

CAPÍTULO III

EL PROTOCOLO INDUSTRIAL COMÚN (CIP) Y LAS REDES DE CAMPO DEVICENET Y CONTROLNET

3.1 Introducción

El Protocolo Industrial Común, CIP (del inglés Common Industrial Protocol), desarrollado por las asociaciones ODVA y ControlNet International, mencionadas en el capítulo I, tiene como objetivo proveer una capa de aplicación, de acuerdo con el modelo OSI, común para diferentes redes. Actualmente, tres son las redes que lo han implementado, a saber, DeviceNet, ControlNet y EtherNet/IP. En el presente capítulo se estudia la relación que existe entre el protocolo CIP y las redes antes mencionadas con el objetivo de sentar las bases del estudio de las redes DeviceNet y ControlNet en particular.

3.2 El problema de la integración de las redes en la industria

De acuerdo con el modelo jerárquico CIM, se utilizan diferentes redes para los distintos niveles de automatización en la industria. Debido a las diferentes tecnologías de redes desarrolladas, cada una con su propio protocolo de comunicación particular, se producen aislamiento de sectores dentro del mismo nivel CIM y, más aún, se hace complicada una integración vertical, es decir, entre diferentes niveles jerárquicos de la empresa.

Por tanto, los requerimientos de la industria moderna consideran una integración total de la empresa, pudiéndose disponer de la información proveniente desde cualquier nivel jerárquico y en cualquier momento. Así, desde el computador de gerencia, sería posible monitorear la producción y e incluso los procesos de planta, de forma que se puedan tomar las decisiones más convenientes de acuerdo con los requerimientos de producción, los que a su vez se ven afectados por el mercado.

3.3 El Protocolo CIP

Diferentes empresas y organizaciones han presentado algunas soluciones para permitir esta integración total de la información en la empresa. Una de estas alternativas es la presentada por las asociaciones ODVA y ControlNet Internacional. Para ello han desarrollado el protocolo llamado *Common Industrial Protocol*, CIP, que se puede traducir al español como Protocolo Industrial Común.

El protocolo CIP define las características de las capas de aplicación para diferentes redes lo que ha dado lugar a llamar la “Familia de Redes CIP”. En la figura 3.1 se muestra la arquitectura de red, de acuerdo al modelo de referencia OSI, para las distintas redes que implementan CIP en sus capas superiores.

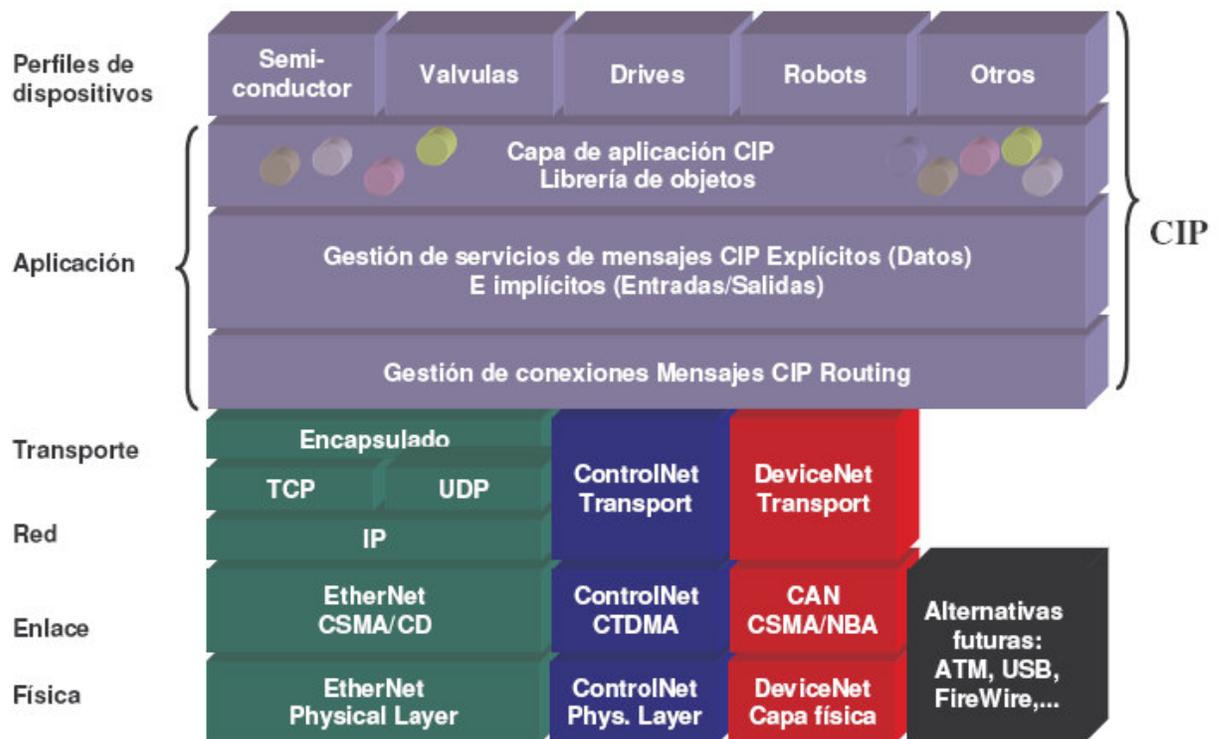


Figura 3.1. Familia de redes CIP.

Se puede observar en la figura 3.1 que tres redes implementan el protocolo CIP en sus capas de aplicación. Estas redes son:

- **DeviceNet**
- **ControlNet**
- **EtherNet/IP**

La *filosofía* del desarrollo del protocolo CIP, es la de proveer de una capa de aplicación que se abstraiga de la tecnología particular de una red industrial, es decir, de las capas más bajas que definen las características físicas y propias del protocolo de comunicación. De esta forma, se puede aplicar esta capa común a distintas tecnologías de redes, cada una con sus capacidades y aplicaciones particulares. Así también, es posible otorgar una capacidad de interconexión entre las redes que implementen CIP en sus capas superiores de una forma transparente.

Esta capacidad de interconexión también la tendrían futuras redes que implementen CIP en sus capas superiores. Esta situación también es ilustrada en la figura 3.1.

3.4 Arquitectura NetLinx

Como se vio en el capítulo I, las asociaciones ODVA y ControlNet Intenational proponen la solución comercial desarrollada por Rockwell-Automation, para la integración de información en la empresa, basada en las redes que implementan el protocolo CIP. Esta solución es llamada NetLinx, la cual divide la empresa en 3 niveles jerárquicos, según el tipo de elementos e información que transporta la red.

En la figura 3.2 se muestra la arquitectura NetLinx.

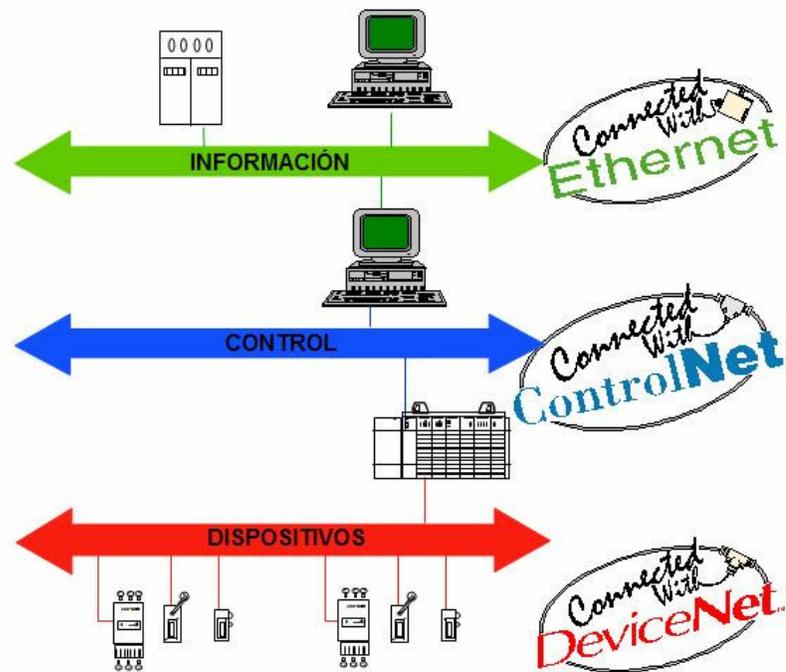


Figura 3.2. Arquitectura NetLinx como solución de integración de la empresa.

Como se desprende de la figura 3.2, la arquitectura NetLinx la componen las redes EtherNet/IP, ControlNet y DeviceNet:

Red EtherNet/IP: Esta red es una adaptación del estándar IEEE 802.3 para su aplicación como red industrial. En el nombre, la sección “IP” significa “Industrial Protocol”, o Protocolo Industrial. Como se observa en la figura 3.1, posee las mismas bases que el EtherNet tradicional (mismas capas física, de enlace, de red y de transporte). Por tanto, dentro de la familia de redes CIP, esta se utiliza como una red del nivel de información, para transportar grandes volúmenes de información.

La red ControlNet: La red ControlNet, dentro de la arquitectura NetLinx, pertenece al nivel de control. Ésta es estudiada en detalle en la parte 4.

La red DeviceNet: La red DeviceNet, dentro de la arquitectura NetLinx, pertenece al nivel de dispositivo. Ésta es estudiada en detalle en la parte 3.

3.5 Organización de las Especificaciones de las Redes CIP

Las especificaciones para cada red, así como para la capa común, son entregadas por ODVA y ControlNet International en la forma de Volúmenes. Así, cada red, se compone de dos volúmenes: un volumen común (capas de aplicación), y un volumen específico (capas inferiores).

A continuación se entrega la organización de las especificaciones CIP:

- La especificación EtherNet/IP consiste en:
 - Volumen 1: Protocolo Industrial Común
 - Volumen 2: Adaptación EtherNet/IP a CIP

- La especificación DeviceNet consiste en:
 - Volumen 1: Protocolo Industrial Común
 - Volumen 3: Adaptación Devicenet a CIP

- La especificación ControlNet consiste en:
 - Volumen 1: Protocolo Industrial Común
 - Volumen 4: Adaptación CIP a ControlNet

- Las especificaciones de seguridad CIP:
 - Volumen 5: Seguridad CIP

Actualmente están en desarrollo, las especificaciones para una red CIP, denominada CompoNet, la cual se compondrá de los siguientes volúmenes:

- Volumen 1: Protocolo Industrial Común
- Volumen 6: Adaptación CompoNet a CIP

El **volumen 1** detalla los aspectos comunes de CIP que se aplican a todas las adaptaciones de la red.

El **volumen 2** es la adaptación EtherNet/IP de CIP, que describe cómo CIP es adaptado a las capas de transporte Ethernet TCP/IP y de UDP/IP. También contiene cualquier extensión necesaria para el Volumen 1 que sea necesaria para EtherNet/IP.

El **volumen 3** es la adaptación DeviceNet de CIP, que describe cómo CIP es adaptado a la capa de enlace de datos CAN. También contiene cualquier extensión necesaria para el Volumen 1 que sea necesaria para DeviceNet.

El **volumen 4** es la adaptación ControlNet de CIP, que describe cómo CIP es adaptado a la capa de enlace de datos ControlNet. También contiene cualquier extensión necesaria para el Volumen 1 que sea necesaria para ControlNet.

El **volumen 5** es seguridad de CIP. Contiene la información necesaria para implementar el protocolo de seguridad CIP en las redes Redes CIP.

3.5.1 Estandarización Internacional

Algunos de estos Volúmenes actualmente están estandarizados internacionalmente a través del Comité Electrotécnico Internacional, o IEC (por sus siglas en inglés). De esta manera, en el contexto de los estándares internacionales los volúmenes se corresponden de la manera que se muestra en la tabla 3.1:

Volumen CIP	Estándar IEC
Volumen 1	IEC-61158 type 2, del año 2000
Volumen 4	IEC-61158 type 2, del año 2000
Volumen 3	IEC-62026-3, del año 2000

Tabla 3.1. Estandarizaciones de CIP.

Se puede observar que el Volumen 1 y 4 poseen la misma estandarización. Esto se debe a que la red ControlNet se estandarizó completamente, de manera que toma los dos volúmenes que lo componen y da lugar a la estandarización IEC.

En los capítulos V y VIII se entrega más información acerca de los estándares internacionales y las redes DeviceNet y ControlNet respectivamente.

3.5.2 Mantenimiento y desarrollo adicional de las especificaciones

Las asociaciones ODVA y ControlNet internacional tienen un sistema de grupos de trabajo que para el mantenimiento de las especificaciones y ampliaciones del protocolo, como por ejemplo, el desarrollo de nuevos objetos o nuevos perfiles de dispositivo. Se dividen en dos grupos que son:

- grupo de interés especial (SIGs) para DeviceNet,
- grupo de interés especial (SIGs) para ControlNet, y
- grupo de interés especial común (JSIGs) para EtherNet/IP (puesto que la tecnología EtherNet/IP es administrada en común por ODVA y los miembros internacionales de ControlNet).

Los resultados de estos SIGs y JSIGs se escriben como DSEs (DeviceNet Specification Enhancements), CSEs (ControlNet Specification Enhancements), ESEs (EtherNet/IP Specification Enhancements), SSEs (Safety Specification Enhancements) o CIPSEs (CIP Specification Enhancements), los cuales son presentados al Comité Examinador Técnico (TRB) para la aprobación y posterior incorporación a las especificaciones.

En la figura 3.3 se muestra el proceso de mejoramiento y ampliación de las especificaciones CIP.

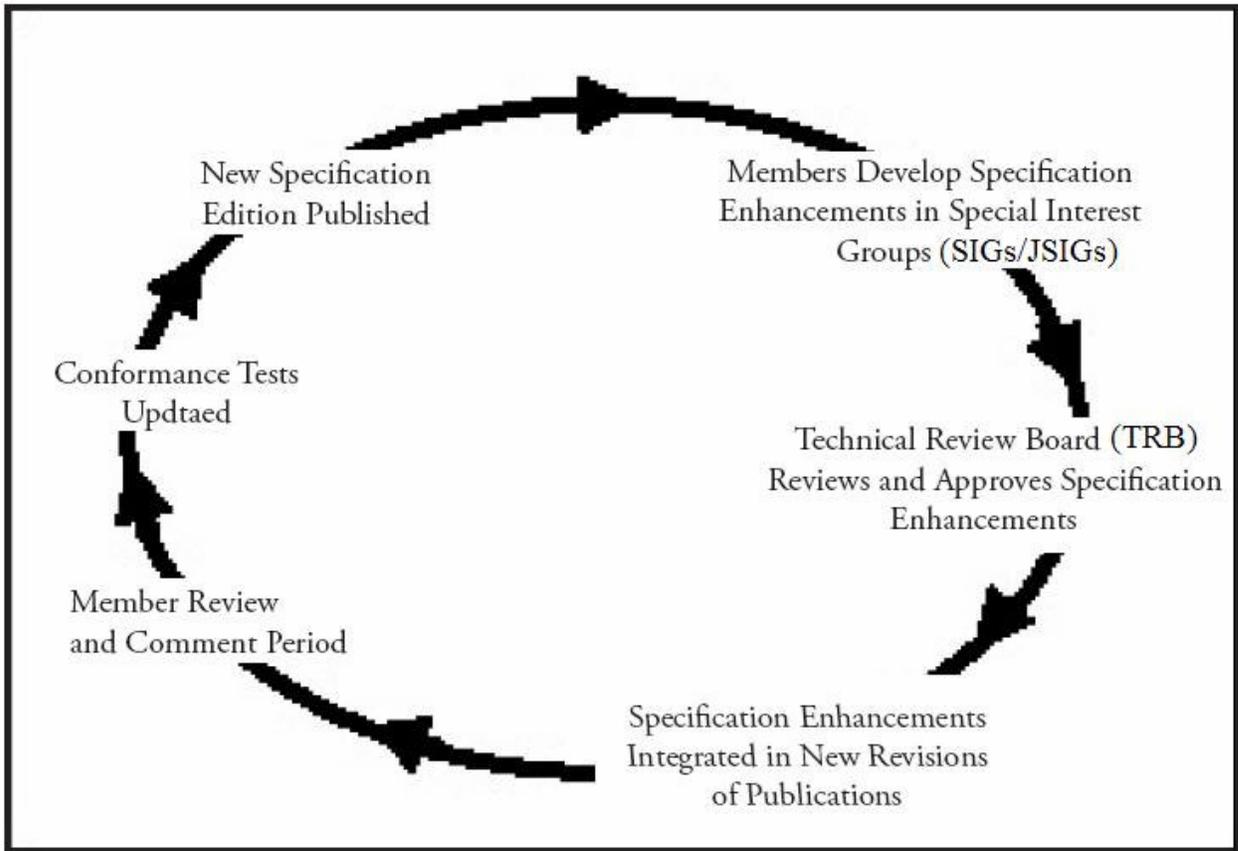


Figura 3.3. Proceso de mejoramiento y ampliación de las especificaciones CIP.

Solamente los miembros de ODVA o de ControlNet pueden trabajar dentro de los SIGs y del JSIGs. Los participantes tienen como ventaja un conocimiento anticipado de cambios técnicos.

CAPÍTULO IV

DESCRIPCIÓN DEL PROTOCOLO INDUSTRIAL COMÚN CIP

4.1 Introducción

Las redes de DeviceNet y ControlNet, dentro de su arquitectura de red, definen sus capas de aplicación a través del único protocolo CIP. Por lo tanto, a ese nivel de la arquitectura de red, comparten todas las características que se definen en CIP Volumen 1. La base del protocolo CIP es el *modelado de objetos abstractos* que le permite una funcionalidad independientemente de la tecnología de la red. En el presente capítulo se hace una descripción del Protocolo Industrial Común, CIP, como definición de las capas superiores para las redes DeviceNet y ControlNet.

4.2 Términos generales utilizados por CIP

En el estudio del protocolo industrial común CIP, es necesario esclarecer algunos términos típicos en el ámbito de los sistemas de control, pero dentro del contexto de CIP. A continuación se entregan las definiciones para algunos términos.

Cliente/Servidor: Es una arquitectura de comunicación, en donde se establece la relación que existe entre los procesos de dispositivos distintos. El *cliente* es el dispositivo que *envía* una solicitud a un servidor. El cliente espera una respuesta del servidor. Por otra parte, el *servidor* es el dispositivo que *recibe* una solicitud de un cliente. Se espera que el servidor dé una respuesta al cliente.

Master/Slave (Maestro/Esclavo): Es una aplicación particular de la arquitectura de comunicación cliente/servidor en que el *Master*, es decir, el cliente, *requiere y envía información* hacia un Slave, es decir, un servidor. Por otra parte, el *Slave*, *se limita a enviar la información solicitada y a efectuar las acciones de control* dadas por el Master.

Modelo Fuente/Destino: Es un modelo de comunicación que establece la forma en que son compartidos los mensajes. En este modelo, el dispositivo **fuelle** envía un mensaje a un solo dispositivo destinatario específico. Por otra parte, el dispositivo destino es el que recibe el mensaje procedente desde el dispositivo fuente. Por esta razón, se incluye en la cabecera de un mensaje, la dirección de red del dispositivo fuente y la dirección de red del dispositivo destino. Esta tipo de comunicación es punto a punto.

Modelo Productor/Consumidor: Es un modelo de comunicación que establece la forma en que son compartidos los mensajes. En este modelo, el dispositivo **productor** coloca un mensaje sobre la red para el consumo por uno o varios consumidores. Generalmente, el mensaje producido no se dirige a un consumidor específico. Por otra parte, el **consumidor** es un dispositivo que recoge (*consume*) un mensaje puesto en la red por un dispositivo productor. El consumidor determina qué mensaje consumir por medio de un identificador en la cabecera del mensaje. Este modelo de comunicación es inherentemente *multicast* y puede soportar además comunicación punto a punto (por lo tanto, puede decirse que incorpora al modelo fuente/destino). En la figura 4.1 se comparan los mensaje de los modelos fuente/destino y de productor/consumidor.

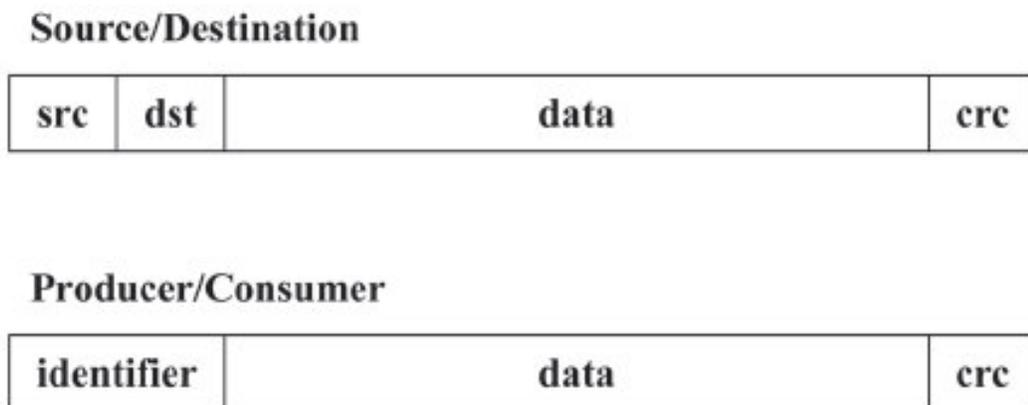


Figura 4.1. Modelos de comunicación Fuente/Destino y Productor/consumidor.

Modelo productor/consumidor para CIP: El protocolo CIP utiliza el modelo de comunicación productor/consumidor, en contraste con el modelo tradicional de fuente/destino. Aprovecha su naturaleza *multicast*. Los nodos sobre la red determinan si se deben consumir los datos de un mensaje basándose en un valor de identificación (*identifier*) de la conexión, el cual está incluido en el paquete.

Mensaje explícito: Los mensajes explícitos contienen la información de dirección y de servicio que lleva al dispositivo receptor a realizar cierto servicio (*una acción*) en una parte específica de un dispositivo (por ejemplo, un atributo).

Mensaje I/O o implícito: Los mensajes I/O o implícitos no llevan la información de dirección y/o de servicio; el nodo consumidor sabe qué hacer con los datos basados en la Conexión ID que fue asignada cuando la conexión fue establecida. Los mensajes implícitos son nombrados así porque el significado de los datos está implícito por la conexión ID.

Modelo de Objetos: El protocolo CIP utiliza el modelo de objetos para la implementación de las capas de aplicación de las distintas redes de campo. El modelo de objetos se basa en representaciones abstractas de las características de los dispositivos conectados a determinada red. Dado que estos objetos se ubican en las capas de aplicación de acuerdo al modelo ISO/OSI, un objeto es creado a través del uso de algún lenguaje de programación apropiado (por ejemplo, C++) siguiendo las pautas de estructura y sintaxis entregadas por las especificaciones de CIP.

Punto a punto: es el envío de información entre un dispositivo emisor (por ejemplo, un Master que envía un mensaje de solicitud de información) hacia un solo dispositivo receptor (por ejemplo, un Slave que recibe el mensaje y luego reenvía la información solicitada).

Multicast: es el envío de información desde un dispositivo emisor (por ejemplo, un cliente que envía un mensaje de solicitud de información) hacia varios dispositivos receptores determinados (por ejemplo, varios dispositivos servidores).

4.3 Modelado de Objetos

El protocolo CIP utiliza un modelo de objetos abstractos para describir:

- El conjunto de servicios de comunicación disponibles.
- El comportamiento visible externamente de un nodo CIP.
- Un medio común para el acceso e intercambio de información entre productos CIP.

Todo nodo CIP es modelado como una *colección de objetos*. Un objeto provee una representación abstracta de un componente particular dentro de un producto. Cualquier cosa no descrita en forma de objeto no es visible a través de CIP.

Los objetos CIP están estructurados en los siguientes elementos:

- **Clases**
- **Instancias**
- **Atributos**

Una **clase** es un conjunto de objetos donde todos representan el mismo tipo de componente de sistema. Un objeto **instancia** es la representación real de un objeto particular dentro de una clase. A su vez, cada instancia de una clase posee los mismos **atributos**, y que a su vez poseen su propio conjunto particular de valores.

En la figura 4.2 se representa el modelado de objetos de un nodo CIP.

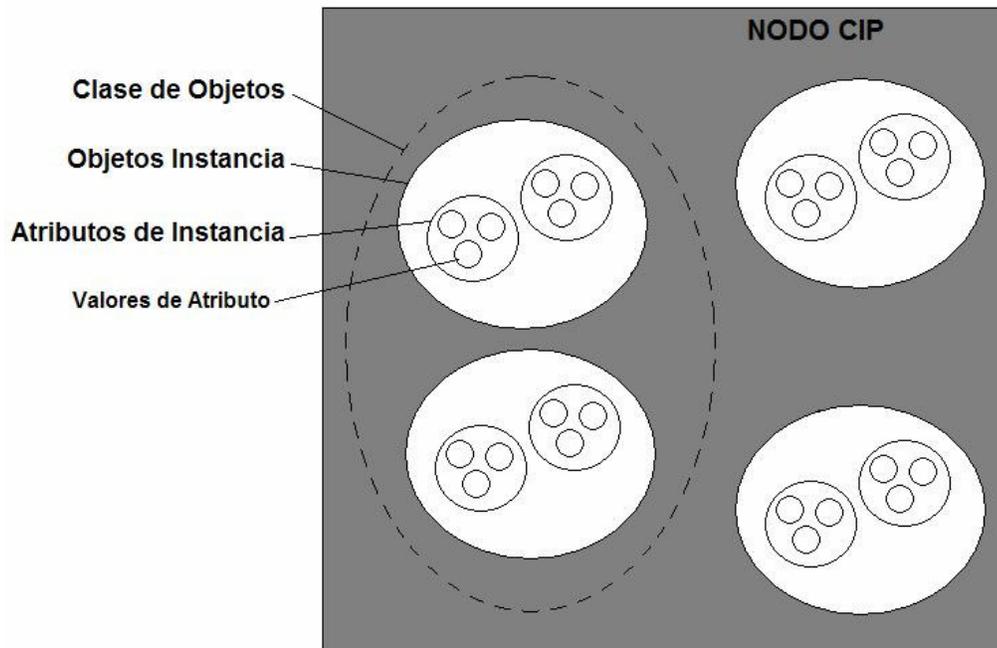


Figura 4.2. Nodo CIP representado según el modelo de objetos.

Tal como se muestra en la figura 4.2 múltiples objetos instancia dentro de una clase particular pueden residir dentro de un nodo CIP.

Los objetos y sus componentes son direccionados a través de un esquema uniforme compuesto de los siguientes elementos:

- **Node Address** (Dirección de Nodo)
- **Class Identifier o Class ID** (Identificador de Clase)
- **Instance Identifier o Instance ID** (Identificador de Instancia)
- **Attribute Identifier o Attribute ID** (Identificador de Atributo)
- **Service Code** (Código de Servicio)

Node Address (o Node ID): Dirección de Nodo. Es un valor entero de identificación asignado a cada nodo sobre una red CIP. En DeviceNet y ControlNet, también es llamado MAC-ID y es el número de nodo del dispositivo dentro de la red. En EtherNet/IP, la dirección de nodo es la dirección IP.

Class ID: Identificador de clase. Es un valor entero de identificación asignado a cada clase de objetos accesible desde la red.

Instance ID: Identificador de Instancia. Es un valor entero de identificación asignada a un objeto instancia que lo identifica entre todas las instancias de la misma clase.

Attribute ID: Identificador de Atributo. Es un valor entero de identificación asignado a una clase o atributo.

Service Code: Código de servicio. Es un valor entero de identificación que denota una solicitud de acción que puede estar dirigido a un objeto instancia particular o objeto clase.

Los valores de *Class ID* se dividen en dos partes para diferenciar dos tipos de objetos, que son los de definición pública, realizados por ODVA y ControlNet International, y los de definición específica de fabricante. En la tabla 4.1 se muestra los rangos de números que identifican a qué tipo pertenece un objeto. El resto de los valores identificadores están reservados por CIP.

Class ID	Rango de números
Definición Pública (ODVA y CI)	0x00 – 0x63 0x00F0 – 0x02FF
Definición Específica de Fabricante	0x64 – 0xC7 0x0300 – 0x04FF

Tabla 4.1. Rango de Class ID para los objetos CIP.

Lo mismo ocurre para los Instance ID y los Attribute ID. En la tabla 4.2 se muestran los rangos de Instance ID para un objeto particular, el Assembly Object. Mientras que en la tabla 4.3 se muestran los rangos para los Attribute ID.

Assembly Object – Class ID = 0x04 (objeto de definición pública)	
Instance ID	Rango de números
Definición Pública (ODVA y CI)	0x01 – 0x63 0x0100 – 0x02FF
Definición Específica de Fabricante	0x64 – 0xC7 0x0300 – 0x04FF

Tabla 4.2. Rango de Instance ID del Assembly Object (Class ID = 0x04).

Attribute ID	Rango de números
Definición Pública (ODVA y CI)	0x00 – 0x63
Definición Específica de Fabricante	0x64 – 0xC7

Tabla 4.3. Rango de Attribute ID para las instancias CIP.

Tanto los objetos como los atributos específicos de fabricante deben seguir las pautas dadas por la especificación CIP.

En la figura 4.3 se ilustra el esquema de direccionamiento definido por CIP.

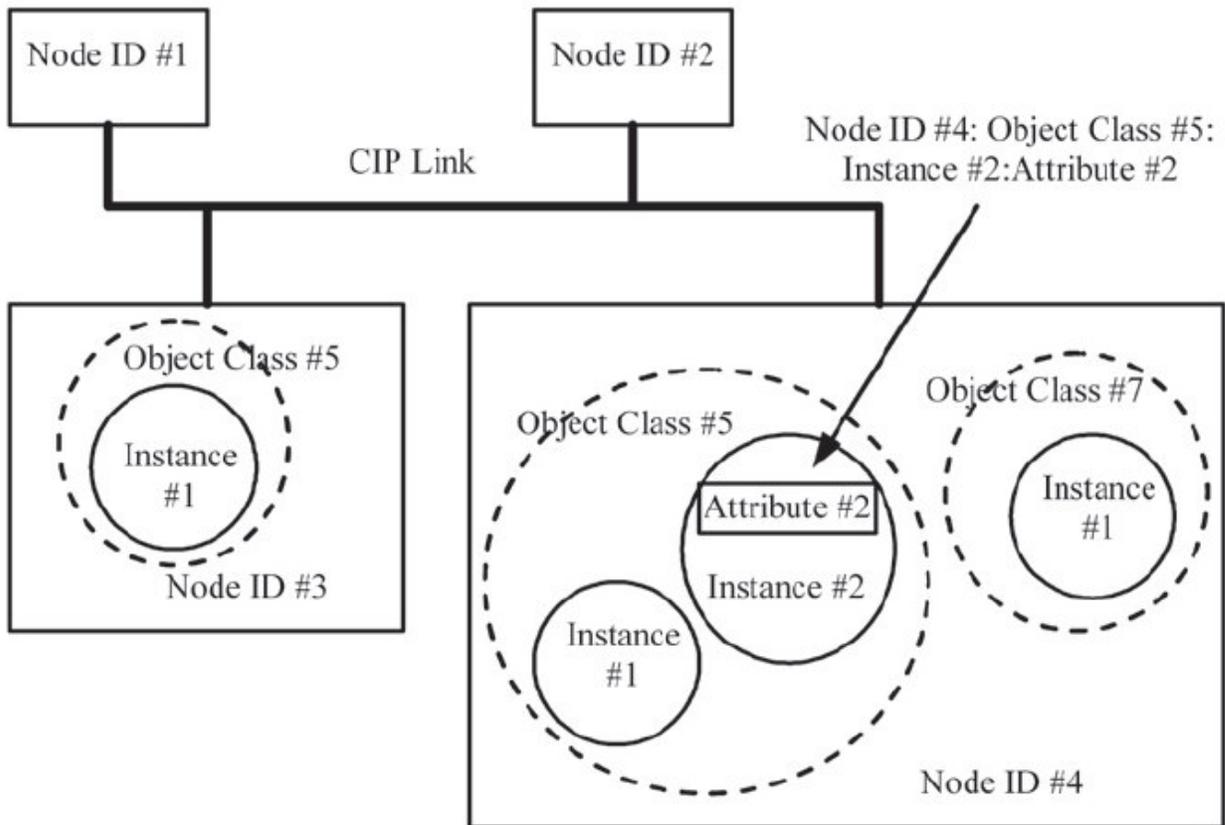


Figura 4.3. Esquema de direccionamiento CIP.

Se puede observar en la figura 4.3 una red CIP (por ejemplo, una red DeviceNet). La red posee cuatro nodos. Poniendo atención en el nodo #4, se puede observar que este posee 3 objetos instancia que se dividen en 2 clases de objetos. Siguiendo el esquema de direccionamiento definido por CIP, se puede observar en la figura 4.4 la siguiente secuencia en el nodo #4:

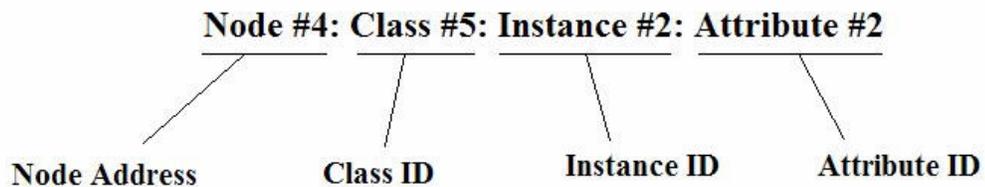


Figura 4.4. Secuencia de direccionamiento de objetos en el nodo #4 de la figura 4.3.

De esta forma, es inequívoca la aplicación requerida. Todos los nodos CIP deben ser direccionados siguiendo este esquema de direccionamiento.

4.4 Servicios

Los códigos de servicio son utilizados para definir la acción solicitada cuando un objeto es direccionado a través de mensajería explícita usando el esquema de direccionamiento descrito en la sección anterior.

Aparte de las funciones de lectura y escritura, se ha definido un conjunto de servicios CIP. Estos servicios CIP son de naturaleza común, lo que significa que se pueden utilizar en todas las redes CIP y son útiles para una variedad de objetos. También, existen códigos de servicio específicos de objetos que pueden tener diferentes significados para el mismo código, dependiendo de la clase a la cual pertenece el objeto.

Además de los códigos de servicio generales y de los códigos específicos de objetos definidos por CIP, los desarrolladores de dispositivos pueden definir otros servicios específicos para sus productos. Aunque esta característica otorga mucha flexibilidad para los fabricantes, tiene como desventaja que los servicios específicos de fabricante no pueden ser entendidos universalmente. Los fabricantes deben entregar, al menos, una descripción pública de estos servicios particulares a sus clientes que necesiten acceder a esta información.

4.5 Protocolo de mensajería

El Protocolo Industrial Común, CIP, es basado en conexión. Esto significa que primero debe ser establecida una conexión entre los nodos antes de comenzar a transmitir (ver secciones 6.4 y 9.4 acerca de las capas de red y transporte DeviceNet y ControlNet respectivamente). Una conexión CIP proporciona una trayectoria entre múltiples objetos de aplicación. Cuando se establece una conexión, las transmisiones asociadas a esa conexión se asignan a una Conexión ID o CID. Si la conexión implica un intercambio bidireccional, entonces son asignados dos valores CID. En la figura 4.5 se representa el esquema de conexión CID.

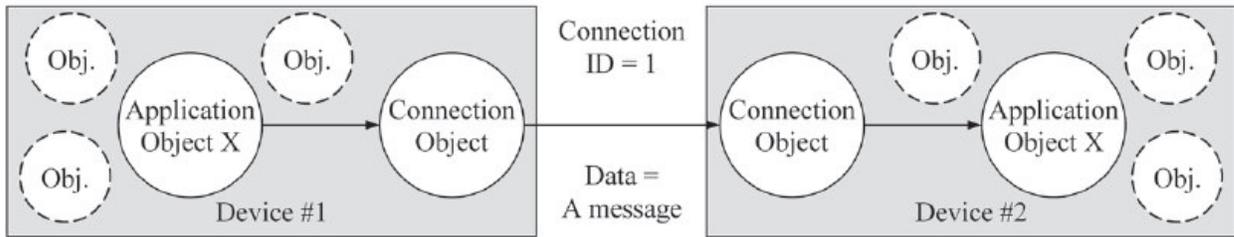


Figura 4.5. Conexiones y Conexión ID CIP.

La definición y el formato de la Conexión ID (o CID) dependen de la red. Por ejemplo, la Conexión ID para conexiones CIP sobre DeviceNet se basa en el *Identifier* (o Identificador) definido en el protocolo CAN, que es el que implementa DeviceNet para su capa de enlace (ver sección 6.3.2), mientras que en la red ControlNet, la Conexión ID está basada en el tipo de paquete de enlace, *Link Packet*, definido por ControlNet para su capa de enlace (ver sección 9.3.2).

Puesto que el intercambio de información en una Red CIP se hace a través de conexiones, se ha definido un proceso para establecerlas cuando los dispositivos aún están desconectados. Esto se hace a través de la función *Unconnected Message Manager (UCMM)*, que puede traducirse al español como el Gestor de Mensajería No Conectada, que es responsable del procesamiento de solicitudes y respuestas explícitas no conectadas.

Para las redes ControlNet y EtherNet/IP, el establecimiento de una conexión CIP se realiza enviando un mensaje de solicitud de servicio para iniciar el proceso de comunicación denominado *UCMM Forward_Open service request message*. La red DeviceNet en cambio utiliza un método simplificado que se describe en las secciones 6.4.3 y 6.4.4.

Una solicitud UCMM Forward_Open contiene toda la información requerida para crear una conexión entre el dispositivo originador (originator) y el dispositivo objetivo (target) y, si es requerido, una segunda conexión entre el dispositivo objetivo y el dispositivo originador.

Particularmente, la solicitud Forward_Open contiene la información sobre lo siguiente:

- Información de time-out para esta conexión
- Red CID para la conexión desde el originador al objetivo.
- Red CID para la conexión desde el objetivo al originador.
- Información acerca de la identidad del originador (identificación del fabricante, y número de serie)
- Tamaños de dato máximos de los mensajes en esa conexión;
- Mecanismos de disparo (*trigger*), por ejemplo, cíclico o cambio del estado (COS);
- Trayectoria de conexión para el dato del objeto de aplicación en el nodo.

La trayectoria de conexión también puede contener un segmento de ruteo que permita a las conexiones existir a través de múltiples redes CIP. La solicitud UCMM Forward_Open puede contener, además, una clave electrónica del dispositivo objetivo (identificación del vendedor, tipo de dispositivo, código de producto y revisión), así como la información de la configuración que será remitida al ensamblaje de configuración.

4.5.1 Tipos de Conexiones de Mensajería CIP

Las conexiones CIP se dividen en las siguientes dos categorías:

- Conexiones de Mensajería I/O o Implícita, y
- Conexiones de Mensajería Explícita.

a) Conexiones de Mensajería I/O o Implícita

Las Conexiones de Mensajería I/O proporcionan trayectorias de comunicación dedicadas de propósito especial entre una aplicación productor y una o más aplicaciones consumidor. Los datos I/O específicos de aplicación se mueven a través de estos puertos, proceso llamado frecuentemente de Mensajería Implícita (debido a que se sobreentiende que son datos I/O). Estos mensajes son típicamente *multicast*.

b) Conexiones de Mensajería Explícita

Las Conexiones de Mensajería Explícitas proporcionan trayectorias de comunicación genéricas, multipropósito entre dos dispositivos. Estas conexiones se refieren a menudo a simples conexiones de mensajería. Los mensajes explícitos proporcionan la tradicional relación solicitud/respuesta orientada a las redes de comunicación. Este tipo de mensajes son punto a punto.

4.6 Objetos de comunicación

Los objetos de comunicación CIP gestionan y entregan el *run-time*, o tiempo de ejecución, para el intercambio de mensajes. Aunque los objetos de comunicación siguen los principios y pautas generales de todos los objetos CIP, éstos poseen la característica única de que son los puntos focales de toda comunicación CIP.

Cada objeto de comunicación contiene una parte de enlace productor, una parte de enlace consumidor o ambos. Las conexiones I/O pueden ser productoras, consumidoras o productoras y consumidoras. Por otra parte, las conexiones de mensajería explícitas son siempre productoras y consumidoras. En las figuras 4.6 y 4.7 se representan las conexiones de Mensajería I/O y de Mensajería Explícita respectivamente.

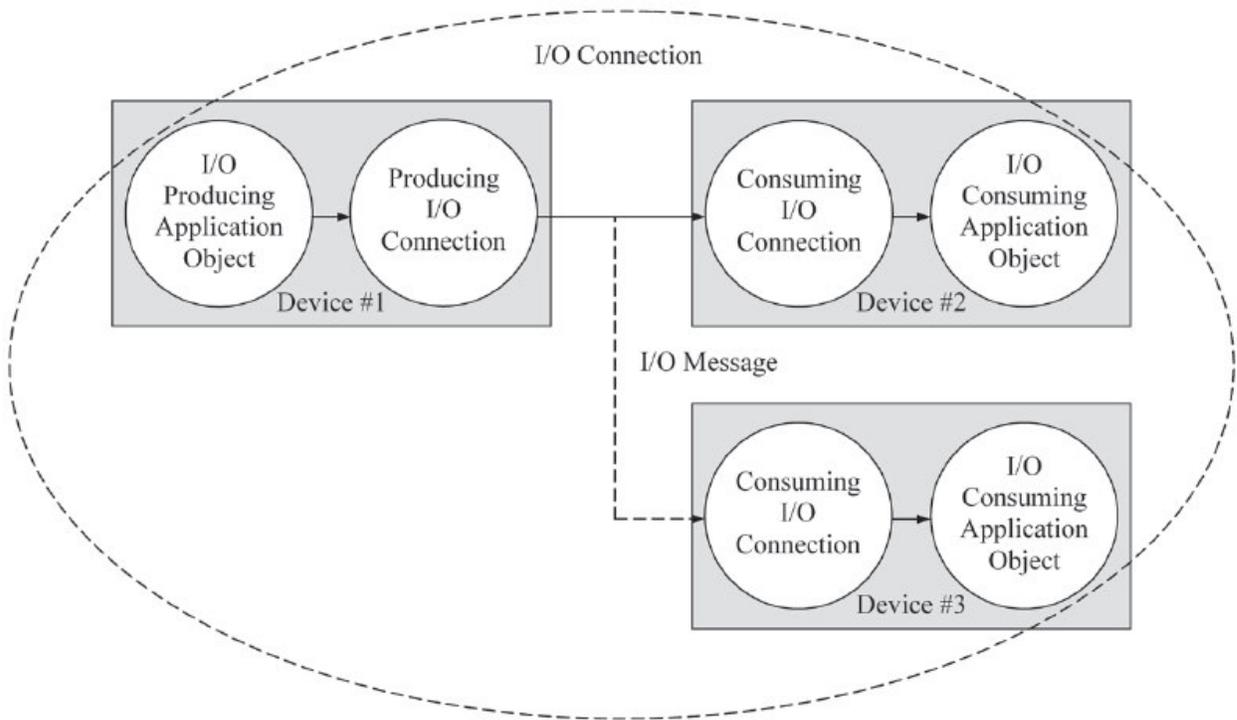


Figura 4.6. Conexión I/O CIP para Mensajería I/O multicast.

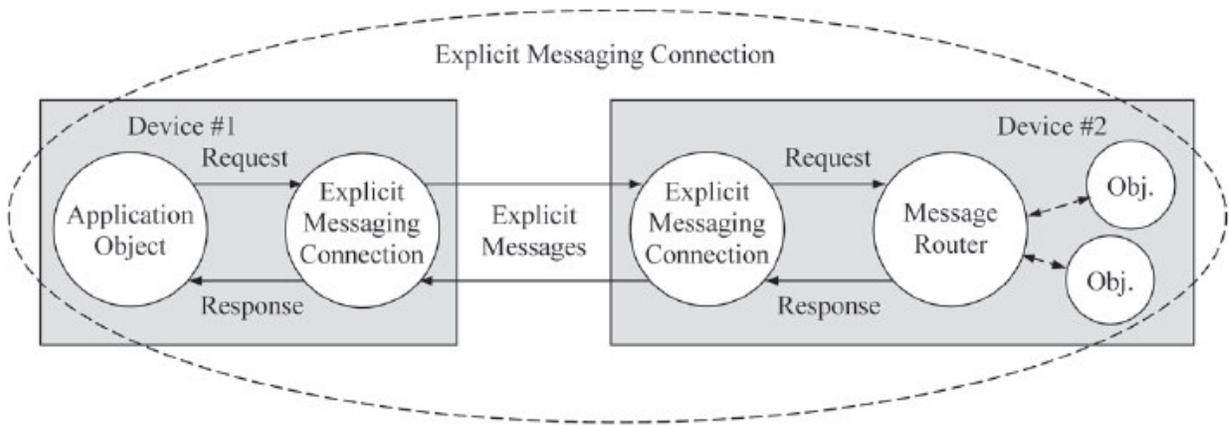


Figura 4.7. Conexión CIP para Mensajería Explícita punto a punto.

Los valores de atributo en los objetos de conexión definen un conjunto de atributos que describen parámetros vitales de esta conexión. Observar que los mensajes explícitos están siempre dirigidos hacia el objeto Router-Message (o Ruteador de Mensaje).

Los valores de atributo de un objeto de conexión especifican si es una Conexión I/O o una Conexión de Mensajería Explícita, el tamaño máximo de los datos que se intercambiarán a través de esta conexión y la fuente y destino de los datos. Atributos adicionales definen el estado y el comportamiento de la conexión. Comportamientos particularmente importantes incluyen cómo son accionados los mensajes (desde la aplicación, a través de cambio de estado o cambio de dato, con eventos cíclicos o por medio de eventos de red) y la sincronización de las conexiones (el *time-out*, o tiempo-fuera, asociado con ésta conexión y la acción predefinida si ocurre el *time-out*). El protocolo CIP permite que conexiones múltiples coexistan en un dispositivo, aunque en DeviceNet, por ejemplo, típicamente se tienen una o dos conexiones activas.

4.7 Librería de Objetos de las redes CIP

La Familia de Protocolos CIP, es decir, el conjunto de redes que implementan el protocolo CIP, posee una amplia colección de objetos definidos comunes. El conjunto total de clases de objeto se puede subdividir en tres tipos:

- **General-use** (o de uso general)
- **Application-specific** (o específico de aplicación)
- **Network-specific** (o específico de red)

Los objetos definidos en el volumen 1 de la Librería de Redes CIP están disponibles para utilizarse en todas las adaptaciones de la red CIP. Algunos de estos objetos pueden requerir cambios o limitaciones específicas cuando están implementadas en algunas de las adaptaciones de red. Estas excepciones se observan en el volumen de especificaciones correspondiente a la red. Por lo tanto, para ver el cuadro completo de implementación de una red en particular, se debe considerar el volumen 1 y el volumen particular de la red (por ejemplo, DeviceNet se compone del Volumen 1 y del Volumen 3 de CIP).

En la tabla 4.4 se entregan los objetos definidos por CIP para uso general.

Objetos de uso general CIP	
Assembly	Message Router
Acknowledge Handler	Parameter
Connection	Parameter Group
Connection Configuration	Port
Connection Manager	Register
File	Selection
Identity	

Tabla 4.4. Objetos CIP de uso general.

En la tabla 4.5 se entregan los objetos definidos por CIP específicos de aplicación.

Objetos específicos de aplicación	
AC/DC Drive	Overload
Analog Group	Position Controller
Analog Input Group	Position Controller Supervisor
Analog Output Group	Position Sensor
Analog Input Point	Presence Sensing
Analog Output Point	S-Analog Actor
Block Sequencer	S-Analog Sensor
Command Block	S-Device Supervisor
Control Supervisor	S-Gas Calibration
Discrete Group	S-Partial Pressure
Discrete Input Group	S-Single Stage Controller
Discrete Output Group	Safety Supervisor
Discrete Input Point	Safety Validator
Discrete Output Point	Softstart Starter
Group	Trip Point
Motor Data	

Tabla 4.5. Objetos CIP de uso específico.

En la tabla 4.6 se enlistan los objetos específicos de red.

Objetos específicos de red
ControlNet
ControlNet Keeper
ControlNet Scheduling
DeviceNet
Ethernet Link
TCP/IP Interface

Tabla 4.6. Objetos específicos de red CIP.

Los objetos de uso general pueden ser encontrados en muchos dispositivos diferentes, mientras que los objetos específicos de aplicación típicamente se encuentran sólo en los dispositivos diseñados para tal aplicación.

4.7.1 Conjunto mínimo de objetos requeridos en un dispositivo

Aunque son varios los tipos de objetos definidos para las redes CIP, un dispositivo típico sólo implementa un subconjunto de todos ellos. De modo que para que un dispositivo puede operar en una red CIP debe implementar como mínimo determinados objetos. En la figura 4.8 se representa un dispositivo típico de acuerdo con el modelo abstracto de objeto.

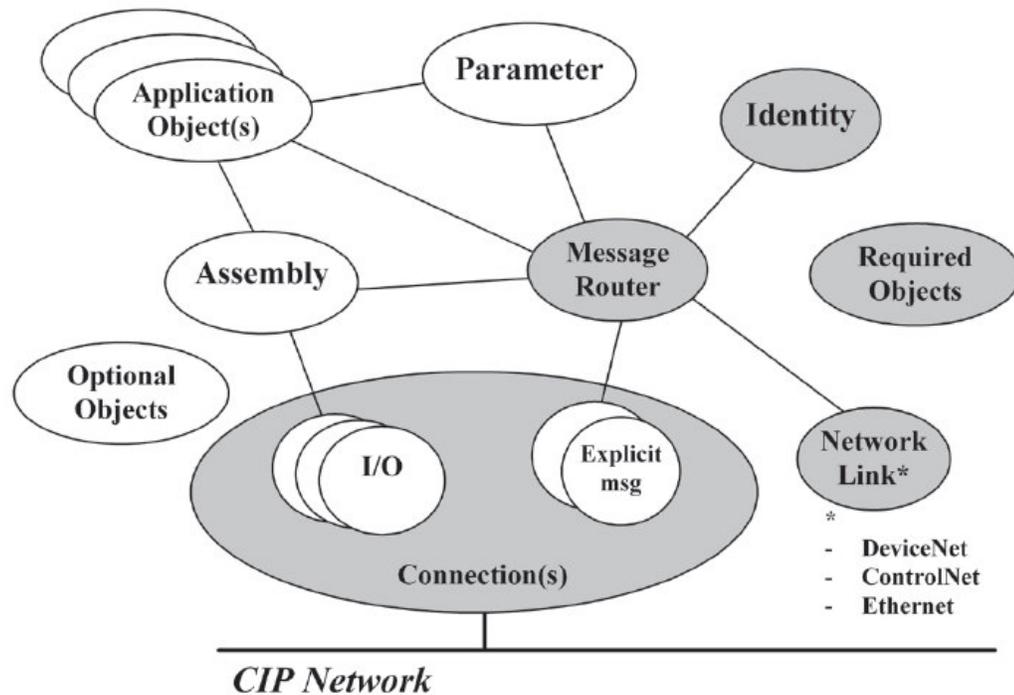


Figura 4.8. Dispositivo típico representado según el modelo de objetos CIP.

Los objetos requeridos para un típico dispositivo CIP (ya sea DeviceNet, ControlNet o EtherNet/IP) son:

- Un **Connection-Objeto** (u objeto de conexión), o bien un **Connection Manager Object** (u objeto gestor de conexión). Estos objetos pertenecen al grupo de objetos de uso general y cumplen las funciones de objetos de comunicación.
- Un **Identity-Object**, u objeto identidad. Éste objeto pertenece al grupo de objetos de uso general.
- Un **Message Router Object**, u objeto ruteador de mensaje (o al menos implementar esta función). Este objeto pertenece al grupo de objetos de uso general.
- Uno o más **objetos específicos de red**. Estos objetos no son del grupo de uso general, sino que dependen de la red.

Este subconjunto de objetos es requerido en todo dispositivo para que éste sea funcional. Es claro que el dispositivo debe incorporar, además, el objeto de aplicación específica según el tipo de dispositivo que sea. Estos objetos requeridos en todo dispositivo típico aparecen sombreados en la representación de un dispositivo con el modelo de objetos que se entrega en la figura 4.8.

Por otra parte, pueden ser agregados a los dispositivos otros objetos desarrollados por los fabricantes de dispositivos, en el contexto de objetos específicos de fabricante, para aumentar las funcionalidades de su producto. Por ello, a través de ODVA y ControlNet Internacional se plantean los nuevos objetos de manera que estos puedan pasar a formar parte de los objetos definidos por CIP a través del grupo de trabajo llamado Special interest Group (JSIGs/SIGs). De esta forma los objetos pasan a ser públicos.

4.8 Descripción de objetos CIP

A través de la descripción de los siguientes 3 objetos se ilustra el principio general de los objetos CIP.

- Identity Object (u objeto identidad)
- Parameter Object (u objeto parámetro)
- Assembly Object (u objeto de ensamble)

4.8.1 Identity Object

El Identity Object, u Objeto Identidad, es requerido por todo dispositivo CIP. Su Class ID es 0x01. La mayoría de los dispositivos soporta solamente una instancia del objeto identidad. Por ello, no hay requerimientos de cualquier clase de atributos que describan detalles de clases adicionales, como por ejemplo, cuántas instancias existen en el dispositivo. Solamente los atributos de instancia son requeridos en la mayoría de los casos.

Los atributos de un Objeto Identidad son:

- Atributos Obligatorios:
 - Vendor ID
 - Device Type
 - Product Code
 - Revision
 - Status
 - Serial Number
 - Product Name

- Atributos Opcionales
 - State
 - Configuration Consistency Value
 - Heartbeat Interval

a) Descripción de los atributos del objeto identidad

Vendor ID: El atributo Vendor ID identifica al fabricante del dispositivo. Corresponde a un valor entero asignado por ODVA o ControlNet Internacional al fabricante que lo solicite. Este valor será el mismo en cada red.

Device Type: El atributo Device Type, o Tipo de Dispositivo, especifica qué perfil se ha utilizado para este dispositivo. Debe ser uno de los Device Type descritos por CIP o por un tipo específico de fabricante.

Product Code: El atributo Product Code, o Código de Producto, es un número entero definido por el fabricante del dispositivo. Este código se utiliza para distinguir múltiples productos del mismo Device Type y del mismo fabricante.

Revision: El atributo Revision, o Revisión, está dividida en dos valores enteros que especifican una Major Revision, o Revisión Principal, y una Minor Revision, o Revisión Menor. Cualquier cambio (o cambios) del dispositivo que dé lugar a modificar su comportamiento en la red debe ser reflejado en un cambio (o cambios) en la parte Minor del atributo Revisión. Cualquier cambio (o cambios) en el dispositivo que requiere una revisión de su Electronic Data Sheet, EDS (Hoja de Datos Electrónica en español) debe ser reflejado en un cambio en la parte Major del atributo Revisión. Los atributos Vendor ID, Device Type, Product Code y Major Revision proporcionan una identificación inequívoca de un EDS para este dispositivo.

Status: El atributo Status proporciona la información sobre el estado del dispositivo, por ejemplo si está siendo controlado por otro dispositivo o configurado, y si ha ocurrido alguna falla menor o importante.

Serial Number: El atributo serial number, o número de serie, se utiliza para identificar únicamente los dispositivos individuales en conjunto con el Vendor ID, es decir, ninguno de dos dispositivos CIP de un determinado fabricante puede tener el mismo número de serie. El serial number posee 32 bits.

Product Name: El atributo Product Name, o Nombre de Producto, permite al fabricante darle un nombre significativo en código ASCII, de hasta 32 caracteres, al dispositivo.

State: El atributo State, o Estado, describe el estado de un dispositivo en un solo valor entero. Se detalla menos que el atributo *Status*.

Configuration Consistency Value: El atributo Configuration Consistency Value, o Valor de Consistencia de Configuración, permite una distinción entre un dispositivo que ha sido configurado y otro que no lo ha sido, o entre distintas configuraciones en un solo dispositivo. Esto ayuda a evitar descargas innecesarias de configuración.

Heartbeat Interval: El atributo Heartbeat Interval, o Intervalo de Pulso (o latido), habilita el Device Heartbeat Message (o traducido al español como el Mensaje de Pulso de Dispositivo). El tiempo máximo entre dos *Heartbeat* puede ser fijado entre 1 y 255 segundos.

4.8.2 Parameter Object

El Parameter Object, u Objeto Parámetro, es un objeto de uso general que tiene asignada un Class ID = 0x0F. Proporciona un medio general para permitir el acceso a diferentes atributos de los distintos objetos en un dispositivo sin requerimiento de alguna herramienta especial, como un *handheld* o terminal portátil, para conocer los objetos específicos en el dispositivo.

Los atributos del Parameter Object contienen información acerca de cuántas instancias existen en el dispositivo, si es soportada un Parameter Object de versión completa (*full*) o liviana (*stub*), si se utiliza un objeto Configuration Assembly y el lenguaje usado en el Parameter Objectc, entre otras informaciones.

La versión liviana (Stub) del Parameter Object utiliza 6 atributos. Estos son:

- **Parameter Value:** Es el parámetro real.
- **Link Path Size y Link Path:** Estos dos atributos describen la aplicación Objeto/Instancia/Atributo desde la cual ha sido recuperado el Parameter Value.
- **Descriptor:** Describe las propiedades del parámetro, como por ejemplo, sólo lectura, parámetro monitor, etc.
- **Data Type:** Describe el tipo de dato (tamaño, rango, etc) usando un mecanismo estándar definido por CIP.
- **Data Size:** Tamaño del dato en bytes.

Estos seis atributos permiten el acceso, la interpretación y la modificación del valor de parámetro. Sin embargo, los atributos restantes, que incluye la versión completa (full) del Parameter Object, hacen más fácil entender el significado del parámetro. En total, la versión completa posee 24 atributos (los 6 anteriores más otros 18). Los atributos restantes, que siguen a los 6 primeros descritos anteriormente, identifican los siguientes aspectos del parámetro.

- Los próximos tres atributos entregan, a través de secuencia ASCII, el nombre del parámetro, sus unidades de ingeniería y un texto asociado de ayuda;
- Otros tres atributos contienen el valor el mínimo, el valor máximo y los valores por defecto del parámetro;
- Cuatro atributos más pueden unirse al *Parameter Value* para que el parámetro pueda mostrarse de una manera más significativa. Por ejemplo, a un valor sin tratar de corriente, pueden unirse estos atributos y mostrarse específicamente en amperes;
- Otros cuatro atributos más pueden unirse otros parámetros de valor variable para aumentar su información. Por ejemplo, porcentaje de acuerdo a un rango completo, el cual es configurado por otro parámetro;
- El atributo #21 define cuantos lugares decimales deben ser mostrados si el Parameter Value ha sido ampliado;
- Los últimos tres atributos pasados son una versión de lenguaje internacional del nombre del parámetro, sus unidades de ingeniería y un texto asociado de ayuda.

4.8.3 Assembly Object

Un Assembly Object, u Objeto de Ensamble, es un objeto de uso general que tiene asignada el Class ID = 0x04. Este objeto proporciona la opción de *mapear* los datos desde los atributos de diversas instancias y de varias clases en un único atributo (#3): un Assembly Object. Este mapeo es utilizado generalmente para los Mensajes I/O para maximizar la eficiencia en el intercambio de datos de control sobre la red. El mapeo Assembly hace que los datos de I/O estén disponibles en un solo bloque; así, hay pocas instancias de Connection Objects y pocas transmisiones en la red. Los datos de proceso normalmente se combinan de los diferentes Objetos de la Aplicación. Un Assembly Object también puede ser utilizado para configurar un dispositivo con un solo bloque de datos, aliviando la necesidad de configurar parámetros individuales.

El protocolo CIP hace una distinción entre un *Input Assembly* (o Ensamble de Entrada) y un *Output Assembly* (o Ensamble de Salida). En este contexto, los términos “Input” y “Output” son vistos desde la perspectiva del elemento controlador (por ejemplo, un PLC).

Un Input Assembly en un dispositivo *recoge* los datos desde la entrada de aplicación (por ejemplo, un sensor de proximidad) y los *produce* en la red, donde son consumidos por el dispositivo controlador y/o la interfaz de operador.

Un Output Assembly en un dispositivo *consume* los datos que el controlador envía a la red y los escribe a la salida de la aplicación (por ejemplo, el control de la velocidad del motor).

Un Assembly también puede ser utilizado para transmitir un completo conjunto de parámetros configurables en vez de tener que accederlos en forma individual. Este tipo de Assembly es llamado Configuration Assembly (Ensamble de Configuración).

Un ejemplo de mapeo Assembly de un dispositivo CIP típico se muestra en la figura 4.9.

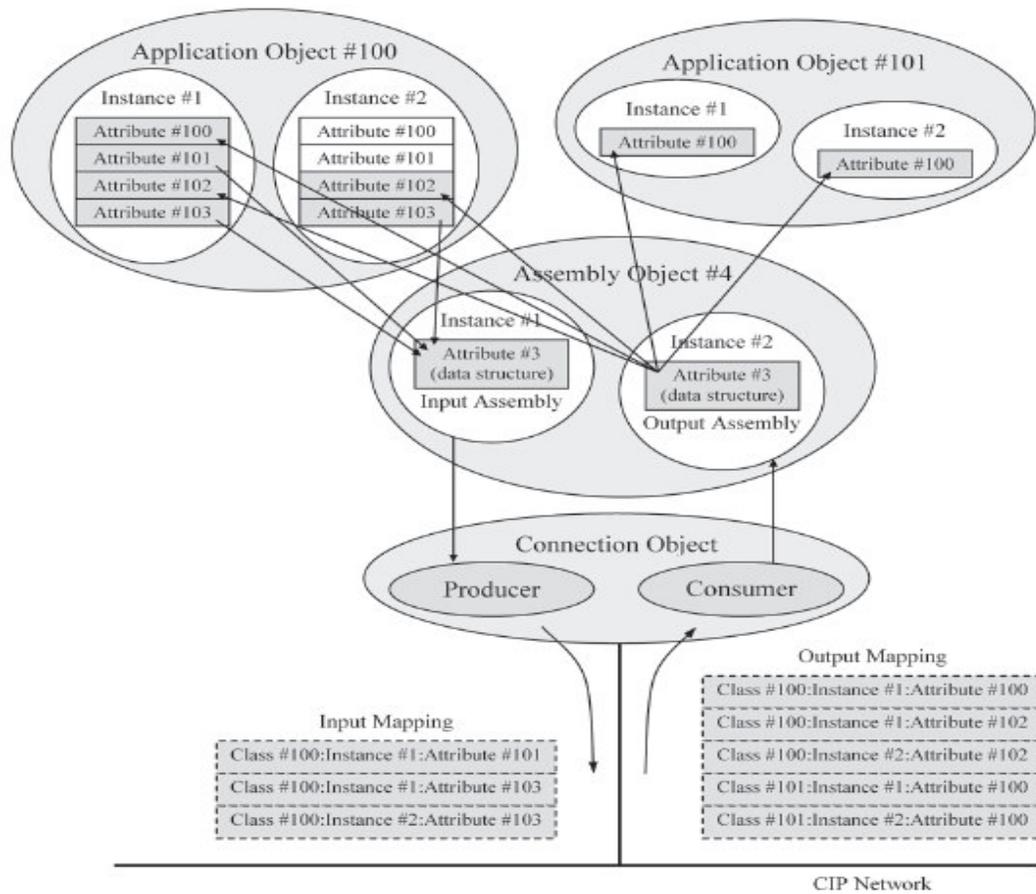


Figura 4.9. Ejemplo de un mapeo Assembly por un típico dispositivo I/O CIP.

Los datos de los objetos de aplicación 100 y 101 son mapeados en dos instancias del objeto Assembly. La Instancia #1 es configurada sobre una Input Assembly para los datos de entrada, y la Instancia #2 como una Output Assembly para los datos de salida. El bloque de datos está siempre accesible a través del atributo #3 de la instancia Assembly pertinente. Los Atributos 1 y 2 contienen la información mapeada.

El mapeo de I/O Assembly es especificado para ciertos perfiles de dispositivo (por ejemplo, partidores de motor) por ODVA. Los desarrolladores de dispositivos pueden elegir qué Assembly soportarán sus productos. Si ningún Assembly de definición pública representa las funcionalidades del dispositivo, entonces el propio fabricante puede desarrollar un Assembly específico de fabricante.

4.9 Perfiles del dispositivo

Es posible diseñar productos para las redes CIP siguiendo solamente las especificaciones de comunicación de la red y el modelo de objetos. Sin embargo esto resultaría en que productos similares finalmente tendrían estructuras y comportamientos muy diferentes unos de otros. Con el objetivo de superar esta situación y permitir una aplicación de dispositivos CIP más simple, se han agrupado los dispositivos de funcionalidad similar y se les ha asociado un determinado perfil. Este *perfil de dispositivo* CIP contiene la descripción completa de la estructura y el comportamiento del objeto.

En la tabla 4.7 se muestran los perfiles de dispositivos, junto a su número identificador, definidos en el Volumen 1 de CIP. Estos perfiles son válidos en todas las redes CIP, tales como DeviceNet y ControlNet.

Perfiles de dispositivos CIP	
AC Drives Device (0x02)	Mass Flow Controller (0x1A)
Communications Adapter (0x0C)	Motor Overload Device (0x03)
Contactora (0x15)	Motor Starter (0x16)
ControlNet Phy. Layer Component (0x32)	Photoelectric Sensor (0x06)
ControlNet PLC (0x0E)	Pneumatic Valve (0x1B)
DC Drives (0x13)	Position Controller (0x10)
DC Power Generator (0x1F)	Process Control Valve (0x1D)
Encoder (0x22)	Residual Gas Analyzer (0x1E)
Fluid Flow Controller (0x24)	Resolver (0x09)
General Purpose Discrete I/O (0x07)	RF Power Generator (0x20)
Generic Device (0x00)	Safety Discrete I/O (0x23)
Human Machine Interface (0x18)	Softstart Starter (0x17)
Inductive Proximity Switch (0x05)	Turbomolecular Vacuum Pump (0x21)
Limit Switch (0x04)	Vacuum/Pressure Gauge (0x1C)

Tabla 4.7. Perfiles de dispositivos CIP.

Los desarrolladores de dispositivos deben usar un perfil. Cualquier dispositivo que no cae dentro de uno de los perfiles especializados enlistados en la tabla 4.7 debe usar un perfil de dispositivo genérico o un perfil específico de fabricante. En la documentación provista por el fabricante al usuario, deben ser descritos el perfil que utiliza y qué partes son implementadas.

Todo perfil de dispositivo se compone un conjunto de objetos, tanto requeridos como opcionales, y un comportamiento asociado con el tipo de dispositivo particular. La mayoría de los perfiles también definen uno o más formatos de dato I/O (Assembly) que definen el significado de sus bits individuales y bytes.

Además de los objetos de definición pública, los vendedores pueden agregar sus propios objetos si sus dispositivos entregan funcionalidades adicionales. Asimismo, los fabricantes pueden desarrollar sus propios perfiles de dispositivos en el contexto de perfil específico de vendedor.

En la tabla 4.8 se enlistan los rangos asignados de los números que identifican los perfiles públicos (es decir, los definidos por ODVA y ControlNet Internacional) y los específicos de vendedor. Todos los otros números de perfil que no aparecen en la tabla están reservados por CIP.

Rangos de perfiles para dispositivos CIP	
Perfiles de dispositivo públicos	0x00 – 0x63
	0x10 – 0x02FF
Perfiles específicos de fabricante	0x64 – 0xC7
	0x0300 – 0x02FF

Tabla 4.8. Perfiles de dispositivos para las redes CIP.

4.10 Configuración de dispositivo y los Electronic Data Sheets

4.10.1 Métodos de Configuración

El protocolo CIP entrega varias opciones para la configuración dispositivos. De ellas se pueden destacar los siguientes tres métodos:

- Por Hoja de datos impresa,
- Por el Parameter Object, y
- Por un Electronic Data Sheet, EDS, (u hoja de datos electrónica),

A los métodos enunciados se agrega también el uso de un objeto Configuration Assembly, y combinaciones de los métodos. De todas las alternativas, la más eficiente es el uso de un Electronic Data Sheet o EDS.

A continuación se da una descripción de los tres métodos de configuración y posteriormente se describe en más detalle lo que es el EDS.

a) Configuración por Hoja de Datos Impresa

Al usar la información de configuración recogida en una hoja de datos impresa, las herramientas de configuración pueden proporcionar solamente las sugerencias para datos de servicio, clase, instancia y atributo y retransmitir esta información a un dispositivo. Mientras que este procedimiento puede hacer el trabajo, es la solución menos deseable puesto que no determina el contexto, el contenido o el formato de los datos.

b) Configuración por Parameter Object

El objeto Parameter Object proporciona una descripción completa de todos los datos configurables en un dispositivo. Puesto que el mismo dispositivo proporciona toda la información necesaria, una herramienta de configuración puede acceder a todos los parámetros y mantener una interfaz de uso amigable. Sin embargo, este método impone una carga para el dispositivo con información completa del parámetro que puede ser excesiva para un dispositivo pequeño, como por ejemplo, dispositivo simple DeviceNet. Por lo tanto, una versión liviana del objeto Parameter Object, llamada Parameter Object Stub, puede ser utilizada. Esta opción todavía permite el acceso a los datos del parámetro, pero no describe el significado de ellos. Se puede utilizar el Parameter Object Stub en combinación con una hoja de datos impresa, pero aún así, la configuración resulta engorrosa.

c) Configuración por EDS

Un EDS, por otra parte, entrega toda la información que contiene un objeto Parameter Object Full, además de la información de la Conexión I/O. De esta forma, el EDS entrega las funcionalidades y facilidades de uso de un objeto Parameter Object Full, sin imponer una carga excesiva sobre un dispositivo individual. Además, un EDS proporciona un medio para que las herramientas de software puedan realizar configuraciones *offline* y descargarla hacia los dispositivos posteriormente.

4.10.2 Descripción del EDS

Un EDS es un archivo de texto escrito en ASCII y que se puede generar en cualquier editor ASCII (por ejemplo, el bloc de notas de Windows). La especificación CIP proporciona un completo sistema de reglas para el diseño y la sintaxis de un EDS, lo cual hace que la configuración de los dispositivos sea más simple. El propósito principal de un EDS es entregar la información sobre varios aspectos de las capacidades del dispositivo, siendo las más importantes las conexiones I/O que soporta y qué parámetros de configuración existen dentro del dispositivo.

a) Secciones de un EDS

Un EDS se estructura en secciones. Cada sección comienza con un nombre de sección entre corchetes []. Las primeras dos secciones son obligatorias para todo el EDSs.

- [File]: Describe el contenido y la revisión del archivo.
- [Device]: Es equivalente a la información del Identity Object y se utiliza para emparejar un EDS a un dispositivo.
- [Device Classification]: Describe con qué red se puede conectar el dispositivo. Esta sección es opcional para DeviceNet, pero requerido para ControlNet y EtherNet/IP.
- [IO_Info]: Describe los métodos de conexión y tamaños de I/O. Requerido para DeviceNet solamente.
- [Variant_IO_Info]:. Describe múltiples configuraciones de datos IO_Info. Requerido solo para DeviceNet
- [ParamClass]: Describe atributos del nivel de clase del Parameter Object.
- [Params]: Identifica todos los parámetros de configuración en el dispositivo, sigue la definición del Parameter Object.
- [EnumPar]: Lista de enumeración de opciones de parámetro para presentar al usuario. Este es un método antiguo especificado solamente para DeviceNet .
- [Assembly]: Describe la estructura de los ítems de dato.
- [Groups]: Identifica a todos los grupos de parámetros en el dispositivo y nombre de grupo de listas y números de instancia de Parameter Object.
- [Connection Manager]: Describe las conexiones soportadas por el dispositivo. Utilizado típicamente en ControlNet y EtherNet/IP.
- [Port]: Describe los varios puertos de red que un dispositivo puede tener.
- [Modular]: Describe las estructuras modulares dentro de un dispositivo
- [Capacity]: Especifica la capacidad de comunicación de los dispositivos ControlNet y EtherNet/IP.

A través de estas secciones se entrega una descripción muy detallada del dispositivo. El fabricante debe determinar qué secciones requiere su dispositivo, de esta forma no es necesario implementar todas las secciones.

Una herramienta de software con una librería de EDS usará primero la sección [Device] para intentar asociar un EDS con cada dispositivo conectado a la red. Una vez que un dispositivo particular ha sido asociado con su EDS, la herramienta de software mostrará sus propiedades y parámetros y permitirá al usuario realizar las modificaciones (si es necesario). Una herramienta de software también puede mostrar qué Conexiones I/O permite un dispositivo y cuál de éstas está en uso.

En los anexos 1 al 4 se muestran el archivo EDS de un dispositivo DeviceNet así como de un dispositivo ControlNet.

4.11 Data Management

La especificación CIP posee una parte denominada Data Management, que puede traducirse al español como el Gestionamiento de los Datos. Ésta describe los modelos de direccionamiento para las entidades CIP y su estructura de datos.

El direccionamiento de la entidad es realizado a través de Segmentos, los que permiten flexibilidad en el uso de diferentes tipos de métodos de direccionamiento.

El primer byte de un Segmento CIP permite una distinción entre los siguientes dos esquemas de direccionamiento:

- Segment Address, o Segmento de direccionamiento (0x00 - 0x9F), y
- Data Type, o Tipo de Dato (0xA0 - 0xDF).

De aquí se desprenden dos usos de este esquema de direccionamiento, que son el Logical Segments, o Segmentos Lógicos y el Data Type, o Tipo de Dato.

4.11.1 Logical Segments

Los Logical Segments, o Segmentos Lógicos, son segmentos de direccionamiento que pueden ser utilizados para direccionar objetos y sus atributos dentro de un dispositivo. Su primer byte se encuentra en el rango 0x20 – 0x3F. Típicamente se estructura de la siguiente manera:

[Class ID] [Instance ID] [Attribute ID]

Cada elemento de esta estructura permite varios formatos: 1 byte, 2 bytes, y 4 bytes. La figura 4.10 muestra un ejemplo típico de este método de dirección.

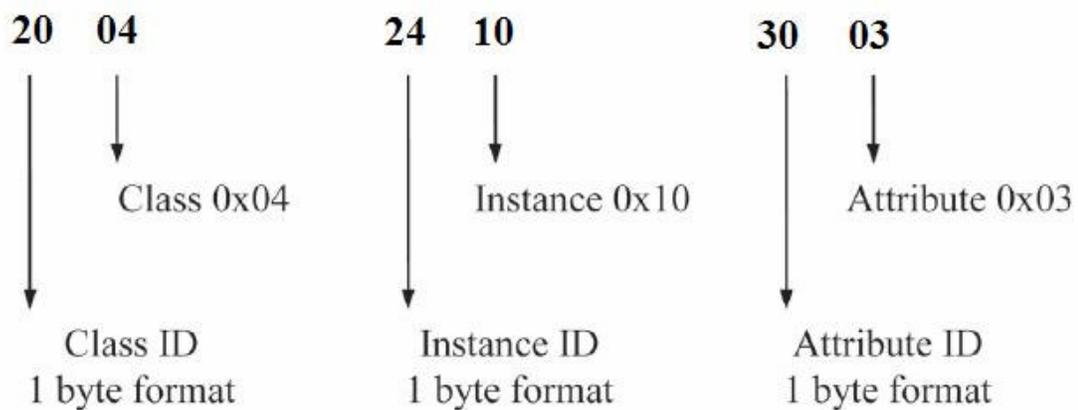


Figura 4.10. Ejemplo de codificación de Logical Segments.

Este tipo de direccionamiento es generalmente usado para apuntar a Assembly, Parameters y otros atributos direccionables dentro de un dispositivo. También se utiliza extensivamente en los EDSs.

4.11.2 Data Types

Los Data Types, o Tipos de Datos, pueden ser de dos formas:

- Data Type Estructurados (primer byte = 0xA0 – 0xA3), o
- Data Type Elementales (primer y único byte 0xC1 – 0xDE).

Los Data Type Estructurados pueden ser arreglos de Data Type Elementales o cualquier composición de arreglos de Data Type Elementales. En la tabla 4.9 se enlistan los Data Type más utilizados.

Data Type más utilizados
<ul style="list-style-type: none"> • 1-bit (encoded into 1-byte): <ul style="list-style-type: none"> ◦ Boolean, BOOL, Type Code 0xC1
<ul style="list-style-type: none"> • 1-byte: <ul style="list-style-type: none"> ◦ Bit string, 8 bits, BYTE, Type Code 0xD1 ◦ Unsigned 8-bit integer, USINT, Type Code 0xC6 ◦ Signed 8-bit integer, SINT, Type Code 0xC2
<ul style="list-style-type: none"> • 2-byte: <ul style="list-style-type: none"> ◦ Bit string, 16-bits, WORD, Type Code 0xD2 ◦ Unsigned 16-bit integer, UINT, Type Code 0xC7 ◦ Signed 16-bit integer, INT, Type Code 0xC3
<ul style="list-style-type: none"> • 4-byte: <ul style="list-style-type: none"> ◦ Bit string, 32-bits, DWORD, Type Code 0xD3 ◦ Unsigned 32-bit integer, UDINT, Type Code 0xC8 ◦ Signed 32-bit integer, DINT, Type Code 0xC4

Tabla 4.9. Data Types más utilizados en CIP.

4.12 Bridging y Ruteo de mensajes CIP

El protocolo CIP define mecanismos que permiten la transmisión de mensajes a través de las diferentes redes que lo implementen. Dado que las capas de aplicación de las redes de la familia CIP comparten una estructura común, los mensajes pueden ser transmitidos desde una red CIP a otra. Por ejemplo, un mensaje producido por un nodo en una red DeviceNet, puede ser transmitido al medio y consumido en por nodo conectado a la red ControlNet. Esto se realiza a través de dispositivos Routers o Bridge.

En la figura 4.11 se ilustra el recorrido de un mensaje CIP desde una red una red DeviceNet hacia una red ControlNet.

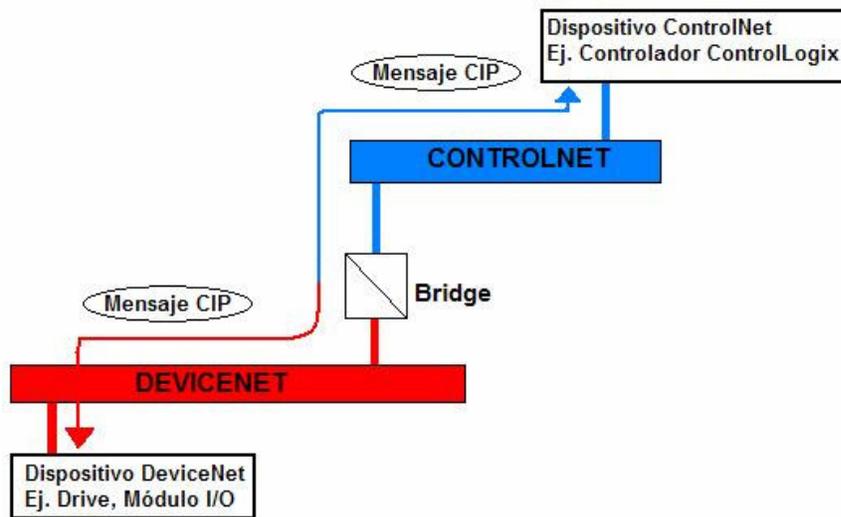


Figura 4.11. Recorrido de un mensaje CIP entre las redes DeviceNet y ControlNet.

El dispositivo *bridge* para redes CIP solo debe traducir la porción correspondiente a la tecnología, es decir, los contenidos relativos a las capas inferiores (física, enlace, de red y transporte). Pero respecto del significado al nivel de aplicación, este es el mismo en ambas redes. Esto optimiza el uso de de redes CIP para implementar los distintos niveles de una arquitectura de comunicación para la integración de los niveles CIM.