la red, es Panel.

Con los pasos anteriores el programa cliente ya conoce qué es lo que hay dentro del servidor, pero no conoce qué valores, ni qué tipo de datos están contenidos en

él, es por esto que debe hacerse una lectura de los atributos de acceso para las variables (*Get variable access attribute*). Los valores que pueden tomar LED y BOT son verdadero y falso, ya que son de tipo boolean.

Cada uno de los paquetes mostrados en el diagrama de flujo de la figura 5.2, corresponde a un MMS PDU. Por ejemplo, la figura siguiente muestra un MMS PDU, este PDU esta formado internamente como se muestra en la figura 5.4. Esta figura es un ejemplo que será utilizado para mostrar la codificación y decodificación de un MMS PDU.

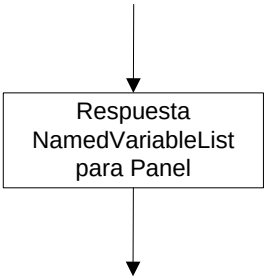


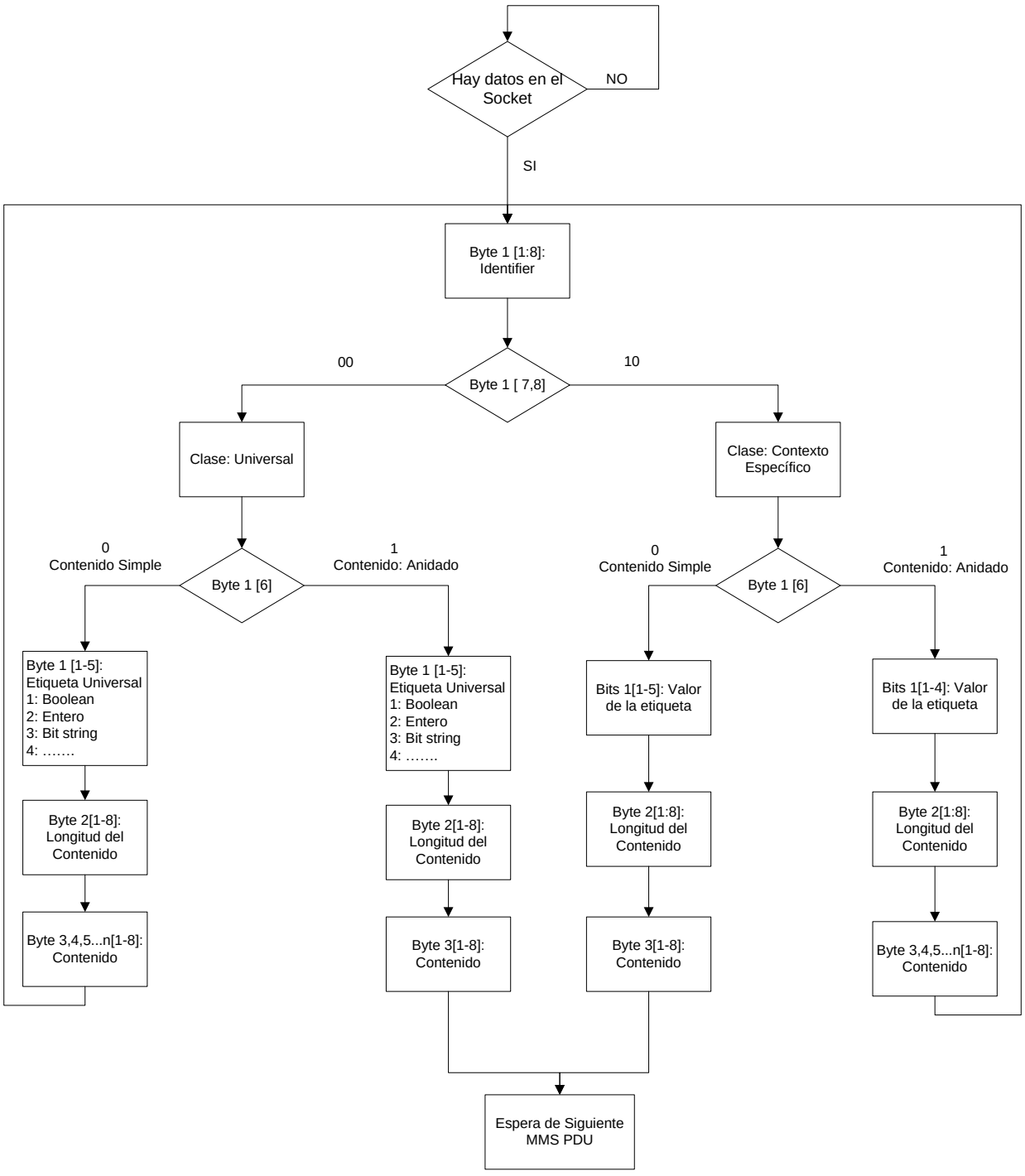
Figura 5.3 Mensaje recibido o generado por el Sistema Generador de Mensajes bajo la Norma IEC

En el capítulo 3, marco teórico, se especifica la información de las capas física, de red, transporte, sesión y presentación en un MMS PDU. En la figura siguiente, se muestra el mensaje que resulta, después de quitar la información correspondiente a las capas de la 1 a la 6 del modelo OSI.

Byte 1	Byte 2	Byte 3	Byte 4	34	4a
78	61	5b	4d	E0	02
…	…	…	…		

Figura 5.4 Mensaje recibido o generado por el Módulo de transporte MMS sobre IP, sin la información de la capa 1 a la 6.

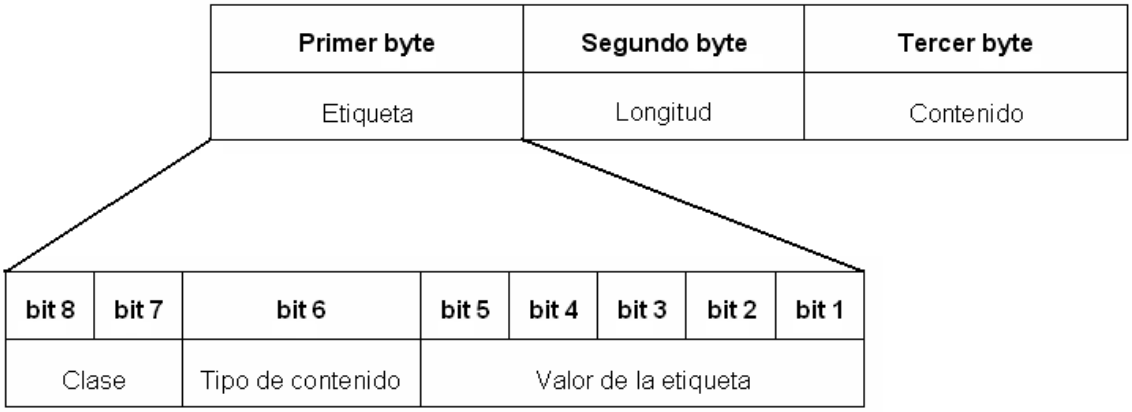
La siguiente figura muestra el algoritmo que se diseñó para codificar y decodificar los MMS PDU ya sea en la *Rutina de Inicialización* o en la de *Lectura*.



Microsoft Office Visio 2007

Figura 5.5 Diagrama de flujo para la codificación y decodificación de mensajes MMS.

Siguiendo el diagrama anterior y como se mencionó en el marco teórico, un mensaje MMS codificado en ASN.1 BER, consta de 3 partes, Etiqueta, Longitud, Contenido. En la siguiente figura se muestra un esquema de esta representación, Etiqueta corresponde al primer byte, Longitud al segundo y Contenido al tercer byte del mensaje de la figura 5.6.



Microsoft Office Visio 2007

Figura 5.6 Esquema de un mensaje MMS codificado en ASN.1 BER

La Etiqueta a su vez está formada por: Clase, Tipo de contenido y Valor de la Etiqueta, a continuación se muestra la distribución de cada bit.

Por lo general, es común que en los mensajes se den valores de 00 y 10 para Clase, esto representa una Clase universal y una Clase de contexto específico, respectivamente.

Si la clase es universal, el valor de la etiqueta en los bits 1 al 5 son interpretados según la siguiente tabla:

Tabla 5.1 Tipo de dato de acuerdo al valor de la etiqueta universal

Valor de la etiqueta (Hexadecimal)	Tipo de dato
01	Boolean
02	Integer
03	BitString
04	OctecString
05	Null
06	Object Identifier
10	Sequence
16	IA5String
17	UTCTime
18	GeneralizaTime
1A	VisibleString

Si la clase es de contexto específico, significa que el valor de la etiqueta debe ser interpretado según el documento en [9]. Este documento en versión digital, contiene todos los servicios y objetos definidos en ASN.1 para MMS, tanto la parte uno y dos de la norma ISO 9506.

Debido a lo extenso que sería presentar cada uno de los PDU, a continuación se presenta el caso específico de confirmed-RequestPDU, cuando se quiere hacer un GetNamedList.

Los mensajes están escritos en Abstract Syntax Notation 1, esto solo es una forma de representación abstracta, no corresponde a ningún lenguaje de programación. El número dentro de los corchetes cuadrados es el número de opción que se elije en cada paso.

Primero se elije el tipo de PDU.

```
MMSpdu ::= CHOICE {
  confirmed-RequestPDU      [0] IMPLICIT Confirmed-RequestPDU,
  confirmed-ResponsePDU     [1] IMPLICIT Confirmed-ResponsePDU,
  confirmed-ErrorPDU        [2] IMPLICIT Confirmed-ErrorPDU,
  unconfirmed-PDU           [3] IMPLICIT Unconfirmed-PDU,
  rejectPDU                 [4] IMPLICIT RejectPDU,
```

```
  cancel-RequestPDU          [5] IMPLICIT Cancel-RequestPDU,
  cancel-ResponsePDU         [6] IMPLICIT Cancel-ResponsePDU,
  cancel-ErrorPDU            [7] IMPLICIT Cancel-ErrorPDU,
  initiate-RequestPDU        [8] IMPLICIT Initiate-RequestPDU,
  initiate-ResponsePDU       [9] IMPLICIT Initiate-ResponsePDU,
  initiate-ErrorPDU          [10] IMPLICIT Initiate-ErrorPDU,
  conclude-RequestPDU        [11] IMPLICIT Conclude-RequestPDU,
  conclude-ResponsePDU       [12] IMPLICIT Conclude-ResponsePDU,
  conclude-ErrorPDU          [13] IMPLICIT Conclude-ErrorPDU
}
```

Cuando se tiene el tipo de PDU se busca como se procede cuando la opción es *confirmed-ResponsePDU*, en este caso se encuentra que se debe incluir el *invokeID* y el *service*.

```
Confirmed-ResponsePDU ::= SEQUENCE {
  invokeID       Unsigned32,
  service        ConfirmedServiceResponse,
}
```

InvokeID corresponde a un número de hasta 32 bits y *service* corresponde a alguno de los servicios mencionados abajo. El *service* que se ocupaba es el getNameList [1].

```
ConfirmedServiceResponse ::= CHOICE {
  status                     [0]    IMPLICIT          Status-Response,
  getNameList                [1]    IMPLICIT    GetNameList-Response,
  identify                   [2]    IMPLICIT          Identify-Response,
  rename                     [3]    IMPLICIT          Rename-Response,
  read                       [4]    IMPLICIT          Read-Response,
  getVariableAccessAttributes [6]   IMPLICIT  GetVariableAccessAttributes
  defineNamedVariable        [7]    IMPLICIT       DefineNamedVariable,
…
```

```
…
…
 additionalService              [78]    AdditionalService-Response,
getDataExchangeAttributes       [80]    GetDataExchangeAttributes-Response,
 }
```

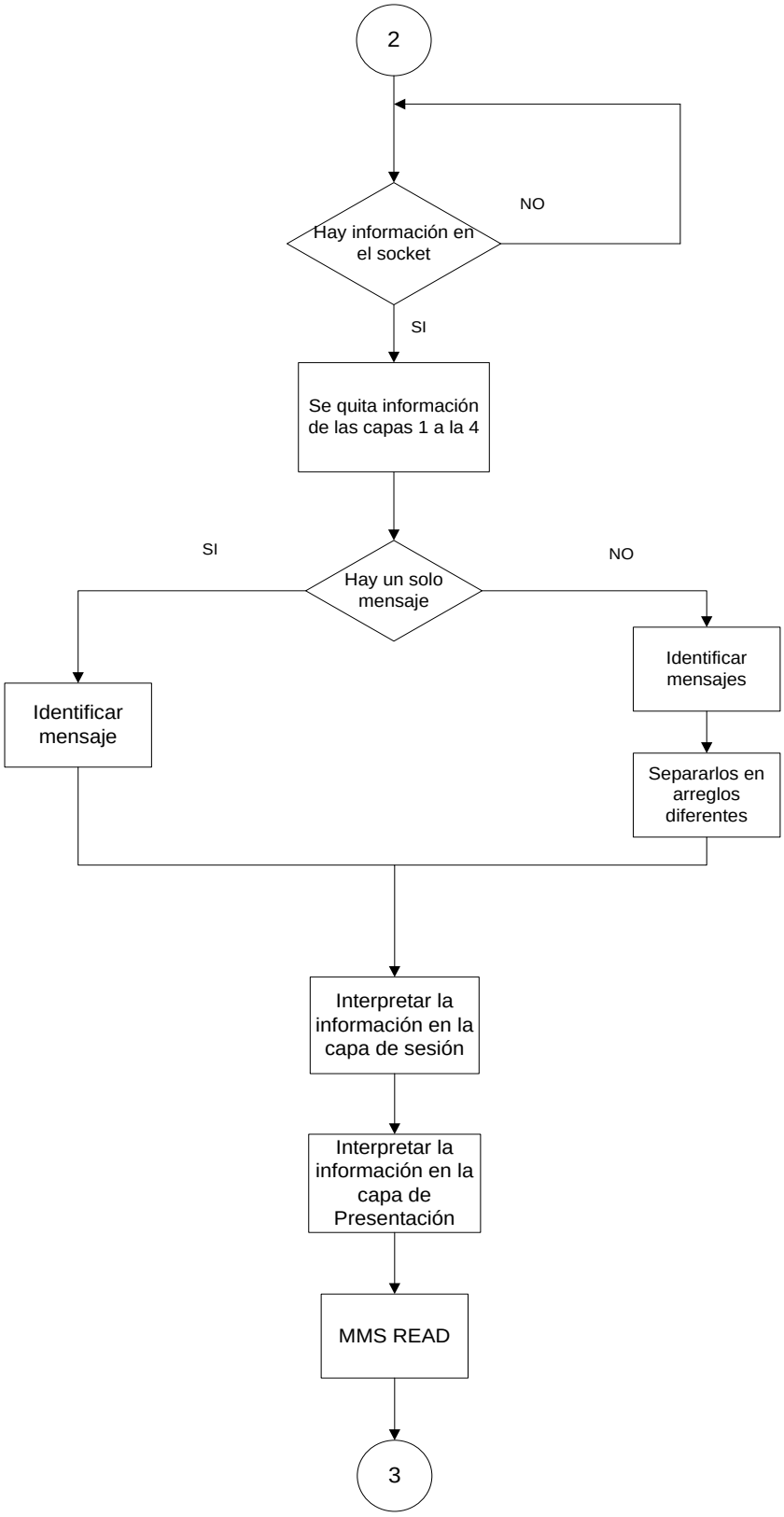
Se dice que es un lenguaje implícito y abstracto ya que dos entidades no necesitan enviarse toda esta información para saber que opciones se eligieron, solo con los números entre corchetes es suficiente.

Ahora, una vez que se define en sintaxis abstracta el contenido del mensaje, se deben usar las reglas de ASN.1 [10] y [11] para expresar el mensaje apropiadamente, las reglas funcionan a nivel de capa de presentación y su nombre es Reglas Básicas de Codificación (BER).

El bit 6 indica si el tercer byte de contenido tiene más información anidada. Un cero es contenido *primitive*, o simple y un uno representa anidamiento. En caso de existir anidamiento los pasos anteriores se siguen recursivamente hasta que no existan más datos.

5.2 Rutina de lectura (*Read*)

Una vez que se logra exitosamente una conexión mediante los pasos mencionados en esta sección, el programa cliente está listo para hacer una lectura de los valores que están dentro del servidor. El diagrama de flujo de la siguiente figura muestra los pasos para realizar una operación de Lectura.

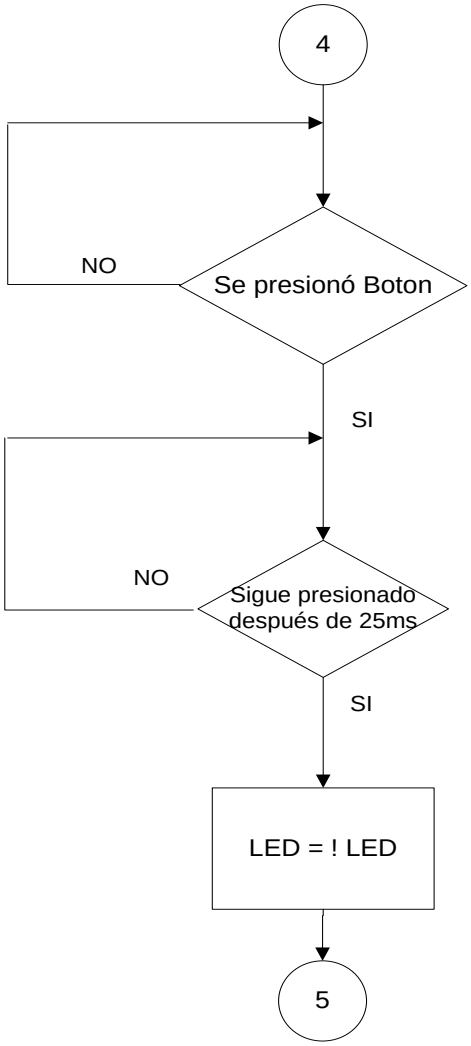


Microsoft Visio 2007

Figura 5.7 Diagrama de flujo para realizar una lectura

5.3 Rutina de Simulación

El otro proceso que se ejecuta paralelamente a la rutina de Lectura, es la rutina de Simulación, esta rutina simula la información contenida en un dispositivo de control, y fue implementada para realizar pruebas. El diagrama de flujo de la siguiente figura muestra las acciones que se realizan.



Microsoft Office Visio 2007

Figura 5.8 Diagrama de flujo de la rutina de lectura

Para evitar el efecto de rebote cuando se presiona el botón, se pregunta si el mismo sigue presionado después de 25ms de haber sido presionado, si la respuesta es sí, entonces, se cambia el estado del LED y este cambio se refleja en la rutina de Lectura.

Capítulo 6: Análisis y Resultados

Para todas las pruebas realizadas con el microcontrolador y el módulo de desarrollo, la dirección IP utilizada fue la 192.168.1.4, mientras que a la interfaz de red de la computadora se le asignó la dirección IP 192.168.1.2.

Rutina de Inicialización

Para verificar que el *Módulo de transporte de MMS sobre IP*, funciona correctamente, se realizaron pruebas con un programa con funciones HMI (*IEDScout* V 1.5).

El *IEDScout* tiene dos ventanas que son de interés. La primera es la ventana de mensajes (*Messages*), en esta ventana se observa el estado de la conexión con un dispositivo de protección o control, mediante la LAN. La segunda venta se llama vista de datos (*Data View*), y en ella se observa el contenido del dispositivo de control o protección al que se este conectado.

Las pruebas realizadas, consistieron en poner el LED y el interruptor en estados conocidos, y registrar los datos en las ventanas del *IEDScout*. También se realizaron capturas de red, para determinar el contenido de los paquetes que se estaban intercambiando.

En la figura 6.1, se presenta una captura hecha con el *MMS Ethereal* utilizando el programa *IEDScout* y el Módulo de transporte MMS sobre IP, en ella se observan los paquetes intercambiados. Donde bajo la columna *Source* está la dirección IP del dispositivo que envía el mensaje y bajo la columna *Destination* está la dirección IP de la interfaz a quien va dirigido el mensaje. Bajo la columna *Protocol* está la información del protocolo utilizado en la operación.

Ahí se observa que los primeros paquetes TCP corresponden al establecimiento de la conexión (three way handshake), los paquetes COTP son los encargados de establecer los servicios de la capa de transporte, entre otras funciones y los

mensajes MMS corresponden a las operaciones MMS sobre los objetos de la norma IEC 61850.

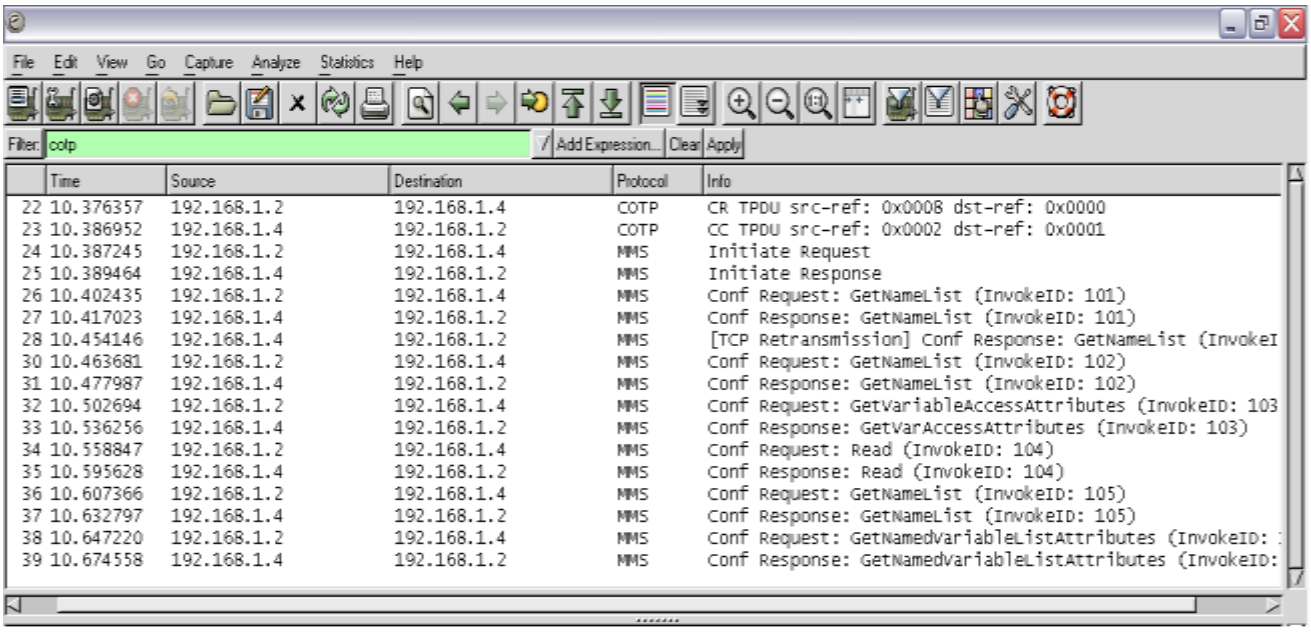


Figura 6.1 Captura del tráfico de red realizada en la interfaz del programa cliente IEDScout.

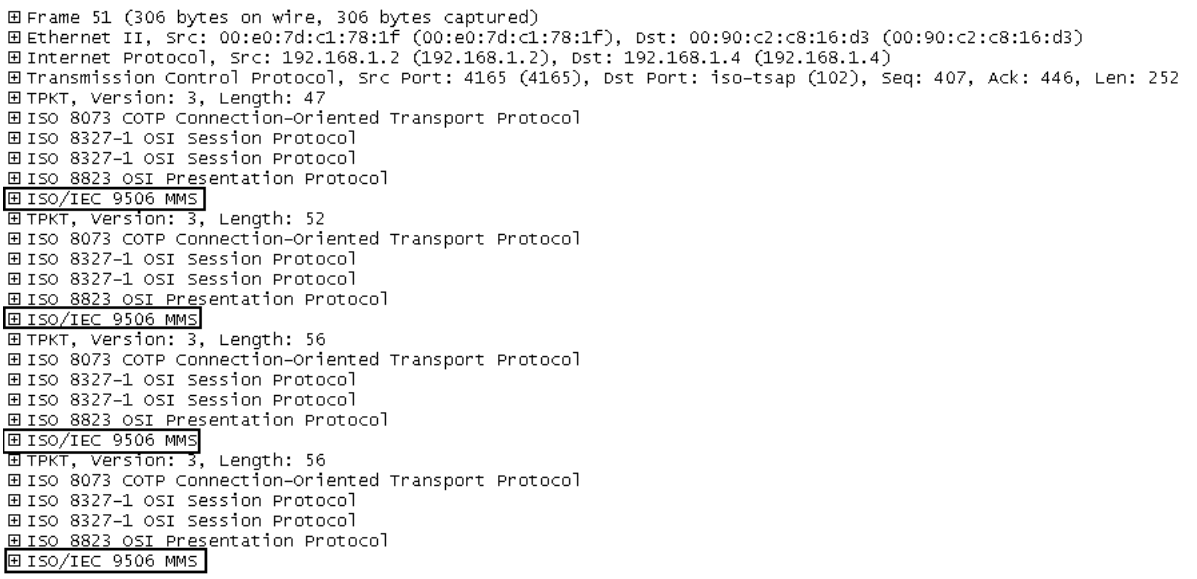
En la siguiente figura se muestra una imagen de la ventana de *Messages* del *IEDScout.* Encerrado en un círculo se muestra el mensaje *Completed*, indicando que la conexión se realizó positivamente. Los mensajes anteriores a este, indican qué acción está realizando el programa cliente en ese momento.

Figura 6.2 Mensaje de Conexión establecida positivamente en la ventana de Messages

Con el resultado obtenido en esta ventana, se asegura que las rutinas escritas para el *Módulo de transporte MMS sobre IP*, generan las tramas de información con el formato necesario, establecido por la norma para que se establezca la comunicación.

Durante la etapa de verificación del funcionamiento del software, el analizador de red mostró, que en un solo paquete que el *IEDScout* enviaba, podía estar contenido más de un mensaje MMS. Estos paquetes debieron ser interpretados de forma diferente. Para procesar este tipo de tramas, primero se quitó la parte que corresponde a las capas de red de la 1 a la 4, luego se comparó la longitud del encabezado de paquete TPKT con la longitud del mensaje almacenado en el buffer, en caso de ser diferentes se estaba en presencia de varios mensajes, por lo que se procedió a separarlos y guardarlos en variables diferentes para luego procesarlos según los algoritmos descritos en la solución detallada.

En la captura de la figura siguiente se muestra el caso de una trama que contiene 4 mensajes MMS, estos están encerrados en un rectángulo. Cada uno tiene su respectiva información en las capas de sesión, presentación y transporte.



Ethereal

Figura 6.3 Captura de una trama enviada por el cliente que contiene 4 mensajes MMS

Rutina de Lectura (*Read)*

En la figura 6.4 se observa un mensaje capturado con el analizador de red, en ella el programa cliente realiza la lectura de *LLNOCODiag$Led* mediante la operación *Read.* El programa cliente también puede realizar lecturas de cada una de las variables, así como de los nombres de lista de las variables:

LLNOCODiag$Led
LLNOCODiag$Bot
LLNOCODiag

Mediante la rutina de *Read* se logró que el programa *IEDScout* hiciera una lectura de los valores del LED y el interruptor.

```
⊞ Frame 62 (116 bytes on wire, 116 bytes captured)
⊞ Ethernet II, Src: 00:e0:7d:c1:78:1f (00:e0:7d:c1:78:1f), Dst: 00:90:c2:c8:16:d3 (00:90:c2:c8:16:d3)
⊞ Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 192.168.1.4 (192.168.1.4)
⊞ Transmission Control Protocol, Src Port: 3940 (3940), Dst Port: iso-tsap (102), Seq: 536, Ack: 553, Len: 62
⊞ TPKT, Version: 3, Length: 62
⊞ ISO 8073 COTP Connection-Oriented Transport Protocol
⊞ ISO 8327-1 OSI Session Protocol
⊞ ISO 8327-1 OSI Session Protocol
⊞ ISO 8823 OSI Presentation Protocol
⊟ ISO/IEC 9506 MMS
    Conf Request (0)
    Read (4)
    InvokeID: InvokeID:  108
  ⊟ Read
    ⊟    List of Variable
      ⊟      VariableSpecification
        ⊟        Object Name
          ⊟          Domain Specific
            ⊟ DomainName:
                DomainName: Panel
            ⊟ ItemName:
                ItemName: LLN0$CO$Diag$Led
```

Ethereal

Figura 6.4 Captura del tráfico de red cuando se realiza una lectura de datos.

Debido a que el tamaño de los paquetes que se intercambian entre el cliente y el servidor es menor que 1024 bytes, no fue necesario dividir un mensaje COTP en varias partes, es por esto que en cada COTP PDU se indica que solo un paquete se envía. En caso de ser necesario un intercambio de tramas de mayor longitud, se debe hacer una modificación a nivel de capa de transporte y hacerlo según está definido en el protocolo ISO 8073 [1].

Aunque para el caso específico de este proyecto los datos manejados para LED y el interruptor son de tipo booleano, en caso de querer manejar otro tipo de datos, debería hacerse un cambio mínimo en la definición de los datos, utilizando la tabla 5.1 que muestra la codificación en ASN.1 para datos.

Capítulo 7: Conclusiones y Recomendaciones

7.1 Conclusiones

Respecto al estándar IEC 61850:

- La pila de protocolos utilizada es: MMS, ASN.1 BER, Protocolo de Sesión, TCP, IP, Ethernet. Enumerados de la capa siete a la uno del modelo OSI

- Mediante la pila de protocolos de la norma IEC 61850, se logra encapsular, un mensaje MMS sobre IP, para su transporte sobre una LAN Ethernet

Respecto al desarrollo de software para el *Módulo de transporte MMS sobre IP:*

- Se desarrolló como un proceso de cinco etapas consecutivas: escritura de los requerimientos del software, diseño del software, programación, integración del software en el hardware y pruebas para verificar el funcionamiento del sistema.

- La información obtenida al final de cada etapa, es una entrada indispensable para la etapa siguiente.

- La escritura de los requerimientos del software, debe estar orientada, a qué debe realizar éste, y no el cómo lo hará.

- Una buena definición de requerimientos, facilita la escritura del código fuente.

Respecto al proyecto en general:

- Se logró desarrollar exitosamente, una alternativa de menor costo, para el transporte de paquetes MMS sobre IP, desde un dispositivo de control, hacia una LAN Ethernet. El costo del hardware utilizado fue de $275.

7.2 Recomendaciones

1. Como una implementación inicial, el *Módulo de transporte MMS sobre IP*, tiene solo funciones básicas, para futuros desarrollos, debería considerarse agregar la función de envío de información cada cierto periodo de tiempo, sin necesidad de que sea el HMI el que haga la petición.

2. Analizar en una aplicación específica, si el uso de un servidor con el protocolo *DHCP,* sería el adecuado para asignar los parámetros de red.

3. Debido a que en la capa de transporte se define una longitud máxima de 1024 bytes para un *TPDU*, en caso de que la información a enviar sea de mayor tamaño, se debe hacer una modificación a nivel de esta capa para manejar tramas dentro de la longitud máxima permitida. Esta modificación debe ser hecha siguiendo la especificación del protocolo *COTP* [1] y [2].

Bibliografía

[1] [2] Protocolos ISO sobre TCP. (en línea). Consultadas 18 feb. 2008. Disponibles en http://www.ietf.org/rfc/rfc0905.txt. y en http://www.faqs.org/rfcs/rfc2126.html

[3] Protocolos en las capas de presentación y sesión (en línea). Consultada 23 de feb. 2008. Disponible en http://www.protocols.com/pbook/iso.htm#ISO-SP

[4] [5] Documentos de la ITU acerca de los servicios de la capa de sesión y presentación. (en línea). Consultadas 26 de feb. 2008. Disponible en http://www.itu.ch/itudoc/itu-t/rec/x/x200-499/x225_32038.html y en http://www.itu.ch/itudoc /itu-t/rec/x/x200-499/x226_25241.html

[6] Association Control Service, protocolo ISO 8650. (en línea)., Consultada el 18 feb. 2008. Disponible en http://www.itu.int/ITU-T/asn1/database/itu-t/x/x227/1995/ACSE-1.html#ACSE-1.AARQ-apdu

[7] IEC 61850: Communications Networks and Systems in Substations, International Standard. 2005.

[8] Información del Estándar ISO 9506 - 1 -2. (en línea). Consultada 12 marzo2008. Disponible en http://www.nettedautomation.com/standardization/ISO/TC184/ SC5/WG2/mms_syntax/index.html

[9] Codificación ASN.1 para MMS. (en línea). Consultada 25 de marzo 2008. Disponible en http://asn1.elibel.tm.fr/asnp/

[10] *Abstract Syntax Notation One (ASN.1)* **Specification of Basic Notation** ITU-T Rec.680 (2002) | ISO/IEC 8824-1:2002

[11] *ASN.1 encoding rules* **Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)** ITU-T Rec. X.690 (2002) | ISO/IEC 8825-1:2002

[12] *Abstract Syntax Notation One (ASN.1)* **Specification of Basic Notation** ITU-T Rec. X.680 (2002) | ISO/IEC 8824-1:2002

[13] *Abstract Syntax Notation One (ASN.1)* **Information Object Specification** ITU-T Rec. X.681 (2002) | ISO/IEC 8824-2:2002

[14] Programa Cliente de demostración BrowzerD y Servidor Cummings, TAMARAK Consulting. (en línea). Disponible en www.nettedautomation.com

[15] Programa Cliente de muestra IEDScout V1.5, OMICRON. (en línea). Disponible en http://www.omicron.at/

[16] Programa Cliente de muestra de la empresa INFOTECH. (en línea). Disponible en http://www.infotech.pl/

[17] Rabbit microprocessor DCR 9.21 Documents (CD - ROOM). 1 CD – ROOM (Contiene software: Dynamic C 9.21).

[18] Programa MMS Ethereal. Consultado 6 feb. 2008. Disponible en www.ethereal.com

[19] Pereira Neto, A. 2006. Redes Ethernet en Subestaciones & La Norma Técnica

[20] Schwarz Consulting Company. 2006. *IEC 61850, IEC 61400-25, and IEC 61970 : Information exchange for electric power systems*.

[21] Systems Integration Specialists Company, Inc. 2007. *New Approach to Substation Automation, Communications, and Integration.*

[22] Network protocol suite directory index, (en línea). Consultada el 12 feb. 2008. Disponible en http://www.javvin.com/protocolsuite.html

[23] Información acerca del proceso de encapsulación y modelo OSI. (en línea). Consultada 8 abr. 2008. Disponible en http://www.cisco.com/web/learning/netacad/

Apéndices

A.1 Descripción del Departamento de Ingeniería de Control y Automatización

El Proyecto de Graduación, se llevó a cabo en el Instituto Costarricense de Electricidad, sector energía, en las instalaciones ubicadas en Sabana Norte.

Acerca del departamento: El Centro de Servicio de Diseño (CSD), es un área de especialidad que cuenta con diseñadores e inspectores de líneas de transmisión y de Subestaciones, dentro del CSD se distinguen áreas, cada área tiene un Coordinador que se encarga de la distribución de las tareas en el personal según sus funciones para logra satisfacer las solicitudes de servicio existentes. Las diferentes áreas son:

- Ingeniería de Control y Automatización: Características asociadas a equipos de potencia.
- Área de Ingeniería Potencia y Plantas: Diseño electromecánicos.
- Área de Ingeniería Geotécnica: Información Geotécnica.
- Centro De Servicio de Construcción: Necesidades durante los procesos constructivos.

El proyecto se desarrolló en el área de Ingeniería de Control y Automatización, que como principal actividad desarrolla la ingeniería de control, que conlleva el diseño, construcción y puesta en operación de centrales de generación eléctrica y Subestaciones, tanto obras nuevas como ampliaciones o remodelaciones.
Dentro de sus actividades se pueden citar:

- Conceptualización del sistema de control según requerimientos de la central o subestación.
- Prepara las especificaciones técnicas de los equipos y cantidades de materiales a utilizar, así como la estimación del alcance del suministro de materiales.
- Realiza la publicación de carteles de licitación.
- Estudia y recomienda las posibles adjudicaciones.

- Da seguimiento a los contratos adjudicados.
- Realiza las revisiones de los planos de los fabricantes de equipos.
- Realiza las pruebas de los equipos.
- Recibe los equipos.
- Se encarga de cancelar económicamente a los suplidores de materiales y equipos.
- Realiza la ingeniería en detalle que determinará la interconexión de los diferentes sistemas de control y los de potencia
- Supervisa la instalación, pruebas, puesta en operación de los sistemas de control y potencia.
- Entrega planos actualizados tal y como se instalaron, junto con las especificaciones de instalación, calibración, mantenimiento a las áreas correspondientes.

A.2 Elección de un programa HMI

Para realizar pruebas y verificar el funcionamiento del *Módulo de transporte MMS sobre IP*, se seleccionó un programa HMI, con funciones que le permiten el control, supervisión y adquisición de datos.

Empresas dedicadas a la comercialización de herramientas para la automatización de subestaciones, brindan de manera gratuita, programas de muestra con características limitadas. Se hizo una evaluación de dos programas de muestra que podían ser utilizados como programa HMI. El primero de ellos, es el *BrowzerD* desarrollado por *TAMARACK Consulting*, y descargado del sitio web de *NettedAutomation* [14].

Según las pruebas realizadas, el programa cliente soporta las siguientes operaciones:

- Lectura (*Read*)
- Escritura (*Write)*
- Mostrar Atributos

- Mostrar Datos
- Muestreo de un servidor cada cierto periodo de tiempo
- Configuración de los parámetros de Red
- Elección de los servicios MMS a soportar

El segundo programa considerado, fue el *IEDScout V1.*5 y desarrollado por *OMICRON*, este programa fue descargado de la página web de la empresa [15]. Como características importantes se observó que soporta las siguientes operaciones:

- Lectura (*Read)*
- Mostrar Atributos
- Mostrar Datos
- Configuración de los parámetros de red
- Elección de los servicios MMS a soportar

Para realizar las pruebas que permitieron determinar qué operaciones soportan, los programas arriba mencionados, se utilizó un programa que simula un IED virtual, el nombre del programa es Cummings, y se descargó de la página de *NettedAutomation* [14].

Como procedimiento para realizar las pruebas, se ejecutó el programa cliente HMI y se hizo una conexión con el IED. En el caso del *BrowzerD* el programa trae una ventana llamada *Status Messages*, ahí se observan qué mensajes se intercambian durante el proceso de inicialización de la conexión, así como el estado de misma. El *IEDScout* tiene una ventana similar, llamada *Messages*, que también presenta qué paquetes se están intercambiando y si la conexión fue positiva. El programa que se eligió como HMI, fue el IEDScout V1.5.

Anexos

B.1 Abreviaturas

ASN.1 BER	(Abstract Syntax Notation One Basic Encoding Rules) Reglas Básicas de Codificación de la Notación de Sintaxis Abstracta versión 1.
CC-TPDU	(Conexion Confirm) TPDU de confirmación de conexión
COTP	(Connection Oriented Transport Protocol) Protocolo de Transporte Orientado a Conexión
DHCP	(*Dinamic Host Configuration Protocol)* Protocolo de configuración de host dinámico.
DT-TPDU	(Data TPDU) TPDU de datos.
IDE	(Integrated Development Environment) Ambiente Integrado de Desarrollo.
IEC	(International Electrotechnical Commission) Comisión Internacional Electrotécnica.
IED	(Inteligent Electronic Device) Dispositivo Electrónico Inteligente
ISO	(International Organization for Standardization) Organización Internacional para la Estandarización.
OSI	(Open System Interconnection) Modelo de referencia de Interconexión de Sistemas Abiertos.
MMS	(Manufacturing Message Specification) Especificación de Mensajes de Fabricantes.
SCADA	(Supervisory Control And Data Acquisition) Control supervisor y adquisición de datos.
SPDU	(Session Protocol Data Units) Unidades de datos del protocolo de sesión.
TPDU	(Transport Protocol Data Unit) Unidades de datos del protocolo de transporte.
TSDU	(Transport Service Data Unit) Unidad de datos del servicio del protocolo de transporte TCP.

B.2 Descripción del microprocesador rabbit 3000 y el módulo de desarrollo RCM 3300

El microcontrolador que se utilizó para el desarrollo del proyecto, fue el *Rabbit* 3000, el mismo se monta sobre un módulo de desarrollo RCM3300 *RabbitCore*.

El módulo RCM3300 se programa con una conexión serial estándar a una computadora, a través del cable de programación que viene con el módulo de desarrollo. En la figura siguiente se muestra como se debe instalar el microcontrolador en la tarjeta de desarrollo. También en esa figura se muestra donde están ubicados el LED y el Botón usados para simular las salidas digitales, el módulo de desarrollo corresponde al dispositivo físico Panel.

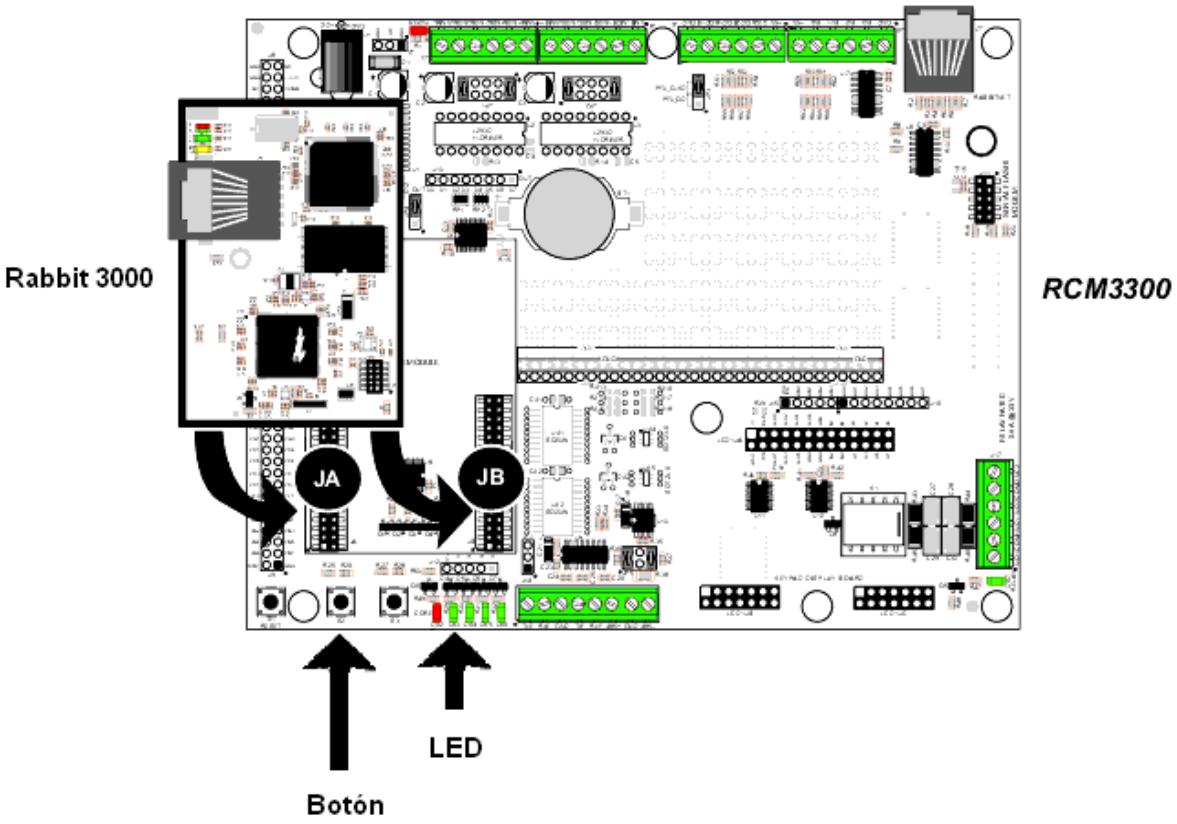


Figura B.2.1 Microcontrolador Rabbit 3000 y Módulo de desarrollo RCM 3300 [17]

Entre las principales características del hardware están:

Posee un puerto Ethernet integrado para conectividad a la red, posee librerías dedicadas al manejo de los protocolos TCP/IP, correspondientes a las capas de la 1 a la 4 del modelo OSI.

La conexión correcta de la alimentación y del cable de programación del módulo de desarrollo se muestra en siguiente figura.

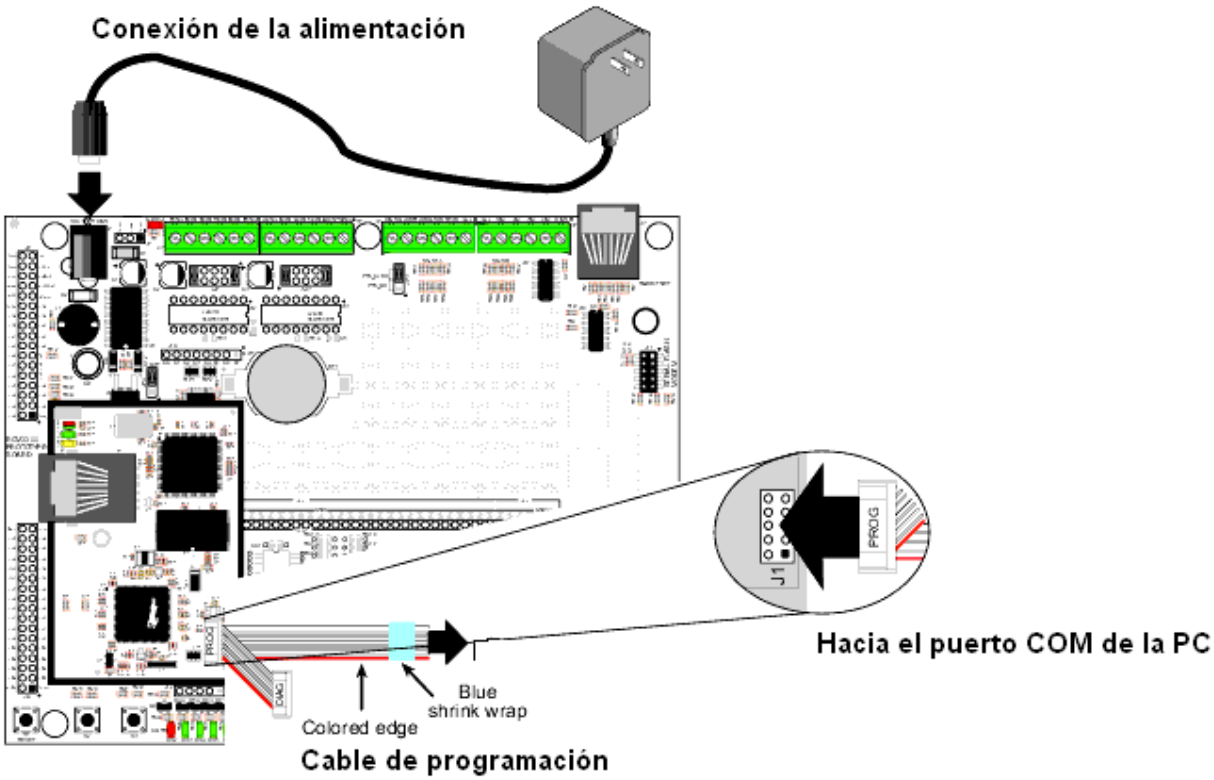


Figura B.2.2 Conexión del cable de alimentación y del cable de programación [17]

B.3 Información de la capa de sesión y presentación

A continuación se muestran los datos que deben ser agregados del lado del servidor durante la **inicialización**, los datos están separados de acuerdo a cada capa de Red.

Capa de Sesión

Servidor initiate response:

ISO 8327-1 OSI Session Protocol
SPDU Type: ACCEPT (AC) SPDU (14)
Length: 138
Connect Accept Item
    Parameter type: Connect Accept Item
    Parameter type: Connect Accept Item
    Protocol Options
        Parameter type: Protocol Options
        Parameter length: 1
        Flags: 0x00
            ....0 = Able to receive extended concatenated SPDU: False
    Version Number
        Parameter type: Version Number
        Parameter length: 1
        Flags: 0x02
            .... ..1. = Protocol Version 2: True
            ....0 = Protocol Version 1: False
Session Requirement
    Parameter type: Session Requirement
    Parameter length: 2
    Flags: 0x0002
    ..0. .... = Session exception report: False
    ...0 .... = Data separation function unit: False
    .... 0... .... = Symmetric synchronize function unit: False
    .... .0.. .... = Typed data function unit: False
    .... ..0. .... = Exception function unit: False
    ....0 .... = Capability function unit: False
    .... 0... .... = Negotiated release function unit: False
    .... .0.. .... = Activity management function unit: False
    ....0. .... = Resynchronize function unit: False
    ....0. .... = Resynchronize function unit: False
    .... 0... = Minor resynchronize function unit: False
    .... .0.. = Expedited data function unit: False
    ....1. = Duplex functional unit: True
    ....0 = Half-duplex functional unit: False
Called Session Selector
    Parameter type: Called Session Selector
    Parameter length: 2
    Called Session Selector: 0001
Session user data
    Parameter type: Session user data
    Parameter length: 12

Capa de Presentación
Servidor initiate response

ISO 8823 OSI Presentation Protocol
CPA-PPDU
    mode-selector
        mode-value: normal-mode (1)
    normal-mode-parameters
        responding-presentation-selector: 00000001
        presentation-context-definition-result-list: 2 items
        Item
            result: acceptance (0)
            transfer-syntax-name: 2.1.1 (joint-iso-itu-t(2) asn1(1) basic-encoding(1))
        Item
            result: acceptance (0)
            transfer-syntax-name: 2.1.1 (joint-iso-itu-t(2) asn1(1) basic-encoding(1))
    user-data: fully-encoded-data (1)
        fully-encoded-data: 1 item
            Item
                presentation-context-identifier: 1
                presentation-data-values: single-ASN1-type (0)

Capa de Aplicación

Servidor initiate response

ISO 8650-1 OSI Association Control Service
aare
    aso-context-name: 1.09506.2.3 (MMS)
        result: accepted (0)
        result-source-diagnostic: acse-service-user (1)
            acse-service-user: null(0)
        user-information:1 item
            Item
                direct-reference:2.1.1 (joint-iso-itu-t(2) asn1(2) basic-encoding(1))
                indirect-reference: 3
                encoding: single-ASN1-type (0)

Para cualquier otra operación después de la inicialización, la información a agregar en cada paquete debe ser la siguiente, también dividida por capa:

Capa de Sesión

ISO 8327-1 OSI Session Protocol
SPDU Type: Give tokens PDU (1)
Length: 0
ISO 8327-1 OSI Session Protocol
SPDU Type: DATA TRANSFER (DT) SPDU (1)
Length: 0

Capa de Presentación

ISO 8823 OSI Presentation Protocol
    user-data: fully-encoded-data (1)
        fully-encoded-data: 1 item
            Item
                presentation-data-values: single-ASN1-type (0)