II Jornadas sobre Tecnologías y soluciones para la Automatización Industrial

DOCUMENTACIÓN ANEXA A LA CONFERENCIA

Transprent Ready:
COMUNICACIONES INDUSTRIALES PARA LA INTEGRACIÓN DE DISPOSITIVOS Y SISTEMAS ABIERTOS.
Introduction to industrial communication networks

Section 1: Basic concepts
Section 2: Requirements and positioning of the main networks
Section 3: The ISO model
Section 4: Physical media
Section 5: Major medium access methods
Section 6: Concepts used at application level
Section 7: Interconnection products
Introduction to industrial communication networks

Section 8: ASi
Section 9: CANopen
Section 10: DeviceNet
Section 11: Ethernet - TCP/IP - Modbus
Section 12: Profibus-DP
Section 13: FIPIO
Introduction to industrial communication networks

Section 14: Interbus
Section 15: Modbus
Section 16: Comparison table for the major networks
Section 17: A look at the IA communication offer
Section 18: How PL7 deals with the communication function
The data comprises physical elements (light, sound, images, electrical voltage, etc.) to which a direction has been attributed.
Transmission methods

Data can be transmitted in **analog** format:
Continuous progression of value

Or in **digital** format:
Discontinuous progression of value (sampling)
Section 1: Basic concepts

Transmission types

**Simplex** transmission: Unidirectional

**Half duplex** transmission: Alternate bidirectional

**Full duplex** transmission: Simultaneous bidirectional
Section 1: Basic concepts

Transmission types

- **Serial transmission:**
  The link usually requires 3 wires: send, receive and earth.
  The bits in a byte are transmitted one after the other.

- **Parallel transmission:**
  The bits in a byte are transmitted simultaneously.
  Used for short distances. As each channel tends to cause interference on
  neighbouring channels, the quality of the signal deteriorates rapidly.
Serial transmission types

- **Synchronous serial transmission:**
  Data is transmitted continuously.
  A synchronization signal is transmitted in parallel with the data signals.

- **Asynchronous serial transmission:**
  Data can be transmitted in an irregular fashion, although the interval between 2 bits is fixed.
  Synchronization bits (START, STOP) encapsulate the data.
For reasons of cost and durability, most communication networks use **half duplex asynchronous serial digital transmission**.
Section 2: Requirements and positioning of the main networks

**Communication requirements**

- **Level 3 Company**
  - **Amount of data to be transmitted**: 1 MB
  - **Speed of reaction**: 1 ms

- **Level 2 Workshop**
  - **Amount of data to be transmitted**: 1 KB
  - **Speed of reaction**: 1 s

- **Level 1 Machines**
  - **Amount of data to be transmitted**: 1 bit
  - **Speed of reaction**: 1 ms

- **Level 0 Sensors and Actuators**

- **Information system**
  - **Production management Supervision**
  - **Control system**
  - **Components**
Section 2: Requirements and positioning of the main networks

Main networks and buses

- Control of process
- Control of machine

Simple
- AS-i

Sophisticated
- CANopen
- FIPIO
- Modbus Plus
- Profibus-DP
- DeviceNet
- Interbus
- Modbus

Local area networks (Field Bus)
- CANopen
- FIPIO
- Modbus Plus
- Profibus-DP
- DeviceNet
- Interbus
- Modbus

Data networks (Data Bus)
- Ethernet
- TCP/IP
- FTP
- HTTP...

Ethernet TCP/IP FTP-HHTTP...
Section 2: Requirements and positioning of the main networks

Network strategy of the Schneider industrial sector

- **Core networks:**

  **Ethernet TCP/IP & Modbus**
  Levels 2 and 3: Information and control system (inter-PLC)
  to be extended to fieldbus level (level 1)

  **CANopen**
  Like an internal device and panel bus (e.g.: Automation Island)

  **ASi**
  For the connection of sensors/actuators (level 0)

  **Modbus RS 485**
  When Ethernet is not suitable (price, topology, etc.)
Section 2: Requirements and positioning of the main networks

Network strategy of the Schneider industrial sector

- **Legacy networks**
  
  FIPIO, Modbus Plus, Uni-Telway, Seriplex

- **Connectivity networks**
  
  A pragmatic approach when the market imposes a solution

  DeviceNet (Allen-Bradley) - Profibus (Siemens) - Interbus (Phoenix) etc.
## ISO model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Physical layer</td>
</tr>
<tr>
<td>2</td>
<td>Link layer</td>
</tr>
<tr>
<td>3</td>
<td>Network layer</td>
</tr>
<tr>
<td>4</td>
<td>Transport layer</td>
</tr>
<tr>
<td>5</td>
<td>Session layer</td>
</tr>
<tr>
<td>6</td>
<td>Presentation layer</td>
</tr>
<tr>
<td>7</td>
<td>Application layer</td>
</tr>
</tbody>
</table>

### Application Layer
- Network administration (starting and stopping the network, message handling)
- Entity used for PC/MAC dialogue

### Presentation Layer
- Organize and synchronize the exchanges between users of the network
- End-to-end checking: restart on errors which have been signalled or otherwise by the network layer

### Network Layer
- Switching in a mesh network: establishment of route
- Sub-layer: error correction, acknowledgement
- Sub-layer: management of access to physical medium

### Session Layer
- Twisted pair, shielded twisted pair, coaxial cable, optical fibre...

---

TCP: Transmission Control Protocol (Layer 4)
IP: Internet Protocol (Layer 3)
Examples of frames in relation to the ISO model

Modbus RTU frame
Request to read words W5 and W6 at slave address 7

<table>
<thead>
<tr>
<th>Bytes</th>
<th>1</th>
<th>1</th>
<th>2</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave address 7</td>
<td>Function code 3</td>
<td>No. of first word 5</td>
<td>No. of words to be read 2</td>
<td>CRC 16</td>
<td></td>
</tr>
</tbody>
</table>

Ethernet TCP-IP frame

<table>
<thead>
<tr>
<th>Bytes</th>
<th>8</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>20</th>
<th>20</th>
<th>46 to 1500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Destin. addr.</td>
<td>Source addr.</td>
<td>LLC</td>
<td>IP</td>
<td>TCP</td>
<td>Application layers</td>
<td>FTP, HTTP, SMTP Modbus etc.</td>
</tr>
</tbody>
</table>

Chapter 3: The ISO model
Physical media

Most popular transmission media
A few electrical standards for twisted pairs
The various topologies
Most popular transmission media

The MEDIA establish the transmission quality:

- speed
- distance
- electromagnetic immunity

Most commonly used media:

**Pair of twisted wires**
The simplest to install, and the **least expensive**.

**Coaxial cable**
This consists of a copper conductor, surrounded by grounding shielding. There is a plastic insulating layer between the conductor and the shielding. The coaxial cable has **excellent electrical properties** and is suitable for **high speed** transmission.

**Optical fibre**
Electrical signals are not carried by a copper cable, but an optical fibre transmits light signals.
This is suitable for use in **harsh industrial environments**. Transmission is reliable over **long distances**.
RS232:
Point-to-point link via 25-pin SUB-D connector.
Distance < 15 meters, speed < 20 Kbps.

RS422A:
Full duplex (simultaneous bidirectional) multi-drop bus on 4 wires.
2 transmission wires, 2 reception wires.
Good immunity to interference. Max distance 1200 meters at 100 Kbps.

RS485:
Same characteristics as RS422A but on 2 wires.
Half duplex (alternate bidirectional) multi-drop bus on 2 wires.


Chapter 4: Physical media

The various topologies

**POINT-TO-POINT** TOPOLOGY

(Between 2 units in communication)

**GRID** TOPOLOGY

(Devices are linked to one another, forming a “spider’s web”. There are a number of possible paths for reaching a node)

**STAR** TOPOLOGY

(Several units communicating via their own line with a Central unit)

**RING** TOPOLOGY

(All the units are connected in series in a closed loop. Communications must pass via all the units to arrive at the receiver)

**TREE** TOPOLOGY

(This is a variant of the star topology)

**BUS** TOPOLOGY

(The network consists of a main line to which all the units are connected)
The main medium access methods

Master - Slave
Token ring
Random access
Section 5: The main medium access methods

**Master - Slave**

Located at the link layer level

The **MASTER** is the entity which grants access to the medium.

The **SLAVE** is the entity which accesses the medium after requesting it from the master.

**Polling**

- What do you want to say?
- Nothing to declare

**Eg: Profibus-DP**
Section 5: The main medium access methods

Located at the link layer level

The members of a logical **RING** gain access to the network upon receipt of a token.

The **TOKEN** is a group of bits that is passed in a rotating address sequence from one node to another.

Eg: Modbus Plus
Section 5: The main medium access methods

Random access

Located at the link layer level

**Carrier Sense Multiple Access**

A set of rules determining how network devices respond when two devices attempt to use the medium simultaneously (called a *collision*).

CSMA/CD is a type of contention protocol: competition for resources.

Informal discussion between undisciplined individuals:

As soon as there’s a silence, the one who wants to talk begins to speak.
Section 5: The main medium access methods

CSMA/CD = Carrier Sense Multiple Access Collision Detect: Destructive collision

1 - Collision detection
2 - Stop of the emitted frame
3 - Scrambling frame emission
4 - Wait a random time
5 - Frame re-emission

Eg: Ethernet

CSMA/CD = Carrier Sense Multiple Access Collision Avoidance: Non destructive collision

1 - Non destructive collision detection
2 - The device with the lower priority stops its transmission
3 - End of the high priority frame transmission
4 - The device with lower priority can send its frame

Eg: CAN
Concepts used at application level

Client - Server
Producer - Consumer
Traffic types
The concept of a profile
Section 6: Concepts used at application level

Client - Server

The **CLIENT** is an entity requesting a service on the network

The **SERVER** is the entity which responds to a request from a client

Request

Please can you send me the configuration of motor starter no. 3?

Response

No problem, here is the whole file!

Eg: Modbus
Section 6: Concepts used at application level

The **PRODUCER** is a single entity which produces information.

The **CONSUMER** is an entity which uses it (several entities can use the same information).

It’s 6 pm

PRODUCER

CONSUMER N°1

I’m going to miss my train!!!

CONSUMER N°2

Let’s go to see a movie...

Eg: CANopen DeviceNet
Traffic types

Cyclical data:
Data that is refreshed periodically according to a pre-determined time.
This is process data.

A small amount of information refreshed frequently.

Acyclic data:
Data that is refreshed according to a request or to an event.
This is used at start-up for configuration and setup, or for diagnostics in the event of a fault.

A lot of information without time constraints.
Section 6: Concepts used at application level

An open system comprises interoperable and interchangeable components.

Interoperability is the ability to communicate intelligibly with other devices.
It is achieved by means of strict adherence to protocol specifications.

Interchangeability is the ability to replace one device with another (possibly supplied by a different manufacturer).
It is achieved by means of adherence to profile specifications.

All manufacturers reserve the right to define whether or not they wish to offer manufacturer-specific functions in addition to those which are part of the minimum profile or core.
A **profile** is a **standardized way** of describing functions which ensure components can be interchanged.

This description adheres to a **strict syntax**. Data is grouped by function:

- Identification: product name, reference, version, family, manufacturer
- Characteristics relating to communication: Speeds supported, type and size of messages exchanged, etc.
- Characteristics relating to the application: Variables which can be accessed in write mode, in read mode, when stopped, when running, etc.

Most profiles **are provided in electronic file format**: EDS file, GSD file, etc. supplied on floppy disk or CD-ROM with the product. This file provides details of the characteristics of the device "offline".
## Section 6: Concepts used at application level

### Extract from TEGO Power Quickfit CANopen EDS file

<table>
<thead>
<tr>
<th>[FileInfo]</th>
<th>[MandatoryObjects]</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreatedBy=Martin Rostan</td>
<td>SupportedObjects=2</td>
</tr>
<tr>
<td>ModifiedBy=Martin Rostan</td>
<td>1=0x1000</td>
</tr>
<tr>
<td>Description=EDS for Tego Power CANopen</td>
<td>2=0x1001</td>
</tr>
<tr>
<td>CreationTime=10:05PM</td>
<td></td>
</tr>
<tr>
<td>CreationDate=01-17-2001</td>
<td></td>
</tr>
<tr>
<td>ModificationTime=10:35PM</td>
<td></td>
</tr>
<tr>
<td>ModificationDate=01-17-2001</td>
<td></td>
</tr>
<tr>
<td>FileName=F:\Produkte\Tego Power\APP1CC00</td>
<td>[1000]</td>
</tr>
<tr>
<td>FileVersion=1</td>
<td>ParameterName=Device Type</td>
</tr>
<tr>
<td>FileRevision=1</td>
<td>ObjectType=0x7</td>
</tr>
<tr>
<td>EDSVersion=4</td>
<td>DataType=0x0007</td>
</tr>
<tr>
<td>[DeviceInfo]</td>
<td>AccessType=ro</td>
</tr>
<tr>
<td>VendorName=Schneider Electric SA (France)</td>
<td>DefaultValue=0x30191</td>
</tr>
<tr>
<td>VendorNumber=90</td>
<td></td>
</tr>
<tr>
<td>ProductName=APP-1CC00</td>
<td></td>
</tr>
<tr>
<td>ProductNumber=1</td>
<td></td>
</tr>
<tr>
<td>RevisionNumber=1</td>
<td></td>
</tr>
<tr>
<td>OrderCode=APP-1CC00</td>
<td></td>
</tr>
<tr>
<td>BaudRate_10=0</td>
<td></td>
</tr>
<tr>
<td>BaudRate_20=0</td>
<td></td>
</tr>
<tr>
<td>BaudRate_50=0</td>
<td></td>
</tr>
<tr>
<td>BaudRate_125=1</td>
<td></td>
</tr>
<tr>
<td>BaudRate_250=1</td>
<td></td>
</tr>
<tr>
<td>BaudRate_500=1</td>
<td></td>
</tr>
<tr>
<td>BaudRate_800=0</td>
<td></td>
</tr>
<tr>
<td>BaudRate_1000=1</td>
<td></td>
</tr>
</tbody>
</table>
Interconnection products

- Repeater
- Hub
- Switch
- Transceiver
- Bridge
- Router
- Gateway
Section 7: Interconnection products

**Repeater**
- Can be used to add segments to a network.
- It amplifies and restores the same type of signal.
- Example = RS485 repeater

**Hub**
- Can be used to extend a star network.
- It amplifies and restores the same type of signal on all ports.
- Example = Ethernet hub
  (does not reduce the number of collisions)

**Switch**
- Can be used to extend a star network.
- It amplifies and restores the same type of signal on a single port.
- Example = Ethernet switch
  (can be used to reduce the number of collisions)
Section 7: Interconnection products

Transceiver - Bridge

Transceiver

Can be used to add different types of segment to a network

Example = RS232/RS485 transceiver

Bridge

Can be used to connect 2 networks using the same protocol but different lower layers

Example = Modbus RS485/Ethernet TCP-IP bridge
Section 7: Interconnection products

**Router - Gateway**

**Router**

- Can be used to connect 2 networks of the same type
- Example = Ethernet TCP-IP router

**Gateway**

- Can be used to connect 2 networks of different types
- Example = FIPIO/Modbus gateway
ASi

History
ASi and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
Section 8: ASi

**History**

- **1990:**
  2 universities and 11 companies (mainly German) create the ASi consortium in order to define a "low-cost" interface for connecting sensors and actuators.

- **1992:**
  The first chips become available.
  Creation of the international ASi association: [http://www.as-interface.net/](http://www.as-interface.net/) based in Germany. Schneider joins the association.

- **1995:**
  Creation of national promotional associations (France, The Netherlands, UK)

- **2001:**
  ASi V2 specifications: 62 slaves, support for analog products, improved diagnostics.
  Integration of safety products: "Safety at work"
### ASi and the ISO model

#### 3 layers used + profiles

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>APPLICATION</td>
</tr>
<tr>
<td>6</td>
<td>PRESENTATION</td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
</tr>
<tr>
<td>3</td>
<td>NETWORK</td>
</tr>
<tr>
<td>2</td>
<td>LINK = LLC + MAC</td>
</tr>
<tr>
<td>1</td>
<td>PHYSICAL</td>
</tr>
</tbody>
</table>

- **Generic discrete I/O interfaces**
- **Discrete sensors**
- **Motor starters**
- **Analog I/O, etc.**
### Physical layer

| Medium:                        | 2-wire yellow flat ribbon cable with polarization  
|                               | An unshielded round cable can also be used. |
| Topology:                     | Free                                          |
|                              | No line terminators                          |
| Maximum distance:             | 100 m without repeaters                      |
|                               | 300 m with repeaters                         |
| Speed:                        | 167 Kbps                                     |
|                              | 1 transaction (data exchange) lasts 150 ms.   |
|                              | Cycle time = 5 ms for 31 slaves               |
|                              | 10 ms for 62 slaves                          |
| Max. no. of devices:          | ASi V1: 1 master + 31 slaves                 |
|                              | ASi V2: 1 master + 62 A/B slaves             |
4 types of connection defined in the Schneider ASi specification

- **IP20**
  - Yellow 2-pin removable connector
  - Screw or spring terminals

- **IP65**
  - IDC connector
  - M12 connector (male on product)
Section 8: ASi

**Link layer**

**Medium access method:** Master/Slave

**Max. size of useful data:**
- 4 output bits for a request
- (3 bits in ASi V2 for A/B slaves)
- 4 input bits for a response

**Transmission security:** Numerous checks at bit and frame level
- Start bit delimiter, half-wave pulses,
- length of pause between 2 bits,
- end-of-frame parity, end bit delimiter,
- length of frame
A dozen standardized requests for:

1. **Network administration**: Addressing, identification, parameter settings, reset.

2. **Cyclic I/O exchange**: Data exchange
   - Max. 4 output bits for standard slaves, 3 for A/B slaves
   - Max. 4 input bits for all slaves
   - Cycle time: 5 ms max. for 31 slaves, 10 ms for 62

3. **Cyclic network monitoring**: Read Status
   - Feedback of I/O errors for ASi V2 slaves
   - Cycle time: 155 ms for 31 slaves, 310 ms for 62 slaves

4. **Parameter data transmission**: Write Parameter
   - Via programming of Write Parameter request
   - Max. 4 output bits for standard slaves, 3 for A/B slaves
   - 155 ms maximum for 31 slaves, 310 ms for 62
To ensure interchangeability between products, every ASi slave is identified by a fixed profile which is engraved in the silicon (read-only).

The profile for ASi V1 slaves is defined using 2 hexadecimal digits.

The profile for ASi V2 slaves is defined using 4 hexadecimal digits.
Profiles

**ASi V1: 2 digits**

Profile = IO\_code . ID\_code

- IO\_code = Indicates the number of inputs and outputs on the device (0 to F)
- ID\_code = Indicates the type of device (0 to F)

**ASi V2: 4 digits**

Profile = IO\_code . ID\_code . ID1\_code . ID2\_code

- IO\_code = Indicates the number of inputs and outputs on the device (0 to F)
- ID\_code = Indicates the type of device (0 to F)
- ID1\_code = Used for customizing the product (0 to F)
- ID2\_code = Indicates the product sub-type (0 to F)
CANopen

History
CANopen and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
**History**

- **1980-1983:**
  Creation of **CAN** as an initiative by the German equipment manufacturer **BOSCH** to meet a requirement in the **automotive industry**. CAN only defines one part of layers 1 and 2 of the ISO model.

- **1983-1987:**
  The prices of drivers and micro-controllers featuring CAN become very attractive as they are used in high volume in the automotive industry.

- **1991:**
  **CIA = CAN in Automation** is born: [http://www.can-cia.de/](http://www.can-cia.de/) to promote industrial applications.
Section 9: CANopen

**History**

- **1993:**
  CAL = CAN Application Layer specifications published by CiA describing transmission mechanisms but not when and how to use them.

- **1995:**
  CiA publishes the **DS-301 communication profile: CANopen**

- **2001:**
  CiA publishes DS-304 which can be used to integrate **level 4 safety components** on a standard CANopen bus (CANsafe).
### CANopen and the ISO model

CANopen is based on CAL

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
<th>Device Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>APPLICATION</td>
<td>CiA DS-301 = Communication profile</td>
</tr>
<tr>
<td>6</td>
<td>PRESENTATION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
<td>EMPTY</td>
</tr>
<tr>
<td>3</td>
<td>NETWORK</td>
<td>EMPTY</td>
</tr>
<tr>
<td>2</td>
<td>LINK = LLC + MAC</td>
<td>CAN 2.0 A and B + ISO 11898</td>
</tr>
<tr>
<td>1</td>
<td>PHYSICAL</td>
<td>CAN 2.0 A and B = ISO 11898-1 and 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ISO 11898 + DS-102</td>
</tr>
</tbody>
</table>

CAL = CAN Application Layer
### Physical layer

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medium</td>
<td>Shielded twisted pair</td>
</tr>
<tr>
<td></td>
<td>2 or 4-wire (if power supply)</td>
</tr>
<tr>
<td>Topology</td>
<td>Bus type</td>
</tr>
<tr>
<td></td>
<td>With short tap links and 120 ohm line termination resistor</td>
</tr>
<tr>
<td>Maximum distance</td>
<td>1000 m</td>
</tr>
<tr>
<td>Speed</td>
<td>9 possible speeds from 1 Mbps to 10 Kbps</td>
</tr>
<tr>
<td></td>
<td>Depends on bus length and cable type: 25 m at 1 Mbps, 1000 m at 10Kbps</td>
</tr>
<tr>
<td>Max. no. of devices</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>1 master and 127 slaves</td>
</tr>
</tbody>
</table>
CiA recommendation DR-303-1 includes a list of suitable connectors divided into 3 categories with a description of their pin configuration.
Example architecture

Line termination resistor

Line termination resistor (120 Ω)

FTB1CN

FTB1CN
Medium access method: CSMA/CA
Every device may send data as soon as the bus is free.
The principle of dominant and recessive bits enables non-destructive bit-by-bit arbitration in the event of a collision.
The priority of a message is indicated by the value of the identifier: The identifier with the lowest value has priority.

Communication model: Producer/Consumer
An identifier coded on 11 bits and located at the start of the message informs the receivers about the type of data contained in each message. Each receiver decides whether or not to accept the data.
This concept permits multiple communication models:
Transmission on change of state, cyclic, SYNC signal, Master_Slave system.
Max. size of useful data: 8 bytes per frame

Transmission security:
One of the best local industrial networks
Numerous signalling and error detection devices ensure high transmission security.
**Application layer**

4 types of standardized service:

1. **Network administration**: Parameter settings, start-up, monitoring (master-slaves)

2. Transmission of **low-volume process data** (<= 8 bytes) in real time: PDO = Process Data Object (producer-consumer)
   PDOs can be transmitted on changes of state, cyclically, on receipt of the SYNC message or at the request of the master.

3. - Transmission of high-volume **parameter data** (> 8 bytes) by segmentation without time restrictions: SDO = Service Data Object (client-server)

4. Predefined messages for managing **synchronization (SYNC), time-based references, fatal errors**: SFO = Special Function Object
The allocation of identifiers on CANopen is based on the division of the identifier into 2 parts:

Function code is used to code 2 receiving PDOs, 2 sending PDOs, 1 SDO, 1 EMCY object, 1 Node Guarding identifier, 1 SYNC object, 1 time stamp object and 1 Node Guard.

Node ID corresponds to the address of the product coded for example using DIP switches.
## Broadcast objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Function code</th>
<th>Node ID</th>
<th>CMS priority group</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMT</td>
<td>0000</td>
<td>0x000</td>
<td>0</td>
</tr>
<tr>
<td>SYNC</td>
<td>0001</td>
<td>0x080</td>
<td>0</td>
</tr>
<tr>
<td>TIME STAMP</td>
<td>0010</td>
<td>0x100</td>
<td>1</td>
</tr>
</tbody>
</table>

## Peer-to-peer objects

<table>
<thead>
<tr>
<th>Object</th>
<th>Function code</th>
<th>Node ID</th>
<th>CMS priority group</th>
</tr>
</thead>
<tbody>
<tr>
<td>Emergency</td>
<td>0001</td>
<td>0x081-0x0FF</td>
<td>0, 1</td>
</tr>
<tr>
<td>Transmit PDO 1</td>
<td>0011</td>
<td>0x181-0x1FF</td>
<td>0, 1</td>
</tr>
<tr>
<td>Receive PDO 1</td>
<td>0100</td>
<td>0x201-0x27F</td>
<td>2</td>
</tr>
<tr>
<td>Transmit PDO 2</td>
<td>0101</td>
<td>0x281-0x2FF</td>
<td>2, 3</td>
</tr>
<tr>
<td>Receive PDO 2</td>
<td>0110</td>
<td>0x301-0x37F</td>
<td>3, 4</td>
</tr>
<tr>
<td>Server SDO</td>
<td>1011</td>
<td>0x581-0x5FF</td>
<td>6</td>
</tr>
<tr>
<td>Client SDO</td>
<td>1100</td>
<td>0x601-0x67F</td>
<td>6, 7</td>
</tr>
<tr>
<td>NODE GUARD</td>
<td>1110</td>
<td>0x701-0x77F</td>
<td>1</td>
</tr>
</tbody>
</table>
CANopen profiles are based on the object dictionary concept: **Device Object Dictionary (OD)**.

The CANopen object dictionary is an **ordered group of objects** which can be accessed via an index of 16 bits and, if required, a sub-index of 8 bits.

Each network node has an OD in an ASCII format **EDS (Electronic Data Sheet)** file (DSP 306 specification).

This dictionary contains all the elements describing the node along with its network characteristics.
### Object dictionary structure

<table>
<thead>
<tr>
<th>Index (hex)</th>
<th>Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>Reserved</td>
</tr>
<tr>
<td>0001 – 009F</td>
<td>Data types area</td>
</tr>
<tr>
<td>00A0 – 0FFF</td>
<td>Reserved</td>
</tr>
<tr>
<td>1000 – 1FFF</td>
<td>Communication profile area</td>
</tr>
<tr>
<td>2000 – 5FFF</td>
<td>Manufacturer-specific profile area</td>
</tr>
<tr>
<td>6000 – 9FFF</td>
<td>Standardized device profile area</td>
</tr>
<tr>
<td>A000 – FFFF</td>
<td>Reserved</td>
</tr>
</tbody>
</table>
CANopen defines 2 types of profile:

**DS-301 communication profile:**
Describes the general structure of the OD and the objects in the "communication profile area" zone. It is valid for all CANopen products.

**DSP-4xx device profiles:**
Describes the various standard objects associated with the different types of product (discrete I/O modules, drives, measuring devices). Some objects are mandatory, others are optional, some are read only, others are read/write.
DeviceNet

History
DeviceNet and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
1980-1983:
Creation of CAN as an initiative by the German equipment manufacturer BOSCH to meet a requirement in the automotive industry. CAN only defines one part of layers 1 and 2 of the ISO model.

1983-1987:
The prices of drivers and micro-controllers featuring CAN become very attractive as they are used in high volume in the automotive industry.

1993-1994:
Allen Bradley (Rockwell Automation Group) develops and launches DeviceNet products.
History

- **1995:**

- **1997:**
  The association includes approximately 200 member companies and offers a hundred different products.

- **2002:**
  The ODVA starts to develop specifications for integrating safety components.
DeviceNet Specifications Volume 1

<table>
<thead>
<tr>
<th>7</th>
<th>APPLICATION</th>
<th>DeviceNet Specifications Volume 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>PRESENTATION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
<td>EMPTY</td>
</tr>
<tr>
<td>3</td>
<td>NETWORK</td>
<td>EMPTY</td>
</tr>
<tr>
<td>2</td>
<td>LINK = LLC + MAC</td>
<td>CAN 2.0 A and B + ISO 11898</td>
</tr>
<tr>
<td>1</td>
<td>PHYSICAL</td>
<td>CAN 2.0 A and B = ISO 11898-1 and 2</td>
</tr>
</tbody>
</table>

DeviceNet and the ISO model

CiA DS-301 = Communication profile

CAN 2.0 A and B + ISO 11898

CAN 2.0 A and B = ISO 11898-1 and 2
Section 10: DeviceNet

Physical layer

Medium: 2 shielded twisted pairs
2 wires for communication and 2 wires for power

Topology: Bus type
With short tap links and 120 ohm line termination resistor

Maximum distance: 1000 m

Speed: 3 possible speeds: 125, 250 or 500 Kbps
Depends on bus length and cable type as well as product consumption

Max. no. of devices: 64 master modes (scanner) included
All connectors must have 5 pins.

The following connectors are recommended:

**Phoenix Combicon**

MSTB 2.5/5-ST-5.08-AU: Network cable side  
MSTBA 2.5/5-G-5.08-AU: Product side, horizontal pins  
MSTBVA 2.5/5-G-5.08-AU: Product side, vertical pins

**Mini Style connector**

ANSI/B93.55M-1981
Section 10: DeviceNet

Connectors

Micro Style connector (M12)
Lumberg RST 5-56/xm or equivalent
Section 10: DeviceNet

IP20 taps

For use as open tap with zero length drop line or for daisy-chain drop line

Trunk or drop line

Open tap with drop line (up to 6 m/20 ft.)

Drop

Trunk or drop line

Daisy chain drop line

Screw connector as shown in Figure 9.17 in Section 9.3.7.2

Trunk or drop line

Drop Lines

Multi-port tap

Trunk or drop line

Drop Lines
Section 10: DeviceNet

IP65 taps

- Sealed mini-style
- Sealed multi-port tap with connectors for four drop lines
- "T" Tap
- Junction box (with cord grips)
- Trunk or drop line
- Cord grips
Section 10: DeviceNet

Example architecture

- Allen Bradley PLC
- Quantum
- 24 V power supply
- Line termination resistor
- Thin cable
- ATV58
- TEGO POWER
- LUF P
- Tesys model U
- FTB
- Modbus
- ATV28
- ATS48
- Line termination resistor
Medium access method: **CSMA/CA**

Every device may send data as soon as the bus is free.

The principle of dominant and recessive bits enables non-destructive bit-by-bit arbitration in the event of a collision.

The priority of a message is indicated by the value of the identifier: **The identifier with the lowest value has priority.**

Communication model: **Producer/Consumer**

An identifier coded on 11 bits and located at the start of the message informs the receivers about the type of data contained in each message. Each receiver decides whether or not to accept the data.

This concept permits **multiple communication models**:

Transmission on change of state, cyclically, on Strobe signal, via Master_Slave system.
Max. size of useful data: **8 bytes per frame**

Fragmentation possible if more than 8 bytes

**Transmission security:**

One of the best local industrial networks.
Numerous signalling and error detection devices ensure high transmission security.
3 types of standard services:

1. **Network administration**: Parameter settings, start-up, monitoring (master-slaves)

2. Transmission of **low-volume process data** in real time: 
   - **I/O messages**
   I/O messages can be transmitted on changes of state, cyclically, on receipt of the Strobe message or via master polling, etc.

3. Transmission of high-volume **parameter data** (> 8 bytes) by segmentation without time restrictions: **Explicit messages** in client/server mode.
### Allocation of Identifiers

<table>
<thead>
<tr>
<th>IDENTIFIER BITS</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 9 8 7 6 5 4 3 2 1 0</td>
<td><strong>Group 1 Messages</strong></td>
</tr>
<tr>
<td>0 0 1 1</td>
<td><strong>Source MAC ID</strong></td>
</tr>
<tr>
<td>0 0 1 1</td>
<td><strong>Source MAC ID</strong></td>
</tr>
<tr>
<td>0 0 1 1</td>
<td><strong>Source MAC ID</strong></td>
</tr>
<tr>
<td><strong>Group 2 Messages</strong></td>
<td></td>
</tr>
<tr>
<td>1 0 0</td>
<td><strong>MAC ID</strong></td>
</tr>
<tr>
<td>1 0 0</td>
<td><strong>Source MAC ID</strong></td>
</tr>
<tr>
<td>1 0 0</td>
<td><strong>Source MAC ID</strong></td>
</tr>
<tr>
<td>1 0 0</td>
<td><strong>Source MAC ID</strong></td>
</tr>
<tr>
<td>1 0 1</td>
<td><strong>Destination MAC ID</strong></td>
</tr>
<tr>
<td>1 0 1</td>
<td><strong>Destination MAC ID</strong></td>
</tr>
<tr>
<td>1 0 1</td>
<td><strong>Destination MAC ID</strong></td>
</tr>
<tr>
<td>1 0 1</td>
<td><strong>Destination MAC ID</strong></td>
</tr>
</tbody>
</table>
DeviceNet uses **object type modelling** for:

- The list of available communication services
- Device characteristics
- A standard means of describing how to access the internal variables of a product

A **DeviceNet node** is modelled as a collection of objects.
DeviceNet uses a **4-level addressing** method:

- MAC ID
- Class ID
- Instance ID
- Attribute ID

The variables of a node can be accessed via a **path** which comprises:

- Class ID
- Instance ID
- Attribute ID

Diagram:

- MAC ID #1
- MAC ID #2
- MAC ID #3
- MAC ID #4
- MAC ID #4: Object Class #5: Instance #2: Attribute #1

Diagram paths:

- Object Class #5: Instance #1
- Object Class #5: Instance #1
- Object Class #5
- Attribute #1
- Attribute #2
- Instance #2
- Instance #1
- Object Class #7: Instance #1

Section 10: DeviceNet
A DeviceNet profile is defined in an **EDS (Electronic Data Sheet)** file **supplied with the product**.

This file provides a precise description of all the component objects of the product.
Extract from DeviceNet LUFP9 gateway EDS file

$ DeviceNet Manager Generated Electronic Data Sheet
[File]
DescText = "LUFP9 Gateway";
CreateDate = 12-08-98;
CreateTime = 10:31:30;
ModDate = 10-07-2002;
ModTime = 16:39:54;
Revision = 1.02;

[Device]
VendCode = 90;    $ Vendor Code
ProdType = 12;    $ Product Type
ProdCode = 60;    $ Product Code
MajRev = 1;       $ Major Rev
MinRev = 3;       $ Minor Rev
VendName = "Schneider Electric Gateways";
ProdTypeStr = "Communications Adapter";
ProdName = "LUFP9";
Catalog = "LUFP9";

$ Parameter Class Section
[ParamClass]
MaxInst = 29;    $ Max Instances - total # configuration parameters
Descriptor = 0x00; $ Parameter Class Descriptor - No parameters
CfgAssembly = 0x00; $ The config assembly is not supported.

[Params]
$ Polled production
$ Polled consumption

Param1=
0, $ parameter value slot
6, "20 05 24 00 30 64",
0x0002, $ descriptor (Scaling)
8, 1, $ USINT, 4 bytes
"Polled production", $ parameter name
"", $ units string
"", $ scaling links not used
0, 5, 0, $ min, max, default (0)
0, 0, 0, 0, $ mult, div, base, offset scaling
0; $ decimal places

Param2=
0, $ parameter value slot
6, "20 05 24 00 30 65",
0x0002, $ descriptor (Scaling)
8, 1, $ USINT, 4 bytes
"Polled consumption", $ parameter name
"", $ units string
"", $ scaling links not used
0, 5, 0, $ min, max, default (0)
0, 0, 0, 0, $ mult, div, base, offset scaling
0; $ decimal places
Modbus Ethernet TCP/IP

History
Modbus Ethernet TCP/IP and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
The DoD finances a project about "packet switching" in 1960.

- Development of the ARPANET network (IBM) in 1970.
- The INTERNET is launched: TCP/IP developed in current formats in 1980.
- Growth rate 15% in 1987.
- Growth rate 60% in 1996.
- IEEE 802.3 standardization of CSMA/CD networks in 1996.
- The INTERNET grows at a rate of 60% in 1999.

Ethernet principles defined by XEROX in 1975.
First specification of Ethernet by XEROX, DEC and INTEL in 1980.

Modbus Schneider Transparent Factory
http://www.transparentfactory.com/
Ethernet only covers the first 2 layers of the OSI model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Function</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PHYSICAL</td>
<td>Ethernet V2 or 802.3</td>
</tr>
<tr>
<td>2</td>
<td>LINK = LLC + MAC</td>
<td>CSMA/CD</td>
</tr>
<tr>
<td>3</td>
<td>NETWORK</td>
<td>IP</td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
<td>TCP</td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>6</td>
<td>PRESENTATION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>7</td>
<td>APPLICATION</td>
<td>Modbus, HTTP, FTP, BootP, DHCP</td>
</tr>
</tbody>
</table>
Physical layer

Topology: Free
Bus, star, tree or ring

Maximum distance: Depends on medium and speed
Minimum: 200 m on 100 base TX
Maximum: 40,000 m on 10 base F

Speed: 10 Mbps - 100 Mbps - 1 Gbps
1 Gbps in office automation

Max. no. of devices: Depends on medium
Minimum: 30 per segment on 10 base 2
Maximum: 1024 on 10 base T or 10 base F
### Transmission media

Ethernet is available on three types of medium:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Speed</th>
<th>Max. length</th>
<th>Max. no. of stations/segment</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coaxial cable</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 base 5</td>
<td>Thick Ethernet</td>
<td>10 Mbps</td>
<td>500 m</td>
<td>100</td>
</tr>
<tr>
<td>10 base 2</td>
<td>Thin Ethernet</td>
<td>10 Mbps</td>
<td>185 m</td>
<td>30</td>
</tr>
<tr>
<td><strong>Shielded twisted pair</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 base T</td>
<td>Twisted pair</td>
<td>10 Mbps</td>
<td>100 m</td>
<td>1024</td>
</tr>
<tr>
<td>100 base TX</td>
<td>Twisted pair</td>
<td>100 Mbps</td>
<td>100 m</td>
<td>? ? ?</td>
</tr>
<tr>
<td><strong>Optical fibre</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10 base F</td>
<td>2 fibres</td>
<td>10 Mbps</td>
<td>2000 m</td>
<td>1024</td>
</tr>
<tr>
<td>100 base FX</td>
<td>2 fibres</td>
<td>100 Mbps</td>
<td>2000 m</td>
<td>? ? ?</td>
</tr>
</tbody>
</table>
Section 11: Modbus Ethernet TCP/IP

Twisted pair

Used increasingly, even at 100 Mbps

**UTP** - Insulated pairs of copper wires twisted together
Multiple colour-coded pairs enclosed in a plastic sleeve
Faster than coaxial cable

**STP** - Indivisible pairs enclosed in a shielding with aluminium foil

**Category 5 (Cat 5)** – The most common for IT networks
Cat 5 = 100 Mbps (specification pending)
Cat 3 = 10 Mbps

Uses RJ45 connector
Optical fibres are popular because they are secure (absence of electrical currents), compact and immune to noise and electromagnetic interference.

They support very long segment lengths (max. 2 km).

They are often used as backbones.

Three component parts:

**Core** – Carries the light beam (glass or plastic)

**Cladding** – Glass tube which reflects any interference light in the core

**Coating** – Protects the core and the optical cladding

*Multimode fibre* is the most popular type as it is the least expensive and easier to use.
Transport network link layers

Medium access method: **CSMA/CD**
*Carrier Sense Multiple Access with Collision Detection*
The stations listen to the transmission medium and wait until it is free to send.
If a collision is detected, each station continues to send in order that the collision is seen by the entire network.
The stations resend their message after a random period of time has elapsed.

**Determinism:**  
Resolved using segmentation
Load factor < 10%

**Transmission method:**  
In packets
or IP datagrams, 64 to 1500 bytes

**Max. size of useful data:**  
1442 bytes per packet (APDU)

**Transmission security:**  
**CRC32** at link layer level
**Acknowledgement** at TCP link level
**Response** at application level (UNITE/Modbus)
The major application protocols

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
</table>
| HTTP     | HyperText Transfer Protocol = Web  
File transfer in HTML format |
| FTP      | File Transfer Protocol  
File transfer based on the client/server model |
| SNMP     | Simple Network Management Protocol  
Network management: Configuration, monitoring, administration |
| DNS      | Domain Name Service  
Translates the symbolic name of a network node into an IP address |
Application protocols

**BOOTP:** Bootstrap Protocol
- IP address assignment by a server

**TELNET:** Terminal interfacing with devices in half duplex mode
- Encapsulated ASCII format

**UNITE:** Protocol based on the client/server model created by Telemecanique

**MODBUS:** Protocol based on the client/server model created by Modicon

**I/O scanning:** Period I/O updated by automatic sending of Modbus requests
Implementation classes define a list of services to be implemented in order to ensure the interoperability of Schneider Transparent Ready products.

These classes are defined for 4 device families:

- **Controllers:** PLC, numerical controllers, etc.
- **Devices:** Drives, motor starters, remote I/O
- **Gateways:**
- **HMI/SCADA**

Implementation classes are identified by:

- A letter from A to Z relating to WEB services
- Followed by a number from 00 to 99 relating to user services and communication
- And an ASCII suffix relating to the physical layer.
Section 11: Modbus Ethernet TCP/IP

Implementation classes

A: without Web
B: Web Basic
C: Web Configurable
D: Web Active
E: Web Distributed server

A: without Web
Z: Web Basic
Y: Web Regular
X: Web Active
w: Web Distributed client

01: modbus Basic access
05: modbus Regular access

10: modbus on TCP-IP basic access
20: modbus on TCP-IP management access
30: modbus on TCP-IP added values access
40: distributed control on TCP-IP

Example of Implementation Class:

A05-SL-RS485 Modbus on RS485, no Web
A00-Can for Can Open: profiles to be defined
C30-Eth100 Modbus on Ethernet TCP-IP (100 Mbs) + com & Web services
Section 11: Modbus Ethernet TCP/IP

**Web services**

**Server**
A: without Web
B: Web Basic
C: Web Configurable
D: Web Active
E: Web Distributed

**Web**

**Client**
A: without Web
Z: Web Basic
Y: Web Regular
X: Web Active
W: Web Distributed

- Web level A
- Web level B
- Web level C
- Web level D
- Web level E
- Web level F
- Web level G

- Maintenance: B R E
- Monitoring: B R E
- Diag: B R E
- Doc: B R E
- Conf: B R E

- optional
- mandatory
Section 11: Modbus Ethernet TCP/IP

User and communication services

User & Communication with TCP-IP

Com without TCP-IP

Communication level 01
Communication level 05
Communication level 10
Communication level 20
Communication level 30

Modbus Messaging B R E
IOScan B
FDR B R E
Net Mgt (Modbus) B
Net Mgt (SNMP) B R
Glob Data B R
Bd. Mgt B R

optional
mandatory

Schneider Electric

Centro de Formación de Schneider Electric
Profibus-DP

History
Profibus-DP and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
In 1987, the German federal minister for technological research and development creates a "Fieldbus" working group comprising 13 organizations including SIEMENS and 5 research institutes.  

Profibus (PROcess FIeldBUS) is born.

PROFIBUS is managed by a user group which includes manufacturers, users and researchers: The PROFIBUS CLUB.

User clubs in 20 of the world's most industrialized countries provide support in native languages. These centres of competence are governed by the "PROFIBUS International" (PI) organization, which has more than 750 members.  

http://www.profibus.com/
The 3 versions of Profibus

- Profibus-DP
- Profibus-PA
- Profinet
Profibus and the ISO model

<table>
<thead>
<tr>
<th>7</th>
<th>Application</th>
<th>FMS = Fieldbus message specif.</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Presentation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Link</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
<td></td>
</tr>
</tbody>
</table>

FDL = Fieldbus data link

RS485 or optical fibre

FMS = Fieldbus message specif.

DP functions

DP profiles
Section 12: Profibus-DP

Physical layer

Topology: Bus with active line terminators

Maximum distance: Depends on medium and speed
Minimum: 100 m at 12 Mbps without repeaters
Maximum: 4800 m at 9.6 kbps with 3 repeaters

Speed: 9.6 Kbps to 12 Mbps

Max. no. of stations: 32 without repeaters
124 with 3 repeaters
Section 12: Profibus-DP

Types of connection

IP20
- 9-pin Sub D
- Female, product side
- with or without line terminator

IP65
- M12 connector
  - 4
  - 3
  - 2
  - 1
- Female, product side
- Han-Brid
- DESINA recommendation
Section 12: Profibus-DP

Example architecture

Quantum

ATV58

TEGO POWER

Repeater

Line terminators

FTB1DP

Line terminator

FTB1DP

FTB1DP

Centro de Formación de Schneider Electric
PROFIBUS uses a **hybrid** access method:

- Communication between active stations is based on the **token ring** concept.
- Passive stations (slaves) use the **master-slave** concept.
The **token ring concept** ensures that access to the bus is provided to each master device in a predefined time window.

The token is a special telegram sent by a master which must be distributed to the other masters on the ring in a maximum configurable period of time.

The **master-slave concept** enables the master in possession of the token to access slaves assigned to it (passive stations) as well as other masters (FMS message handling).

Messages destined for slaves and responses to them are called **PPOs: Parameter Process Objects**.

Profibus-DP can operate with a single master (mono-master mode).

The Profibus-DP Premium master module does not support master-to-master communication (FMS).
### Description of PPO

The master sends a **cyclic request** to the slave.

<table>
<thead>
<tr>
<th>1st word</th>
<th>Optional aperiodic exchanges zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKW</td>
<td>PKE</td>
</tr>
<tr>
<td>PWE</td>
<td>PZD</td>
</tr>
<tr>
<td></td>
<td>PZD1</td>
</tr>
<tr>
<td></td>
<td>PZD2</td>
</tr>
<tr>
<td></td>
<td>PZD3</td>
</tr>
<tr>
<td></td>
<td>PZDn</td>
</tr>
</tbody>
</table>

The master receives a **cyclic response** from the slave.

<table>
<thead>
<tr>
<th>1st word</th>
<th>Optional aperiodic exchanges zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>PKW</td>
<td>PKE</td>
</tr>
<tr>
<td>PWE</td>
<td>PZD</td>
</tr>
<tr>
<td></td>
<td>PZD1</td>
</tr>
<tr>
<td></td>
<td>PZD2</td>
</tr>
<tr>
<td></td>
<td>PZD3</td>
</tr>
<tr>
<td></td>
<td>PZDn</td>
</tr>
</tbody>
</table>

Last word

All words are exchanged cyclically although aperiodic exchanges are used when necessary.

- **PKW** = Parameter - Kennung - Wert = Parameter - Address - Value
- **PKE** = Parameter - Kennung = Parameter address
- **PWE** = Parameter - Wert = Value of the parameter whose address is contained in the PKE
- **PZD** = Prozeßdaten = Process data
## Using PKWs

### Output data

<table>
<thead>
<tr>
<th>PKW</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word 1</td>
<td>PKE output</td>
</tr>
<tr>
<td>Word 2</td>
<td>R/W output</td>
</tr>
<tr>
<td>Word 3</td>
<td>0</td>
</tr>
<tr>
<td>Word 4</td>
<td>Output</td>
</tr>
</tbody>
</table>

### Input data

<table>
<thead>
<tr>
<th>PKW</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word 1</td>
<td>PKE input</td>
</tr>
<tr>
<td>Word 2</td>
<td>R/W/N input</td>
</tr>
<tr>
<td>Word 3</td>
<td>0</td>
</tr>
<tr>
<td>Word 4</td>
<td>PWE input</td>
</tr>
</tbody>
</table>

**PKE output:**
- Bits 0 to E: Address of variable
- Bit F: 0 Single read or write
- 1 Continuous read or write

**R/W output:**
- = 16#0052 = Read
- = 16#0057 = Write

**PWE output:**
- = If write: Write value

**PKE input:**
Copy of the PKE output value

**R/W/N input:**
- = 16#0052 Read correct
- = 16#0057 Write correct
- = 16#004E Read or write error

**PWE input:**
- If read correct value of variable
- If write correct copy of PWE output value
- If error
  - = 0: Address incorrect
  - = 1: Write refused
**Application layer**

**Data exchanges:**
- Process: Cyclic exchanges
- Parameters, diagnostics: Aperiodic (PKW)

**Max. size of data:** 244 bytes of PPO

**Interoperability:**
- Product certified by the Profibus organization

**Interchangeability:**
- Communication and application profiles
Three types of station are defined:

**DP master class 1 (DPM1):** Programmable controllers such as PLCs, PCs, etc.

**DP master class 2 (DPM2):** Development or diagnostics tool

**DP slave:** Peripheral device performing cyclic exchanges with "its" active station

The Profibus-DP TSX PBY 100 Premium module is a subset of DPM1
Application profiles complete the standard for a given area of application.

Examples:

- **Numerical controllers and robots**
  Based on sequential diagrams, movements and commands are described from the point of view of the control system.

- **Encoders**
  Based on the connection of rotary, angle and linear encoders, and based on the definition of functions (scaling, diagnostics, etc.).

- **PROFIDRIVE variable speed drives**
  Based on the basic functions of the drive: drive commands and states are described.

- **Process control and supervision (HMI)**
  Specifies how control (and supervision) devices are linked with higher-level control system components. Uses the extended functions of PROFIBUS-DP relating to communication.
The characteristics of a PROFIBUS device are described in the form of an "electronic device data sheet" (GSD) in a predefined format.

GSD files must be provided by all PROFIBUS device manufacturers.

**General specifications**
This section contains information about the manufacturer, the product name, hardware and software versions, speeds supported, etc.

**Specifications relating to masters**
This section contains all the parameters relating to masters, such as the maximum number of slaves and up/downloading options. This section does not exist for slave devices.

**Specifications relating to slaves**
This section contains the specifications relating to slaves, such as the number and type of I/O variables, diagnostic texts, information about modules for modular products, etc.
Section 13: FIPIO

FIPIO

History
FIPIO and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
History

- It all started with a working group managed by the Science and Technology Department of the Ministry for Industry and Research (France) including the manufacturers TELEMECANIQUE, MERLIN GERIN, CGEE, ALSTHOM and CSEE. This group worked on the FIP specification during the years 1983-1985.

- The WorldFIP users and manufacturers group was created in 1987 under the name CLUB FIP.

http://www.worldfip.org/

WorldFIP meets the requirements of standards EN 50170 and IEC 61158.
### FIPIO and the ISO model

#### CAN 2.0 A and B + ISO 11898

<table>
<thead>
<tr>
<th>Level</th>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>APPLICATION</td>
<td>Process data channel + PCP message handling</td>
</tr>
<tr>
<td>6</td>
<td>PRESENTATION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
<td>EMPTY</td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
<td>EMPTY</td>
</tr>
<tr>
<td>3</td>
<td>NETWORK</td>
<td>EMPTY</td>
</tr>
<tr>
<td>2</td>
<td>LINK = LLC + MAC</td>
<td>Master-slave with a single frame (shift register)</td>
</tr>
<tr>
<td>1</td>
<td>PHYSICAL</td>
<td>RS 485</td>
</tr>
</tbody>
</table>
Physical layer

Medium: Shielded twisted pair or optical fibre

Topology: Bus type
With tap link or daisy chain connections + line terminators

Maximum distance:
- 1000 m for an electrical segment
- 3000 m for an optical segment
- 15 000 m with electrical repeaters
No. of repeaters + no. of stations <= 36
No. of repeaters x 0.5 + total length in km < 22

Speed: 1 Mbps
Regardless of cable length

Max. no. of devices: 127
1 master and 126 slaves
Max. 32 devices per segment
Standardized 9-pin SUB D connector

male 9-pin SUB D product side

To FIPIO main cable or tap junction
Section 13: FIPIO

Example architecture

Premium

Momentum I/O

Magelis

TBX IP20

LUFP1

Model U starter-controllers

ATS48

ATV28

Modbus

Line terminator

ASi gateway

Micro

Electrical-optical converters

TBX IP67

Line terminator

ASi

Line terminator

TBX IP20

Line terminator

24 V

Line terminator

24 V

Line terminator

Modbus

ASi

Line terminator

24 V

Line terminator
Medium access method: Master/Slaves (bus arbitrator)
The bus arbitrator derives the list of variables (identifiers) to be scanned as well as their periodicity (data contained in the device profile) from the system configuration.

Communication model:
Periodic exchanges: Producer/Consumer
When the bus arbitrator requests the distribution of a variable (identifier), the unique producer of this variable detects this and distributes the variable.
The consumer station or stations detect the variable and the bus arbitrator moves to the next identifier.

Aperiodic exchanges: Client/Server
Once the periodic exchanges are complete, the bus arbitrator processes the aperiodic requests stored in a separate buffer (list of identifiers).
Each variable is scanned at its own pace without being disturbed by aperiodic exchanges.
3 profile families are defined:

- **FRD** = FIPIO *Reduced* Device Profile
- **FSD** = FIPIO *Standard* Device Profile
- **FED** = FIPIO *Extended* Device Profile

The profile selected depends on:

- The number of cyclic variables to be exchanged
- The number of configuration variables
- The number of adjustment variables
- The number of diagnostic variables
- The structure of the device
### Profile overview

<table>
<thead>
<tr>
<th>Standard profile</th>
<th>FRD</th>
<th>FSD</th>
<th>FED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cyclic variables</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input acquisition</td>
<td>2 words</td>
<td>8 words</td>
<td>32 words</td>
</tr>
<tr>
<td>Output control</td>
<td>2 words</td>
<td>8 words</td>
<td>32 words</td>
</tr>
<tr>
<td>Configuration variables</td>
<td>-</td>
<td>16 words</td>
<td>30 words</td>
</tr>
<tr>
<td>Adjustment variables</td>
<td>-</td>
<td>32 words</td>
<td>30 words</td>
</tr>
<tr>
<td><strong>Commands</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specific command</td>
<td>-</td>
<td>-</td>
<td>8 words</td>
</tr>
<tr>
<td><strong>Diagnostics</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Validity of inputs</td>
<td>1 byte</td>
<td>1 byte</td>
<td>1 byte</td>
</tr>
<tr>
<td>Specific status</td>
<td>-</td>
<td>-</td>
<td>8 words</td>
</tr>
</tbody>
</table>
4 fields are used to identify a profile:

- **Family**
  - FRD Reduced
  - FSD Standard
  - FED Extended

- **Structure**
  - C Compact
  - M Modular

- **No. of I/O**
  - 2 words
  - 8 words
  - 32 words

- **Parameters can/cannot be set**
  - P Parameters can be set
  - - Parameters cannot be set

**Profile denomination syntax**

FSD C8 P
Interbus

History
Interbus and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
History

- Protocol specifications by Phoenix Contact (1983)
- First prototypes (1985)
- First products launched (1987)
- Creation of the international Interbus Club (1990)
- First profiles (1993)
- EN 50254 approval (2001)
  - 350,000 networks installed
  - 4 million nodes connected
  - 2700 products
Section 14: Interbus

Interbus and the ISO model

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>APPLICATION</td>
</tr>
<tr>
<td>6</td>
<td>PRESENTATION</td>
</tr>
<tr>
<td>5</td>
<td>SESSION</td>
</tr>
<tr>
<td>4</td>
<td>TRANSPORT</td>
</tr>
<tr>
<td>3</td>
<td>NETWORK</td>
</tr>
<tr>
<td>2</td>
<td>LINK = LLC + MAC</td>
</tr>
<tr>
<td>1</td>
<td>PHYSICAL</td>
</tr>
</tbody>
</table>

- Process data channel + PCP message handling
- Master-slave with single frame (shift register)
- RS 485

CiA DS-301 = Communication profile

CAN 2.0 A and B + ISO 11898

CAL = CAN Application Layer

PROCESS DATA CHANNEL + PCP MESSAGE HANDLING

DRIVECOM drives
HMI MMI COM
Welding robots
Etc.
Section 14: Interbus

Physical layer

**Medium:**  Shielded twisted double pair
1 pair for receiving, 1 pair for sending

**Topology:**  Ring type
Viewed from the outside, resembles a bus topology with the connecting cable containing the signal loop-back.

**Maximum distance:**  400 m between 2 devices
12.8 km total

**Speed:**  500 Kbps

**Max. no. of devices:**  512
The different types of bus

**Remote bus**
(main bus)
- RS 485 point-to-point
- Max. 256 devices
- Max. 400 m between 2 devices
- Total length: 12.8 km

**Remote bus branch**
Bus terminal module: 170 BNO 671 00 (IP20)

**IP20 bus terminal module**
for local bus

**Local bus TTL**
(designed for a cost-effective installation of a remote sub-station in an enclosure)
- Max. 8 devices
- Max. 1.5 m between 2 devices
- Total length: 10m
- Max. current: 800 mA

**Interbus sensor loop**
(direct connection of digital and analog sensors on Interbus-S via a bus terminal module)
- 1 unshielded pair + 24 V
- Max. 32 devices
- Max. 10 m between 2 devices
- Total length: 100 m

**Installation bus**
(variant of remote bus + sensor power supply voltage)
- RS 485
- With 24 V, 4.5 A max. power supply
- Max. 40 I/O modules
- Max. 50 m between 2 devices
- Total length: 50 m

**IP 65 bus terminal module**
for installation bus
- Regenerates data
- Supplies 24 V/4.5 A
Bus terminal module: 170 ENO 396 00 (IP65)

No Schneider devices on the local bus or "sensor loop"
Section 14: Interbus

Types of connection

**IP20**

- **9-pin Sub D IN**
  - Male, product side

- **9-pin Sub D OUT**
  - Female, product side

**IP65**

- **M12 IN connector**
  - Male, product side

- **M12 OUT connector**
  - Female, product side
Section 14: Interbus

Example architecture

Max. 400 m between each product

ATV50

TEGO POWER

Bus terminal module

Remote bus

IN

OUT

Installation bus

24 V

max. 50 m

Remote bus

max. 400 m

Installation bus

24 V

max. 50 m
Medium access method: Master/Slaves
Transmission of a single frame containing both sensor (input) and actuator (output) data.

This single frame is managed like a shift register with a maximum of 256 words. Each slave (station) is a component of the register.

The frame structure is hybrid, enabling 2 data classes to be supported (maximum 32 words per device):

- **Cyclic process data** (periodic slave I/O words)
- **Acyclic parameter data** (fixed memory space)
Acyclic exchanges

Acyclic data is transmitted using PCP

PCP = Peripherals Communication Protocol

which fragments parameter data.
Interbus profiles define for a product family:

- The recognition of a device by means of its ID code
- The format of command data (outputs) and status words (inputs) exchanged
- The status chart

A new device can only be integrated into the **CMD Tool** network configuration tool by adding it to a database managed by **PHOENIX CONTACT** (no EDS file).
Modbus

History
Modbus and the ISO model
Physical layer
Link layer
Application layer
Profiles
Strengths - Weaknesses
The **MODBUS protocol** is a message handling structure created by **MODICON** in **1979** to connect PLCs to programming tools.

Today, this protocol is mainly used to set up master/client type communications with slaves/servers between intelligent devices.

MODBUS is **independent of the physical layer**.

It can be implemented on **RS232, RS422, or RS485** links as well as on a **wide variety of other media** (e.g.: optical fibre, radio, etc.).
MODBUS on a serial link operating at 1200 to 56 Kbps with a master/slave access method.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
<th>Modbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Application</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Presentation</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Link</td>
<td>Master/Slave</td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
<td>RS485</td>
</tr>
</tbody>
</table>
MODBUS PLUS is a bus operating at 1 Mbps based on a token ring access method which uses the MODBUS message handling structure.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Application</td>
</tr>
<tr>
<td>6</td>
<td>Presentation</td>
</tr>
<tr>
<td>5</td>
<td>Session</td>
</tr>
<tr>
<td>4</td>
<td>Transport</td>
</tr>
<tr>
<td>3</td>
<td>Network</td>
</tr>
<tr>
<td>2</td>
<td>Link</td>
</tr>
<tr>
<td>1</td>
<td>Physical</td>
</tr>
</tbody>
</table>

- **Modbus**
- **802.4 token ring**
- **RS485**
MODBUS Ethernet TCP/IP uses TCP/IP and Ethernet 10 Mbps or 100 Mbps to carry the MODBUS message handling structure.

![Diagram of the MODBUS Ethernet TCP/IP protocol stack](image-url)
### Section 15: Modbus

#### RS485 physical layer

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Medium:</strong></td>
<td>Shielded twisted pair</td>
</tr>
<tr>
<td><strong>Topology:</strong></td>
<td>Bus type</td>
</tr>
<tr>
<td></td>
<td>With tap links and line terminators</td>
</tr>
<tr>
<td><strong>Maximum distance:</strong></td>
<td>1300 m without repeaters</td>
</tr>
<tr>
<td><strong>Speed:</strong></td>
<td>19,200 bps (56 Kbps on some products)</td>
</tr>
<tr>
<td><strong>Max. no. of devices:</strong></td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>1 master and 31 slaves</td>
</tr>
</tbody>
</table>
Connectors recommended by Schneider

TIA/EIA-485/RJ45

- Female, product side

TIA/EIA-485/9-pin SUB-D

- Female, product side

- Male, product side
Link layer

Medium access method: Master/slave

Transmission method: Client/server

Max. size of useful data: 120 PLC words

Transmission security: LRC or CRC
Start and stop delimiters
Parity bit
Continuous flow
There are 2 versions of the MODBUS protocol:

- **ASCII mode**
  Each byte in the frame is sent in 2-character ASCII format.

- **RTU mode**
  Each byte in the frame is sent in 2-character 4-bit hexadecimal format.

The main advantage of RTU mode is that it sends data more quickly.

ASCII mode allows the insertion of a time interval of one second between 2 characters without generating a transmission error.
The structure of a Modbus frame is the same for requests (message from the master to the slave) and responses (message from the slave to the master).

**Modbus RTU**

- **silence**
- **Address**
- **Function**
- **Data**
- **Checksum**
- **silence**

Silence \( \geq 3.5 \) characters

**Modbus ASCII**

- **:**
- **Address**
- **Function**
- **Data**
- **Checksum**
- **CR**
- **LF**

\( 3A \) _Hex_ 0D _Hex_ 0A _Hex_

Centro de Formación de Schneider Electric
### Example frame in RTU mode

- **Function code = 3: Read n words**

#### Request:

<table>
<thead>
<tr>
<th></th>
<th>1 byte</th>
<th>1 byte</th>
<th>2 bytes</th>
<th>2 bytes</th>
<th>2 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Slave address</td>
<td>Funct. code = 3</td>
<td>Address of 1st word</td>
<td>Number of words to read</td>
<td>CRC16</td>
</tr>
</tbody>
</table>

#### Response:

<table>
<thead>
<tr>
<th></th>
<th>1 byte</th>
<th>1 byte</th>
<th>2 bytes</th>
<th>2 bytes</th>
<th>2 bytes</th>
<th>2 bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Slave address</td>
<td>Funct. code = 3</td>
<td>Number of bytes read</td>
<td>Value of 1st word</td>
<td>Value of last word</td>
<td>CRC16</td>
</tr>
</tbody>
</table>
**Implementation classes**

Modbus message handling implementation classes are a subset of the Transparent Ready project which defines a list of services to be implemented in order to ensure the interoperability of Schneider products.

3 classes are defined for the server device family (drives, motor starters, remote I/O, etc.).

The classes correspond to a list of Modbus requests to be supported.

- **Basic:** Access to words and identification
- **Regular:** Basic + bit access + network diagnostics
- **Extended:** Regular + other types of access
## Section 16: Comparison table for the major networks

### Comparison at physical level

<table>
<thead>
<tr>
<th>Medium</th>
<th>ASi</th>
<th>CANopen</th>
<th>DeviceNet</th>
<th>Ethernet TCP/IP Modbus</th>
<th>Profibus-DP</th>
<th>FIPIO</th>
<th>Interbus</th>
<th>Modbus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yellow flat ribbon cable Round unshielded cable Round shielded cable</td>
<td>Shielded twisted pair</td>
<td>Double shielded twisted pairs</td>
<td>Coaxial cable: 10base2 - 10base5 Shielded twisted pair 10baseT - 10baseTX Optical fibre 10baseF - 10baseFX</td>
<td>Shielded twisted pair Optical fibre</td>
<td>Shielded twisted pair Optical fibre</td>
<td>Double shielded twisted pairs</td>
<td>Shielded twisted pairs</td>
</tr>
<tr>
<td>Max. distance without repeaters</td>
<td>100m</td>
<td>Acc. to speed: 25m to 1 Mbps 1km to 10 Kbps</td>
<td>Acc. to speed: 100m to 500Kbps 500m to 125Kbps</td>
<td>Twisted pair 100m Optical fibre 2000m</td>
<td>Acc. to speed: 100m to 12Mbps 1.2km to 10Kbps</td>
<td>1000 m twisted pair 3000 m optical fibre</td>
<td>400m</td>
<td>1300m</td>
</tr>
<tr>
<td>Max. distance with repeaters</td>
<td>300m</td>
<td>Depends on the type of repeater</td>
<td>Depends on the type of repeater</td>
<td>10km optical fibre</td>
<td>400 to 4800m acc. to speed</td>
<td>15km</td>
<td>12.8km</td>
<td>Depends on the type of repeater</td>
</tr>
<tr>
<td>Speed</td>
<td>166 Kbps</td>
<td>9 possible speeds from 10 Kbps to 1 Mbps</td>
<td>125, 250 or 500 Kbps</td>
<td>10/100Mbps</td>
<td>9.6 Kbps to 1 Mbps</td>
<td>1 Mbps</td>
<td>500 Kbps</td>
<td>up to 19200 bps</td>
</tr>
<tr>
<td>Max. number of devices</td>
<td>ASi V1: 1 master + 31 slaves ASi V2: 1 master + 62 slaves</td>
<td>128 1 master and 127 slaves</td>
<td>64 1 master and 63 slaves</td>
<td>64 I/O scanning and Modbus</td>
<td>Mono or Multi-masters 126 devices max</td>
<td>1 manager + 126 devices</td>
<td>512</td>
<td>32 1 master and 31 slaves</td>
</tr>
</tbody>
</table>
## Section 16: Comparison table for the major networks

### Comparison at link and application level

<table>
<thead>
<tr>
<th></th>
<th>ASi</th>
<th>CANopen</th>
<th>DeviceNet</th>
<th>Ethernet TCP/IP Modbus</th>
<th>Profibus-DP</th>
<th>FIPIO</th>
<th>Interbus</th>
<th>Modbus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Medium access method</strong></td>
<td>Master Slaves</td>
<td>CSMA/CA</td>
<td>CSMA/CA</td>
<td>CSMA/CD</td>
<td>Token ring and master/slaves</td>
<td>Bus manager</td>
<td>Master Slaves Single frame</td>
<td>Master Slaves</td>
</tr>
<tr>
<td><strong>Type and size of data exchanged</strong></td>
<td>ASi V1: Cyclic: 4 IN bits 4 OUT bits Acyclic: 4 P bits ASi V2: Cyclic: 4 IN bits 3 OUT bits Acyclic: 3 P bits</td>
<td>Cyclic I/O: PDO 8 IN bytes 8 OUT bytes Acyclic: SDO Param./adjust. &gt;8 bytes due to fractioning of information</td>
<td>Cyclic I/O: I/O messages 8 IN bytes 8 OUT bytes or &gt;8 if fragmentation Acyclic: Explicit messages Param./adjust. &gt;8 bytes due to fractioning of information</td>
<td>Cyclic I/O: I/O scanning 125 IN words 125 OUT words</td>
<td>Cyclic I/O: PZD 244 IN words 244 OUT words Acyclic: Param./adjust. via asynchronous messaging 507 words PKW = 1 word at once</td>
<td>Cyclic I/O: 32 IN words 32 OUT words Acyclic: Param. = 30 words Adjust. = 30 words</td>
<td>Cyclic I/O: 256 I/O words Acyclic: 256 words via fragmentation</td>
<td>Acyclic variables 1920 bits 120 words</td>
</tr>
</tbody>
</table>
## Section 16: A look at the IA communication offer

### PLCs

<table>
<thead>
<tr>
<th></th>
<th>ASi</th>
<th>CANopen</th>
<th>DeviceNet</th>
<th>Ethernet TCP/IP Modbus</th>
<th>Profibus-DP</th>
<th>FIPIO</th>
<th>Interbus</th>
<th>Modbus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zelio</strong></td>
<td></td>
<td></td>
<td>Slave</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Twido</strong></td>
<td>Master V2</td>
<td>2004</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Master or Slave</td>
<td></td>
</tr>
<tr>
<td><strong>Micro</strong></td>
<td>Master V1</td>
<td></td>
<td>Yes</td>
<td></td>
<td>Yes Agent</td>
<td></td>
<td>Master or Slave</td>
<td></td>
</tr>
<tr>
<td><strong>Premium</strong></td>
<td>Master V1, Master V2</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes Manager</td>
<td>Yes</td>
<td>Master or Slave</td>
<td></td>
</tr>
<tr>
<td><strong>Quantum</strong></td>
<td>Master V1</td>
<td>Pending</td>
<td>Yes Third-party module</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Master or Slave</td>
<td></td>
</tr>
</tbody>
</table>
## Industrial control

### Section 16: A look at the IA communication offer

<table>
<thead>
<tr>
<th></th>
<th>ASi</th>
<th>CANopen</th>
<th>DeviceNet</th>
<th>Ethernet TCP/IP Modbus</th>
<th>Profibus-DP</th>
<th>FIPIO</th>
<th>Interbus</th>
<th>Modbus</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Motion control</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEXIUM MHDA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>TEGO Quickfit</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tesys U</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Remote I/O</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASI IP20 and IP 67</td>
<td>Yes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP20 Momentum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advantys IP20</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Advantys IP67</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
<td></td>
<td></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>HMI</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XBT-H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XBT-P</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XBT-E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XBT-F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Drives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATS46</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATV28</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATV58</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ATV68</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Section 18: How PL7 deals with the communication function

Setup procedure

1. Wire the installation
2. Configure the slaves: Address, communication speed, etc.
3. Declare the master module in the PLC
4. Configure the master module
5. Save and transfer the configuration to the PLC

Via switches, rotary switch or console. Some products automatically detect the communication speed and format.

PL7 is the programming software for Micro and Premium PLCs

- With PL7 for ASi, Ethernet, FIPIO and Modbus
- With SycCon for CANopen and Profibus
- With CMD Tool for Interbus
Section 18: How PL7 deals with the communication function

**Setup procedure**

1. **Use debug screen to check communication functioning OK**
2. **Development of application program**
3. **Program test**
Adding a communication module to a PLC increases possible object applications, which can be of 2 types:

**Implicit objects:**
These input or output variables are updated automatically by the PLC CPU and the communication module asynchronously.

**Explicit objects:**
These input or output variables are updated at the request of the user program.

It is also possible to exchange data directly between the application and remote devices using communication functions (Read_var, Write_var, Send_Req, etc.)
Section 18: How PL7 deals with the communication function

**Implicit objects**

- **PLC processor**
  - Zone %IMod
  - Zone %I or %IW
  - Zone %Q or %QW

- **Communication module**
  - Diagnostic data
  - Input memory zone
  - Output memory zone

- **PLC cycle time**

- **Asynchronism**

- **Automatic cyclic exchanges**

- **Network cycle time**

- **Bus**
  - Device 1
  - Device 2
  - Device n
Section 18: How PL7 deals with the communication function

Explicit objects

Device 1
Device 2
Device n

Bus

PLC processor
Zone %Mwxy*

Status parameters
Command parameters
Current adjustment parameters
Initial adjustment parameters

Communication module

Status parameters
Command parameters
Current adjustment parameters

Exchanges activated by the program

Exchanges activated by the module following a prog. request

READ_STS
WRITE_CMD
WRITE_PAR
READ_PAR
SAVE_PAR
RESTORE_PAR

* %Mwxy: Where x = Rack number - y = Communication module slot number
Section 18: How PL7 deals with the communication function

Communication functions

PLC processor

Internal application memory slot %MW parameter in request

Device 1
Device 2
Device n

Communication module

Buffer memory

Bus

Exchanges activated on request*

Exchanges activated by the module following a prog. request

READ_VAR
WRITE_VAR
SEND_REQ

* The request enables a parameter to be set defining the device being addressed and where the data is stored.