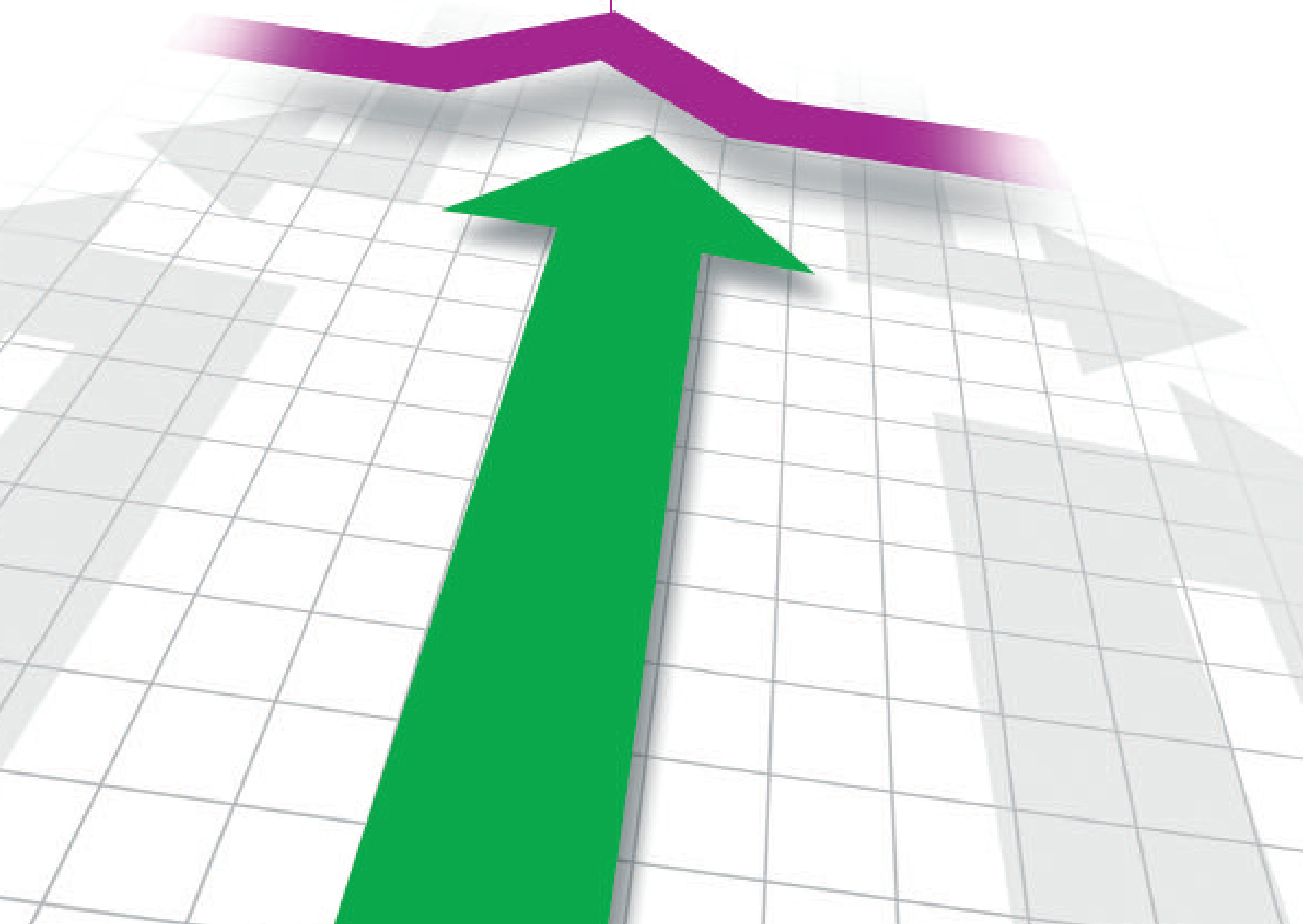
9 chapter
Indutrial networks

Presentation:

- *Needs and available components*
- *Technologies*
- *Schneider Electric policy*



Summary

9. Industrial networks

In this part we discuss the electrical links required for operating automation equipment. These usually involve two categories:

- ***High current** links connecting the power components between the mains supply and the load. We shall not be dealing with this topic here but refer the reader to the sections on power supply and implementation.*
- ***Low current** links connecting all the capture, dialogue, processing and power control components with the machine and process environment.*

9.1 Introduction

Electrical equipment systems are traditionally hard wired.

The international machine standard IEC 60 204-1 and individual country standards have precise stipulations for sections, the quality of the insulating agent and colour markings. Most of these links are made from flexible wire units with a section of 1.5-2.5 mm^2 (AWG 16 and 14), protected at each end.

Until a decade ago, these solutions covered all requirements, both for discrete signals and analogue signals for servocontrol, the latter sometimes requiring shielded cables to prevent electromagnetic interference.

Influenced by IT and automotive industry standards, the advent of digital technologies in other industries has had a considerable impact on the design and construction of electrical equipment.

Digital data exchange entails links by communication networks requiring the use of connectors and ready-made connections. This makes it much simpler to build electrical equipment as wiring errors are reduced and maintenance is more straightforward.

As conventional link technologies are already well known, we shall devote this section to the communication networks used in industry.

9.2 History

In 1968, the company Modicon invented the concept of the programmable logic controller, a single unit to handle a wide range of needs and provide economies of scale. Its high flexibility in use offers many advantages throughout every stage in the lifetime of a plant. Networks came in gradually, initially as serial links. Exchanges were formalised by protocols, such as Modbus (1979, short for MODicon Bus), which has become a standard by its very existence.

Within the last few years, many applications have adopted the field bus. This backbone of automation system architecture is an extremely powerful means of exchange, visibility and flexibility in the devices connected to it. The field bus has gradually led to an overhaul in architecture:

- input/output wires eliminated;
- input/output interfaces superseded or decentralised;
- intelligence decentralised and distributed;
- Internet interconnection.

The 1970s saw the emergence of the Xerox PARC Ethernet – a contraction of ether and net(work) – which 10 years later became the international standard native equipment in practically all computers. Its initial applications were file and message transfer and web page transmission. The spread of information technology to all parts of industry by the 1990's led to the need for industry-wide connection.

The World Wide Web invented by the CERN in 1989 was originally developed to enable different work teams scattered throughout the world share information. The WWW system involves sharing documents and links using HTTP, a simple protocol used by a browser to access web pages stored on a server. These pages are programmed with languages such HTML or XML. The World Wide Web Consortium (W3C) was set up in 1994 to manage technical web developments (see the site http://www.w3.org).

In 1996 Schneider Electric promoted the industrial Ethernet to connect the "management" and "shop floor" sides of businesses by PLC's and then developed the "Transparent Ready" concept based on the addition of industrial tools and protocols, including Modbus, to existing standard Ethernet elements.

9.3 Market requirements and solutions

With the combined effects of user, technological and standards requirements, architectures are now structured into four separate levels interconnected by networks *(⇨ Fig. 1)*.

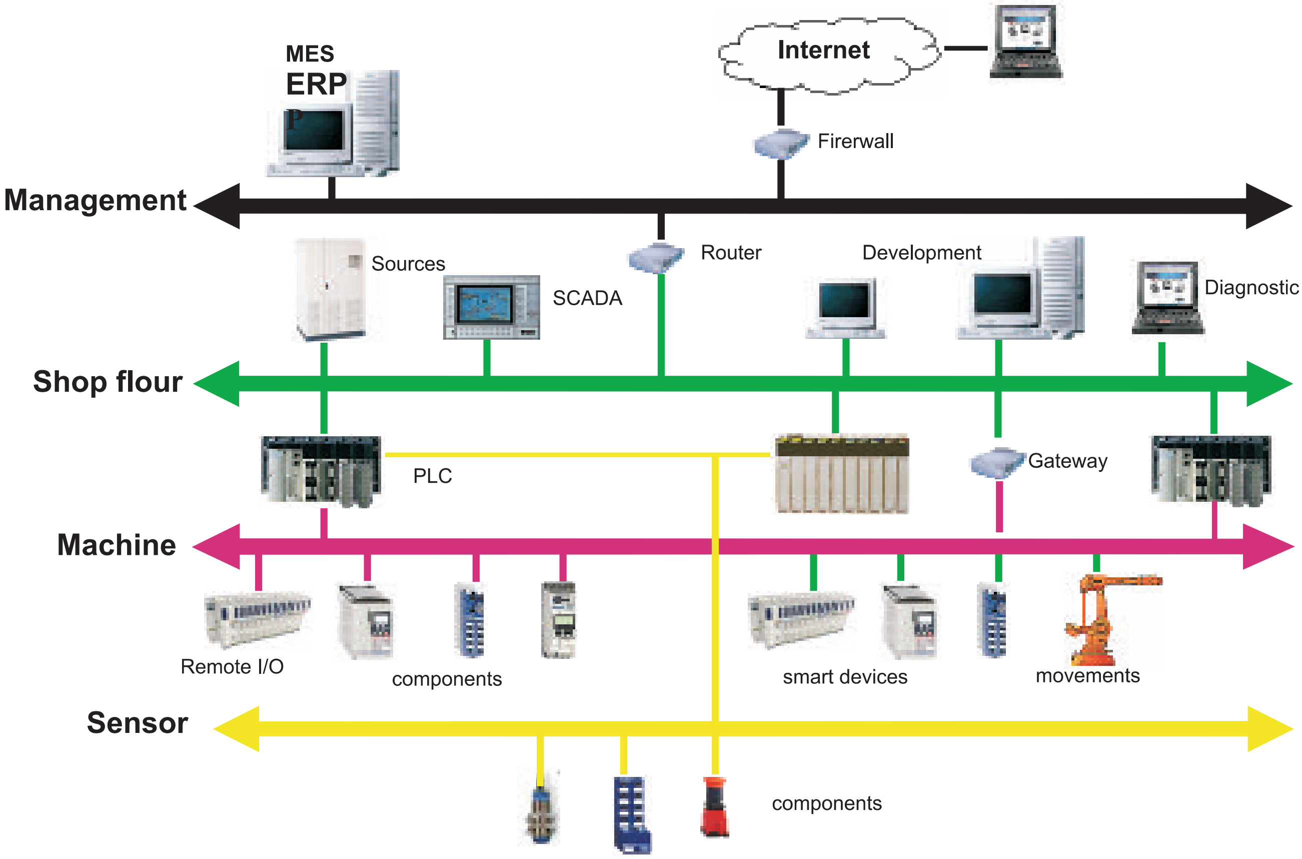


Fig. 1 Example of architecture levels

Before analysing communication network technologies, there should be a breakdown of the main requirements for which these levels provide a relevant solution. The characteristics in the table in *fig. 2* are detailed in the paragraphs which follow.

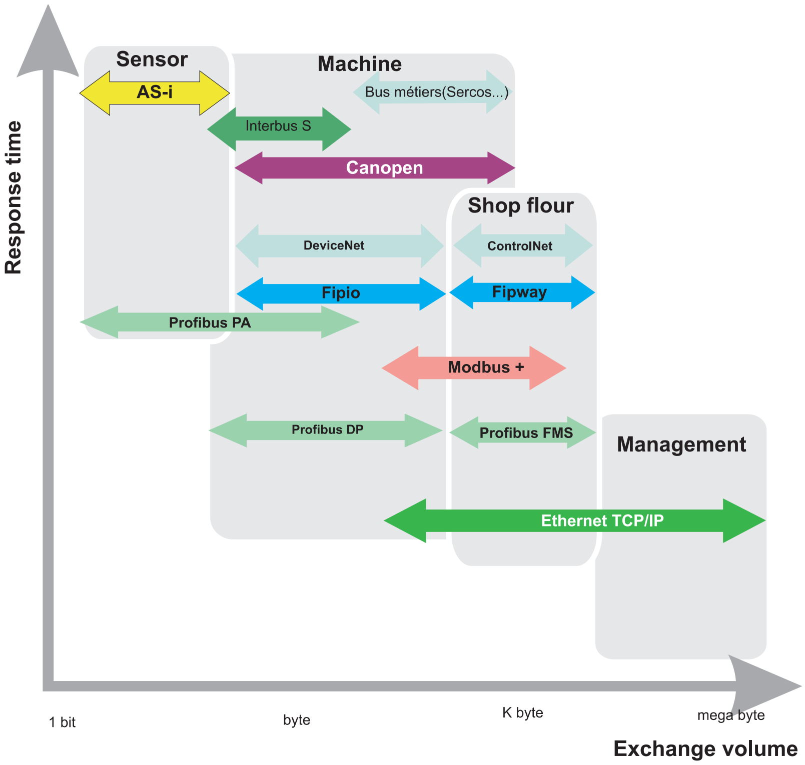
Level	Requirement	Volume to data to transmit	Response time	Distance	Network topology	Number of addresses	Medium
Management	Data exchange. Computer security . Standards between software packages.	Files Mbits	1 min	World	Bus, star	Unlimited	Electrical, optic, radio
Shop floor	Synchronisation of PLC's in the same data exchange automation cell in client/server mode with the control tools (HMI, supervision). Real-time performances.	Data Kbits	50-500 ms	2-40 Km	Bus, star.	10-100	Electrical, optic, radio
Machine	Distributed architecture. Embedded functions and exchange. Transparency. Topology and connection costs.	Data Kbits	5-100 ms (PLC cycle)	10 m to 1K m	Bus, star	10-100	Electrical, optic, radio
Sensor	Simplification of distribution wiring for power supply to sensors and actuators. Optimised wiring costs.	Data Bits		1- 100m	No constraint	10-50	Electrical, Radio

↑ *Fig. 2* *Communication requirements and constraints*

An initial approach is to adopt the two main focuses from this table of requirements:

- the amount of information to transmit;
- the response time needed.

This helps to position the main networks *(⇨ Fig.3)*.



↑ *Fig. 3* *Main industrial networks*

9.4 Network technologies

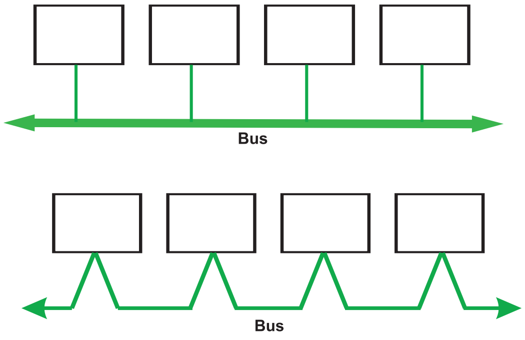
The concepts are described in brief; for further reading, there are many works devoted to this subject.

Network topology

An industrial network is made up of PLC's, human-machine interfaces, computers and I/O devices linked together by communication links such as electric cables, optic fibres, radio links and interface elements such as network cards and gateways. The physical layout of a network is the hardware topology or network architecture.

For the circulation of information the term used is **software topology.**

Topologies are usually divided as follows:
- bus,
- star,
- tree,
- ring,
- hub.



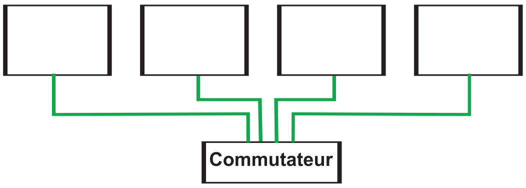
↑ *Fig. 4* *Network topology*

• Bus topology

This is one of the simplest layouts; all the elements are wired together along the same transmission line. The word bus refers to the physical line. This topology is easily implemented and the failure of a node or element does not prevent the other devices from working.

Machine and sensor level networks, otherwise known as field buses, use this system.

The bus topology is implemented by linking devices together in a chain or to the main cable via a connection box (TAP) *(⇨ Fig. 4)*.

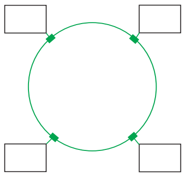
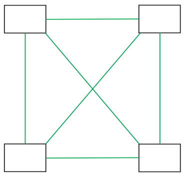


↑ *Fig. 5* *Star network topology*

• Star topology

This is the Ethernet topology, the most common at management and shop floor levels *(⇨ Fig. 5)*. It has the advantage of being very flexible to run and repair. The end stations are linked together via an intermediate device (repeater, switch). Failure of a node does not prevent the network as a whole from working, though the intermediate device linking the nodes together is a point of weakness.

• Other topologies *(⇨ Fig. 6)*
- **The ring topology** uses the same hardware layout as the star topology but ensures greater network availability.
- **The hub topology** is not very widespread in industry and has the disadvantage of a large number of links.

Ring	Hub
	
The ring topology uses the same hardware layout as the star topology but ensures greater network availability.	The hub topology is not very widespread in industry and has the disadvantage of a large number of links.

↑ *Fig. 6* *Other topologies network*

Protocol

A communication protocol specifies a set of rules for a given type of communication. Initially, protocol was the word meaning what was used to make dissimilar devices communicate on the same level of abstraction. The term now extends to the rules of communication between two layers on the same device.

The OSI (Open System Interconnection) model was created by ISO (International Standards Organisation) which published standard ISO 7498 to provide a common basis for all computer network descriptions. In this model, the suite of protocols in a network is divided into 7 parts called OSI layers, numbered 1 to 7. OSI layers work on the following principles:

- every layer supports a protocol independently of the other layers;
- every layer provides services to the layer immediately above it;
- every layer requires the services of the layer immediately below it;
- layer 1 describes the communication medium;
- layer 7 provides services to the user or an application.

In a communication, the network user calls on the services of layer 7 via a program. This layer formats and enriches the data the program gives it according to its protocol and sends it to the layer below it when a service is requested. Each layer formats the data and adds to it according to the protocols used. Finally it is sent to the medium and received by another network node. It goes back through all the layers of this node and ends up in the correspondent's program, divested of all the protocol-related additions.

The OSI 7-layer model *(⇨ Fig. 7)* has been implemented by several manufacturers but was never a commercial success as the market preferred the 4-layer TCP/IP model which is easier to understand and use and which had already been implemented in the mobile domain. The model does however have a certain theoretical advantage, even though the frontiers of the 4 TCP/IP layers do not have an exact equivalent in OSI. These layers will be described in the subsection on Ethernet.

N°	OSI layer	Function of layer	Examples
7	**Application**	The interface with the user; sends requests to the presentation layer.	HTTP, SMTP,POP3, FTP, Modbus.
6	**Presentation**	Defines how data will be represented. Converts data to ensure that all systems can interpret it.	HTML, XML.
5	**Session**	Ensures correct communication and links between systems. Defines session opening on network devices.	ISO8327, RPC, Netbios.
4	**Transport**	Manages end-to-end communication, data segmentation and reassembly, controls flow, error detection and repair.	TCP, UDP, RTP, SPX, ATP.
3	**Network**	Routes data packets (datagrams) through the network.	IP, ICMP, IPX, WDS.
2	**Data-link**	Creates an error-free link from the hard medium.	ARCnet, PPP, Ethernet, Token ring.
1	**Physical**	Defines the protocols for the bit stream and its electrical, mechanical and functional access to the network.	CSMA, RS-232, 10 Base-T, ADSL.

↑ *Fig. 7* *OSI layers*

Frame

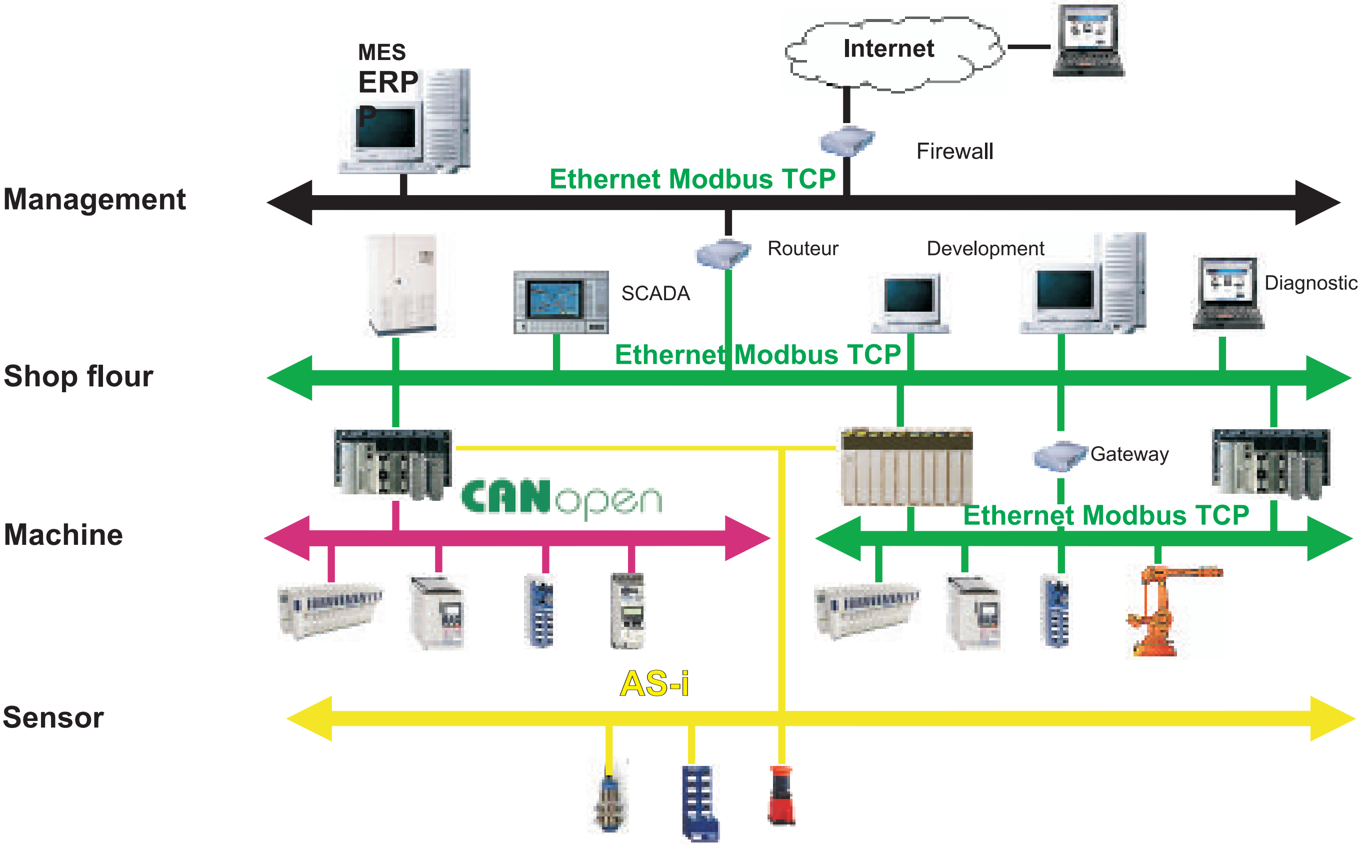
A frame (⇨ Fig.8) is a set of data sent via a network in a single block. It is also known as a packet. Every frame has the same basic layout and contains control information such as synchronisation characters, workstation addresses, an error control value and a variable amount of data.

Header	Frame delimiter	Target address	Source address	Data size	Data	Frame sequence control

↑ Fig. 8 *Format of a frame*

9.5 Networks recommended by Schneider Electric

To answer all requirements with a rational offer, the company has selected three communication networks (⇨ Fig.9) to implement the preferred systems described in the introduction to this document.



↑ Fig. 9 *Communication levels chosen by Schneider Electric*

Ethernet Modbus TCP

The widespread use of Ethernet in business and on the Internet has made it a more or less mandatory communication standard. It helps to cut connection costs and enhance performance, reliability and functionality. Its speed does not slow down applications and its architecture makes upgrading easy. Products and software are compatible, so systems are durable. The "Modbus" protocol, standard usage in industry, provides a simple cost effective application layer.

CANopen

CANopen is the industrial version of the CAN bus developed for automotive purposes. This network has proved its flexibility and reliability for over 10 years in a wide range of applications such as medical equipment, trains, lifts and many machines and plant installations. Schneider Electric's choice of this network is upheld by its widespread distribution.

As-Interface

Modern machines have a great many actuators and sensors and often have safety constraints as well. AS-Interface is the network at sensor level which meets industrial automation requirements. It has the advantage of fast connections and a single cable to convey data and power.

9.6 Ethernet TCP/IP

General description

Ethernet works on the principle of media access controlled by a collision detection mechanism. Each station is identified by a unique key, or MAC address, to ensure that every computer on an Ethernet network has a different address. This technology known as Carrier Sense Multiple Access with Collision Detection (CSMA/CD) ensures that only one station can transmit a message on the medium at a time.

Successive Ethernet upgrades have given rise to the IEEE 802.3 standard (see www.ieee.org) which only defines the characteristics of the physical layers; the way the data accesses the network and the data frame must be defined by further layers. As these notions often get confused, *figure 10* places them and the protocols mentioned are explained in the following paragraphs.

For many years, Ethernet was present in industry but had little success. Suppliers and customers felt it was non-deterministic. Their need for real-time control made them prefer proprietary networks. It was the combination of industry and Internet protocols that finally led them to accept it.

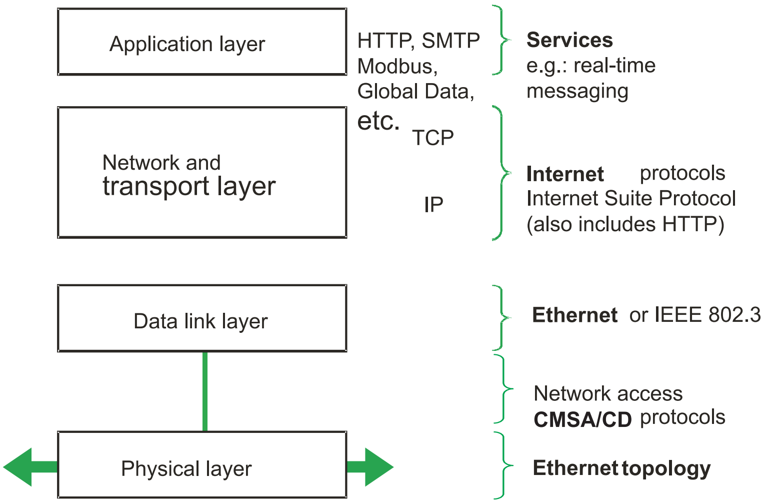


Fig. 10 *Ethernet topology*

Physical layer

The physical layer describes the physical characteristics of communication such as the type of medium conventionally used (electric cables, fibre optic or radio links) and all related details like connectors, types of encoding and modulation, signal levels, wavelengths, synchronisation and maximum distances.

Data link layer

The data link layer specifies media access control and how the data packets are conveyed on the physical layer, in particular the frame structure (i.e. the specific sequences of bits at the start and end of the packets). For example, Ethernet frame headers contain fields indicating which machine on the network a packet is to go to.

Network layer

In its original definition, the network layer solves the problem of conveying data packets across a single network. Further functions were added to it when networks became interconnecting, especially data transmission from a source network to a target one. In general this means that packets are routed across a network of networks, otherwise known as Internet.

In the suite of Internet protocols, IP transmits packets from a source to a target anywhere in the world. IP routing is made available by defining an IP addressing principle to ensure and enforce the uniqueness of every IP address. Each station is identified by its own IP address. The IP protocol also includes other protocols, such as ICMP used for transferring IP transmission error messages and IGMP which manages multicast data. ICMP and IGMP are located above IP but join in the functions of the network layer, thereby illustrating the incompatibility of the Internet and OSI models.

The IP network layer can transfer data for many higher level protocols.

Transport layer

The transport layer protocols can solve problems such as the reliability of data exchange ("Did the data reach the target?"), automatic adaptation to network capacity and data stream control. It also ensures that the data arrive in the right order. In the suite of TCP/IP protocols, transport protocols determine which application each data packet is to be delivered to.

TCP is a connection-oriented transport protocol which delivers a reliable stream of bytes ensuring the data arrive unaltered and in order, with retransmission in the event of loss and elimination of duplicate data. It also handles "urgent" data to be processed in random order (even though they are not technically emitted out of band). TCP tries to deliver all the data correctly and in order – this is its purpose and main advantage over UDP, even though it can be a disadvantage for real-time transfer applications, with high loss rates in the network layer. UDP is a simple, connection-free, "unreliable" protocol. This does not mean it is actually unreliable, but that it does not check that the packets have reached their target and does not guarantee they arrive in order. If an application requires these guarantees, it has to ensure them itself, or else use TCP. UDP is usually used for broadcasting applications such as Global Data or multimedia applications (audio, video, etc.) where there is not enough time for managing retransmission and packet ordering by TCP, or for applications based on simple question/answer mechanism like SNMP queries, where the higher cost of making a reliable connection is disproportionate to needs.

TCP and UDP are used for many applications. Those that use TCP or UDP services are distinguished by their port number. Modbus TCP uses TCP services. UDP can be used for the Factorycast plug-in.

■ Application layer

Most network application functions are located in the application layer.

These include HTTP (World Wide Web), FTP (file transfer), SMTP (messaging), SSH (secured remote connection), DNS (matching IP names and addresses) and many others.

The applications generally work below TCP or UDP and are usually linked to a well-known port. Examples:

- HTTP port TCP 80 or 8080;
- Modbus port 502;
- SMTP port 25;
- FTP port 20/21.

These ports are allocated by the Internet Assigned Numbers Authority.

□ The HTTP protocol (HyperText Transfer Protocol)

It is used to transfer web pages between a server and a browser. HTTP has been used on the web since 1990.

Web servers embedded in Transparent Ready automation devices provide easy access to products anywhere in the world via an Internet browser such as Internet Explorer, Netscape Navigator or others.

□ BOOTP/DHCP

It automatically provides product IP address settings. This avoids having to find the individual address of each device by offloading the task onto a dedicated IP address server.

The DHCP protocol (Dynamic Host Configuration Protocol) automatically allocates device configuration parameters. DHCP is an extension of BOOTP. The BOOTP/DHCP protocol has 2 components:

- the server to provide the IP network address;
- the client which requests the IP address.

The Schneider Electric devices can be:

- BOOTP/DHCP clients which automatically retrieve the IP address from a server;
- BOOTP/DHCP servers for the device to distribute the IP addresses to network stations.

The standard BOOTP/DHCP protocols are used to provide the faulty device replacement service (FDR).

□ File Transfer Protocol (FTP)

It provides the basic means for file transfer. FTP is used by many systems to exchange files between devices.

□ TFTP: Trivial File Transfer Protocol

It is a protocol to simplify file transfer and download codes to devices. For example, it can be used to transfer the boot code in a workstation without a drive unit to connect and download firmware updates for network devices. Transparent Ready devices implement FTP and TFTP to transfer certain data between devices.

□ **NTP (Network Time Protocol)**

It is used to synchronise the time on devices (client or server) via a provider server. Depending on the network used, it provides universal time (UTC) with a precision of a few milliseconds on a local area network (LAN) to several dozen milliseconds on a wide area network (WAN).

□ **SMTP (Simple Mail Transfer Protocol)**

It provides an e-mail transmission service. It is used to send e-mails from a sender to a recipient via an SMTP server.

□ **SNMP (Simple network management protocol)**

The Internet community developed this standard to manage different network components via a single system. The network management system can exchange data with SNMP agent devices. This function enables the manager to view the status of the network and devices, alter their configuration and return alarms in the event of a fault. Transparent Ready devices are SNMP-compatible and can integrate naturally into a network administered via SNMP.

□ **COM/DCOM (Distributed Component Object Model) or OLE (Object Linking and Embedding)**

It is the name of the Windows object component technology used for transparent communication between Windows applications. These technologies are used in OFS data server software (OLE for Process Control Factory Server).

9.7 Web services and Transparent Ready

As already explained, as universal services are not suited to industrial usage, component manufacturers have completed the Internet universal service offer with specific functions for automation systems.

Schneider Electric has developed an offer for “transparent” communication between the web and all the levels described above, defining it as web technology embedded in products and services. This offer has a dual basis:

- Industrial Ethernet;
- WEB components.

The aim is to offer **"Services"** with functions enabling the customer to perform specific tasks such as sending data from one PLC to another or trigger an alarm.

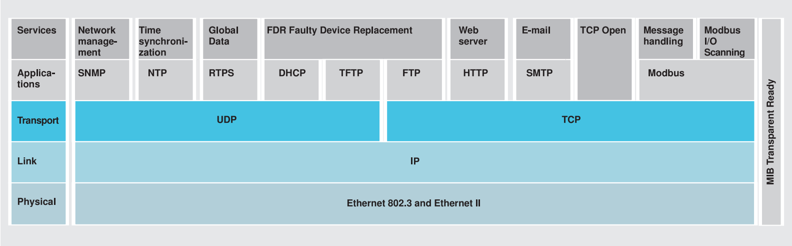
"Web technology" means the same as "Internet technology" and comprises: Internet protocols, programming languages such as Java, html, xml, etc. and the tools which have completely changed the ways of sharing information.

Industrial Ethernet services

In addition to universal Ethernet services (HTTP, BOOTP/DHCP, FTP, etc.), eight other types of Ethernet communication services can be provided with:

- Modbus TCP messaging service;
- remote I/O exchange service: I/O Scanning;
- faulty device replacement service: FDR;
- network administration service: SNMP;
- global Data distribution service;
- bandwidth management service;
- time synchronisation service: NTP;
- event notification service: SMTP (e-mail).

Table 11 shows the position of these services in relation to the layers on the network.



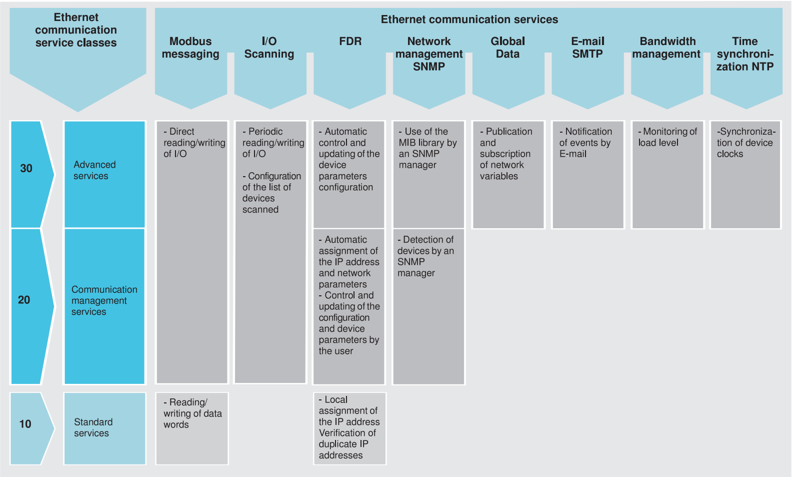
Services	Network management	Time synchronization	Global Data	FDR Faulty Device Replacement			Web server	E-mail	TCP Open	Message handling	Modbus I/O Scanning	MIB Transparent Ready
Applications	SNMP	NTP	RTPS	DHCP	TFTP	FTP	HTTP	SMTP		Modbus		
Transport	UDP					TCP						
Link	IP											
Physical	Ethernet 802.3 and Ethernet II											

Fig. 11 *Position of Ethernet communication services*

These communication services are divided into three classes:

- Class 10: basic Ethernet communication;
- Class 20: Ethernet communication management (network and device levels);
- Class 30: advanced Ethernet communication.

Table 12 gives a brief summary of the services.



Ethernet communication service classes		Ethernet communication services							
		Modbus messaging	I/O Scanning	FDR	Network management SNMP	Global Data	E-mail SMTP	Bandwidth management	Time synchronization NTP
30	Advanced services	- Direct reading/writing of I/O	- Periodic reading/writing of I/O - Configuration of the list of devices scanned	- Automatic control and updating of the device parameters configuration	- Use of the MIB library by an SNMP manager	- Publication and subscription of network variables	- Notification of events by E-mail	- Monitoring of load level	-Synchronization of device clocks
20	Communication management services			- Automatic assignment of the IP address and network parameters - Control and updating of the configuration and device parameters by the user	- Detection of devices by an SNMP manager				
10	Standard services	- Reading/ writing of data words		- Local assignment of the IP address Verification of duplicate IP addresses					

Fig. 12 *Summary of Ethernet services*

Messaging service: Ethernet Modbus TCP

Modbus, the industrial communication standard since 1979, has been applied to Ethernet TCP/IP to make Ethernet Modbus TCP, a fully open Ethernet protocol. Developing an Ethernet Modbus TCP connection does not require any proprietary component or licence purchase. The protocol can be applied to any device that supports a standard TCP/IP communication stack. Specifications are available free of charge from the website: www.modbus-ida.org.

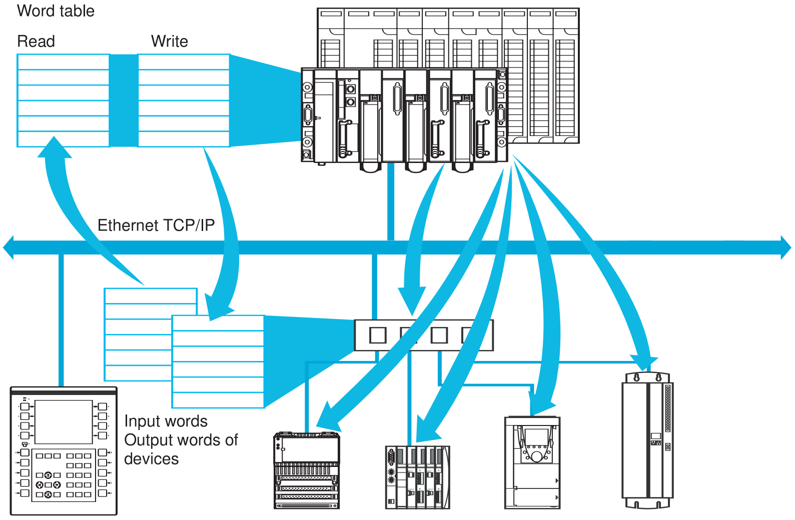
Its simplicity enables any field device, such as an I/O module, to communicate via Ethernet without requiring a powerful microprocessor or a lot of internal memory. Ethernet Modbus TCP has a very simple protocol and high output of 100 Mbps which guarantee its excellent performance enabling this type of network to be used for real-time applications such as I/O scanning.

As the application protocol is identical on Modbus serial link, Modbus Plus and Ethernet Modbus TCP, messages can be routed from one network to another without having to change protocols. Modbus is implemented above the TCP/IP layer, so users also benefit from IP routing which enables devices anywhere in the world to communicate regardless of the distance between them.

IANA (Internet Assigned Numbers Authority) has assigned the Ethernet Modbus TCP with the fixed port TCP 502, thus making Modbus an Internet group standard. The maximum data size is 125 words or registers in read mode and 100 words or registers in write mode.

Remote I/O exchange service: I/O Scanning

This service is used to manage status exchange between remote I/Os via Ethernet. After simple configuration with no specific programming, I/Os are transparently scanned by read/write queries using the Ethernet Modbus TCP client/server protocol. This scanning method via a standard protocol is used to communicate with any device that supports Ethernet Modbus TCP. The service offers definition of two word zones, one to read inputs and the other to write outputs *(⇨ Fig.13)*. The refresh periods are independent of the PLC cycle.



↑ Fig. 13 *Remote I/O exchange service: I/O Scanning*

In operation, the module ensures:

- management of TCP/IP connection IP with each remote device;
- product scanning and I/O copying in the configured word zone;
- feedback of status works to monitor service operation from the PLC application:
- use of preconfigured default values in the event of communication problems.

An offer for hardware and software to implement the I/O Scanning protocol on any device that can be connected to Ethernet Modbus TCP can be found on the Modbus-IDA website (www.modbus-ida.org).

Faulty Device Replacement service (FDR)

The faulty device replacement service uses standard address management technology (BOOTP, DHCP) and the FTP or TFTP (Trivial File Transfer Protocol) file management service. This facilitates maintenance of devices connected to Ethernet Modbus TCP.

It replaces a faulty device by a new device and ensures its detection, reconfiguration and automatic restart by the system. The main steps in replacement are:

- a device using the FDR service has a fault;
- a similar product is taken from the maintenance stock, preconfigured with the device name of the faulty device and reinstalled on the network. Depending on the device, it can be addressed with rotary selectors (e.g. Advantys STB distributed I/Os or Advantys OTB) or with the device's integrated keyboard (e.g. Altivar 71 speed controller);
- the FDR server detects the new device, assigns an IP address to it and transfers the configuration parameters;
- the substitute device checks that all the parameters are compatible with its own characteristics and switches to operation mode.

Network administration service: SNMP

SNMP (Simple Network Management Protocol) monitors and controls all the Ethernet architecture components from a network management workstation to make a quick diagnostic of problems that arise. It is used to:

- interrogate network components such as computers, routers, switches, bridges and terminal devices to view their status;
- obtain statistics on the network the devices are connected to.

This network management software uses the traditional client/server model. However, to prevent confusion with other communication protocols using the same terminology, it is referred to as a network manager or SNMP agent.

Transparent Ready devices can be managed by any SNMP agent, including HP Openview, IBM Netview and, of course, the Transparent Ready ConnexView network management tool. The standard SNMP protocol (Simple Network Management Protocol) provides access to the configuration and management object in the device MIB's (Management Information Bases). MIB's must comply with certain standards to make them accessible for all management tools, though depending on the complexity of the devices, manufacturers may add some objects to the private MIB. The Transparent Ready private MIB has specific management objects for Transparent Ready communication services such as Modbus, Global Data, FDR, etc. These objects facilitate device installation, implementation and maintenance.

Transparent Ready devices support 2 SNMP network management levels:

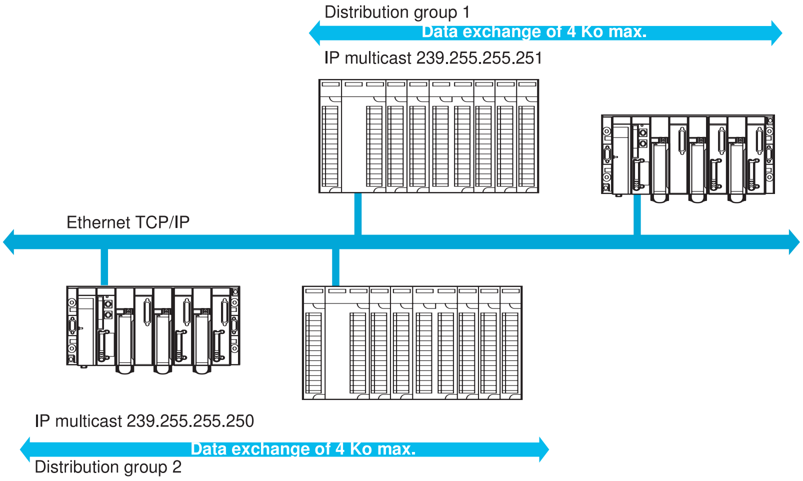
- MIB II Standard interface: a basic network management level is accessible via this interface. The manager uses it to identify architecture component devices and retrieve general information on the configuration and operation of Ethernet TCP/IP interfaces;
- Transparent Ready MIB interface: this interface enhances Transparent Ready device management. The MIB has a set of information enabling the network management system to supervise all the Transparent Ready services. It can be downloaded from the FTP server of any Transparent Ready Ethernet module on a PLC.

■ Global Data distribution service (⇨ *Fig.14)*

The Global Data service ensures multicast data distribution in real time between stations in the same distribution group. It can synchronise remote applications or share a common database amongst distributed applications. Exchanges are based on a standard Publisher/Subscriber protocol guaranteeing optimal performance with a minimum network load. The RTPS protocol (Real Time Publisher Subscriber) is promoted by Modbus-IDA (Interface for Distributed Automation) and is already a standard adopted by several manufacturers. 64 stations can take part in exchanges via Global Data within the same distribution group. Each station can:

- publish a variable of 1024 bytes. The publishing period can be configured for 1 to n periods of the processor master task;
- subscribe from 1 to 64 variables.

The validity of each variable is controlled by Health Status bits linked to a refresh timeout configurable from 50 ms to 1 s. Access to a variable element is not possible. The total size of subscribed variables reaches 4 contiguous Kbytes. To optimise Ethernet performance even further, Global Data can be configured with the multicast filtering option which, combined with the switches in the ConneXium range, multicasts data only on the Ethernet ports with a station subscribing to the Global Data service. If the switches are not used, Global Data are multicast on all the switch ports.


Distribution group 1
Data exchange of 4 Ko max.
IP multicast 239.255.255.251
Ethernet TCP/IP
IP multicast 239.255.255.250
Data exchange of 4 Ko max.
Distribution group 2

↑ *Fig. 14* *Global Data distribution service*

■ NTP time synchronisation service

The time synchronisation service is based on NTP (Network Time Protocol) to synchronise Ethernet TCP/IP client or server time from a server or any other time reference source (radio, satellite, etc.).

The Ethernet Modbus TCP communication modules: – 140 NOE 771 11 on the Modicon Quantum Unity V2.0 (or higher) automation platforms; TSX ETY 5103 on the Modicon Premium Unity V2.0 (or higher) automation platforms – have an NTP client component. These modules can connect to an NTP server using a client query (unicast) to set their local time. Every so often (1 to 120 seconds), the module clock is updated with an error of less than 10 ms for regular processors and 5 ms for high-performance processors. If the NTP server cannot be contacted, the Ethernet Modbus TCP module uses a standby NTP server.

■ SMTP e-mail notification service

This simple e-mail notification service can be programmed. The PLC application uses it to notify an event with conditions. The PLC creates the e-mail automatically and dynamically to alert a defined local- or remote-connected recipient. It should be noted that this service is available with the latest Ethernet communication modules for Modicon Premium and Modicon Quantum PLC's, and with the latest processors with Ethernet connection on the same PLC's used with Unity Pro software. There is also a more complete service independent of the PLC application available with the active Web server module FactoryCast HMI.

The mechanism is simple and effective: predefined message headers are linked to the e-mail body which is created dynamically from the latest information from the PLC application. The PLC application prepares the message according to preset conditions. A function block is used to select one of the 3 predefined headers, create the e-mail with the variables and text (up to 240 bytes) and send it directly from the PLC. The three headers each contain the following predefined elements:

- list of e-mail recipients;
- name of sender and subject.

This information is defined and updated by an authorised administrator using configuration web pages.

■ Web services *(⇨ Fig.15)*

The level of a Web Server service is defined by 4 service classes identified by a letter:

□ Class A

Transparent Ready devices with no web services.

□ Class B

Basic web level for managing static web pages pre-configured in a Transparent Ready device. It offers device diagnostic and monitoring services using a standard web browser.

□ Class C

Configurable web level for customising the website of a Transparent Ready device with web pages defined by the user for the needs of an application. The client procedure diagnostic and monitoring can be run from a standard web browser. The Factorycast offer includes this level of web functionality as well as tools to facilitate management and modification of embedded websites:

□ Class D

Active web level for running specific processes in the Transparent Ready Web Server device itself. This processing capacity can be used for pre-calculation, real-time database management, communication with relational databases and sending e-mails. Communication between the browser and the server is thus reduced and optimised. The Factorycast offer includes this level of web functionality as well as tools to configure processes to run in the Web Server device.

Web server class		Web services			
		Maintenance	Monitoring and IT link	Diagnostics	Optional
D	Active Web server	- User website update	- Autonomous execution of specific services (e.g. alarm notification by E-mail, exchange with databases, calculations, ...) - SOAP/XML (client/server)	- User-defined states	- User documentation
C	Configurable Web server		- PLC variables editor - Remote commands - User Web pages - SOAP/XML (server)	- Communication service diagnostics - State of internal device resources	
B	Standard Web server	- Remote device software update - Remote auto-tests	- Device description - Data viewer	- Device status - Device diagnostic	- Configuration of network parameters and Ethernet communication services - Device documentation
A	No Web server	- No Web service			

↑ *Fig. 15* *Web services*

□ Transparent Ready devices

These are identified by a letter defining the level of web services followed by a number defining the level of Ethernet communication service. E.g.:

- Class A10: a device with no web service and with basic Ethernet services;
- Class C30: a device with a configurable Web Server and advanced Ethernet communication services.

The services offered by a higher class include all those supported by a lower one. The range of Transparent Ready devices is divided into 4 major families:

- field devices (simple or intelligent) like sensors and pre-actuators.
- controllers and PLC's;
- HMI (Human/Machine Interface) applications;
- dedicated gateways and servers.

The selection table in *figure 16* can be used to choose Transparent Ready devices according to the requisite service classes.

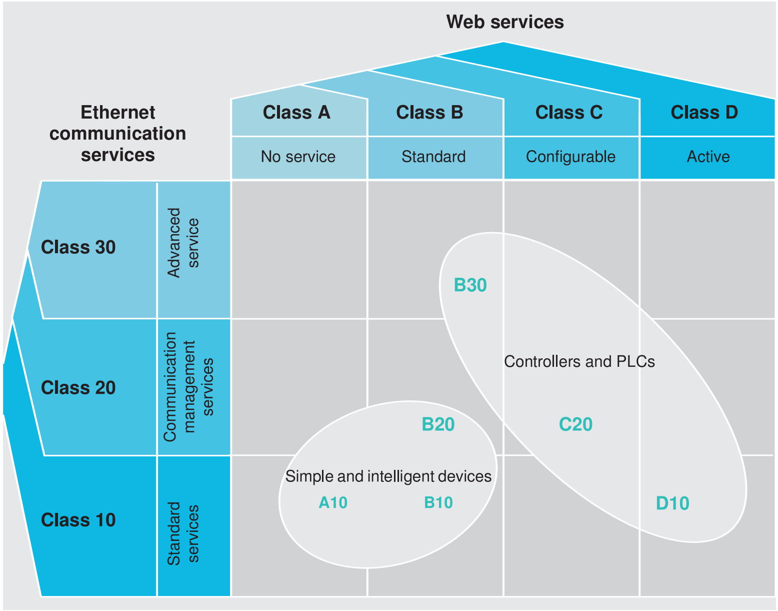


Fig. 16 Selection table

9.8 CANopen bus

General description

CAN (Controller Area Network) is a serial system bus developed by Bosch for the automotive industry. It was presented with Intel in 1985 and designed to reduce the amount of wiring in a vehicle (there can be as much as 2 km of wires in a car) by making control organs communicate via a single bus rather than dedicated lines, thereby reducing the weight of the vehicle.

High immunity to electromagnetic interference combined with reliable real-time transmission caught the attention of industrials. In 1991, the CiA (CAN in Automation) consortium was set up to promote the use of CAN in industry (see the site: http://www.can-cia.de/).

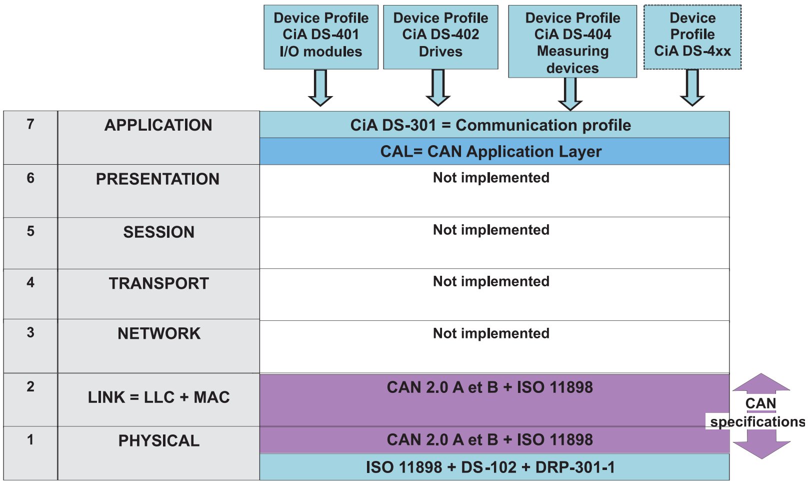
In 1993 the CiA published the CAL (CAN Application Layer) specifications describing transmission mechanisms without giving details on when and how to use them. In 1995 the CiA published DS-301 communication profile: CANopen.

Several applications level 7 layers as in *figure 17* are defined to the CAN standard:

- CANopen;
- DeviceNet;
- CAL;
- SDS;
- CAN Kingdom.

In 2001 the CiA publication of DS-304 enabled integration of level 4 safety components on a standard CANopen bus (CANsafe).

A description of CANopen technical features follows.


Device Profile CiA DS-401 I/O modules
Device Profile CiA DS-402 Drives
Device Profile CiA DS-404 Measuring devices
Device Profile CiA DS-4xx
CAN specifications

7	APPLICATION	CiA DS-301 = Communication profile CAL= CAN Application Layer
6	PRESENTATION	Not implemented
5	SESSION	Not implemented
4	TRANSPORT	Not implemented
3	NETWORK	Not implemented
2	LINK = LLC + MAC	CAN 2.0 A et B + ISO 11898
1	PHYSICAL	CAN 2.0 A et B + ISO 11898 ISO 11898 + DS-102 + DRP-301-1

Fig. 17 *CAN bus layers*

Advantages of CANopen

CANopen uses short frames

Because it has high immunity to electromagnetic interference (EMI), CANopen enables a machine or plant to work with precision, even in an atmosphere of high interference. The short CANopen frames and CANground connection offer the same capacities for every device connected to the network and protect them from electromagnetic interference (EMI).

CANopen for reliable transmission

When a CANopen device transmits data, the system generates and automatically prioritises the message. A telegram cannot be lost because of a collision problem and time is not lost waiting for the network’s next idle status. With CANopen data transmission is absolutely reliable. This is one reason why CANopen is used in medical equipment requiring reliable networks.

CANopen eliminates time loss

Time losses always waste time and money. CANopen is designed to cut time losses to an absolute minimum. With a Hamming bit length of 6, CANopen has a high error detection capacity and a very good correction mechanism. An undetected error probability of 1000 years makes CANopen the most reliable network for machines and plants.

1 bit of error every 0.7s at 500Kbps, 8hrs a day, 365 days a year.

When the network detects an error condition, first device status monitoring feature is the watchdog. Each diagnostic message contains the source and cause of the error, thus enabling a rapid response and a less time lost.
A further diagnostic is developed to improve complex CANopen device diagnostics and uphold the network. In addition, there is an error log to help detection of random errors.

□ CANopen: Performance and flexibility

The main reason for using a network is its performance and flexibility in adapting exactly to the needs of the application. CANopen offers a unique device for data transmission adaptation. Based on the consumer / producer model, CANopen can transfer data in general broadcast, point-to-point, status change and cyclic modes. This means data are transferred only if necessary or on a specific time scale. Process data objects (PDO) can be configured individually. Parameters can be changed at any time.

- Performance

Though CANopen is highly flexible, the network response is very fast. 256 digital I/O points can be processed at 1 Mbps in less than 1 ms Source: Grid Control. Profibus-DP typically requires about 2 ms at 12 Mbps for the same data transfer. In addition to fast response, message priority control can be changed.

With CANopen, data transmission can be adapted to suit application requirements.

□ CANopen cuts costs

CANopen offers ease of installation and low-cost devices. It does not require an equipotential link between devices like many field buses do. A poor connection not only causes communication errors, it also damages field bus devices.

Furthermore, CANopen components are produced in great quantity and this lowers their cost. Schneider Electric passes this advantage on to customers.

Prices of 10 to 20% less than for other field buses can be expected.

■ How CAN works

CAN is a serial bus based on a publisher/subscriber model in which a publisher sends a message to subscribers. CAN was developed with broadcast architecture.

The sender (publisher) sends the message with an identifier. The recipients (subscribers) filter messages from the bus based on their send criteria so if a message is intended for them, they read and process it. The recipient then becomes a sender *(⇨ Fig. 18)*.

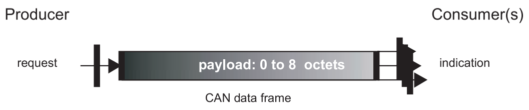


Fig. 18 CAN operation

The diagram shows the push (send) mode of the publisher/subscriber model. CAN also supports its pull (receive) mode. A client can send a message based on a remote transmission request (RTR), which is a CAN frame with RTR flags (status bits). When the producer receives such a request, it transmits the related answer *(⇨ Fig. 19)*.

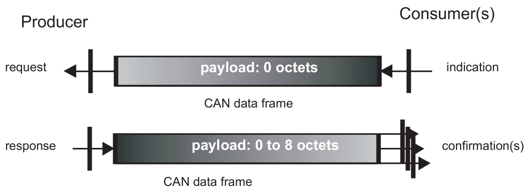
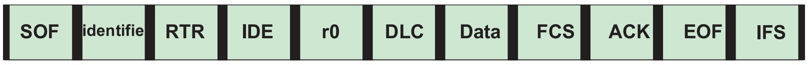


Fig. 19 CAN push / pull (publisher/subscriber model)

In a broadcast architecture, the network nodes can transmit at the same time. CAN has 2 mechanisms to deal with this: first, a sender surveys the communication artery to check if another node is already transmitting. If the artery is free, the node starts to transmit. Several nodes can start transmitting but never at the same time. This problem is overcome by a priority system.

A CAN frame (⇨ Fig.20) starts with a start of frame bit (SOF) followed by eleven identification bits, from the most to the least significant. The next bit is the remote transmission request bit, followed by 5 control bits and up to 8 bytes of working data. The control bits are: ID extension (IDE), a reserve bit and 3 bits of working data length code (DLC) in bytes. A frame check sequence (FCS) of up to 8 bytes follows the working data. The transmitter sends a recessive acknowledgement bit (ACK) which is replaced by a dominant bit by receivers which have received the frame with no error.



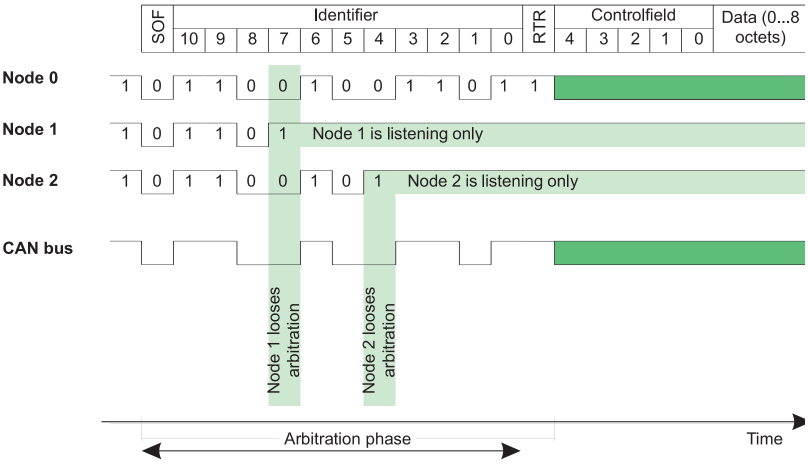
↑ Fig. 20 CAN Frame

The end of frame (EOF) bit denotes the end of frame transmission.

The bus's intermission frame space (IFS) bit must be recessive before the next frame starts. If no node is ready to transmit, the bus stays as it is. Bit codes have 2 values, dominant and recessive. If 2 nodes transmit at the same time, the receiver will only see the dominant value. In binary code, '0' is dominant and '1' is recessive. When a node transmits, it is always heard on the bus. If it transmits a recessive bit and receives a dominant one, it stops transmitting so it can continue receiving the dominant bit. This simple system prevents collisions on the CAN bus. The message with the smallest identifier has priority on the bus.

CAN is a system bus with carrier sense multiple access, collision detection and arbitration of message priority (CSMA/CD+AMP). As collisions never occur, the CAN bus is often said to be CSMA/CA (carrier sense multiple access and collision avoidance).

The message frame described in figure 21 is the base frame. For applications requiring more identifiers, there is the CAN extended frame format. The extended frame has 18 extra identifier bits in the header, after the control bits. This extends the range from 211 to 229 different identifiers. The two frame types can coexist in a single bus.



↑ Fig. 21 Typical CAN message

CAN has several means of detecting wrong messages:

- the frame check sequence (FCS) contains the frame's cyclic redundancy check (CRC). The receiver checks the frame's CRC and compares the result against the FCS. If they are not the same, the frame has a CRC error;
- the receiver detects errors in the frame structure. If the frame structure is faulty, the frame has a format error;
- the receiver of a frame publishes a dominant acknowledgement bit (ACK) if it has received an error-free frame. If the transmitter does not receive this bit, it sends an error acknowledgement;
- CAN uses non return to zero (NRZ) coding with bit stuffing. If the sender has to transmit 5 consecutive bits of the same type, it inserts another bit of the opposite type. Bit stuffing enables the receiver to synchronise with the bit chain. The receiver removes the stuffing bits from the data frame. If there are more than 5 consecutive bits of the same type, the receiver detects a stuffing error.

There are several levels of protocol application that can be used with CAN, such as DeviceNet and CANopen. CAN itself does not define a protocol application level.

■ Overview of CANopen

CANopen defines an application layer and a communication profile based on CAN.

□ CANopen defines the following communication objects (messages)

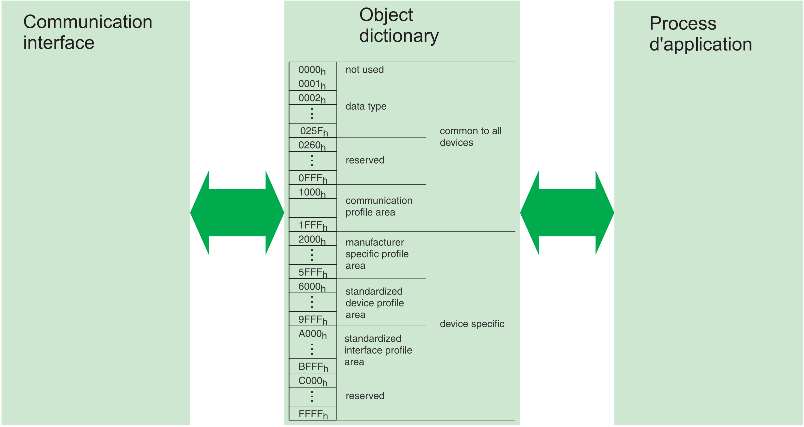
- process data object (PDO);
- service data object (SDO);
- network management object (NMT);
- special function object (SYNC, EMCY, TIME).

□ Properties

- serial data transmission based on CAN;
- up to1 Mbps;
- efficiency approx. 57%;
- up to 127 nodes (devices);
- several masters allowed;
- interoperability of devices of different brands.

□ Object dictionary

The object dictionary *(⇨ Fig.22)* is an interface between the application program and the communication interface.



↑ *Fig. 22* *Object dictionary*

- **Process data object (PDO)**

Process data objects (PDO) are used for their speed of process data transmission. A PDO can carry up to 8 bytes of working data, the maximum for a CAN frame. PDO transmission uses the CAN producer/consumer model extended by synchronised transfers. Synchronised PDO transfer relies on SYNC message transfer on the CAN bus. A PDO is sent in cyclic mode after a number (configurable from 1 to 240) of SYNC messages received. It is also possible to await availability of application process variables and send a PDO after the next SYNC message is received. This is called **acyclic synchronised transfer.**

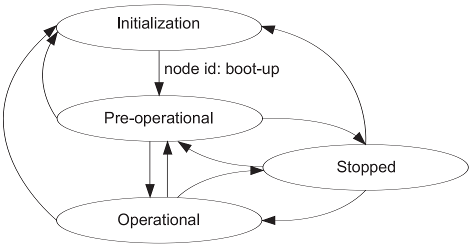
- **Service data objects (SDO)**

Service data objects (SDO) transmit parameters. SDOs give remote devices access to the object dictionary. There is no limit for the length of an SDO. If the working data cannot adapt to the CAN frame, they are divided into several CAN frames. Each SDO is acknowledged.

SDO communication uses the point-to-point mode, with one point acting as server and the others as clients.

- **Network management (NMT)**

Network management objects (NMT) change or check the status of CANopen devices *(⇨ Fig.23)*. An NMT message has a CAN 0 identifier. This gives NMT messages top priority.



↑ *Fig. 23* *Network management*

An NMT message always has 2 bytes of working data in the CAN frame. The first byte contains the encoded NMT command and the second the ID of the addressed node.

A CANopen device starts at initialisation status when the ON button is pressed. When the device has completed its initialisation, it delivers a starting NMT object to notify the master.

The collision detection protocol for monitoring device status is implemented with NMT objects.

- **Special function objects (SYNC, EMCY, TIME)**

CCANopen must have a SYNC producer to synchronise CANopen node actions. The SYNC producer periodically transmits the SYNC object. The SYNC object identifier is 128. This can lead to a delay caused by the priority of this message.

An internal device error can trigger an emergency message (EMCY). The response of EMCY clients depends on the application. The CANopen standard defines several emergency codes. The emergency message is transmitted in a single CAN frame of 8 bytes.

A CAN frame with the ID CAN 256 and 6 bytes of working data can be used to transmit the time to several CANopen nodes.

The TIME message contains the date and time in an object of Time-Of-Day type.

- **Watchdog systems**

CANopen has 2 device status monitoring methods. One is a network manager which regularly scans every device at configured intervals. This method is called "Node guarding" and has the drawback of consuming a lot of bandwidth.

The other is a message sent regularly by each device. This method uses up much less bandwidth than node guarding.

- **Network length and output rate**

The length is restricted by the output rate due to the bit priority procedure *(⇨ Fig.24)*.

Output (Kbps)	1000	800	500	250	125	50	20	10
Max. length (m)	20	25	100	250	500	1000	2500	5000

↑ *Fig. 24* *Network length and output rate*

In documents on CANopen, the most common maximum length mentioned for an output rate of 1 Mbps is 40 metres, calculated without electrical insulation such as is used in Schneider Electric CANopen devices. When this insulation is included, the minimum bus length is 4 metres at 1 Mbps. However, experience shows that, in practice, the maximum length is 20 metres.

Baud rate (kbps)	**1000**	**800**	**500**	**250**	**125**	**50**	**20**	**10**
L max. (m) (1)	0,3	3	5	5	5	60	150	300
ΣL max. (m) local star (2)	0,6	6	10	10	10	120	300	600
Interval min. (m) 0,6 x ΣL local (3)	□	3,6	6	6	6	72	180	360
ΣL max. (m) on all bus (4)	1,5	15	30	60	120	300	750	1500

↑ *Fig. 25* *Length limitations for branching devices*

Limitations on branching devices must be taken into account and are set by the parameters in *figure 25*.

(1) L max.: maximum length of branching device.

(2) EL max. local star: maximum value of total length of branching devices at the same point when a multiport distribution box is used to create a local star topology.

(3) Interval min.: Minimum distance between 2 distribution boxes. Maximum length of branching devices at the same point. This value can be calculated individually for each device: the minimum interval between two branching devices is 60% of the total length of devices at the same point.

(4) EL max. (m) of total bus: maximum value of the total length of all intervals and branching devices on the bus.

Combinations according to compliance classes

Schneider Electric has defined compliance classes for CANopen master and slave devices similar to Ethernet Modbus TCP and web services classification. The compliance classes specify which systems a device can support and ensure the upward functional compatibility of each class *(⇨ Fig.26)*.

Characteristics

Conformance classes			**M10**	**M20**	**M30**
Layer settings	Slave ID		1-16	1-63	1-127
	Data rate	**kbps**	125, 250, 500	M10 + 50, 1000	M20 +10, 20, 800
	LSS		–		Master
Devices supported			16	63	126
NMT (Network Management object)	NMT Master		NMT Master , according to DS301		–
	CANopen Manager		–		NMT master, according to DS301. Configuration Manager according to DSP302
	Boot-up procedure		according to DSP302		
	Time stamp		–		Producer
	Auto configuration		–		support
SDO (Service Data Object)	SDO Client		1	1	2
	SDO Server		–	1	1
	SDO Manager		–		1
	SDO data transfer		Expedited, segment transfer		Expedited, segment block transfer
PDO (Process Data Object)	COB-ID		ReadOnly	Read/Write	Read/Write
	PDO TT		254, 255	M10 +0, 1-240	
	PDO Inhibit Time		–	TPDO's (Read/Write)	
	PDO Event Timer		–	TPDO's (Read/Write)	RPDO's & TPDO's (Read/Write)
SYNC	SYNC		–	producer	producer/consumer
	TRIGGER		–		producer
EMCY			consumer	producer/consumer	
HEALTH	Heartbeat		16 consumers 1 producer	63 consumers 1 producer	126 consumers 1 producer
	Node guarding		no	yes	yes
Parameters	Store parameters		no	yes	yes
Conformance classes			**S10**	**S20**	**S30**
Layer settings	Slave ID		1-63	1-127	1-127
	Data rate	**kbps**	125, 250, 500	S10 + 50, 1000	S20 +10, 20, 800
	LSS		–		Slave
Diagnostic devices	Diagnostic local		–	LED or display	
NMT (Network Management object)	NMT slave		b Start remote node b Stop remote node b Enter pre-Operational b Reset node b Reset communication		
	Time stamp		–		consumer
SDO (Service Data Object)	SDO Client		–		1
	SDO Server		1		2
	SDO data transfert		Expedited, segment transfer		Expedited, segment, block transfer
PDO (Process Data Object)	COB-ID		ReadOnly	Read/Write	Read/Write
	PDO TT		254, 255	S10 +0, 1-240	
	PDO mapping parameters		FIX (Read)		–
	Connection set		Predefined connection set	Free	
	PDO Inhibit Time		–	TPDO's (Read/Write)	
	PDO Event Timer		–	TPDO's (Read/Write)	RPDO's & TPDO's (Read/Write)
SYNC	SYNC		–	consumer	producer/consumer
	TRIGGER		–		consumer
EMCY			producer	producer	consumer/producer
HEALTH	Heartbeat		1 consumer 1 producer		
	Node guarding		no	yes	yes
Parameters	Store parameters		no	no	no

Nota : S00 and M00 are for products not 100% compliant to the conformance class.

↑ *Fig. 26* *CANopen compliance*

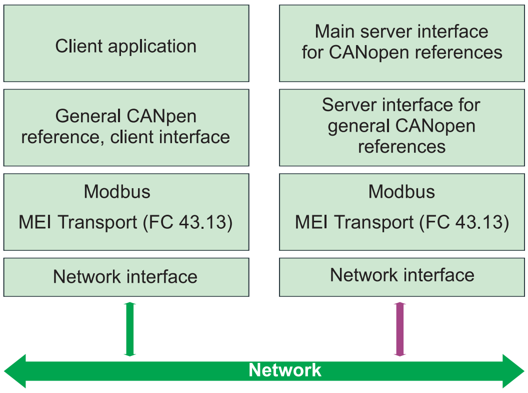
Table 27 shows the best possible product combinations based on the compliance classes.

Compliance class	S10	S20	S30
M10	Possible combination	Usage restriction	▓
M20	░	░	▓
M30	░	░	░

↑ *Fig. 27* *Product combinations*

It is however possible to use a slave device with a master of a lower compliance class (e.g. S20 with M10) or a master device with a slave of a higher compliance class (e.g. M10 with S20), by using only devices supported by the lower compliance class.

9.9 Ethernet and CANopen synergy


Client application
Main server interface for CANopen references
General CANpen reference, client interface
Server interface for general CANopen references
Modbus MEI Transport (FC 43.13)
Modbus MEI Transport (FC 43.13)
Network interface
Network interface
Network

↑ *Fig. 28* *Ethernet / CANopen synergy*

A common communication profile (DS-301) defines amongst other things assignment of COB-ID identifiers for every type of message.

Profiles specific to each product family such as discrete I/Os (DS-401), analogue I/Os, speed controllers (DS 402) and encoders describe the combined objects.

CAN in Automation and Modbus-IDA have worked together to create a standard for complete transparency between CANopen and Ethernet Modbus TCP. The result of this collaboration is the CiA DSP309-2 specification defining the communication standard between an Ethernet Modbus TCP network and a CANopen bus. The specification defines mapping services enabling CANopen devices to communicate in an Ethernet Modbus TCP network via a gateway *(⇨ Fig. 28)*.

Access to information on a CANopen device is available in read/write mode for a great many device control functions.

9.10 AS-Interface (AS-i) Bus

■ General description

Today, machine are fitted with many sensors and actuators. Upgrading possibility, maintenance and safety must be taken into consideration. AS-Interface is the floor level network filling the needs of industrial automatism's.

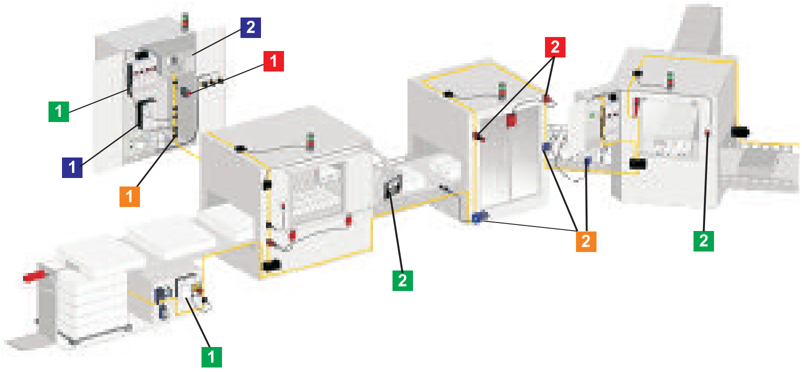
AS-Interface carry data and power in a two wire single cable. Components connected to the network can easily be replaced for maintenance purposes. The new device receives automatically the address of the former product.

AS-Interface is a highly efficient networking alternative to the hard wiring of field devices, like sensors, actuators and PLC's.

All Schneider Electric offer complies with the AS-i standard as defined by the AS-International association. It is an "open" technology supported by leading automation vendors which guaranty interchangeability and interoperability between products.

AS-Interface, as shown *(⇨ Fig. 29)*, is a mature protocol which, from more then 10 years, has proven to be a user friendly and reliable system in hundreds of thousands of applications, including conveyors, process control, bottling plants, electrical distribution systems, airport carousels, elevators and food production.

9.10 AS-Interface (AS-i) Bus



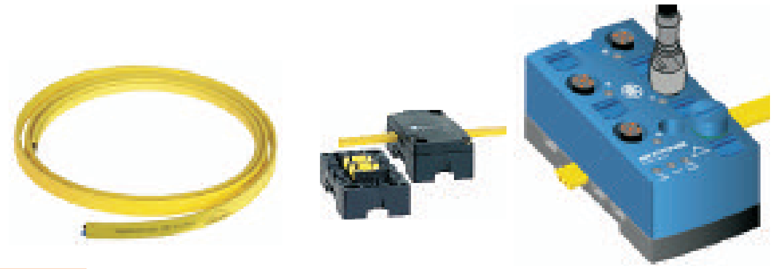
↑ *Fig. 29* *AS-Interface*

1 Interface IP20
2 Interface IP67
1 Control component
2 Dialog component

1 Safety monitor
2 Safety interface
1 Power supply
2 AS-i Master

AS-Interface is identify by a yellow cable *(⇨ Fig.30)* of a particular shape which makes inversion impossible. This cable is self sealing and sensors / actuators are equipped with punch through connectors allowing tool less connection or displacement.

AS-Interface is exclusively a field bus of the master / slave type, master being a PC, a PLC or a controller which receives information form sensors and controls the actuators thorough the installation. AS-Interface has other benefits as a free topology which allows to operate in a star , point to point, line, tree, ring technology network.



↑ *Fig. 30* *AS-Interface components*

During 10 years, AS-Interface was only suitable for discrete I/O. A few vendors had slow analogue devices i.e. temperature sensors; level sensors, but any time these were proprietary products and the number of addresses 0 to 31 was a major restriction.

AS-Interface consortium has launched a new version (V2). With this one, the number of addresses has doubled with a possibility of 62 discrete I/O's per master. But the major change is the capacity to connect any analogue sensors / actuators to any master, trough an AS-Interface. Its is also possible to mix discrete and analogue devices. Although the number of slaves will be reduced, operation is still manageable.

This new version introduced changes at the diagnostic level. The former version was only able to detect faults of the network. V2 version takes into account all defects including defects into the devices.

Obviously, V2 and V1 operating on the same network are compatible.

AS-Interface benefits *(⇨ Fig.31)*

Simplicity	**The simplicity of the wiring system is due to:** • a single cable to connect all actuators and sensors in an automation system; • built-in communication management.
Cost reduction	**Cost can be cut by up to 40% by:** • shorter design, installation, commissioning and upgrade times; • smaller housings due to more compact devices and elimination of intermediate housings now that most functions are controlled ex-machine; • elimination of control cable ducts.
Safety	**AS-Interface helps increase reliability, operational availability and safety:** • wiring errors are impossible; • no risk of poor connections; • high immunity to electromagnetic interference (EMC); • machine safety functions can be fully integrated into AS-Interface.

↑ Fig. 31 *AS-Interface benefits*

AS-Interface components

These are grouped into families *(⇨ Fig.32)* For more information, please refer to the Schneider Electric product catalogues.

Generic device interfaces	These enable any standard device (sensor, actuator, starter, etc.) to be connected to the AS-Interface. They offer wide freedom of choice and are especially suited to machine modifications and improvements previously done by conventional wiring. These interfaces are available for mounting in housings (IP20) or directly on the machine (IP67).
Dedicated interfaces and components	Dedicated interfaces (communication modules, etc.) are used for communication with the AS-Interface cable. Dedicated components are embedded in an interface and can be connected directly the AS-Interface cable. This makes short work of wiring but the choice is not as wide as with generic components.
Master	This is the central component in the system; its function is to manage data exchanges with the interfaces and components (also called slaves) throughout the plant. It can take: - 31 interfaces or components in version V1 (cycle time 5 ms); - 62 interfaces or components in version V2 (cycle time 10 ms). The master is: - either embedded in a PLC, e.g. as an extension, - or connected the field bus, where it acts as a gateway.
AS-Interface power supply	Extra-low voltage of 29.5 to 31.6V for interfaces and components powered via the AS-Interface cable. It is protected against over-voltage and short circuits. This is the only type of supply that can be used on an AS-Interface line. As the AS-Interface cable has restricted current, it is sometimes necessary to add a further supply for some circuits, in particular for actuators.
Flat cable	The yellow cable connected to the power supply ensures two functions: - data transmission between master and slaves; - powering sensors and actuators. The black cable connected to the auxiliary 24V supply powers the actuators and the sensors with insulated inputs. The mechanical profile of the cables makes polarity inversion impossible; the materials used allow for fast reliable connection of the components. When a device is disconnected, e.g. for alteration purposes, the cable recovers its initial shape by self-sealing. These cables support 8A maximum and are available in two versions: - rubber for standard applications; - TPE for applications where the cable may be splashed with oil.
Safety solutions AS-Interface (See section 6 on safety)	Standard process information can be transmitted at the same time and by the same media as information safety up to level 4 of standard EN 60954-1. Integration into AS-Interface by adding a monitor and safety-related components connected to the yellow AS-Interface cable. Safety information is only exchanged between the safety monitor and its components and is transparent for the other standard functions. This means a safety system can be added to an existing AS-Interface network.
Addressing terminal	As the components are connected in parallel on the AS-Interface bus, a different address must be assigned to each. This function is ensured by a terminal connected individually to each components.

↑ *Fig. 32* *AS-Interface components*

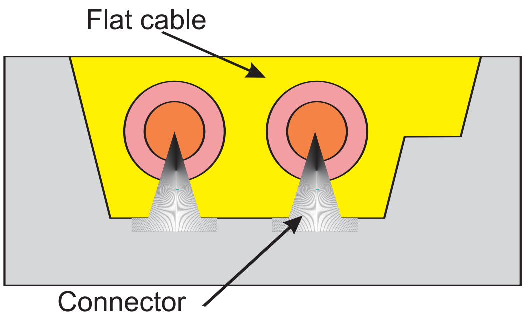


Fig. 33 *AS-Interface connection*

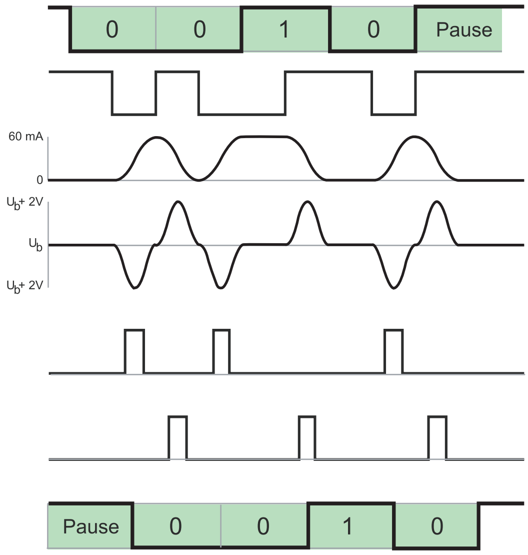


Fig. 34 *Voltage and current waveforms*

Operating principles AS-Interface network

Connection

Connection hardware uses punch trough connector also named “vampire connector”. The connector has two pins which make the connection through the insulating material of the cable. The two halve parts of the connector are then screwed together to make a reliable connection.

This system *(⇨ Fig. 33)* is standardised and any type of installation can be made up to IP67 protection.

Signal modulation

AS-Interface has been designed to run without a terminal plug in any configuration. Operation principle uses current modulation based on Manchester encoding. Two chokes, inserted in the power supply convert this current in a sine wave. The shape of the generated signal avoid the use of shielded cables *(⇨ Fig.34)*.

Length of the network

Length is linked to signal distortion and voltage drop. Maximum length between two slaves shall not exceed 100 m *(⇨ Fig.35)*. On can increase this distance by the use of repeaters with the following limits:

- no more than two repeater per line,
- maximum distance to the master shall not exceed 300 m,
- a passive terminal extends the distance from 100 m to 200 m,
- an active terminal extends the distance to 300 m.

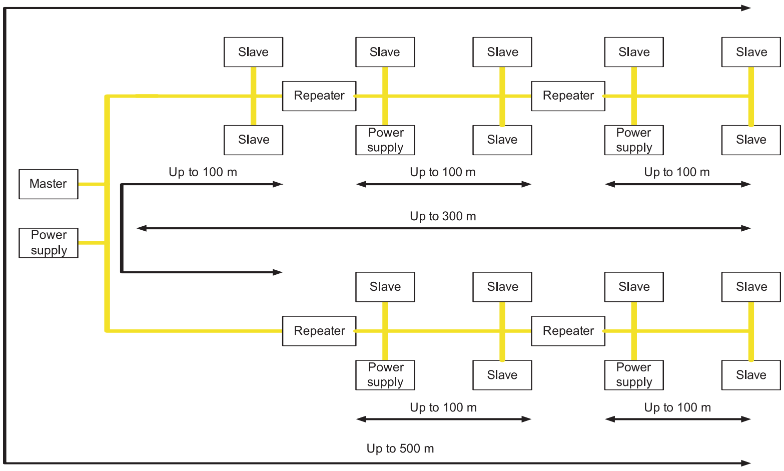
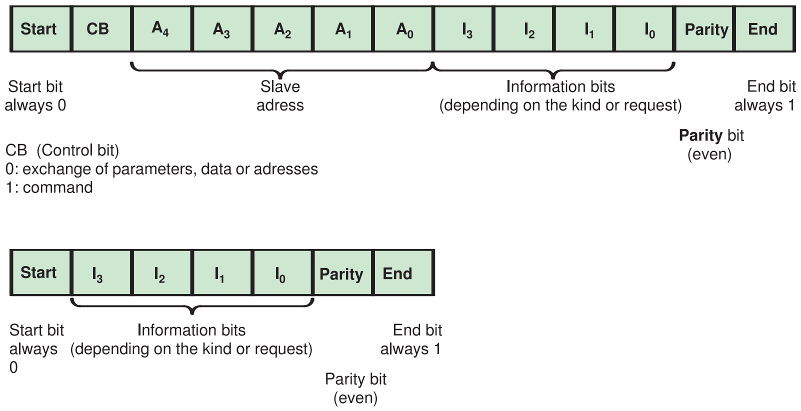


Fig. 35 *AS-Interface limits*

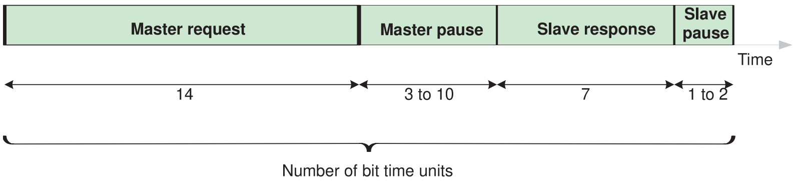
□ Protocol principle

Protocol principle is base on a single master protocol. The master send a request to all slaves in a row, these ones send the requisite data *(⇨ Fig.36)*. When all slaves have send an answer, a new cycle begins and so on. Cycle time depends upon the number of slaves and is easy to calculate.



↑ Fig. 36 *Master slaves frame*

AS-Interface use several means to guaranty the dependability of the data transmission. The signal is checked by the receiver; if the form is incorrect, the message is discarded. A check sum bit, added to a short message (7 and 14 bits), secure the logic content of the information. The master dead time causes the acknowledgement *(⇨ fig.37)*.



↑ Fig. 37 *Response time constitution*

Length of a bit is 6 ms. At a rate of 166.67 Kbits/s, adding all the dwell bits, the cycle time cannot exceed 5082 µs.

- **Each cycle can be divided in 3 parts**
  - data exchange,
  - system supervision,
  - updating / slave insertion.

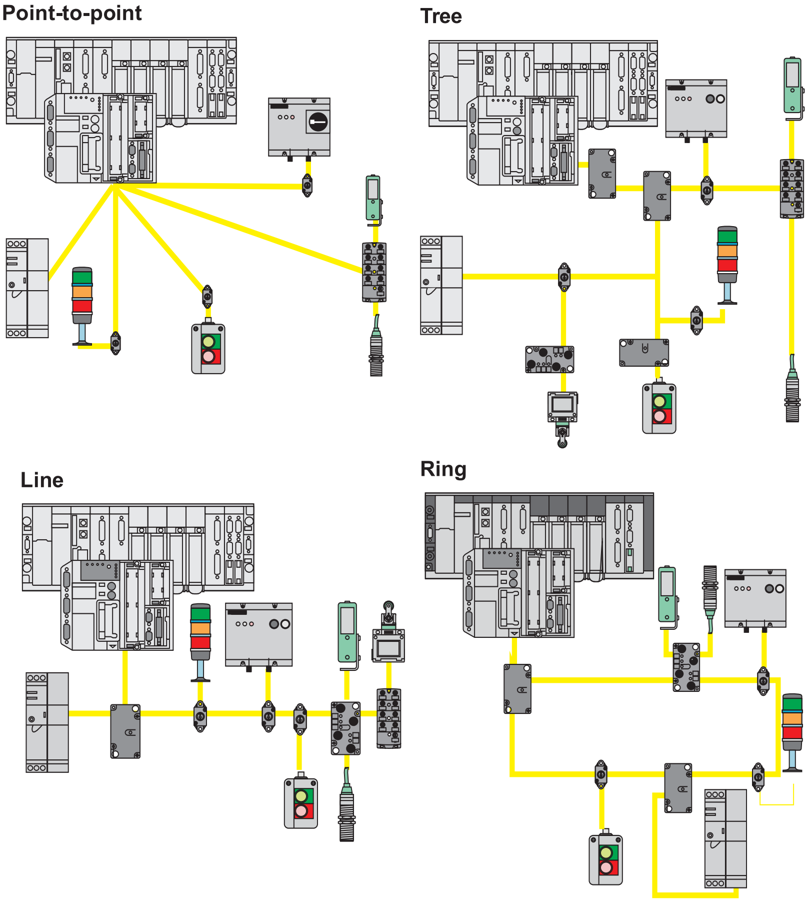
Master's AS-Interface profile tailors its actual capabilities. In general, it has the following functions:

- initialise the system,
- identify the connected slaves,
- send the slaves parameters to the slaves,
- check the integrity of the process data with the slaves,
- monitor the system diagnostics (status of the slaves, status of the power supply etc.),
- transmit all detected fault to the system supervisor (PLC, etc.),
- reconfigure the system if any modification is made to it.

Slaves decipher requests issued from the master and sent the answer with no delay. However, as slave will not answer to an incorrect or inappropriate request. Functional capacity of a slave is defined by its AS-Interface profile.

Topology and AS-Interface wiring

The absence of restrictions allows for all sorts of system configurations, some of which are illustrated below *(⇨ fig.38)*.



↑ Fig. 38 *System configuration*

AS- interface versions

The fisrt one (V1) has been updated to V2.1 which adds the following improvements:

- capacity to connect 62 slaves (V1 limit is 31);
- capacity to transmit a slave fault message without disconnecting the slave which remains able to communicate when continuity of service is a critical issue;
- support of analogue slaves.

AS- interface profile

AS-Interface equipment profile tailors its capabilities. Two AS Interface devices, made by any manufacturer, having the same function and profile operate identically on the same network. They are interchangeable within the network. Profile is factory set in the electronic of the product by two or three characters and cannot be changed.

Today more then 20 profiles have been defined by the AS-i consortium. They are described hereunder.

Table 39 shows the compatibility between V1 and V2.1.

	Slave V1	Slave V2.1 with standard addressing	Slave V2.1 with extended addressing	Analogue slave
Master V1	Compatible	Compatible but slave defects are not forwarded	Not compatible	Not compatible
Master V2	Compatible	Compatible	Compatible	Compatible

↑ *Fig. 39* *V1 / V2.1 compatibility*

□ Master profiles

Master profiles define individual capacities of every AS-Interface master. There are four profile types: M1, M2, M3, M4, the last one is compatible with the former versions.

□ Slave profiles

All slaves have a profile, which means they are seen as ASIC equipped AS-Interface peripheral devices. Dedicated products as smart actuators, interfaces connecting traditional devices to the AS-Interface network are in this family. Profiles, similar to ID cards, have been defined to sort actuators and sensors in large categories. This is particularly useful when a slave has to be replaced i.e. tow actuators made by different manufacturers can be installed on the network with no change in the program or the address.

9.11 Conclusion

The use of networks for communication in industrial automation architectures increases their flexibility so they can fulfil the requirements for adapting machines or plants. To do so involves making choices necessitating specific knowledge of the right solutions out of a wide range of communication networks. Simple criteria should be used: products should be open, standardised and suitable.

- An open network, as opposed to a proprietary one, leaves one free to choose suppliers of automation devices.
- An internationally standardised network guarantees durability and upgradeability.
- A suitable choice balanced between machine or plant requirements and network performance is the way to optimise the investment.

The last point is the one which evidently requires exact knowledge of what is offered for communication networks, which have long been thought of as complicated to select, implement and maintain. Schneider Electric has decided to focus its offer on genuinely open networks based on international standards and adapted to requirements at all levels of automation architecture by defining implementation classes which keep choices simple and optimal.