

Dr. Truong Dinh Chau
Department of Automatic Control Engineering
Faculty of Electric – Electronic Engineering
Ho Chi Minh City Polytechnic University
Mobile: 091 543 74 40
E-mail: tanchau@mail.ru

Supervisory Control & Data Acquisition



SCADA Systems

Modern Automation Control Systems

Integrated Automated Control Systems

Modern Process Control Systems

Complex Manufacturing Systems

Computer Integrated Manufacturing Systems (CIMS)

- 
- Hardware,
 - Software,
 - Algorithmization,
 - Networking,
 - Process Control,
 - PLC,
 - Automatic Control

Description of the Discipline

The course provides theoretical reductions about possibilities, structures, technical features and trend of Supervisory Control And Data Acquisition (SCADA) system, I/O devices and I/O devices networking (DAQ Card, PLC, ...) and industry communication technology.

The crucial issue of the course is analysis of components in the SCADA systems, the mechanisms of integration of I/O devices with SCADA software, MS Office the other applications.

In the course use industry components: Siemens, Intouch, GeniDAQ, DDE Servers, OPC Servers and MS Office for practical demonstration of the theoretical material.

Syllabus

Chapter	Content
<i>SCADA system, structure & components</i>	<ol style="list-style-type: none"> 1. Introduction 2. What is SCADA? 3. Components of SCADA system 4. Information structure of hierarchical control systems – electric systems 5. SCADA GSM – water supply, gas, electric systems.
<i>Data acquisition devices & networking of data acquisition devices</i>	<ol style="list-style-type: none"> 1. Data acquisition cards & accessories 2. PLC 3. Typical configurations of I/O device connection with IPC 4. PLC networking & DCS
<i>Industry communication technology</i>	<ol style="list-style-type: none"> 1. Communication technology 2. Typical mechanisms of data exchange 3. Basic type of automation networks 4. Example: Modbus protocol
<i>Concepts in SCADA software, HMI & control strategies design</i>	<ol style="list-style-type: none"> 1. Tag – abstract of parameters of process control 2. Device – abstract of I/O device 3. Task – HMI & real-time control strategies design 4. Advantech GeniDAQ, Intouch Wonderware
<i>Approaches of device connection with SCADA software</i>	<ol style="list-style-type: none"> 1. Typical configurations of I/O device connection with IPC 2. DLL mechanism 3. DDE mechanism 4. OLE mechanism 5. OPC mechanism 6. Applications – SCADA in Electric Systems

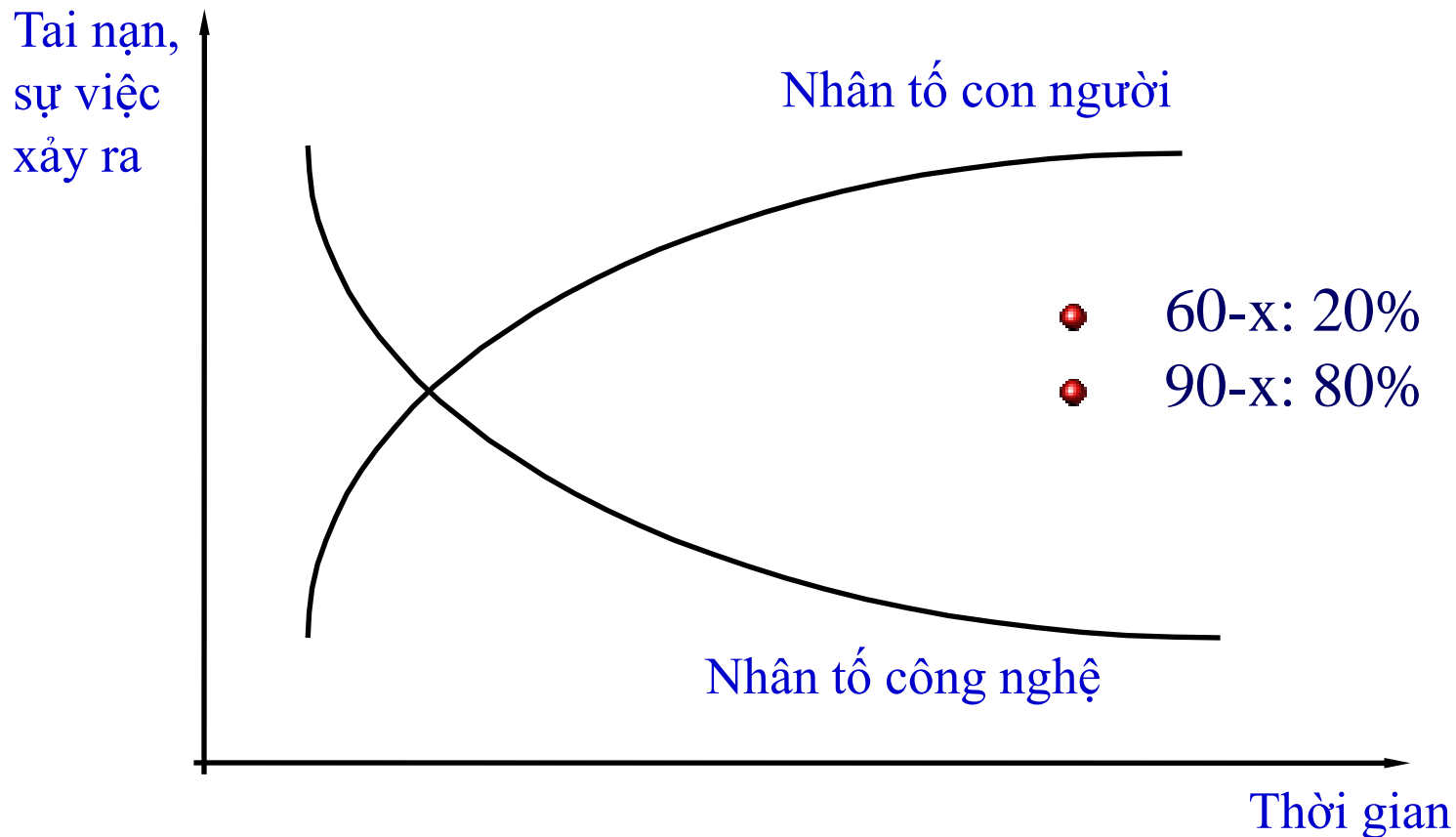
Tài liệu tham khảo

- John Pack, Steve Mackay. Practical Data Acquisition for Instrumentation and Control Systems. Elsevier. 2003. 407 p.
- David Bailey, Edwin Wright. Practical SCADA for Industry. Elsevier. 2003. 288 p.

Mở đầu

- Supervisory Control And Data Acquisition (SCADA) = công nghệ tự động hóa + điều khiển
- Đang phát triển và có khuynh hướng phát triển ở trong mọi lĩnh vực của công nghiệp.
- Trong mười năm trở lại đây ở các nước tiên tiến quan tâm mạnh mẽ.
- Sự phát triển mạnh mẽ của công nghệ thông tin đã làm tăng mức độ tự động hoá và phân bố lại chức năng giữa con người với thiết bị và sinh ra vấn đề tương tác giữa con người – quan sát viên với hệ thống điều khiển.
- SCADA đã được đặt lên hàng đầu ở những nước phương tây vào những năm 80 thế kỷ XX. Ở Việt Nam 90-x.

Xu hướng của nguyên nhân tai nạn trong hệ thống tự động hóa phức tạp



- Áp dụng phương pháp truyền thống cũ trong việc xây dựng hệ thống tự động hoá.
- Chưa đánh giá đúng vai trò cần thiết phải xây dựng hệ thống và giao diện người máy (Human Machine Interface – HMI)

Các lĩnh vực ứng dụng SCADA

- Hệ thống SCADA ứng dụng hiệu quả nhất trong vấn đề tự động hoá điều khiển quá trình liên tục và phân bố.
- Công nghệ dầu khí.
- Điều khiển sản xuất, chuyển tải và phân phối năng lượng điện.
- Cung cấp nước, làm sạch nước và phân phối nước.
- Điều khiển những đối tượng vũ trụ;
- Điều khiển trong giao thông (tất cả các dạng giao thông: hàng không, đường sắt, đường bộ, đường thuỷ, tàu điện ngầm).
- Viễn thông.
- Quân sự.

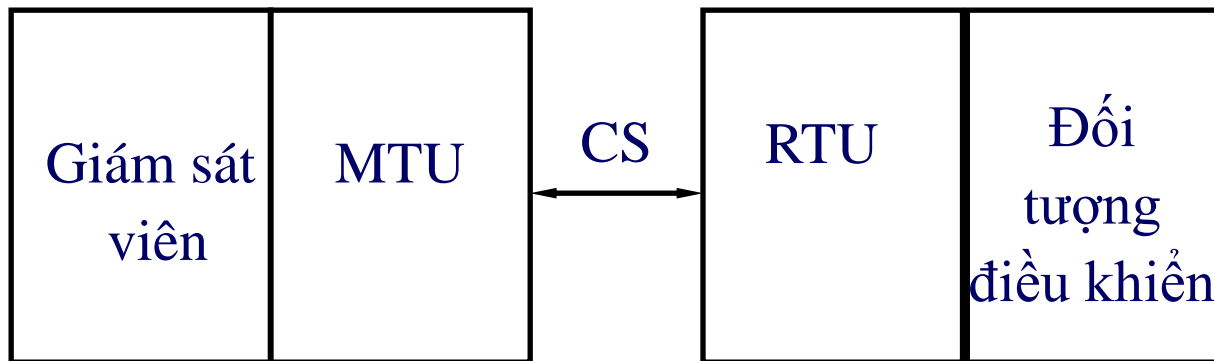
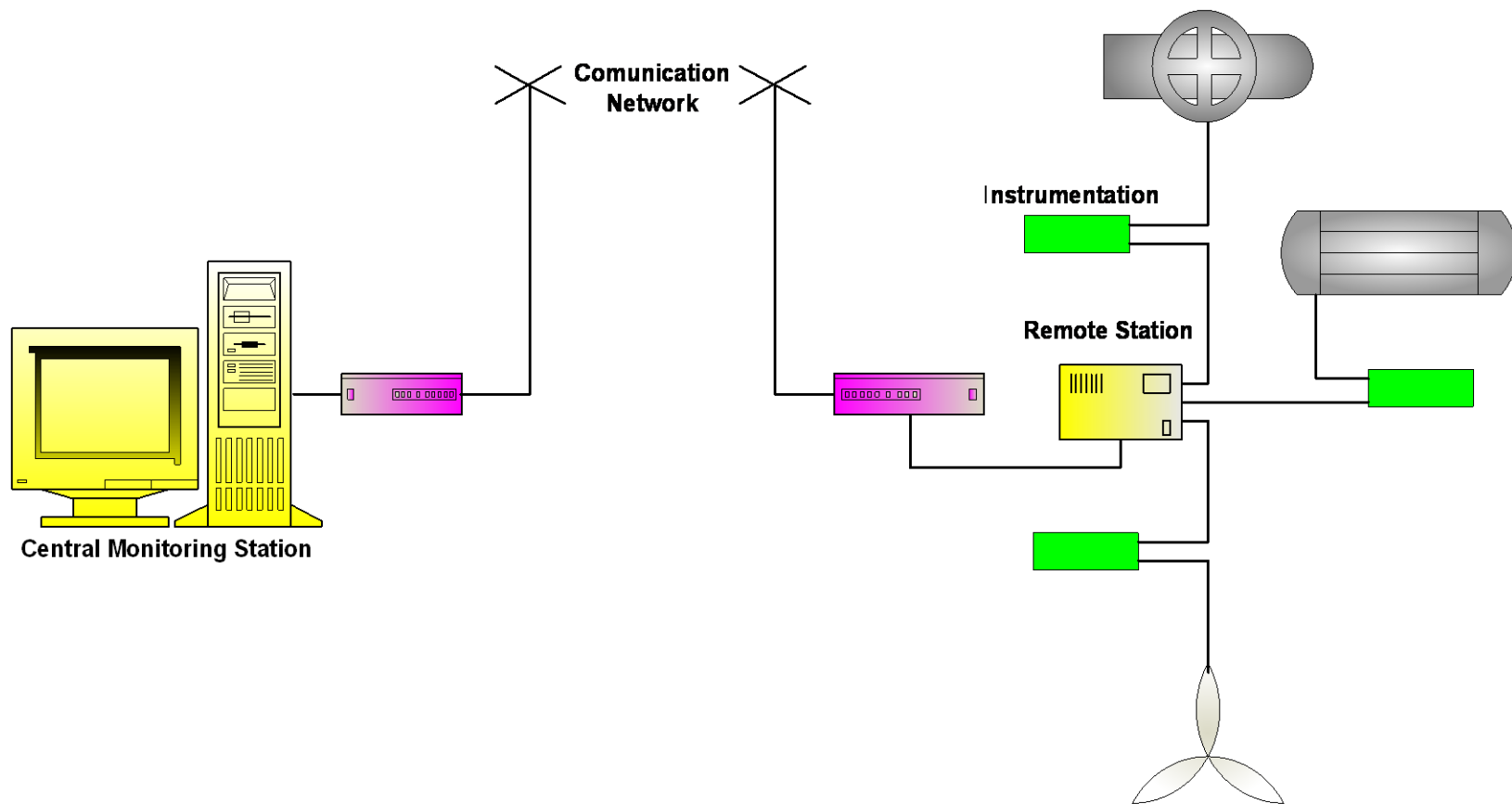
Các hệ thống SCADA

SCADA	Company	Country
InTouch	Wonderware	USA
GeniDAQ	Advantech	Taiwan
Genesis32	Iconics	USA
Trace Mode	AdAstra	Russia
Vijeo Look	Schneider Electric	France
Citect	Ci Technologies	Australia
Factory Link	United States DATA Co.	USA
RSView	Rockwell Software Inc.	USA
LabView	National Instruments	USA
iFIX	Intellution	USA
WinCC	Siemens	Germany
Master SCADA	InSAT	Russia
CIMPLICITY	GE Fanuc	USA
Contour	Obedinenie Uig	Ucraina
Wizcon	Axeda	USA
Crug-2000	Crug	Russia
Elipse SCADA	Elipse Software	USA

What is SCADA?

- SCADA – quá trình thu thập dữ liệu thời gian thực từ các đối tượng để xử lý, biểu diễn, lưu trữ, phân tích và có khả năng điều khiển những đối tượng này.
- Các hệ thống SCADA hiện đại là một giai đoạn phát triển hệ thống tự động hoá trước đây, chính là hệ thống truyền tin và báo hiệu (Telemetry and Signalling).
- Trong những hệ thống SCADA dù ít hay nhiều cũng được thực hiện những nguyên tắc như: làm việc với thời gian thực, sử dụng một khối lượng tương đối lớn thông tin thừa (tần số cập nhật dữ liệu cao), cấu trúc mạng, nguyên tắc hệ thống và mô đun mở, có thiết bị dự trữ để làm việc trong trạng thái “dự trữ nóng”, ...

Thành phần cấu trúc cơ bản của hệ thống SCADA



Thành phần cấu trúc cơ bản của hệ thống SCADA

Remote Terminal Unit (RTU) – thiết bị đầu cuối từ xa – thực hiện các công việc xử lý và điều khiển ở chế độ thời gian thực.

RTU rất đa dạng – từ những cảm biến nguyên thủy thực hiện thu thập thông tin từ đối tượng cho đến những bộ phận máy móc đa xử lý thực hiện xử lý thông tin và điều khiển trong chế độ thời gian thực. Việc sử dụng RTU có bộ xử lý cho phép làm giảm được yêu cầu đối với tốc độ của kênh truyền kết nối với trung tâm điều khiển.

Có 2 loại hệ thống thời gian thực: hệ thống thời gian thực cứng và hệ thống thời gian thực mềm

Thành phần cấu trúc cơ bản của hệ thống SCADA

- **Master Terminal Unit (MTU)** – trung tâm điều phối, thực hiện công việc xử lý dữ liệu và điều khiển ở mức cao ở chế độ thời gian thực mềm.

Một trong những chức năng cơ bản của MTU là cung cấp giao diện giữa con người – quan sát viên với hệ thống. MTU có thể bằng những dạng khác nhau, từ một máy tính đơn lẻ với các thiết bị cũ cho đến hệ thống máy tính lớn bao gồm các Server và Client.

- **Communication System (CS)** – hệ thống truyền thông (kênh liên kết) cần thiết để truyền dữ liệu từ các địa điểm ở nơi xa đến MTU và truyền tín hiệu điều khiển đến RTU.

SCADA = Hardware + Software + Telecommunication + Brainware

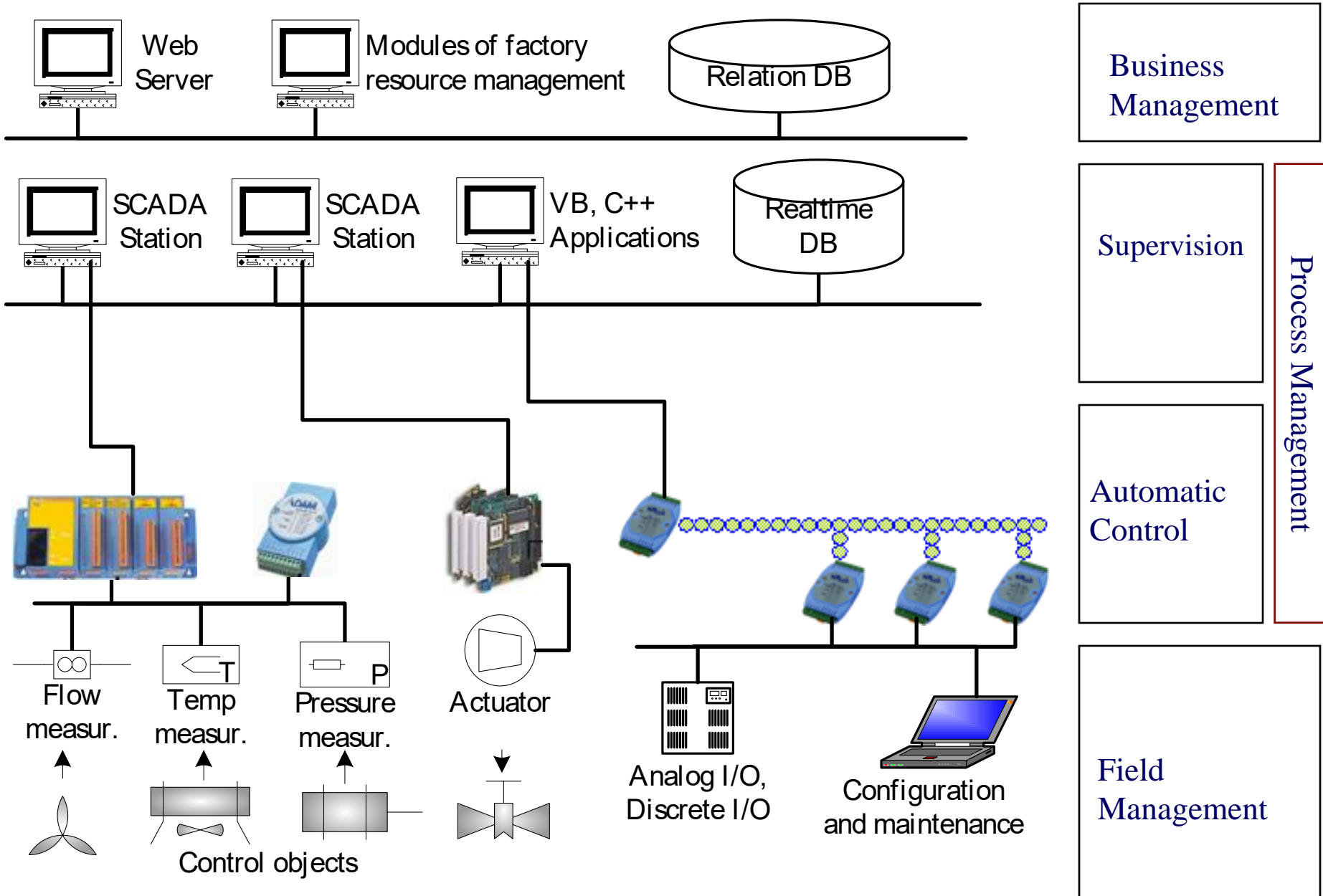
Phân chia nhiệm vụ trên hệ thống SCADA

- Tồn tại 2 loại điều khiển thiết bị:
 - điều khiển tự động;
 - điều khiển bằng những thao tác gán ban đầu của con người.
- Có 4 thành phần chức năng cơ bản:
 - con người,
 - máy tính tương tác với con người,
 - máy tính tương tác với đối tượng điều khiển,
 - đối tượng điều khiển.

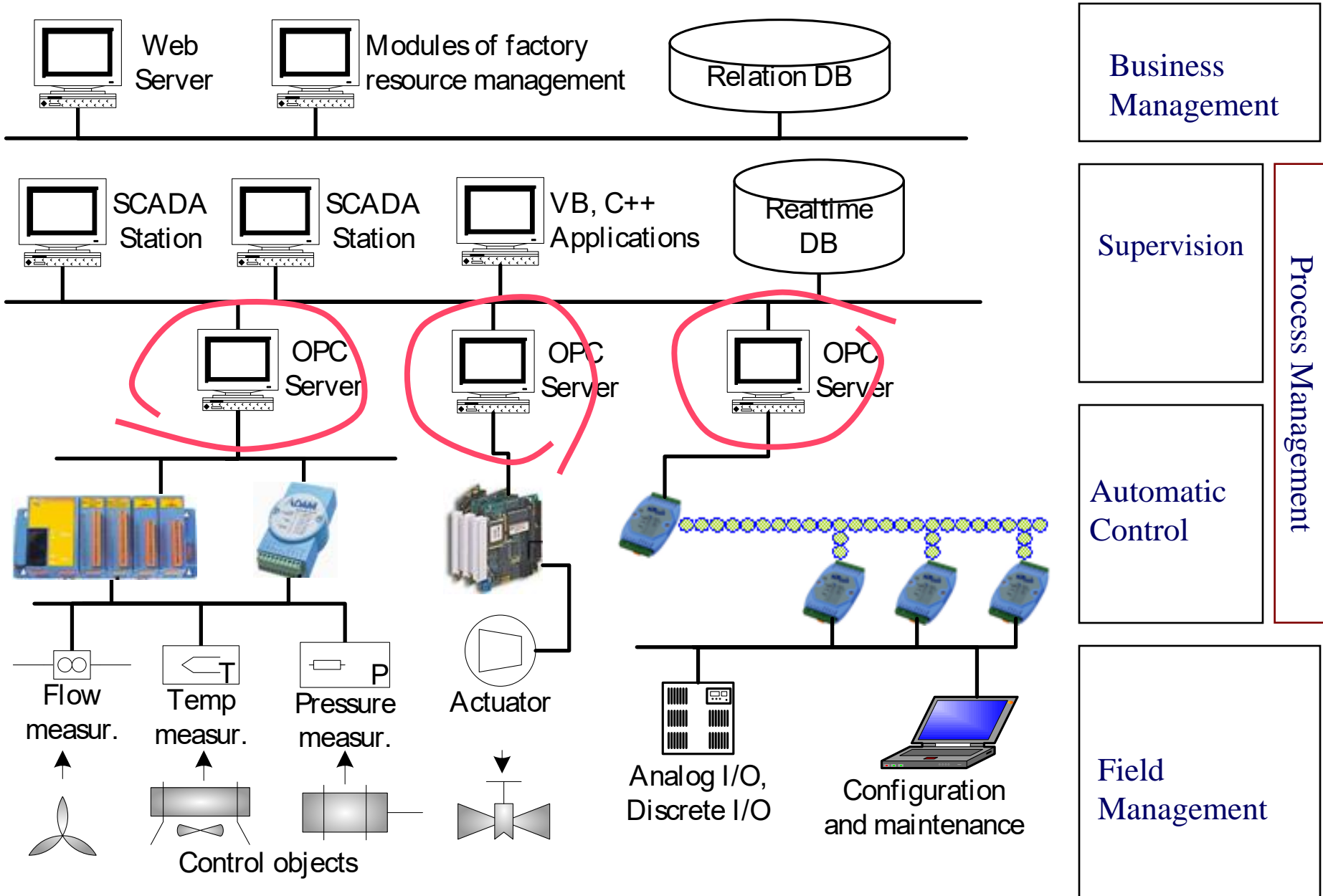
Chức năng của operator

- dự kiến những thao tác nào cần thiết để thực hiện tiếp theo;
- lập chương trình cho những thực hiện tiếp theo;
- theo dõi kết quả làm việc bán tự động hoặc tự động của hệ thống;
- liên can vào quá trình trong những trường hợp có sự việc trầm trọng xảy ra, khi hệ thống tự động không trả lời, hoặc khi cần thiết phải thay đổi thông số quá trình;
- thu nhận kinh nghiệm từ quá trình làm việc.

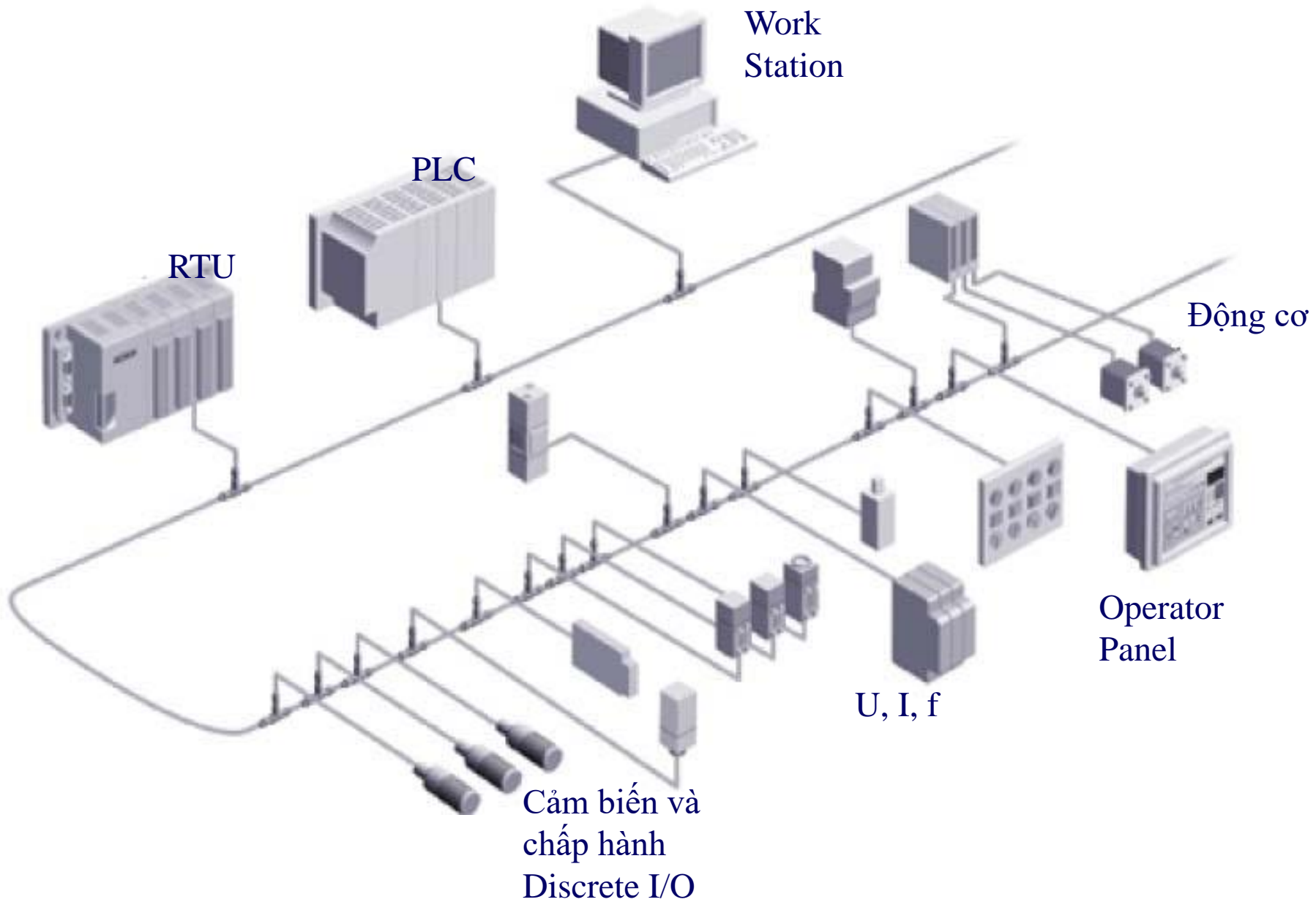
Cấu trúc của hệ thống điều khiển hiện đại



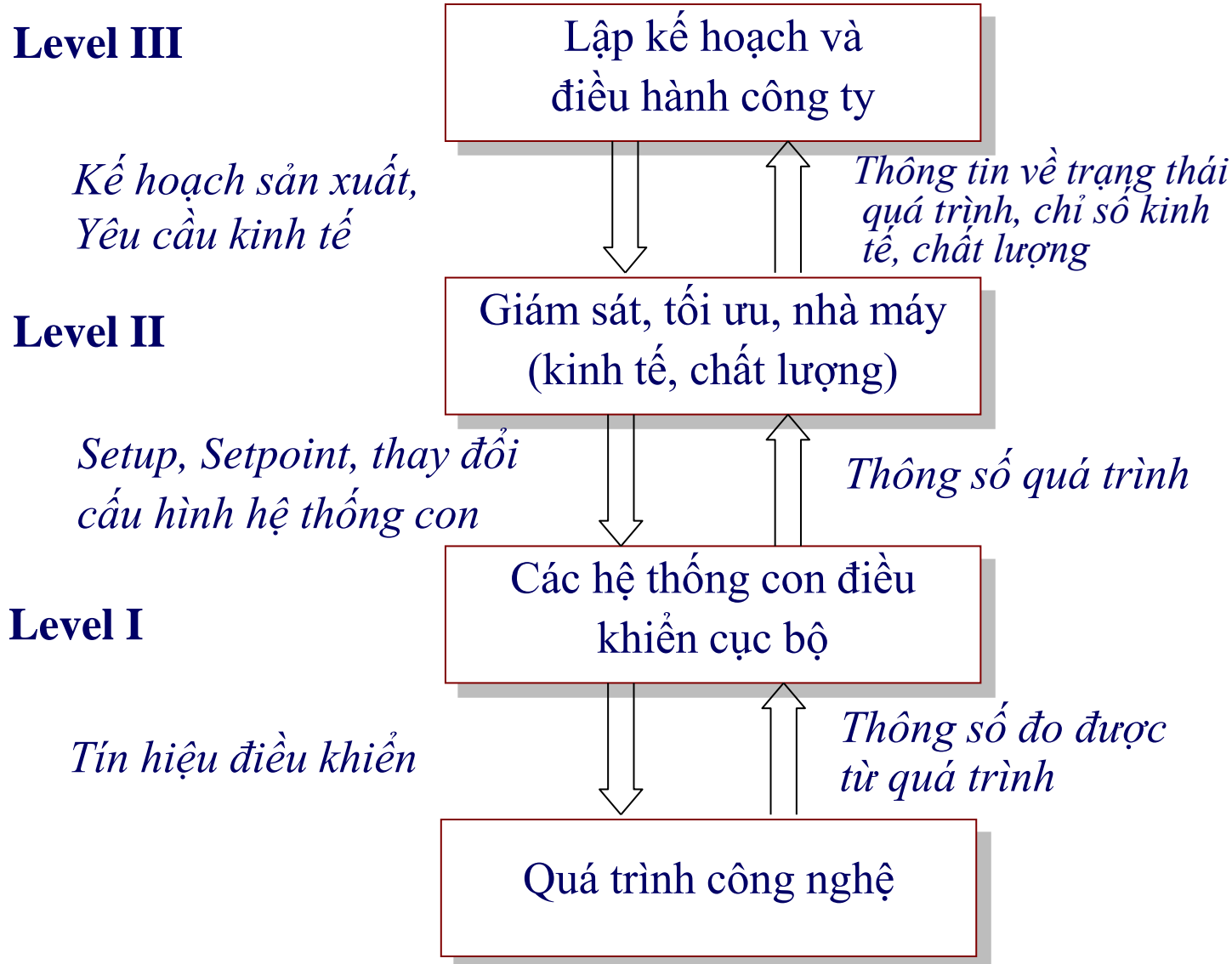
Cấu trúc của hệ thống điều khiển hiện đại



DCS



Luồng thông tin trong hệ thống điều khiển tích hợp



Chức năng và nhiệm vụ của Level I

- Thu thập dữ liệu quá trình công nghệ thời gian thực;
- Tính toán theo algorithm và đưa ra tín hiệu điều khiển theo qui luật cho trước;
- Báo hiệu về việc vượt quá ngưỡng cho phép của các thông số;
- Block những hành động lỗi của Operator và thiết bị điều khiển;
- Ngăn ngừa xảy ra Alarm.

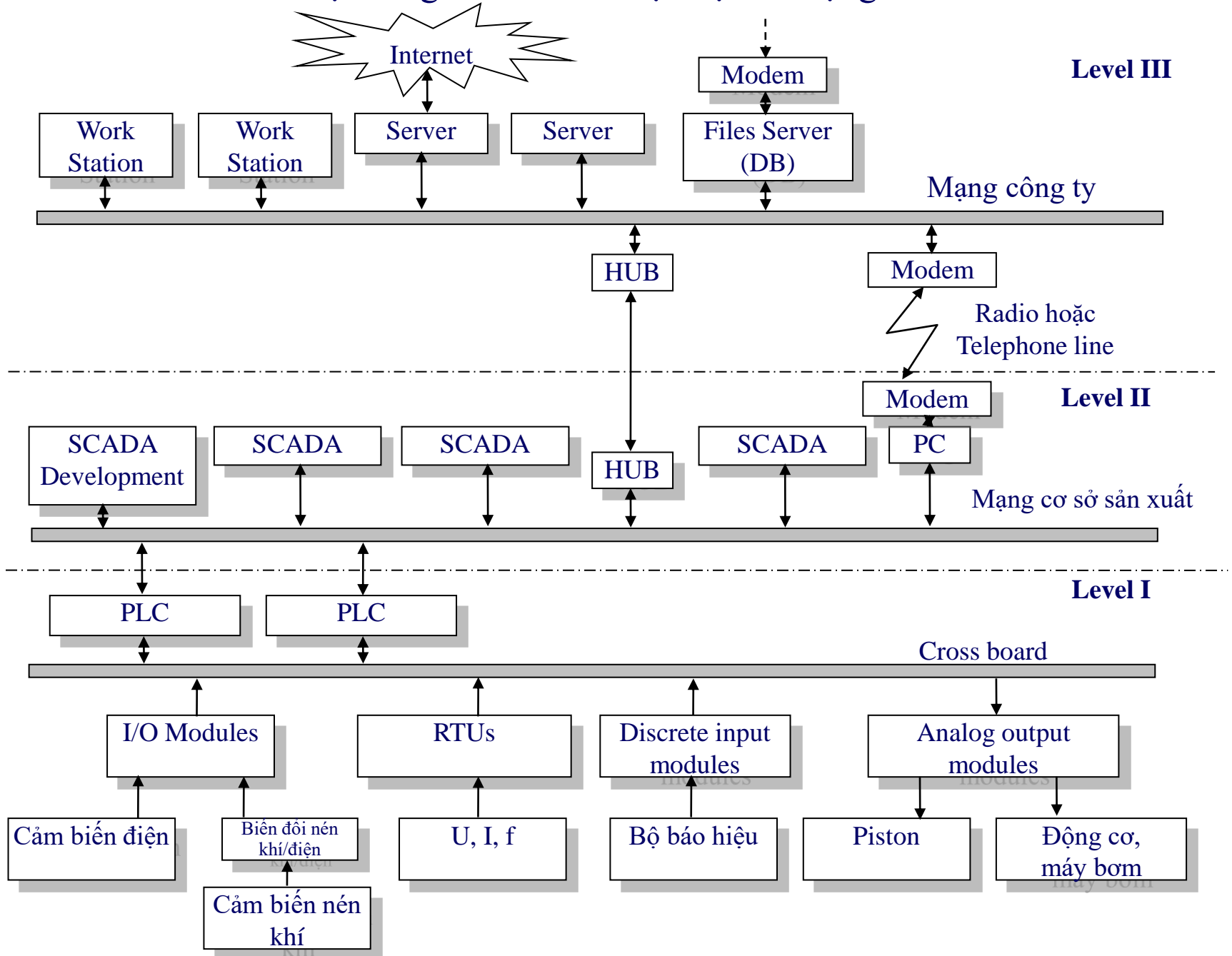
Chức năng và nhiệm vụ của Level II

- Thu thập thông tin từ cấp dưới, xử lý, lưu trữ và monitoring;
- Đưa ra tín hiệu điều khiển trên cơ sở phân tích thông tin;
- Chuyển thông tin về việc sản xuất ở các xưởng, xí nghiệp cho cấp cao hơn;
- Tính toán những thông số thứ cấp, trong đó, chỉ số chất lượng sản phẩm, chỉ số kinh tế-kỹ thuật;
- Lưu trữ thông tin;
- Đưa ra các report;
- Chuẩn đoán về sự hư hỏng của các phần tử trong hệ thống;
- Xác định thông số, cấu hình của các thiết bị điều khiển và những bộ điều khiển cục bộ của Level 1;
- Thay đổi cấu trúc các hệ thống điều khiển cục bộ, thay đổi trạng thái làm việc của các thiết bị điều khiển.

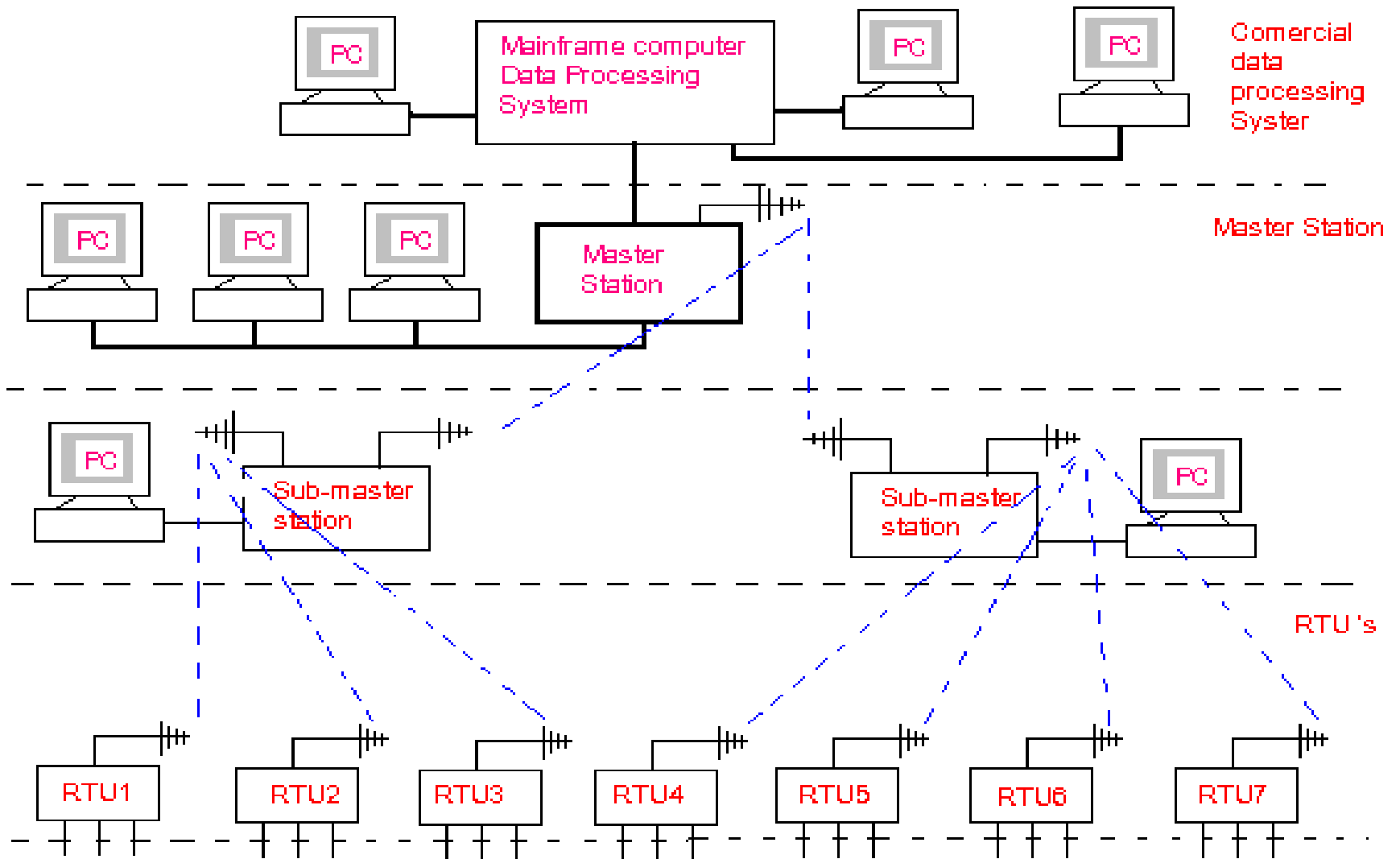
Chức năng và nhiệm vụ của Level III

- Tối ưu các chỉ số kinh tế về sản xuất;
- Điều khiển theo các chỉ số kinh tế, kinh tế-kỹ thuật;
- Quản lý tài nguyên của công ty;
- Lưu trữ thông tin;
- Đưa ra kế hoạch sản xuất.

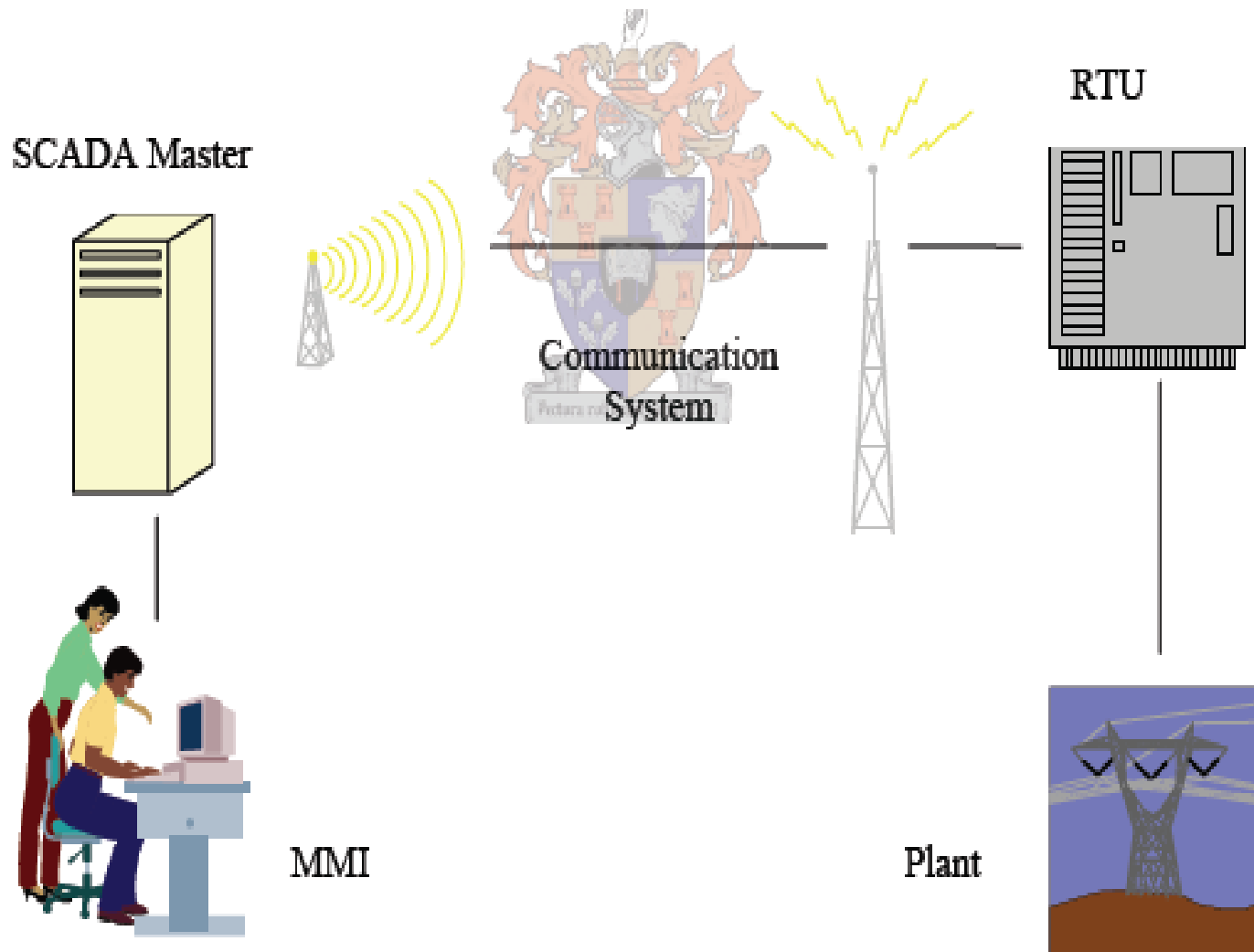
Hệ thống điều khiển hiện đại mở rộng



SCADA on GSM Base



SCADA on GSM Base



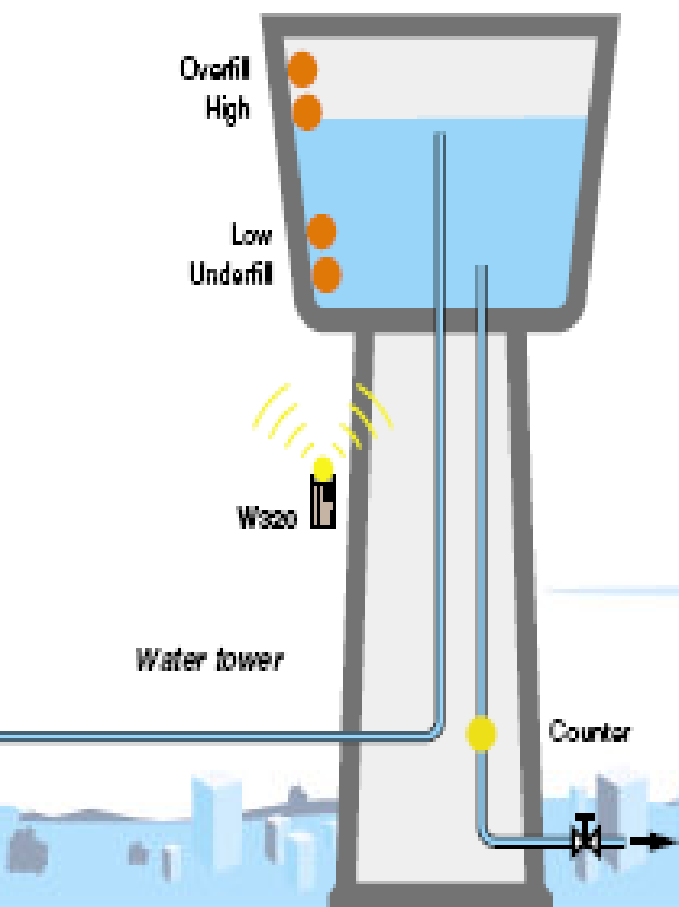
Radio modems

- Radio modem hoạt động ở tần số 400Mhz đến 900Mhz, bán kính phủ sóng 25 -100 km.
- Chúng có thể hoạt động trong một Network nhưng phải có phương thức quản lý network (Protocols) để xử lý và sửa lỗi truyền thông.
- Radio modems dùng giao tiếp theo chuẩn RS-232, tuy nhiên chuẩn RS-422 và RS-485 cũng được sử dụng.
- Tín hiệu số đưa vào modem, được điều chế thành sóng sóng vô tuyến và phát đi trong không gian.

RADIO MODEM



Ứng dụng trong lĩnh vực cung cấp nước



Application software supplied

Water
(more than 4 pumps,
please consult us)

Others

Pumping station

Recirculation station

Wastewater lift station

Reservoir

RTU (Site surveillance)

Programmable controller remote maintenance bridge

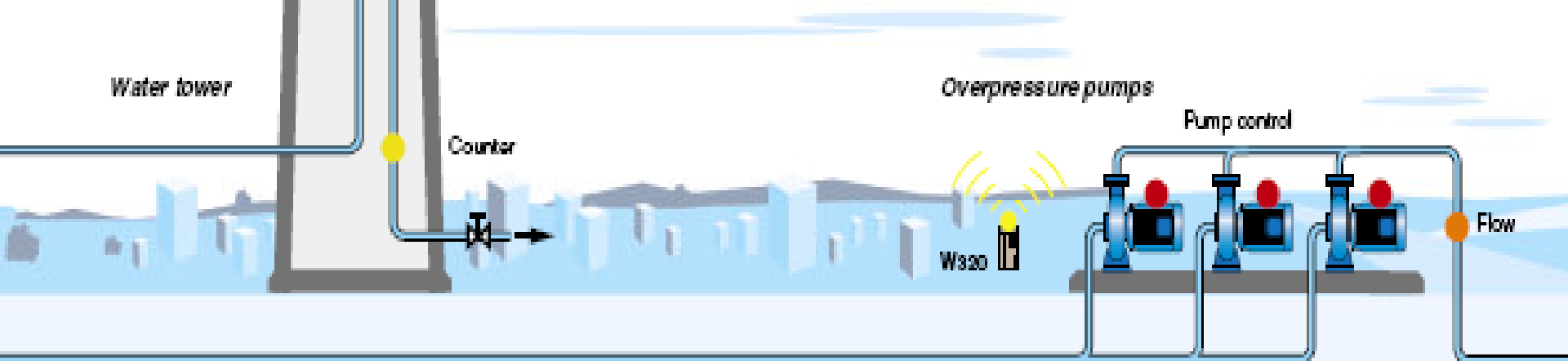
Wind power remote maintenance

Overpressure pumps

Pump control

W320

Flow



Ứng dụng trong lĩnh vực phân phối gas



Autonomous Spill Detection Sensor Over Grated Sump
Remote Gas Compressor Station, West Virginia

Ứng dụng trong lĩnh vực dầu khí



SCADA Applications

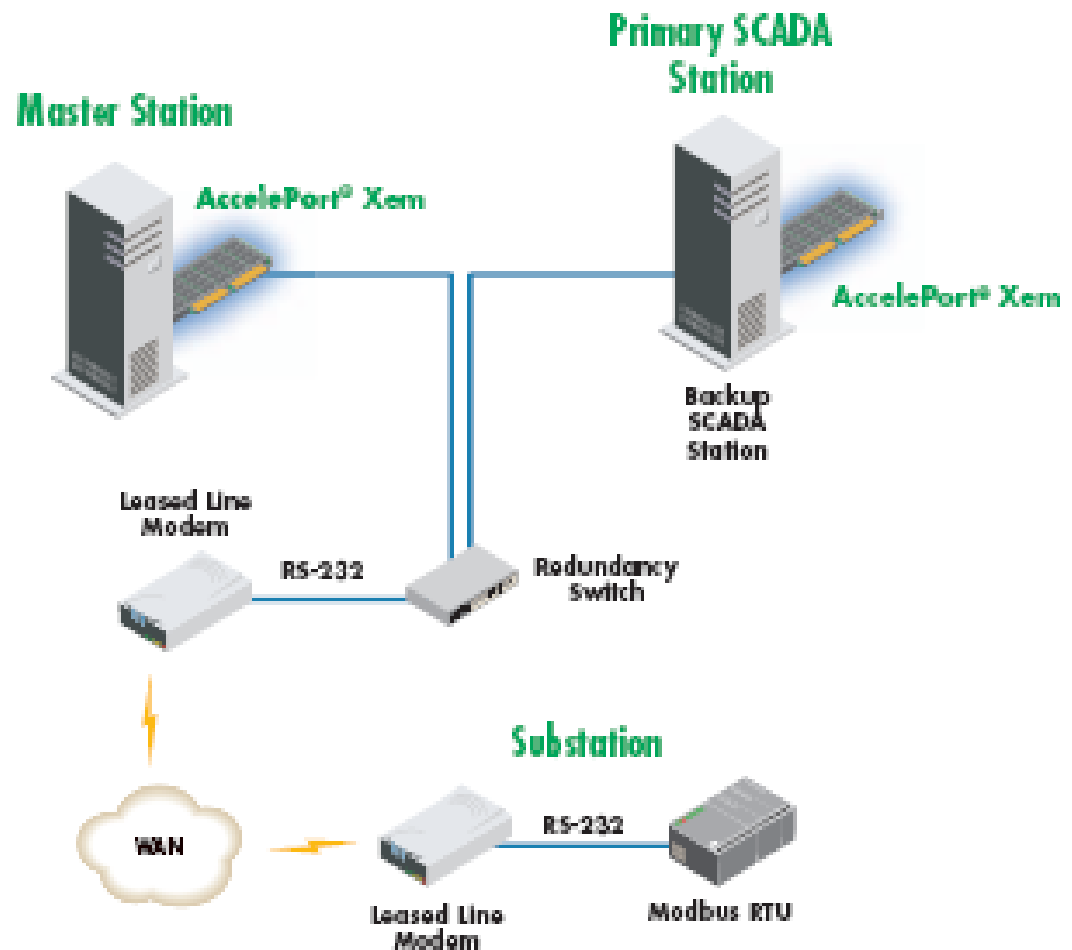
Petróleo Brasileiro S.A. (Petrobras)



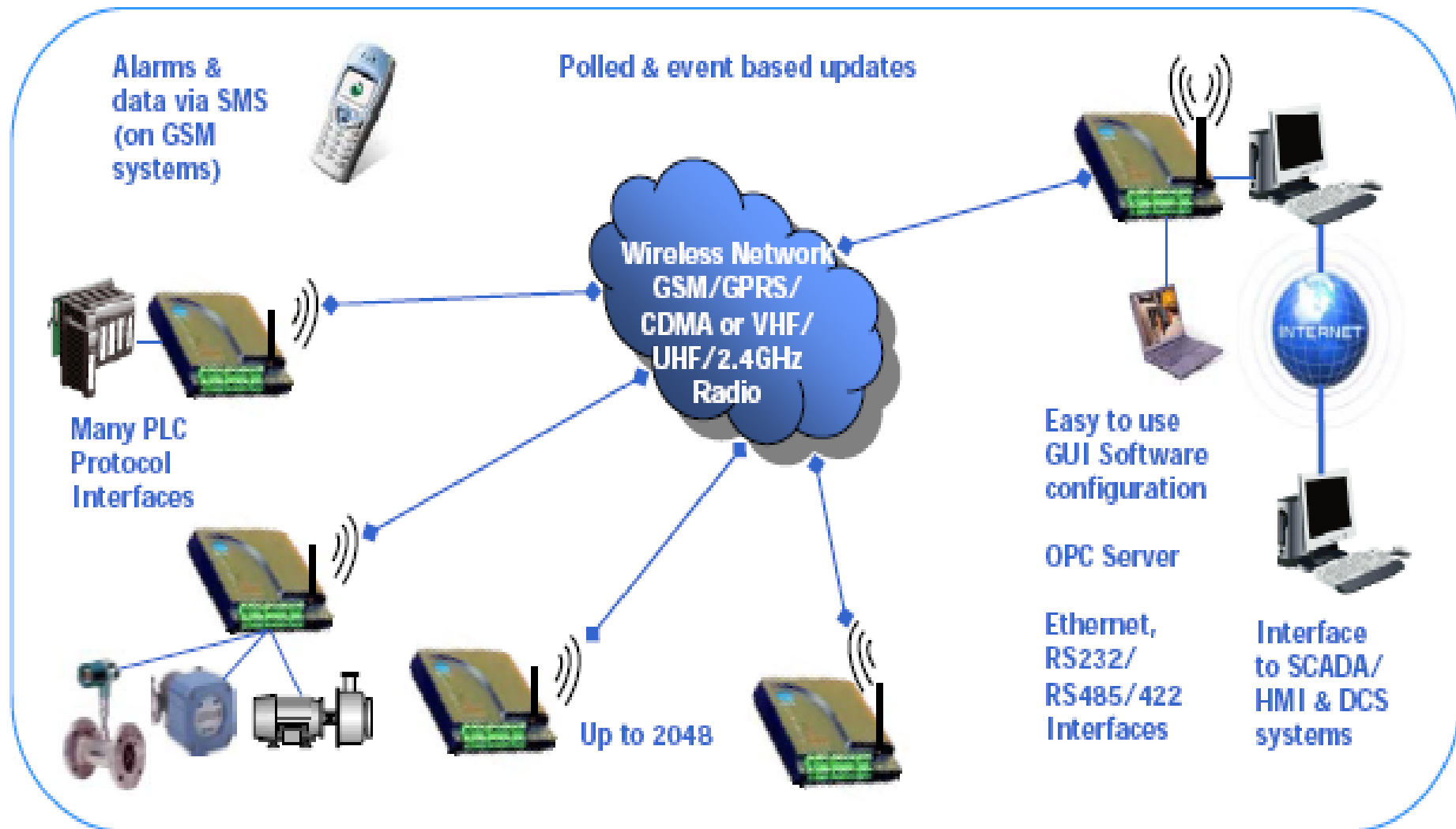
Ứng dụng điều khiển hệ thống chiếu sáng



Ứng dụng điều khiển hệ thống điện năng



SCADA qua vệ tinh



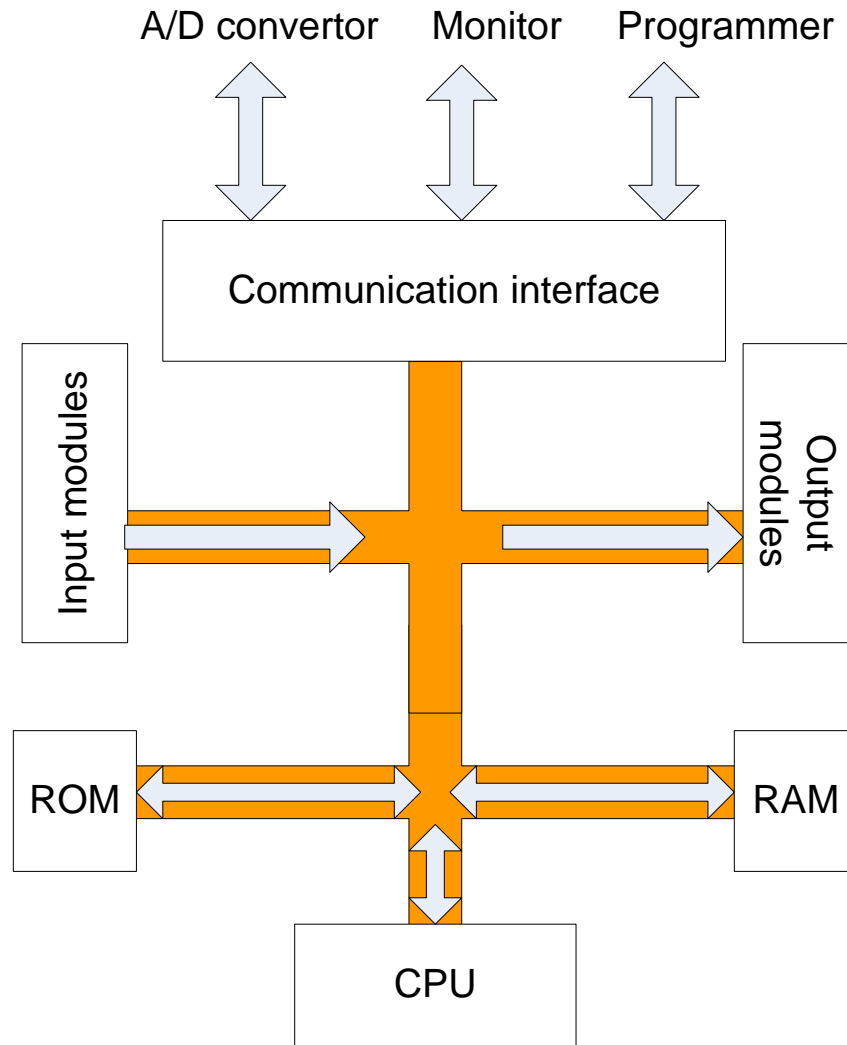
Supervisory Control & Data Acquisition

PLCs

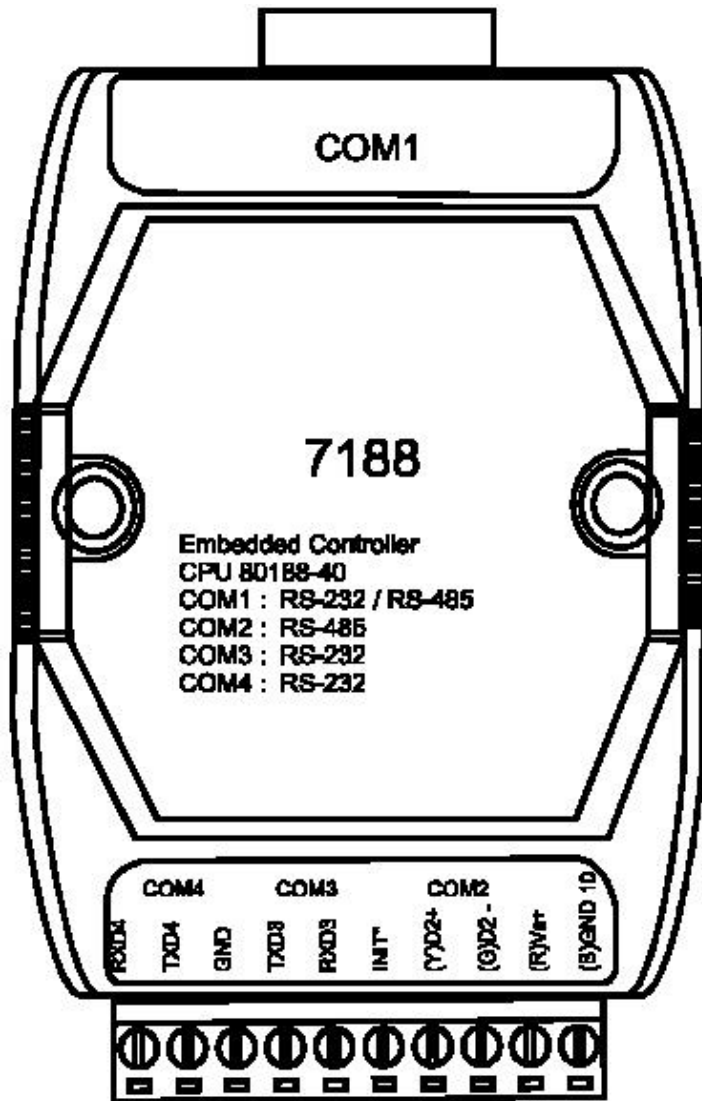


PLC

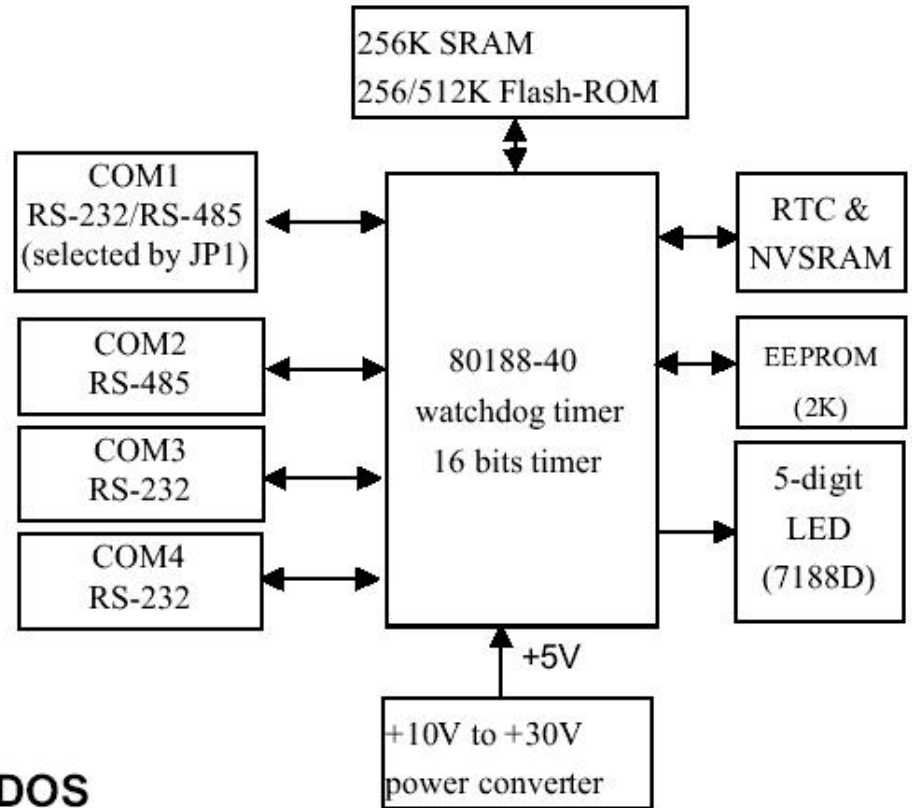
(cấu trúc)



PLC (cấu trúc)



7188D/DOS

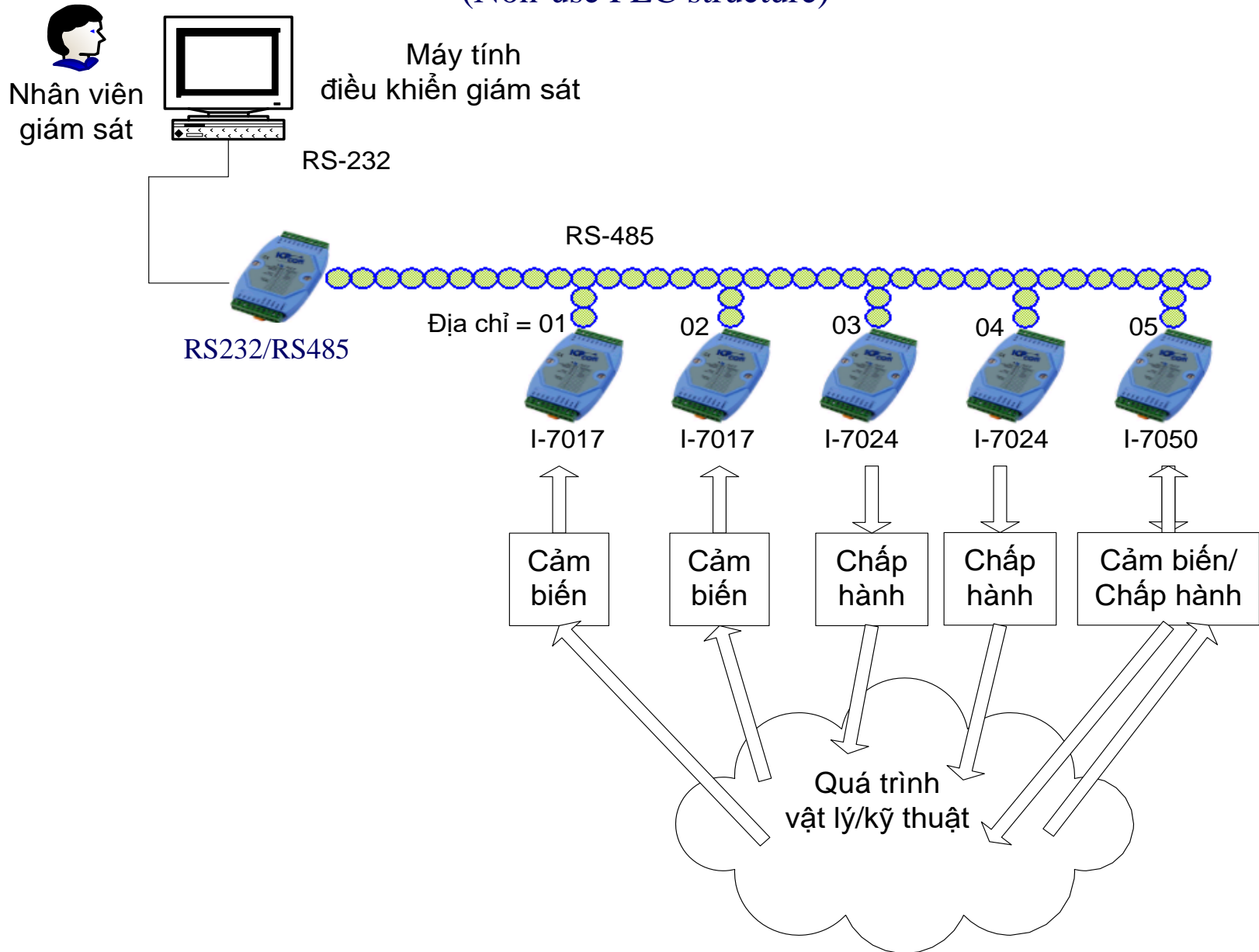


PLC (View)



PLC

(Non-use PLC structure)

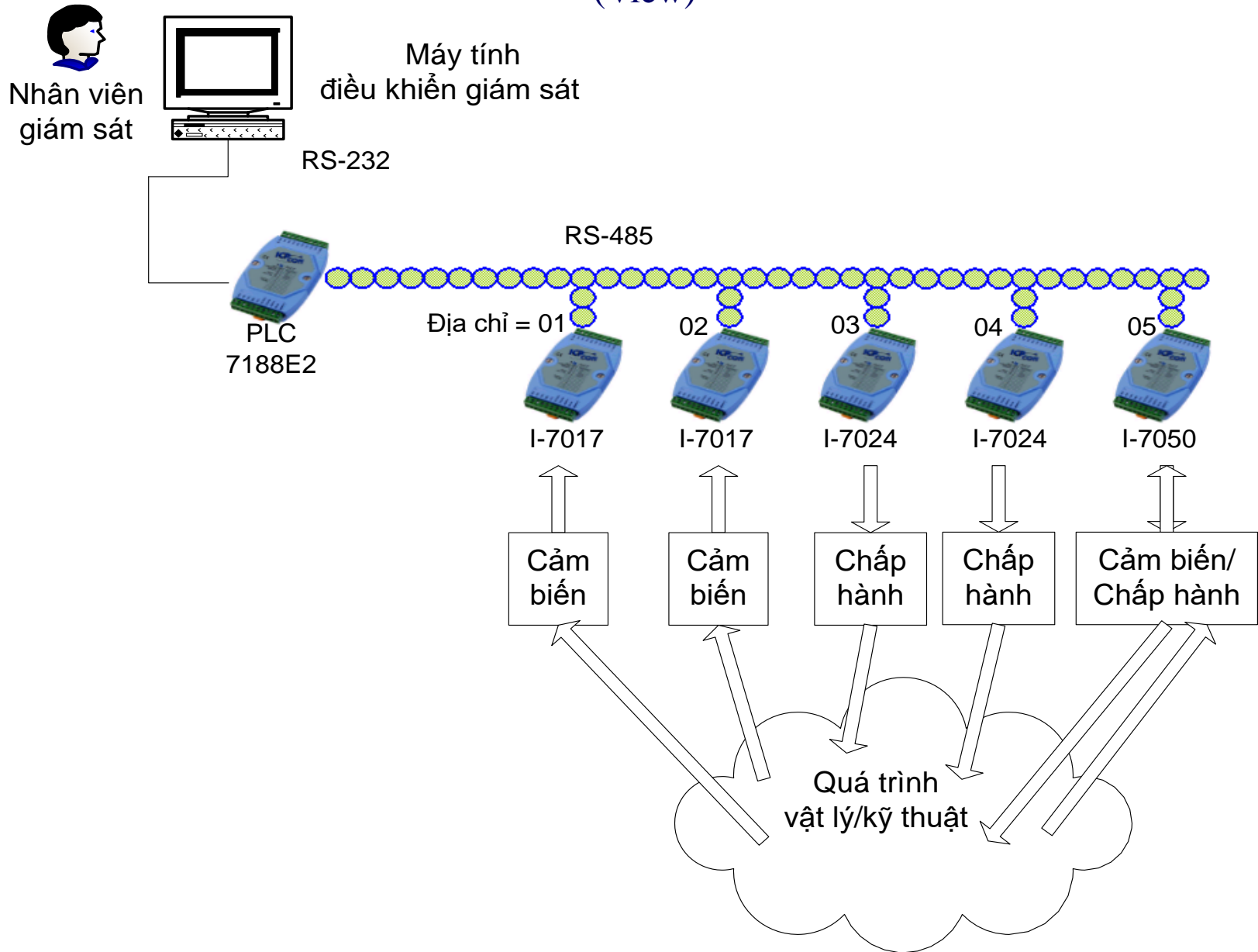


PLC

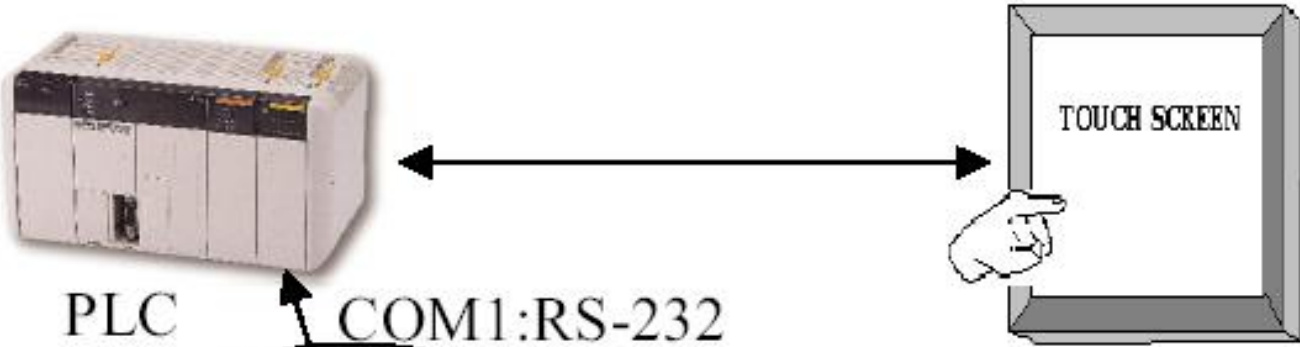
(RS232/RS485)



PLC (View)



PLC
(View)

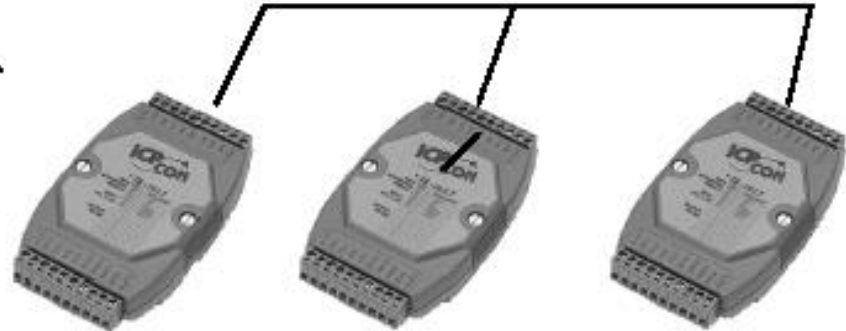


PLC COM1:RS-232



COM2:RS-485

7000 series



PLC (View)

Inline controllers



Remote field controller
for Ethernet



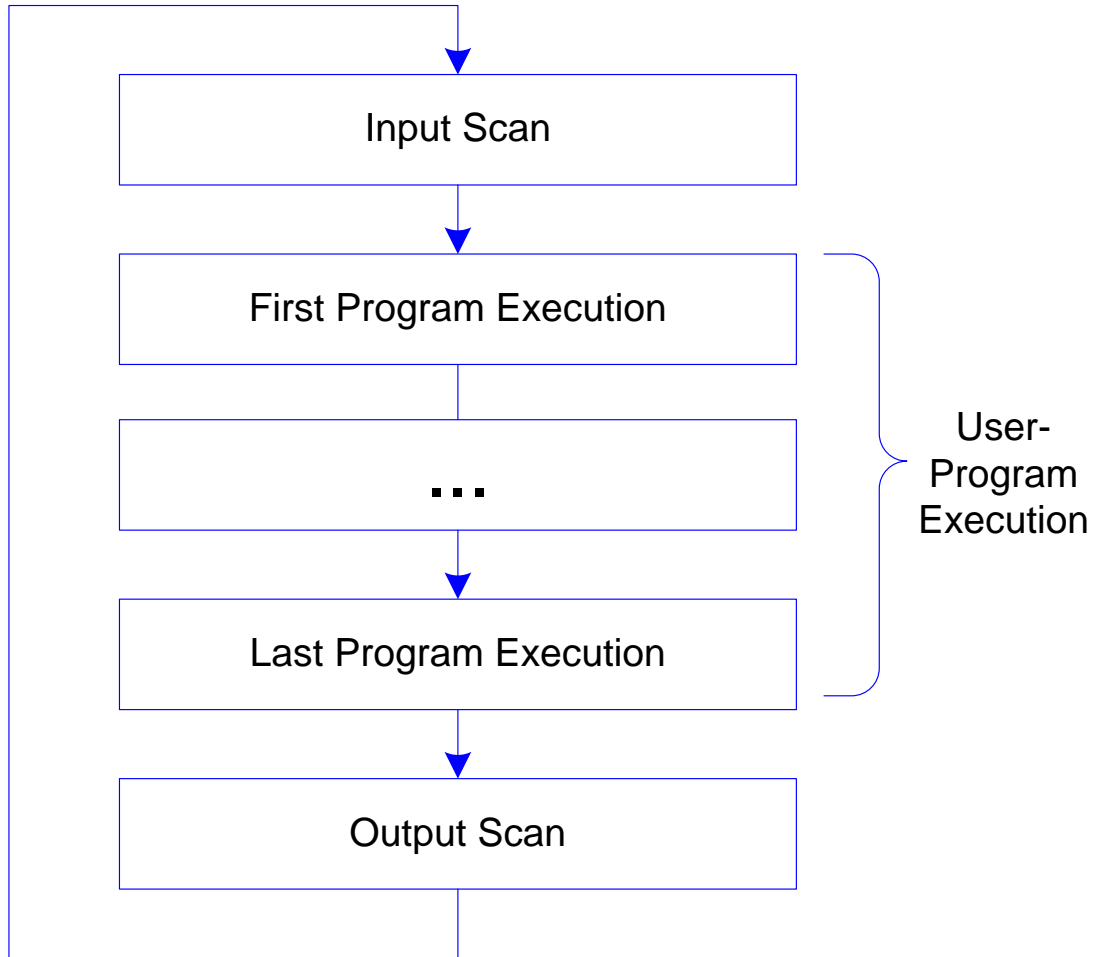
PLC (View)



A powerful combination
of PC platform, PLC,
without display,
MODBUS TCP master

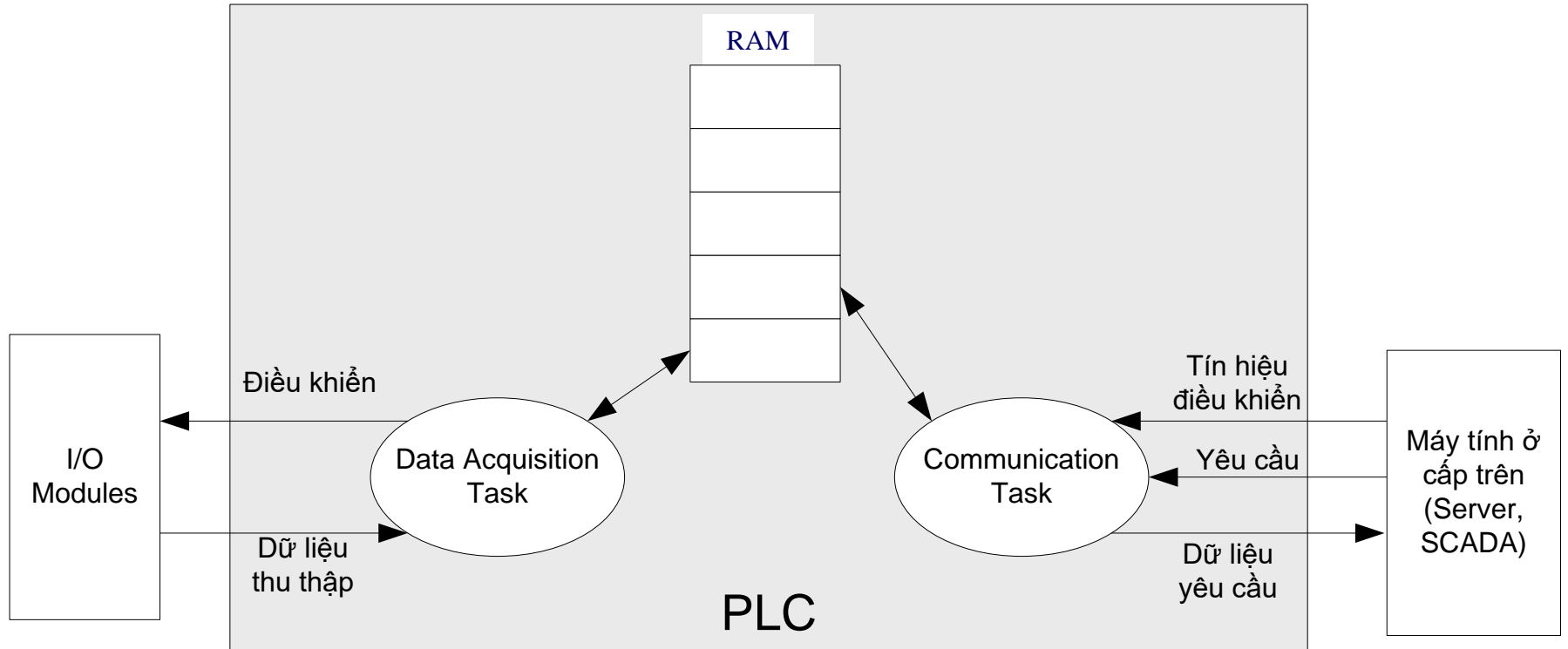
PLC

(DAQ Task)



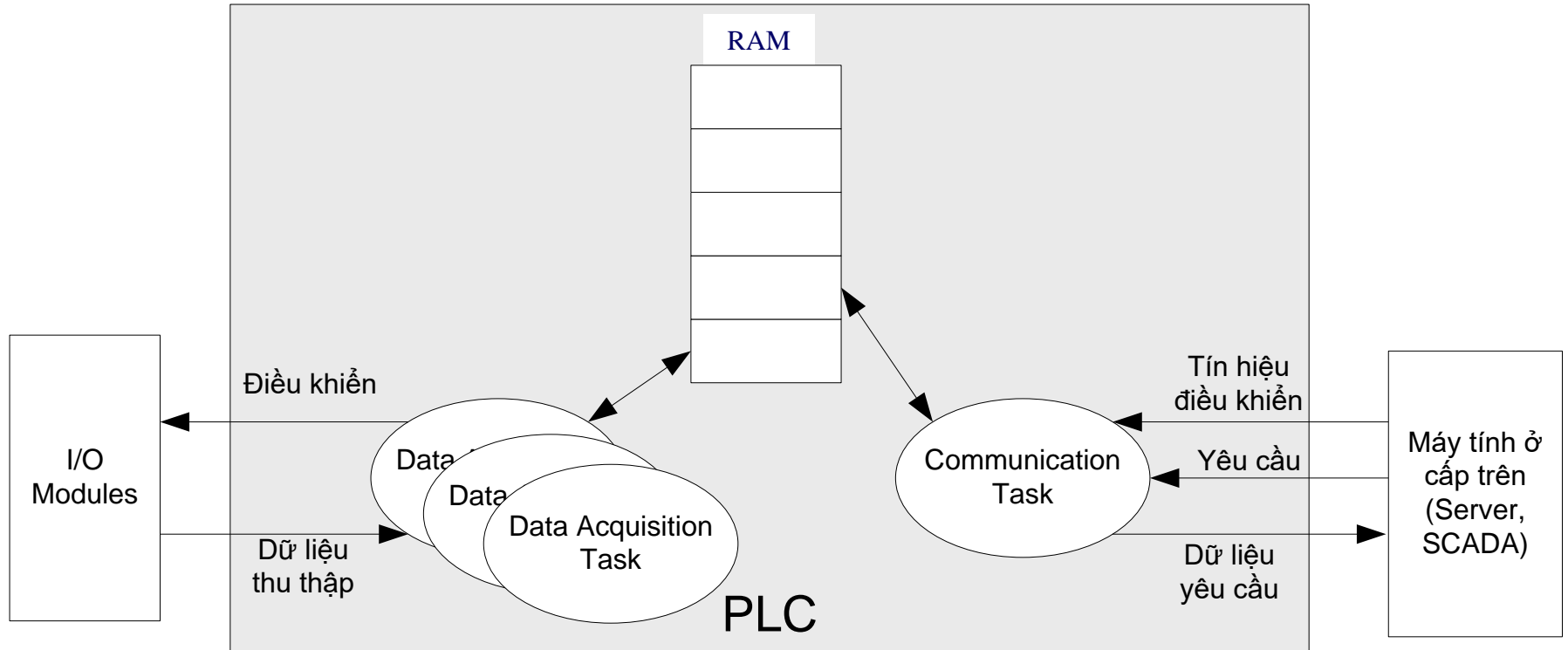
PLC

(DAQ Task và Comm Task)

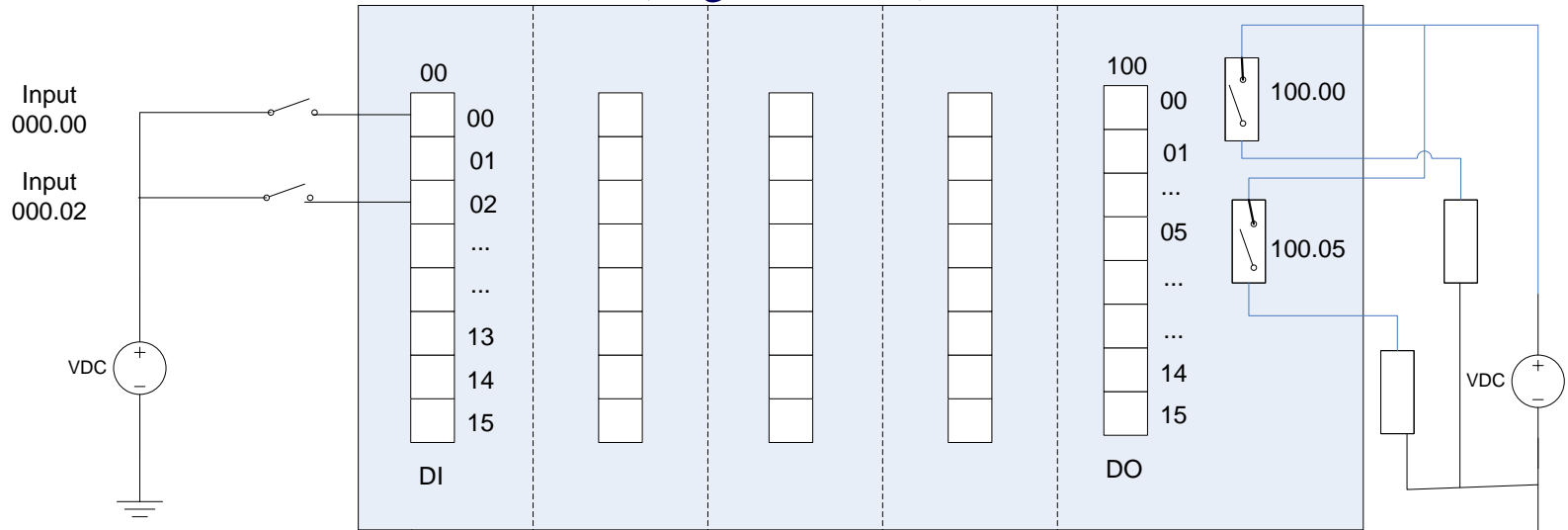


PLC

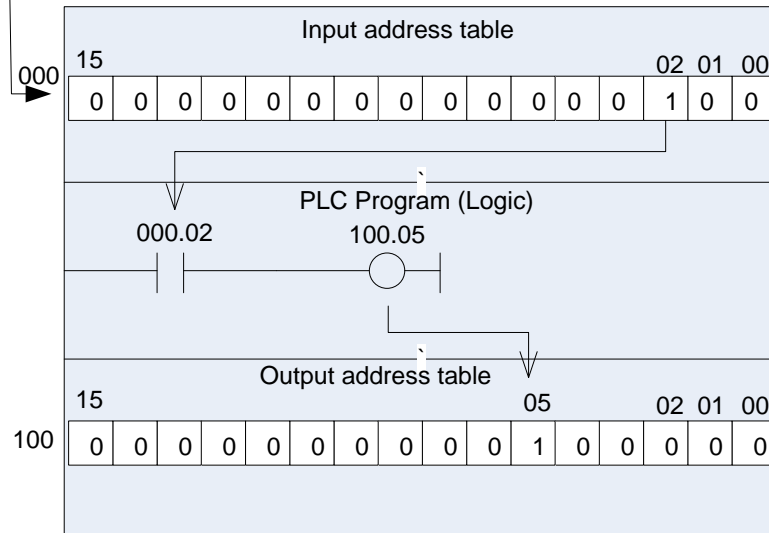
(DAQ Task và Comm Task)



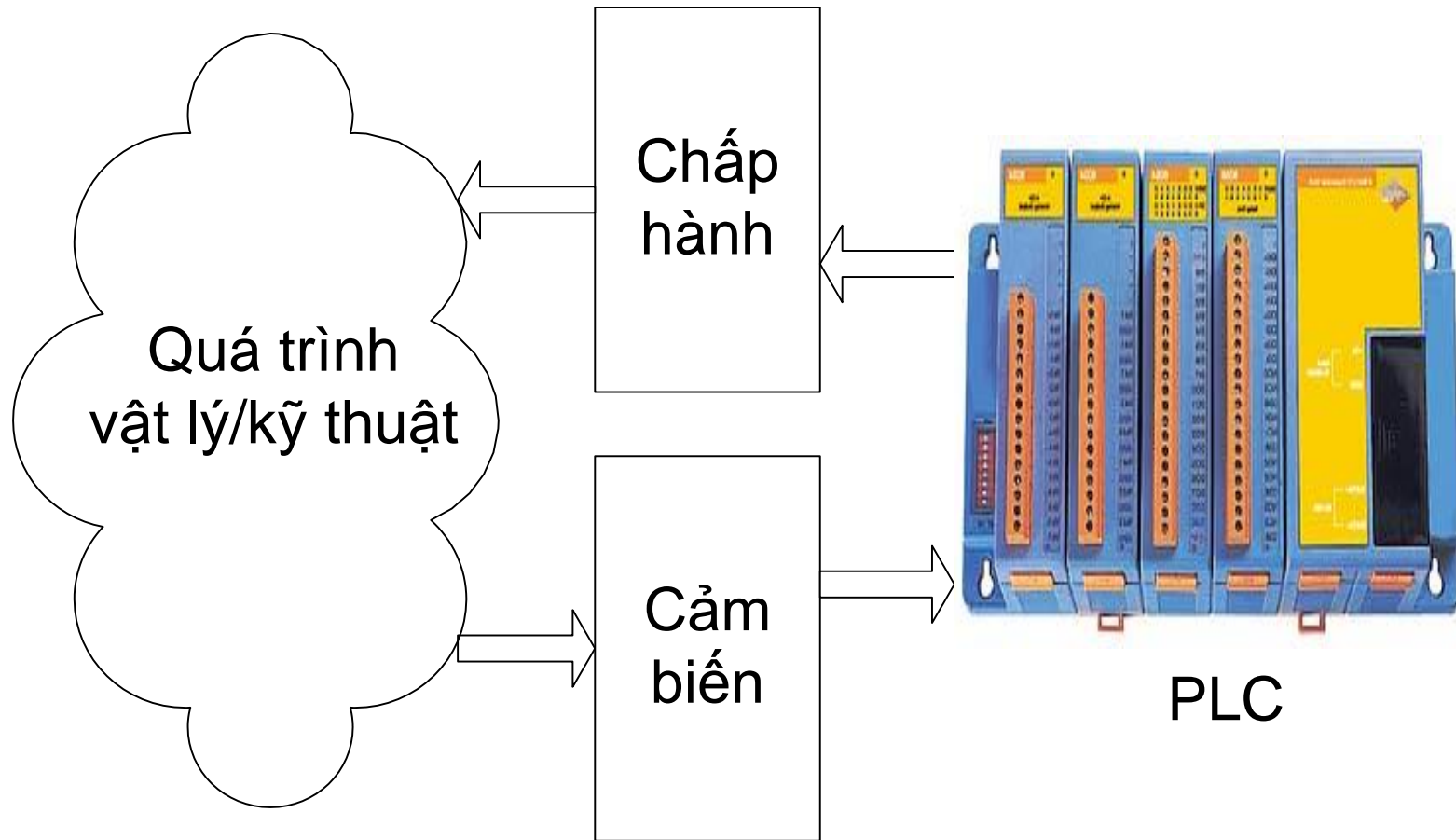
PLC (Logic Control)



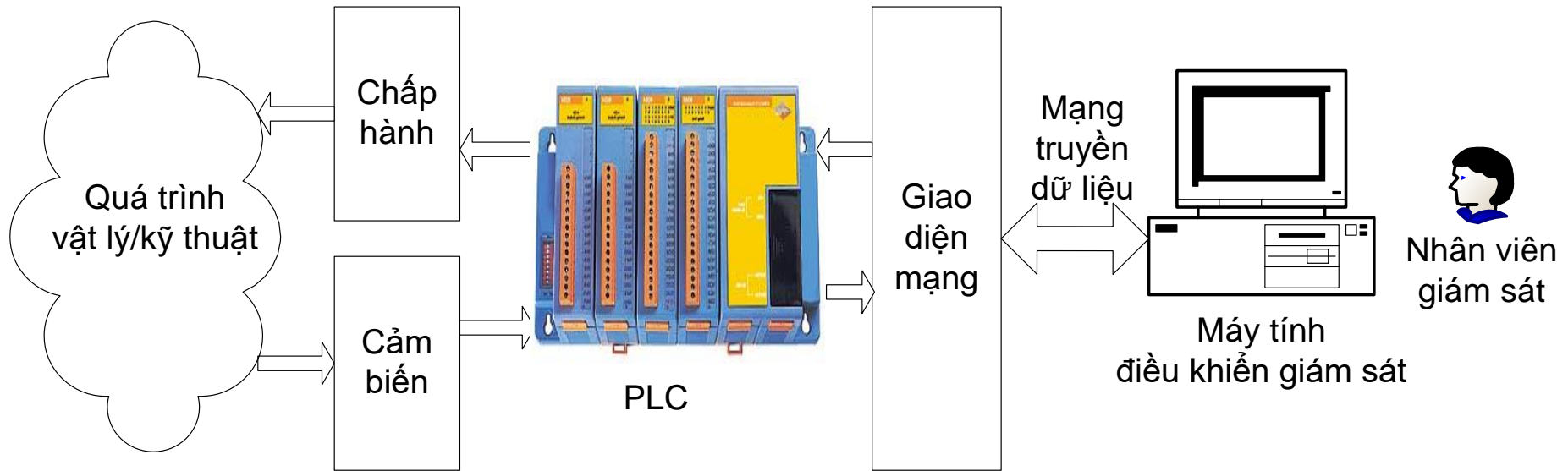
Memory in PLC



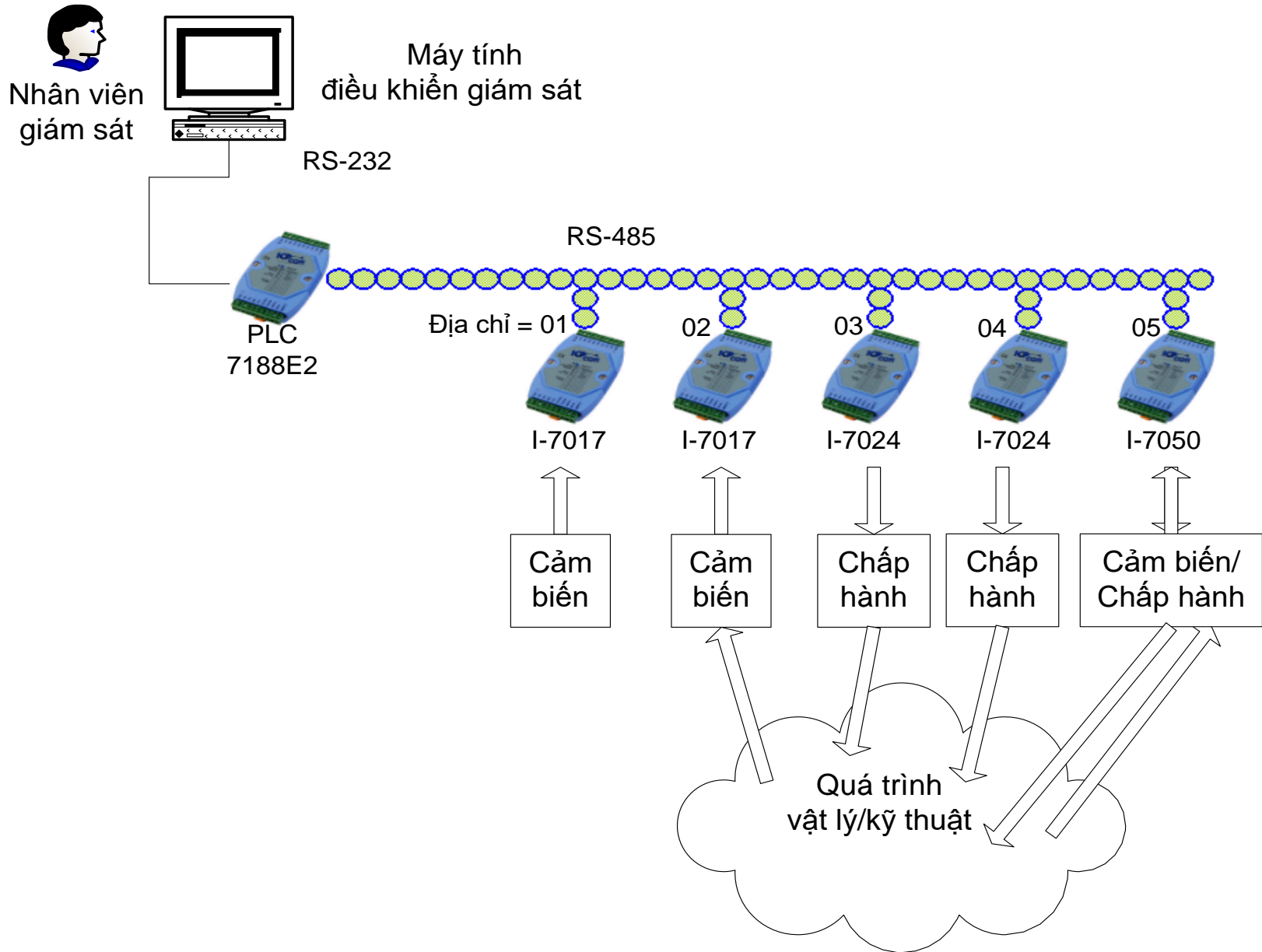
Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



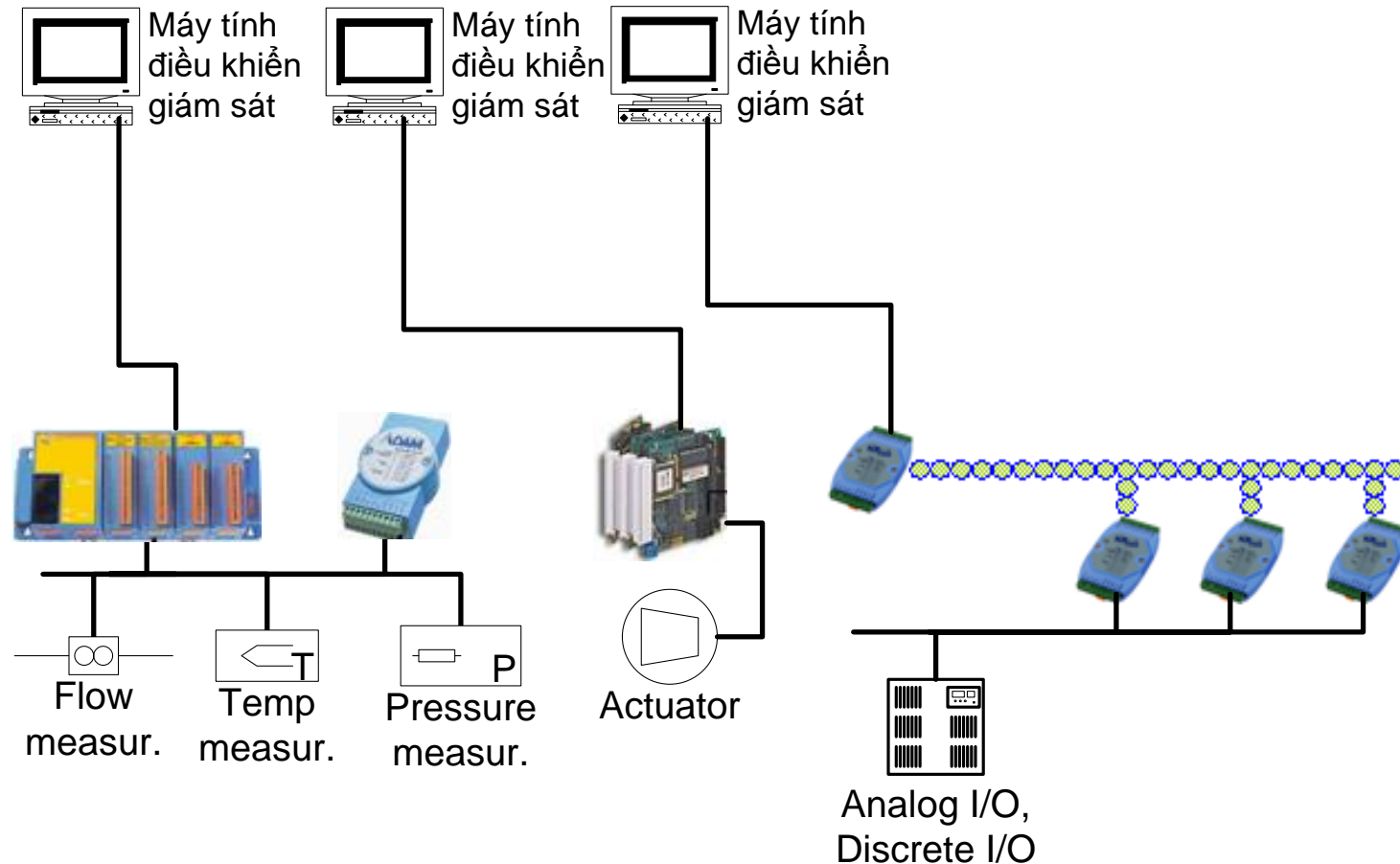
Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



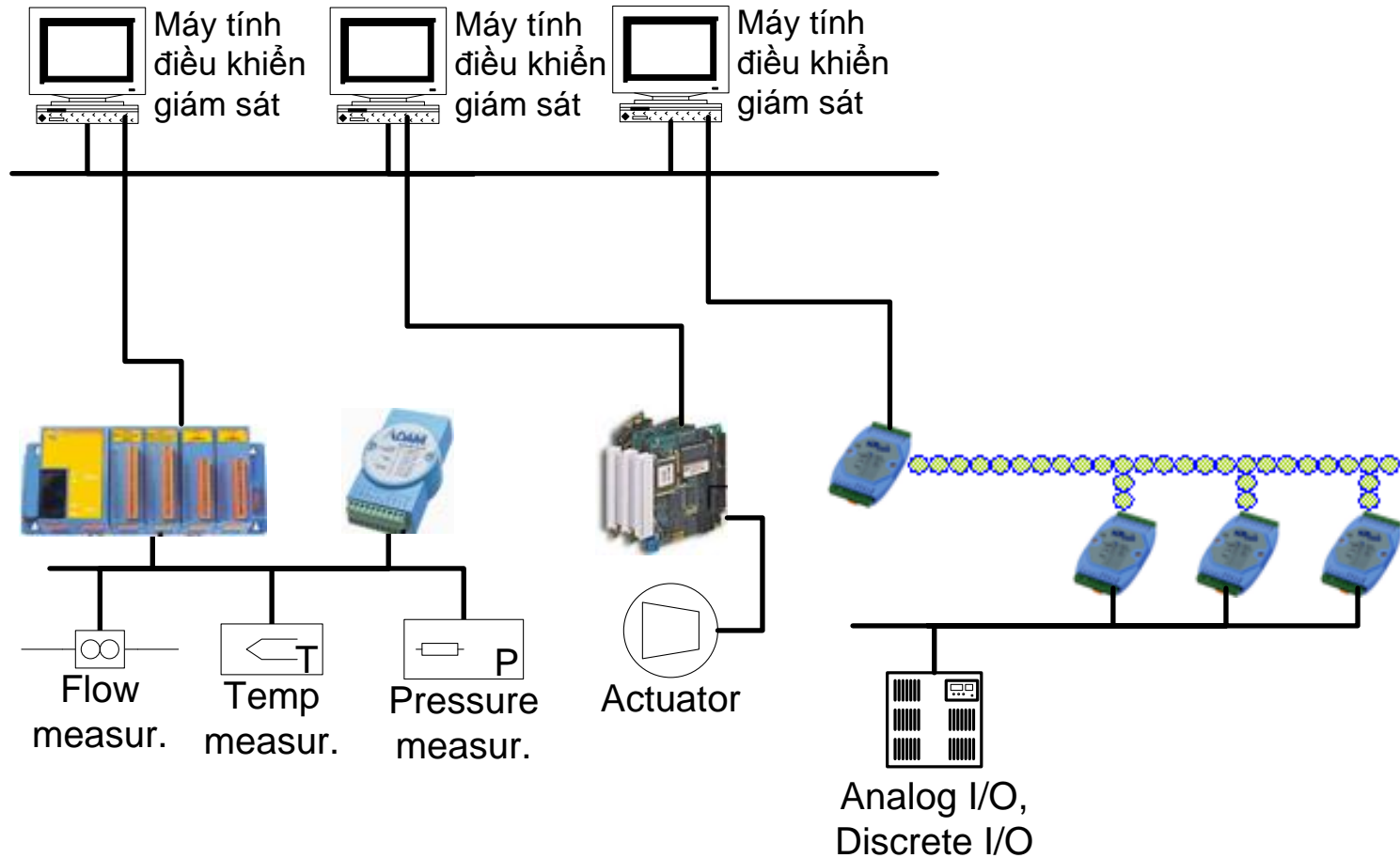
Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



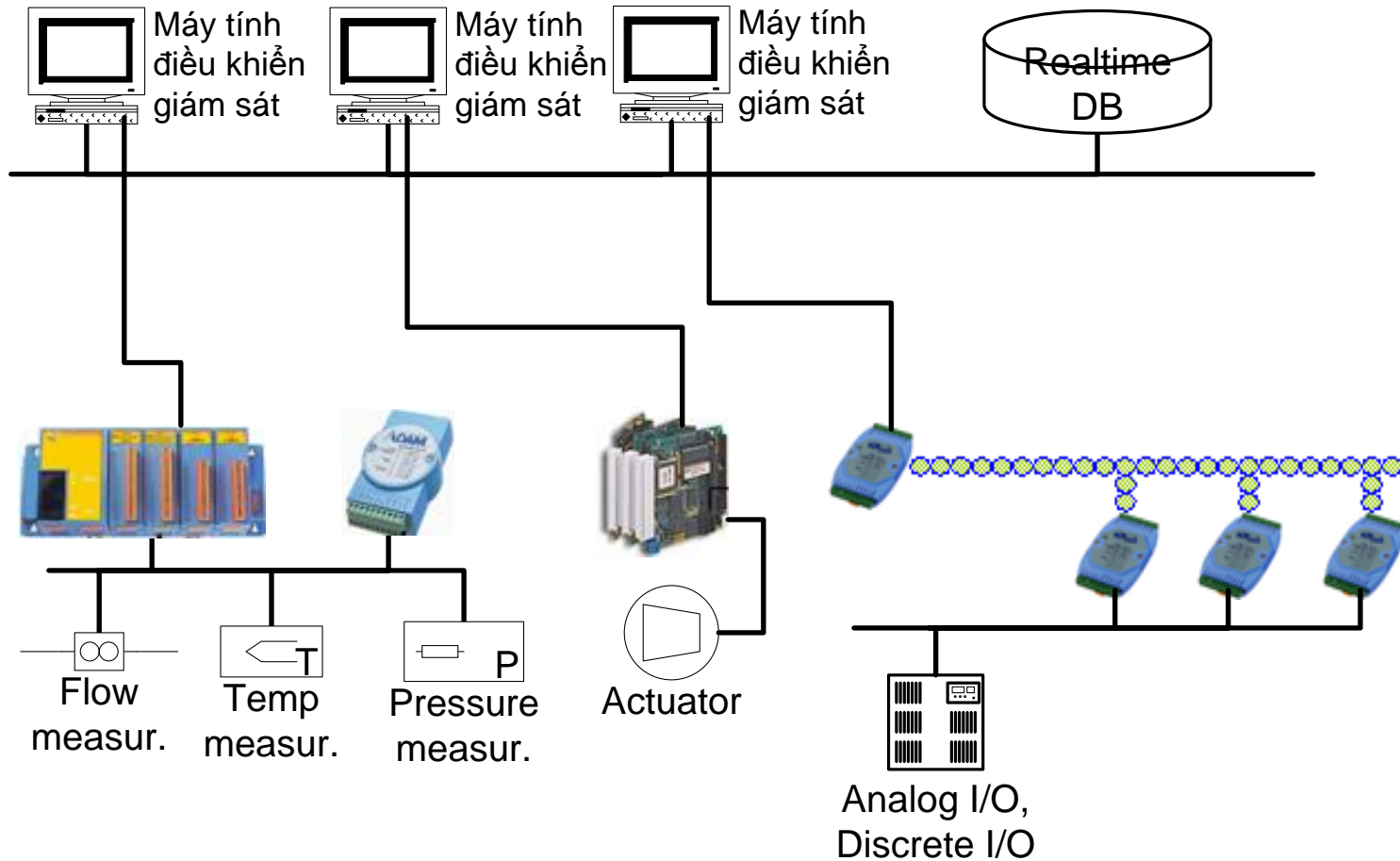
Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



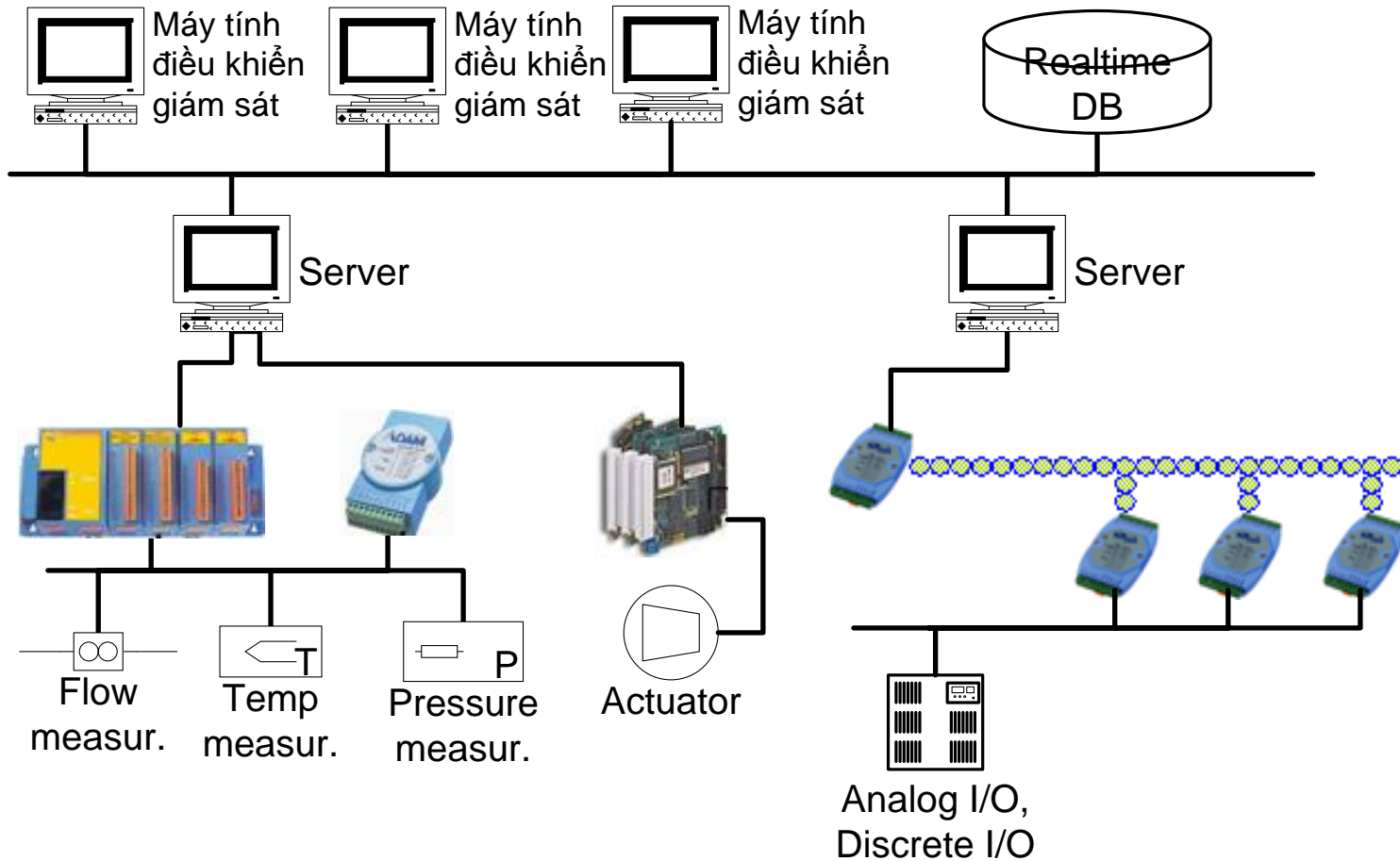
Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



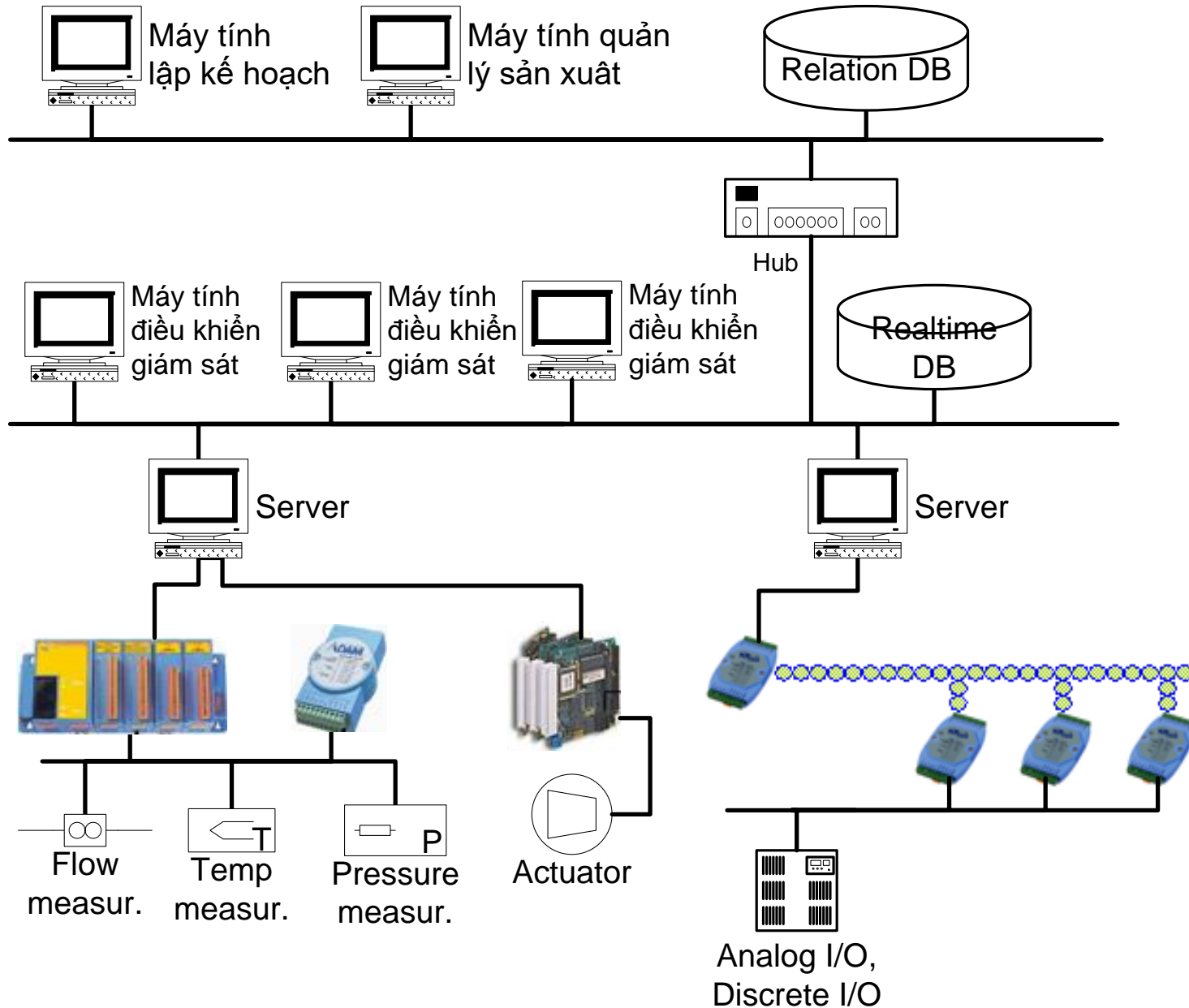
Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



Các cấu trúc hệ thống thu thập dữ liệu & điều khiển bằng máy tính (Evolution)



Máy tính công nghiệp (IPC)

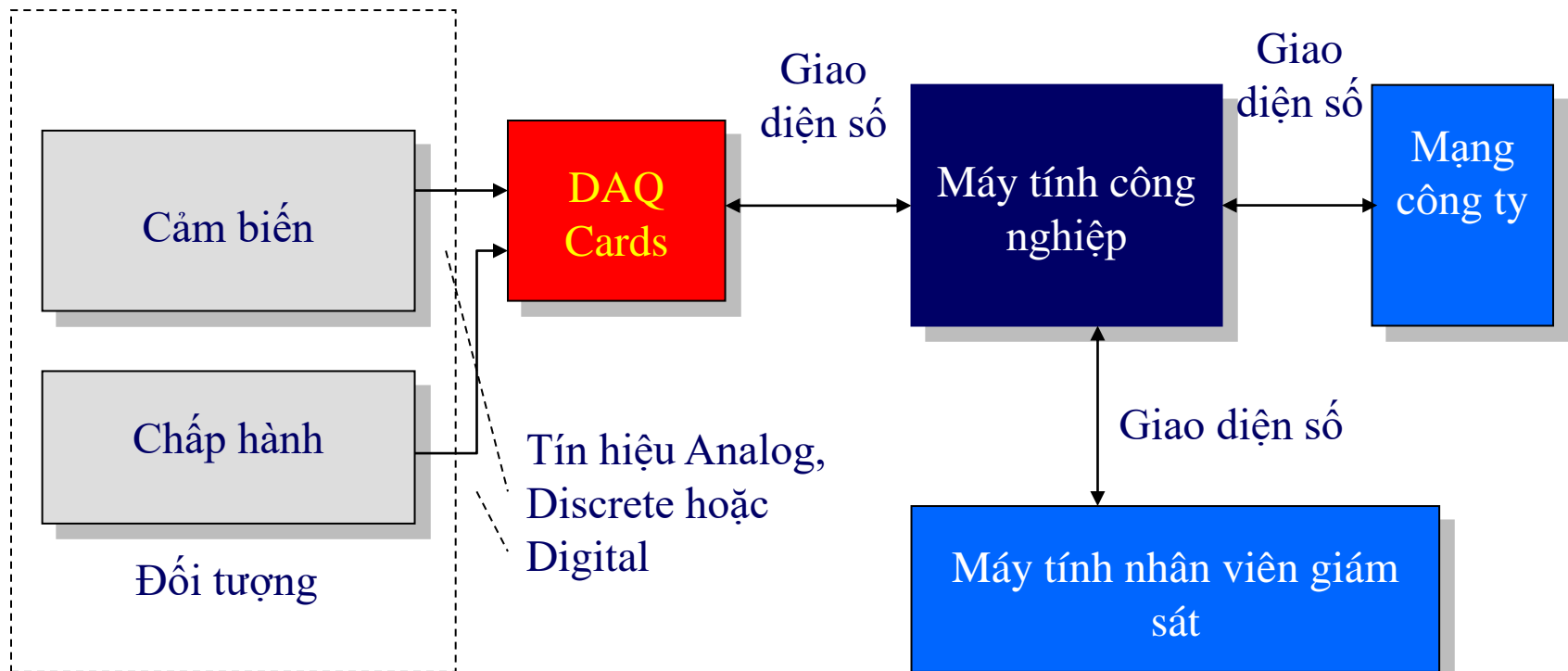


Supervisory Control & Data Acquisition

Data Acquisition Cards & Accessories

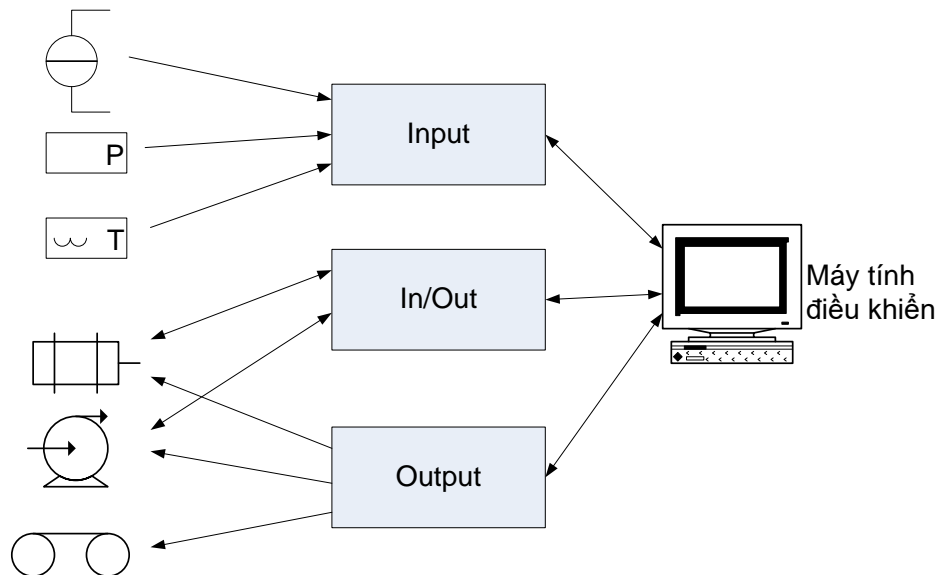
Sự cần thiết của DAQ Cards

- Hầu hết các thông số quá trình công nghệ của các đối tượng công nghiệp thực có dạng analog hoặc discrete.
- Tồn tại rất nhiều loại cảm biến mà chỉ có thể biến đổi đại lượng đo được thành dạng analog.
- Tồn tại rất nhiều loại cơ cấu truyền động, chấp hành chỉ hỗ trợ tín hiệu vào là tín hiệu analog.
- Các thành phần khác của hệ thống tự động đều sử dụng phương thức số để thu nhận, xử lý và điều khiển.

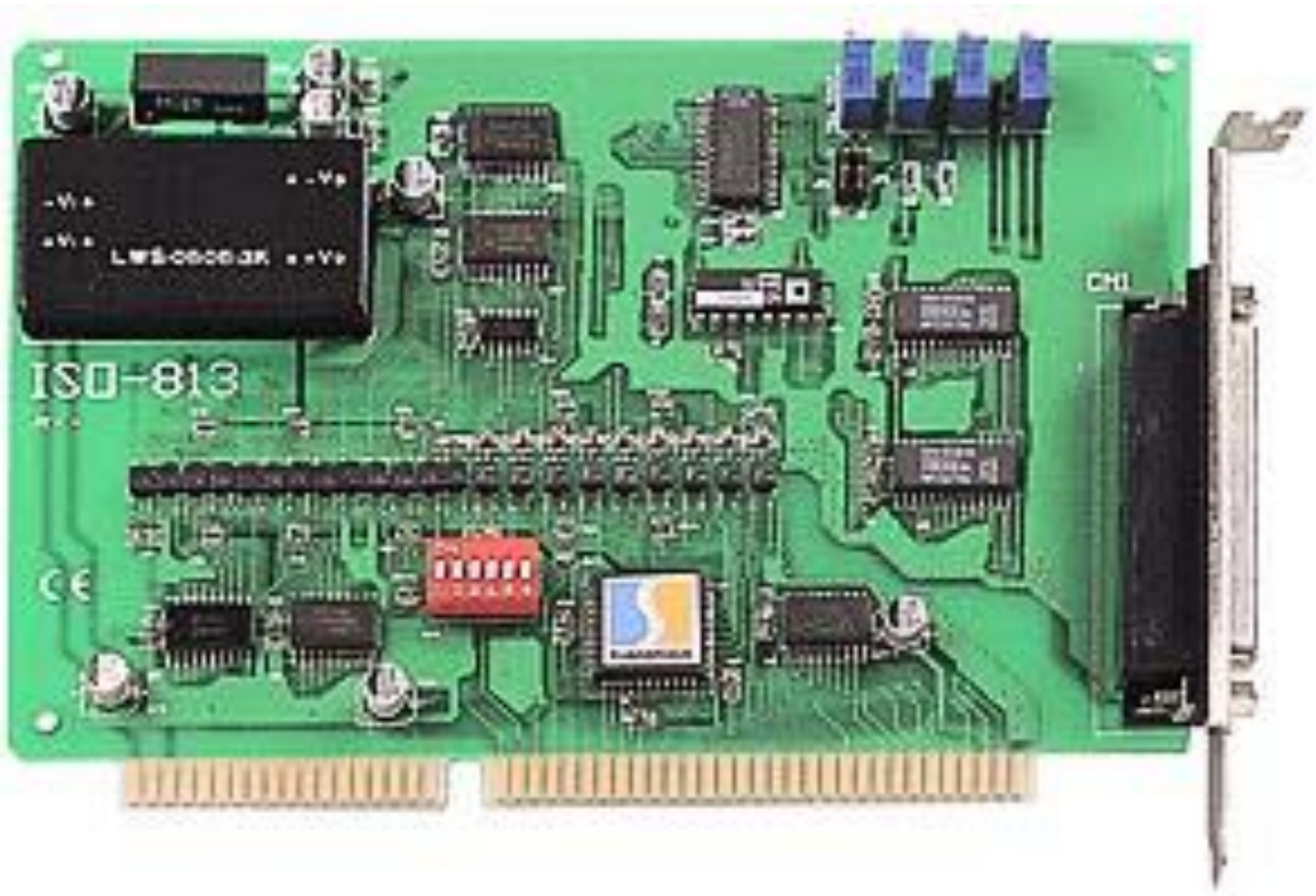


Chức năng của DAQ Boards

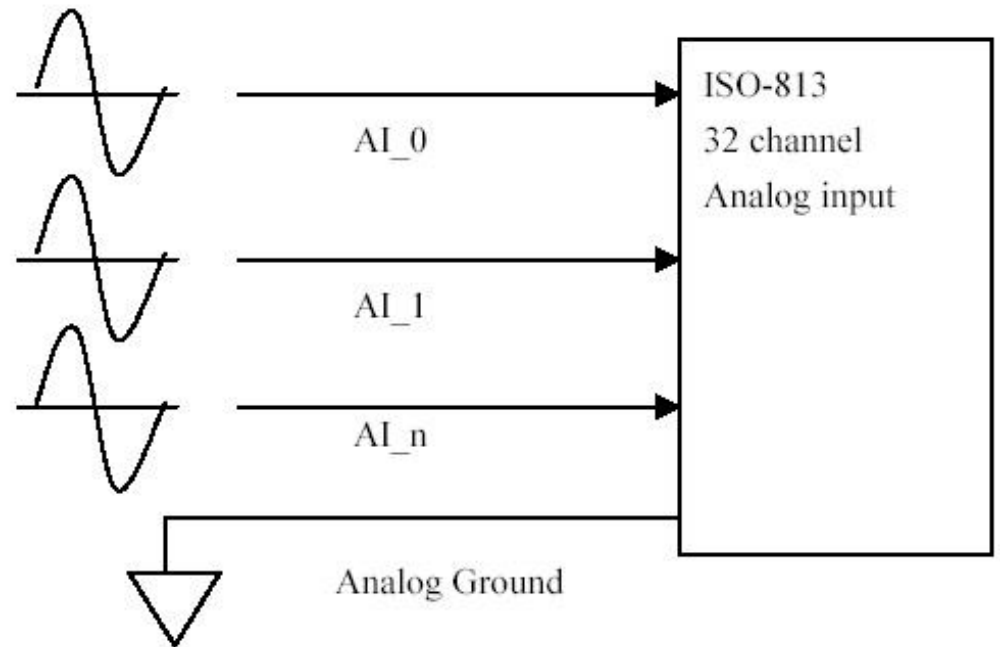
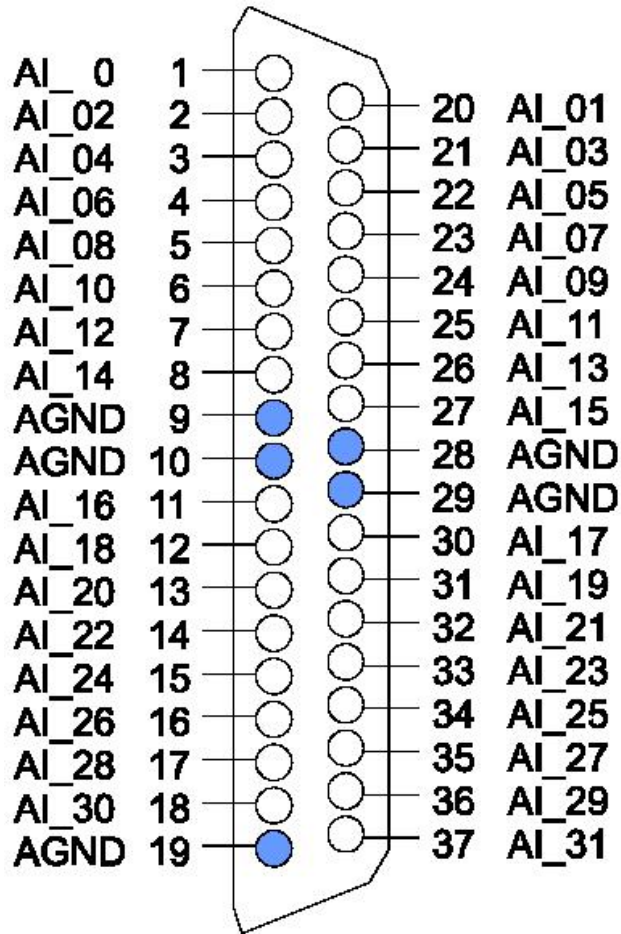
- thu thập và chuẩn hoá tín hiệu analog (dòng và thế);
- thu thập trạng thái ngõ vào discrete;
- lọc tín hiệu thu nhận được từ ngõ vào;
- đưa ra các tín hiệu analog (dòng và thế) và tín hiệu discrete;
- biến đổi analog-digital (module analog input);
- biến đổi digital-analog (module analog output);
- biến đổi thang đo đối với các thông số analog;
- tạo thông tin và truyền theo địa chỉ chỉ định trước.



ADC



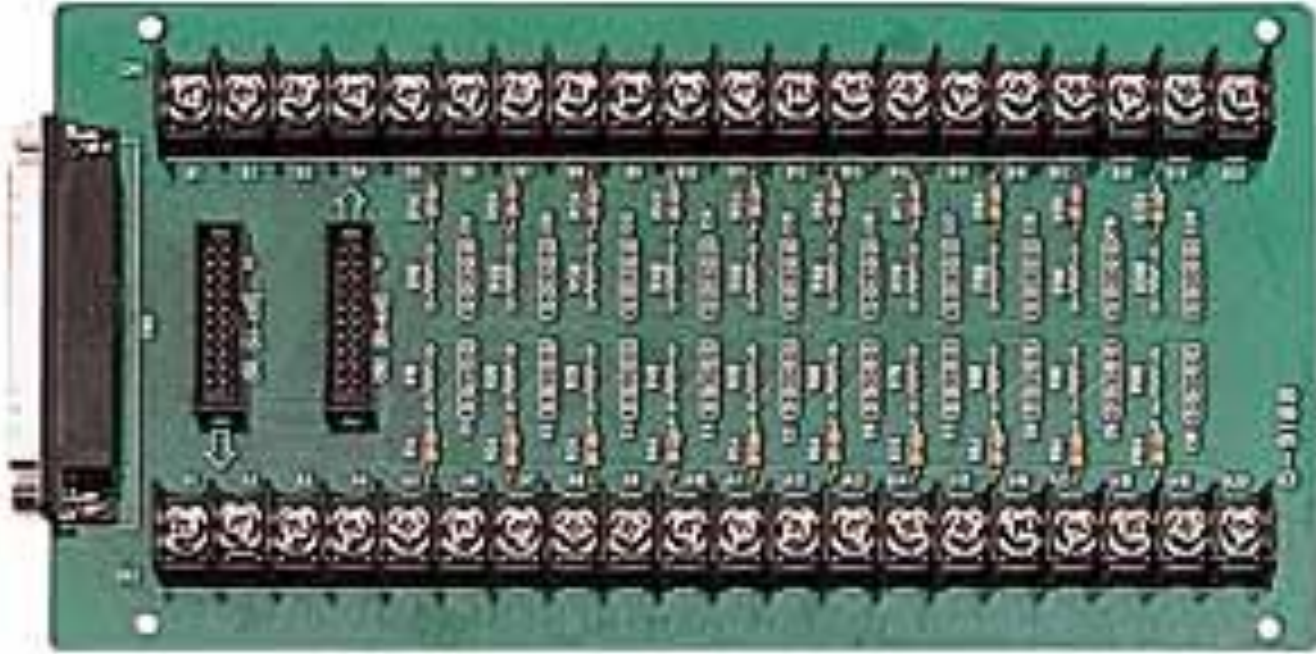
ADC



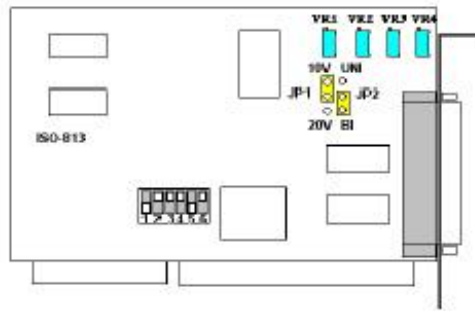
DAC



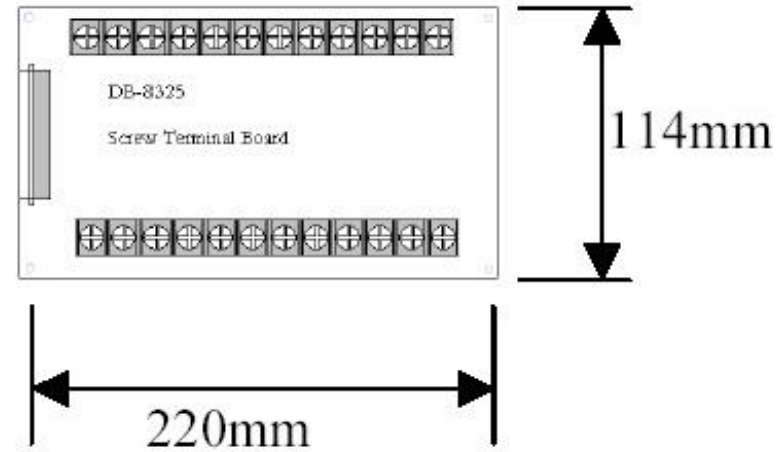
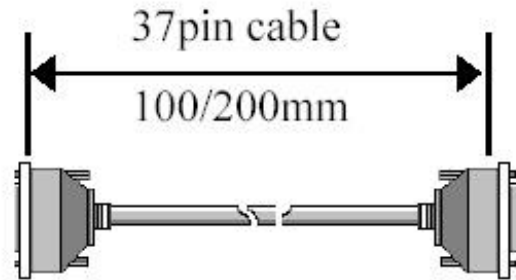
Terminal Board và dây nối với DAQ Cards



Terminal Board & DAQ Cards



ISO-813



Cables



Twisted Pair



Coaxial

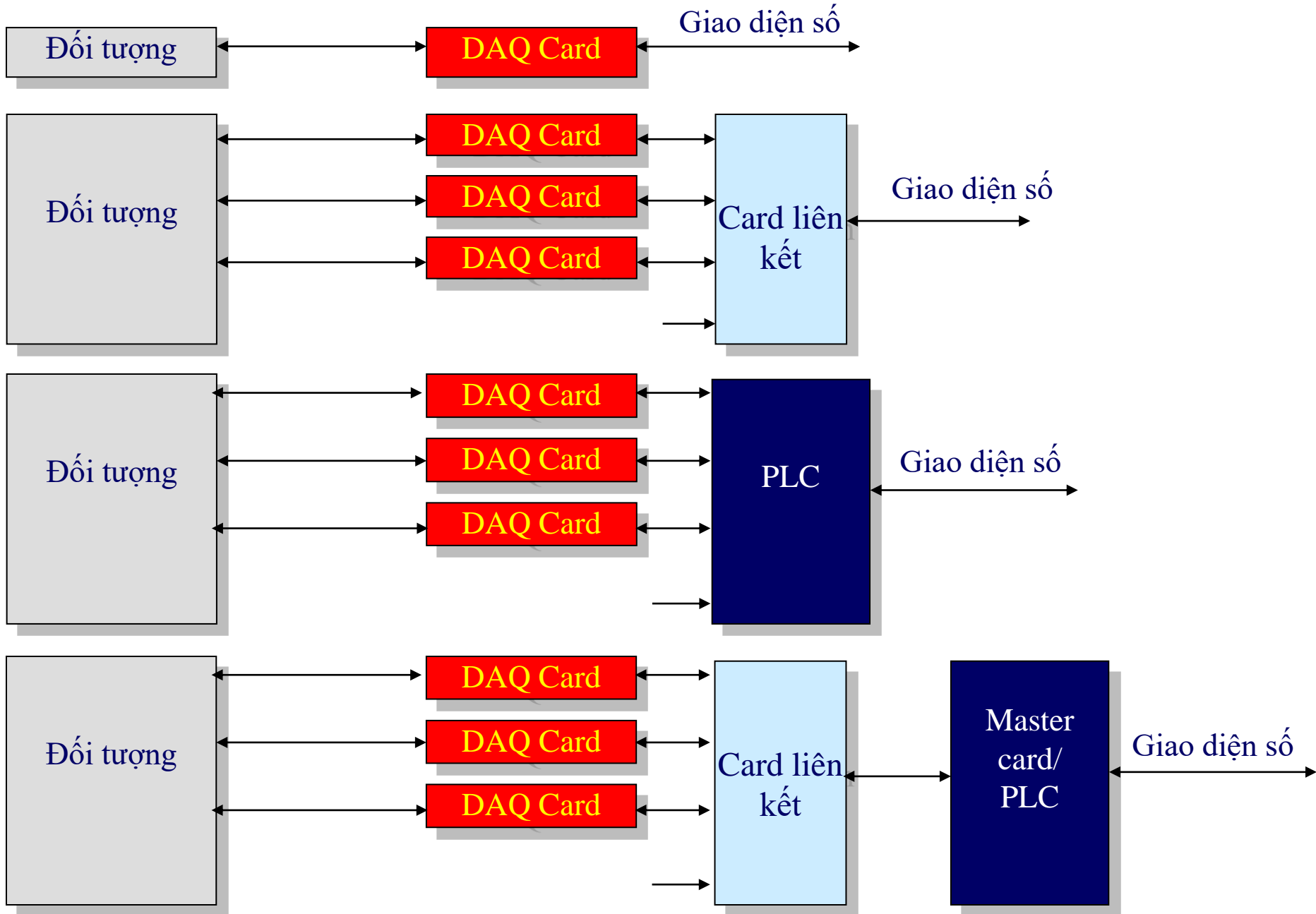


Ribbon

Fire Optics



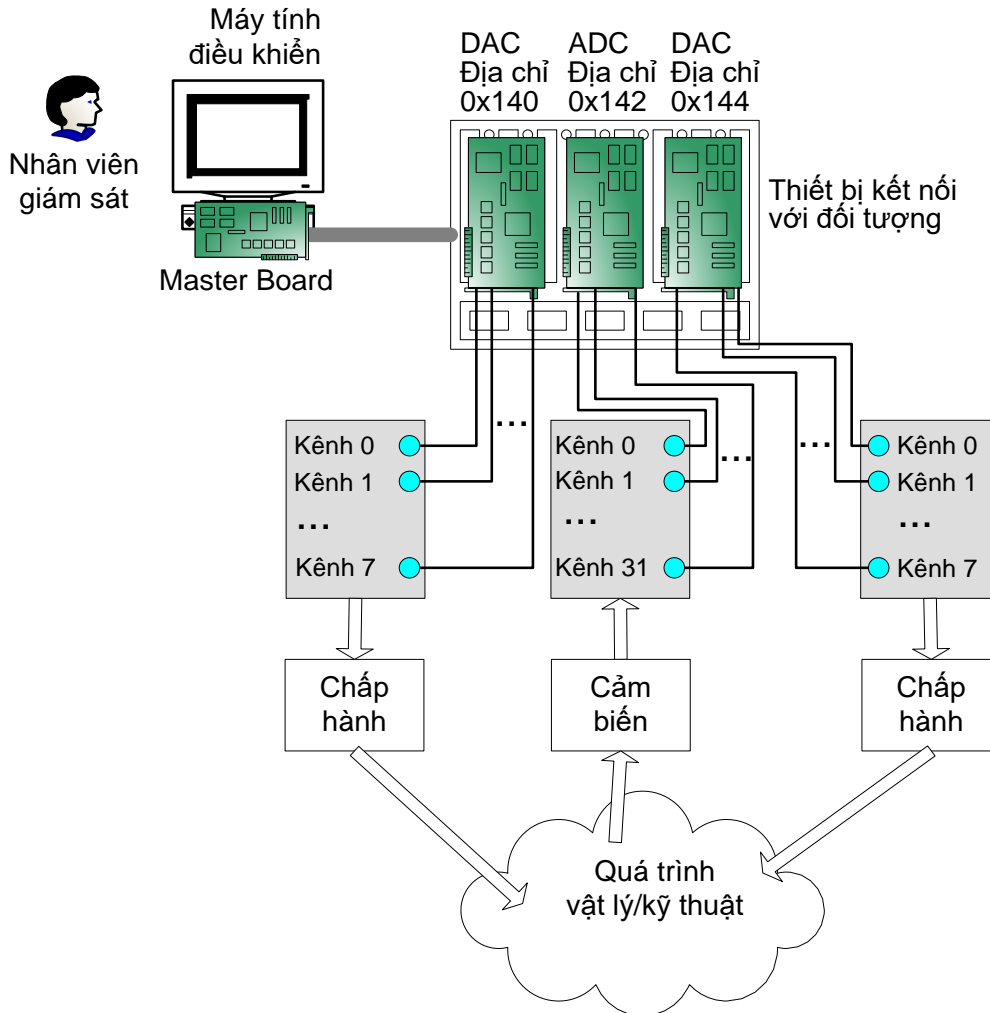
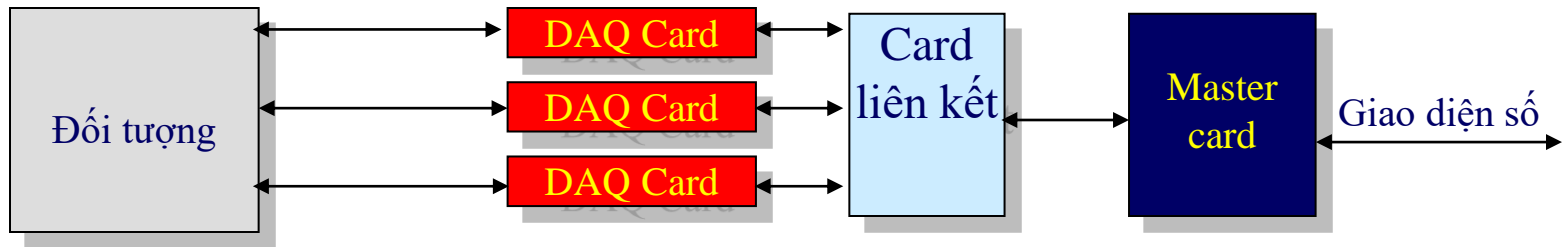
Cấu hình kết nối DAQ Card với các thiết bị khác



Card liên kết



Cấu hình kết nối DAQ Card với các thiết bị khác - ví dụ





Supervisory Control & Data
Acquisition

DAQ Networking

Elements in DAQ Networking

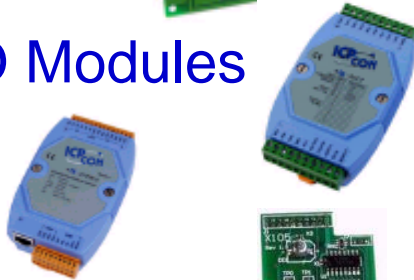
◆ Data Acquisition and Control Cards



◆ Master boards



◆ Remote I/O Modules



◆ Converter



◆ I/O Expansion Board



◆ Embedded Controllers

◆ Compact Embedded Controller



◆ Communication Controller



Feature of Remote I/O Modules

- ◆ Variant I/O Modules
- ◆ RS-485 Network
- ◆ 115.2K Baud Rate Maximum
- ◆ Self-Turner
- ◆ Good Expansibility
- ◆ Wide Voltage Source (10~30VDC)
- ◆ 3000 VDC Isolated Protection

Variant Remote I/O Modules

◆ Analog Input

- V
- mV
- mA
- Thermocouple

◆ Analog Output

- V
- mV
- mA

◆ Digital Input

- Isolated
- Non-Isolated

◆ Digital Output

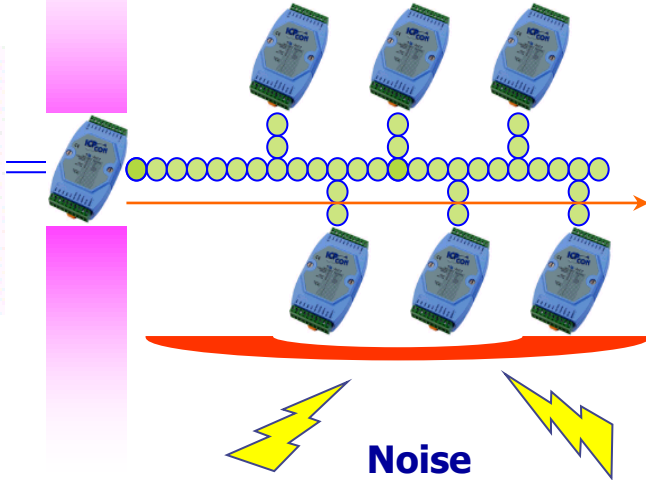
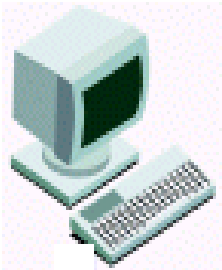
- Photo
- Open Collector
- Relay

◆ Counter

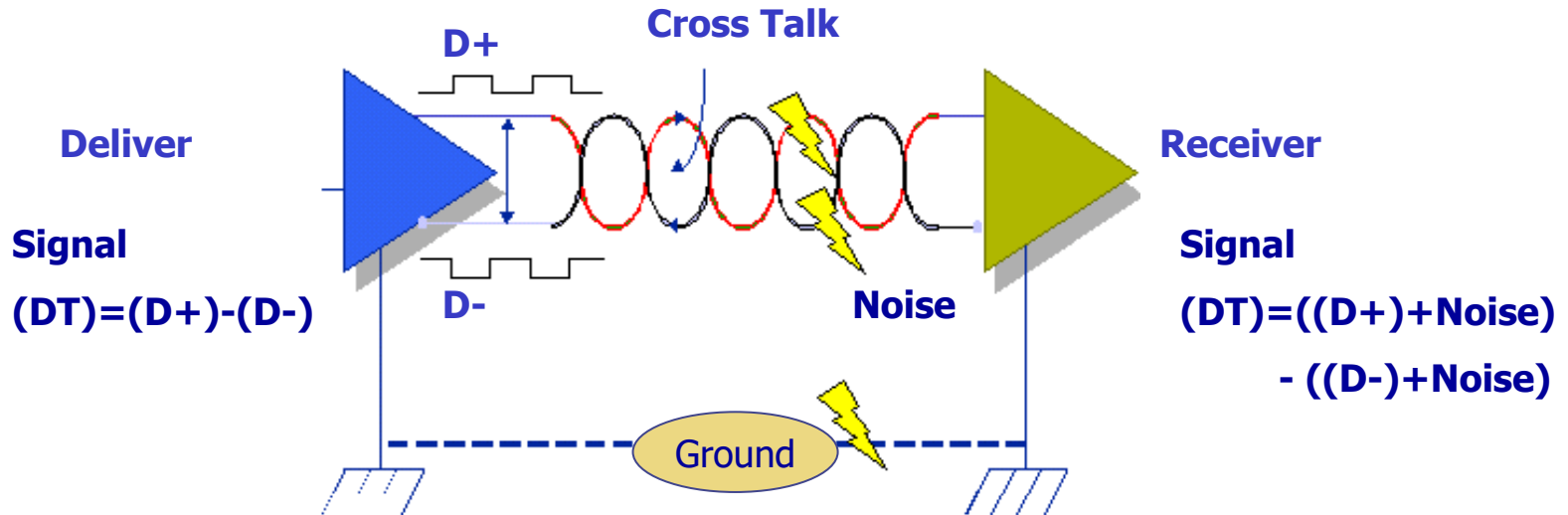
- Counter
- Frequency

Noise Resistance

Isolated Protection



1.2Km (up to 4.8Km by Repeater)



Self-Tuner

- ◆ Variant Baud Rate
- ◆ Variant Data Format
- ◆ Auto-Change Direction



RS-485 type PLC
Data bit: 12 bits
38400 bps



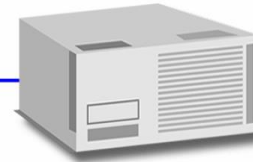
RS-485 type PLC
Data bit: 12 bits
9600 bps



Remote modules
Data bit: 10 bits
can be 1200, 2400
....., 115200 bps



Wireless modem
Data bit: 10 bits
19200 bps



RS-232 Device
Data bit: 11 bits
9600 bps

Interface converter, for converting
RS-232 to RS-422 and RS-485,
with electrical isolation, 2 channels



Bidirectional, wireless, 2400 MHz transmission system (European version), consisting of 2 transceivers (transmitter and receiver) and two antennas with connecting cable



Factory Line Wireless Bluetooth Access Point



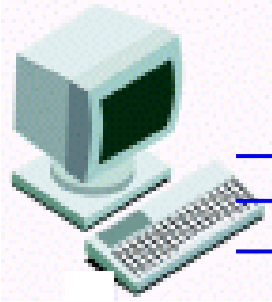
Fieldline Modular Wireless IO set, consisting of: Wireless IO base station for one Wireless IO terminal and Wireless IO device, 8 digital inputs, 8 digital outputs



Wireless-IO module, 16
digital inputs, 16 digital
outputs, 2 analog inputs,
2 analog outputs



Expansibility



COM1
COM2
COM3

RS-232

RS-485

Converter



#01



#02

.....



#FF

RS-232

RS-485

Converter



#01



#02

.....



#FF

⋮

RS-232

RS-485

Converter



#01



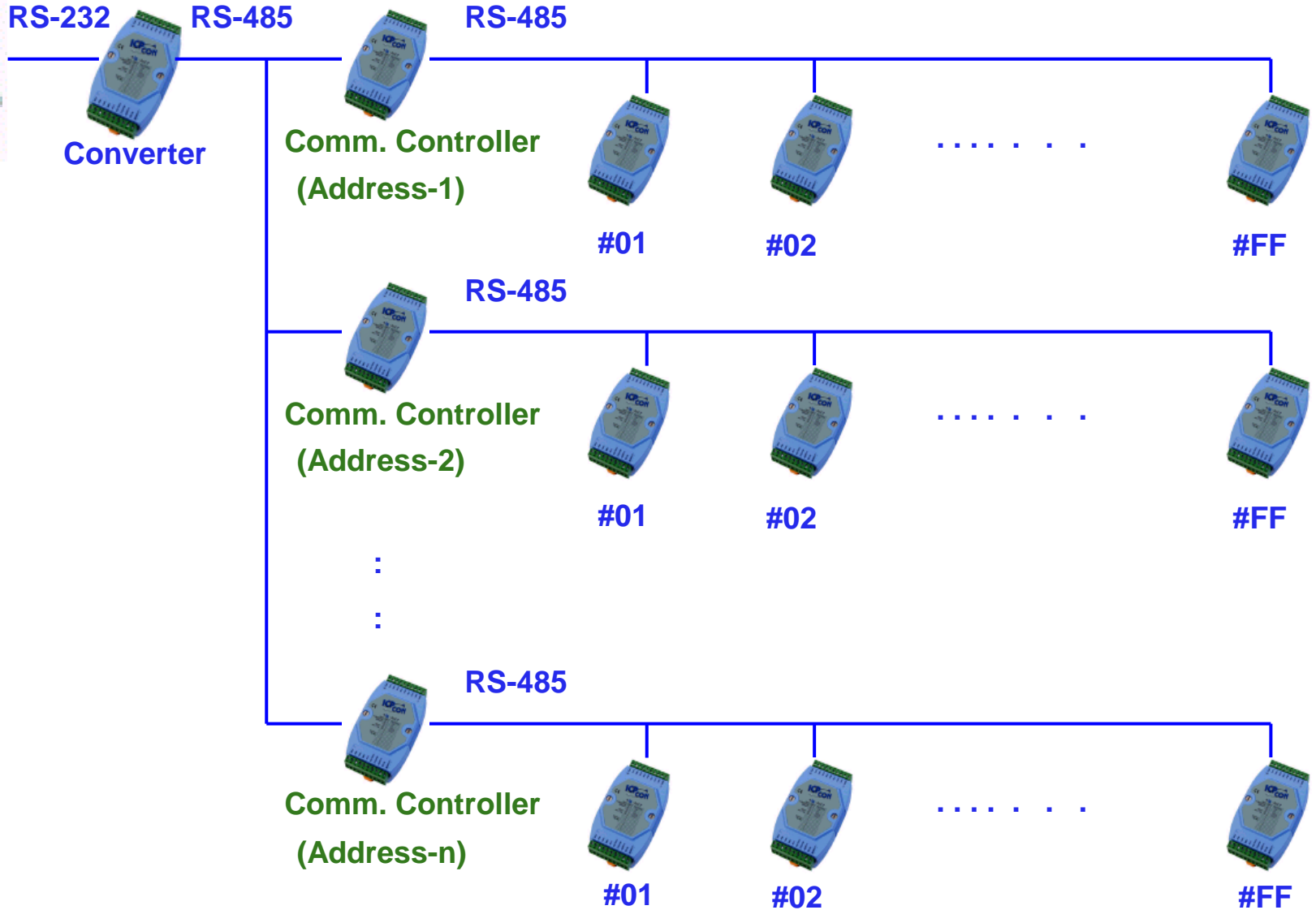
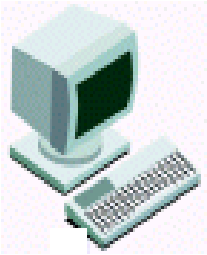
#02

.....

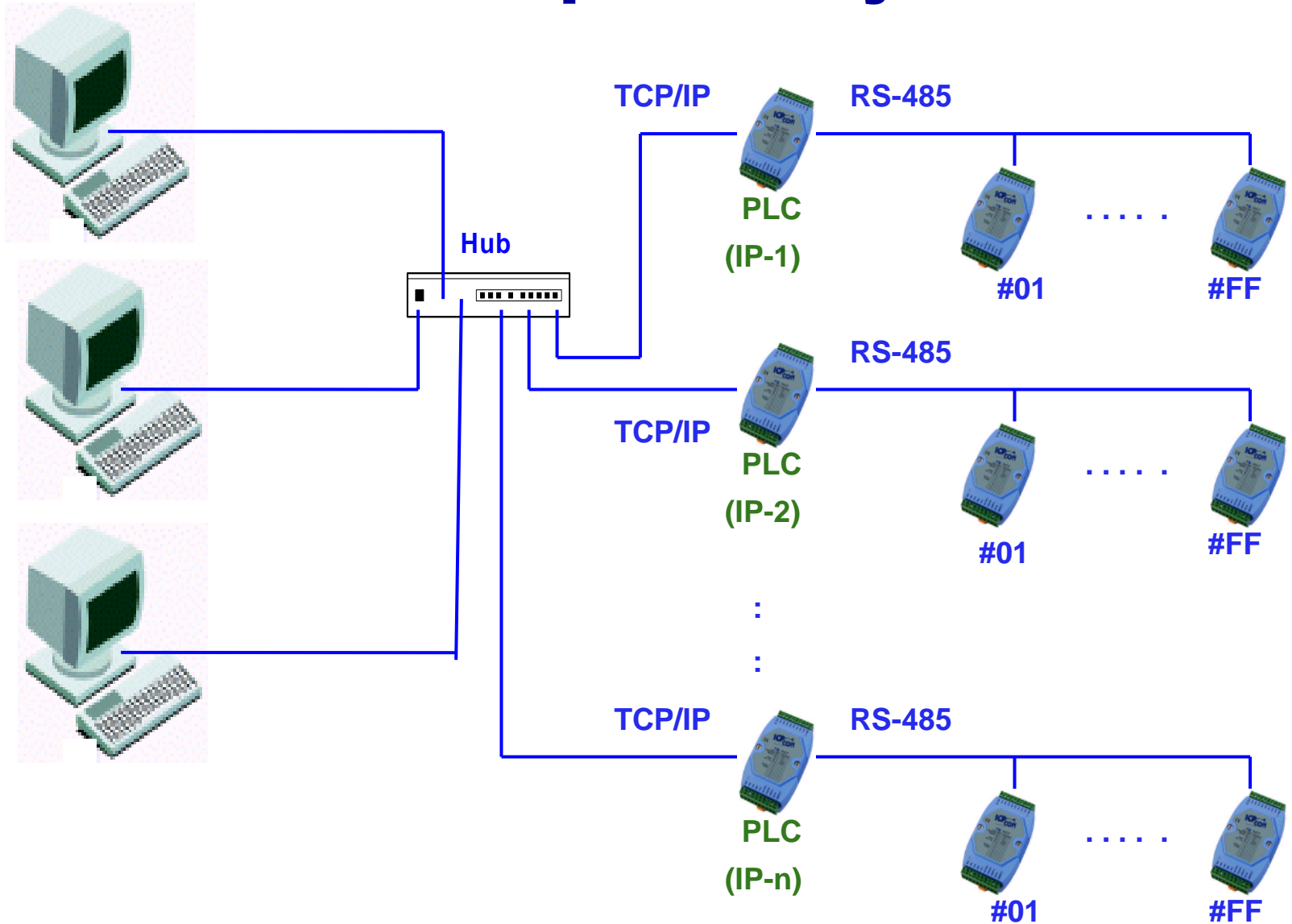


#FF

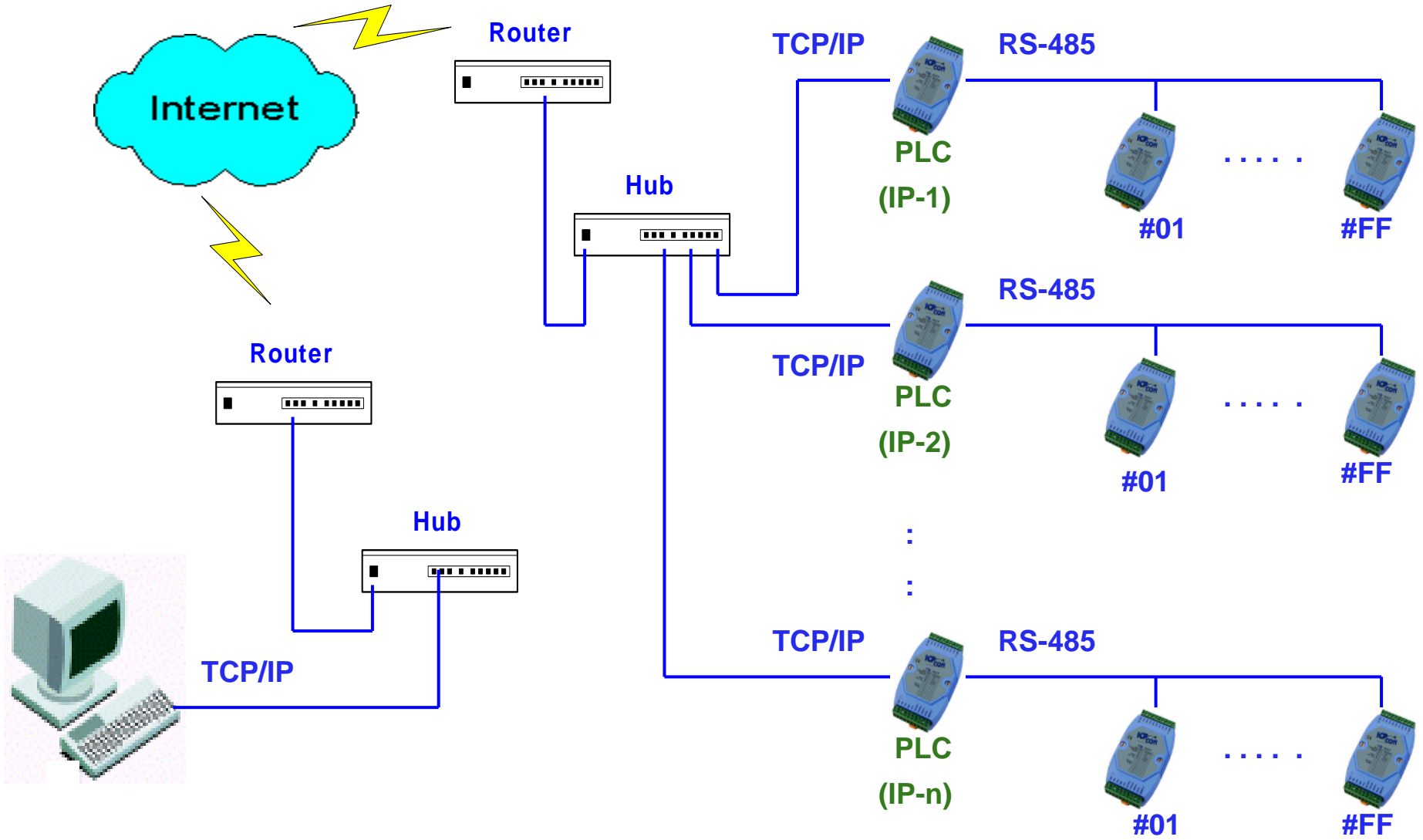
Expansibility



Expansibility



Expansibility



Command Protocol of Modules

- ◆ Use ASCII Command
- ◆ Use Asynchronous Protocols
 - 1 Start Bit
 - 8 Data Bits
 - Non Parity
 - 1 Stop Bit

◆ Typical Command Format

- Leading Character
- Address Character (00~FF)
- Command and Parameter
- Checksum (2 Character)
- CR (0x0D) for End of Command

◆ Example

- \$01M[Chk](CR)Read Module ID (Name)
- C: "\$01M...\r"

Develop Tool Kits

- ◆ Win32 DLL Developing Application
 - VB, VC, BCB, Delphi
- ◆ Dos Library
 - TC, BC
- ◆ ActiveX (OCX)
 - VB, VC, BC, Delphi
- ◆ Library for SCADA (LabView, Intouch, Citect, WinCC,...)
- ◆ Windows/DOS Based Diagnostic/Utility Tool
- ◆ DDE Server
- ◆ OPC Server

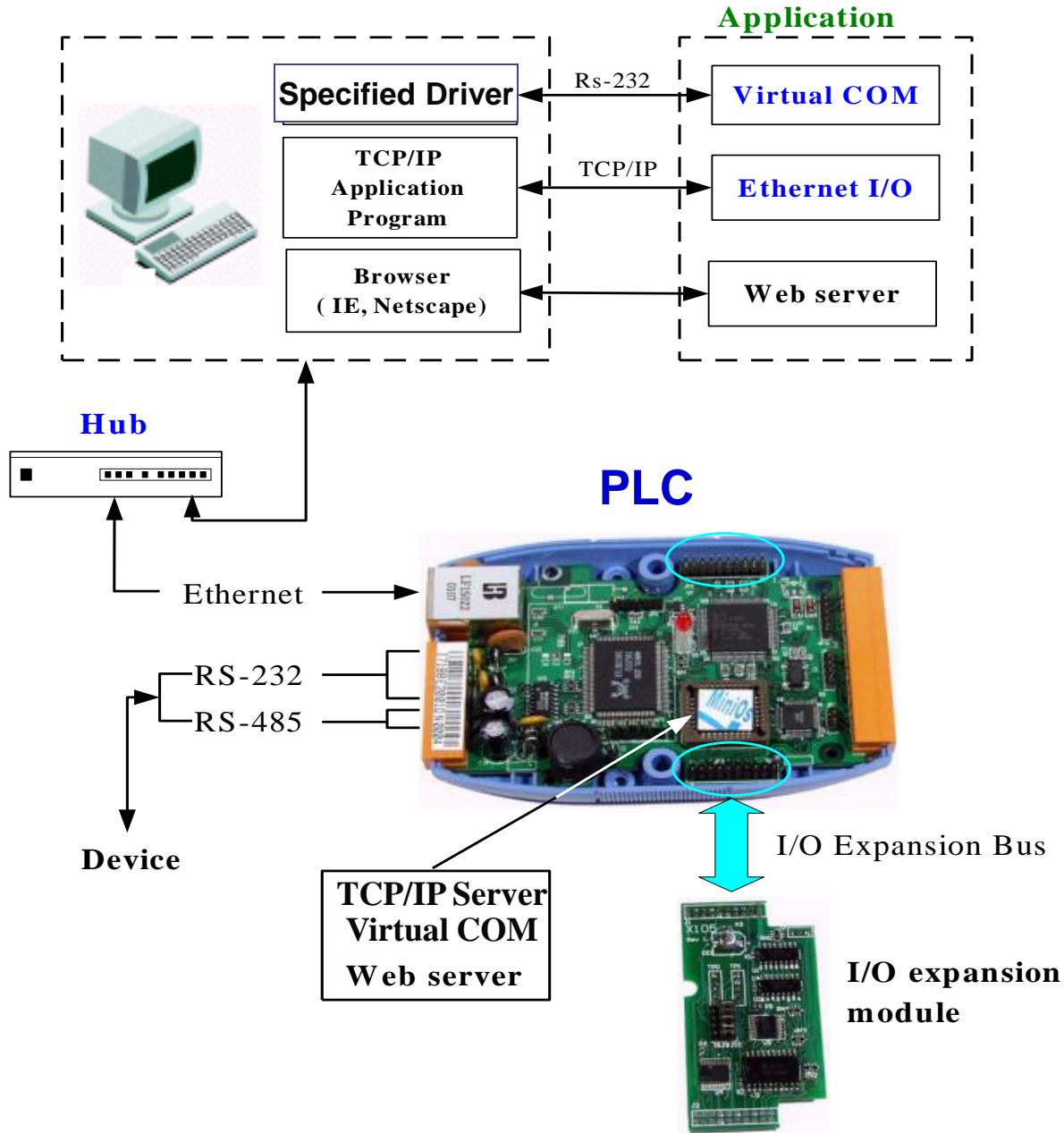
Features of DAQ Remote Module

- ◆ Built-In DOS, MiniOS7, Windows XP Embedded, Linux Embedded...
- ◆ Program Download Port: COM1 or Ethernet Port
- ◆ Remote Configuration, Diagnostics
- ◆ Operating Temperature: -25 °C ~ 75 °C
- ◆ Storage Temperature:-40 °C ~ 80 °C

MiniOS7

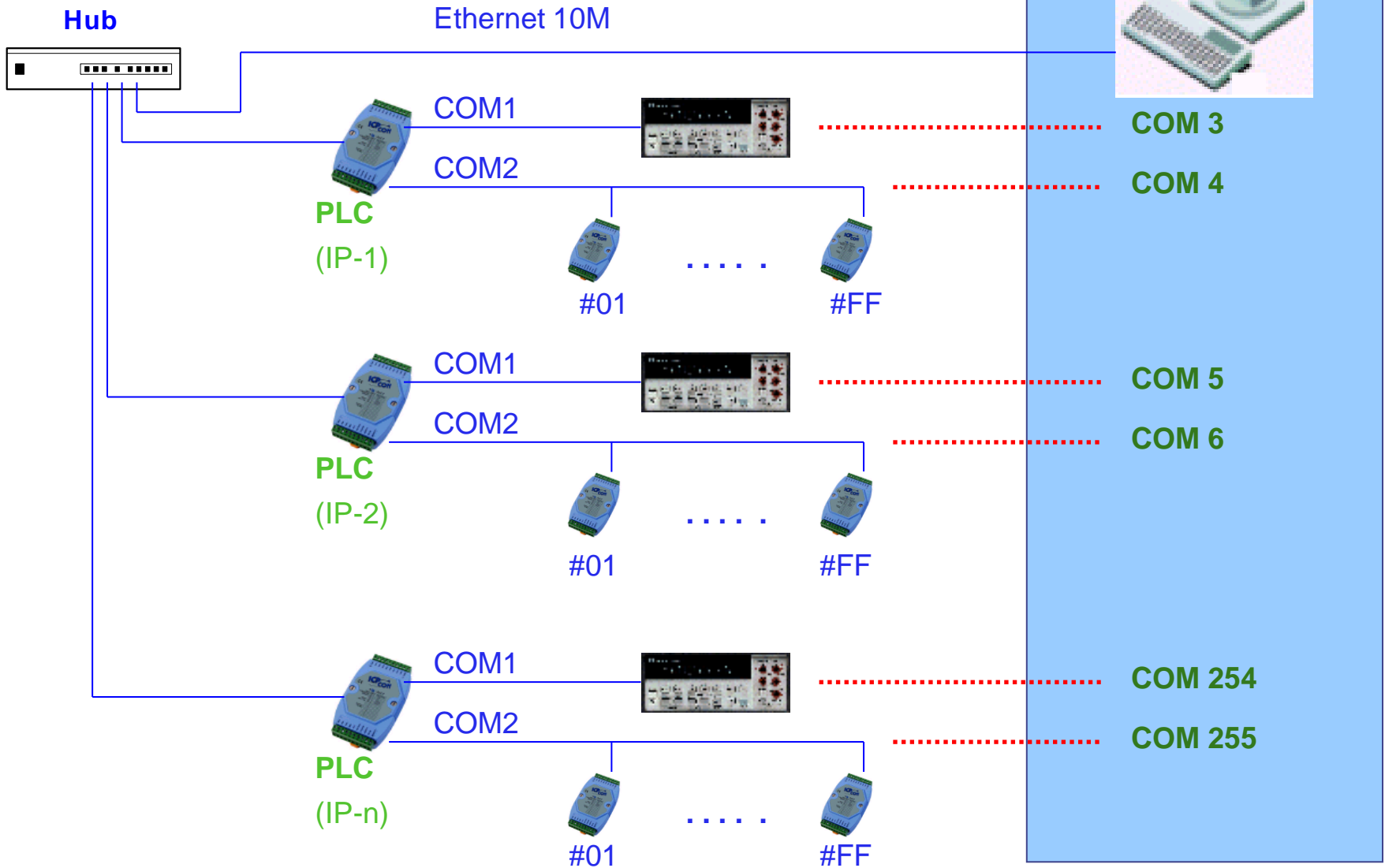
- ◆ Files Download and Execution (.exe, .com, .bat)
- ◆ Use Autoexec.bat
- ◆ Can't Run Another Process (Function system() of C)
- ◆ Warm Up Time
 - 0.1 Second for 20 MHz CPU
 - 0.05 Second for 40 MHz CPU
 - 0.5 Second for 8k Main Unit

Typical Application

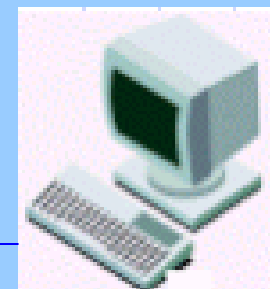
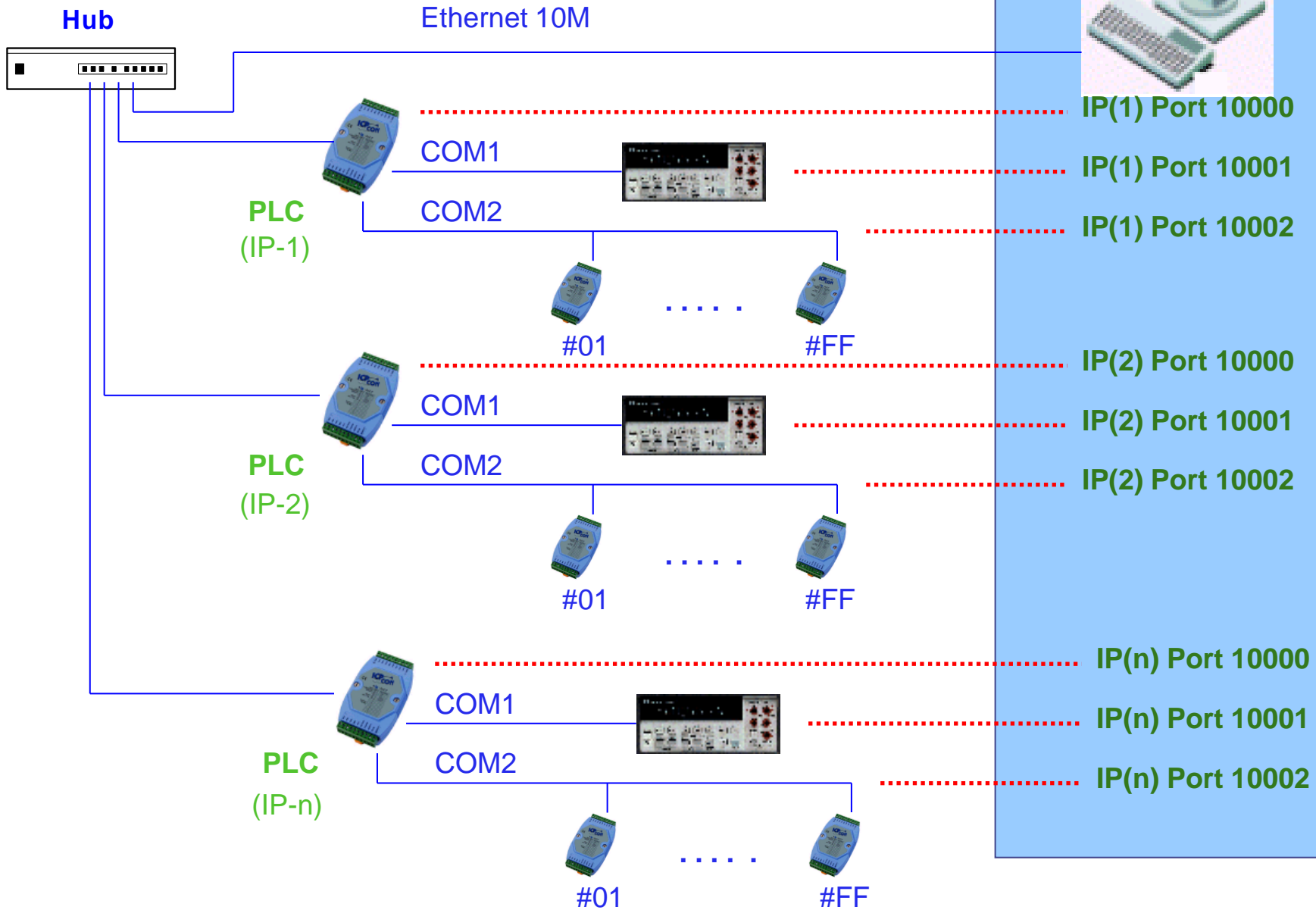


Virtual COM Application

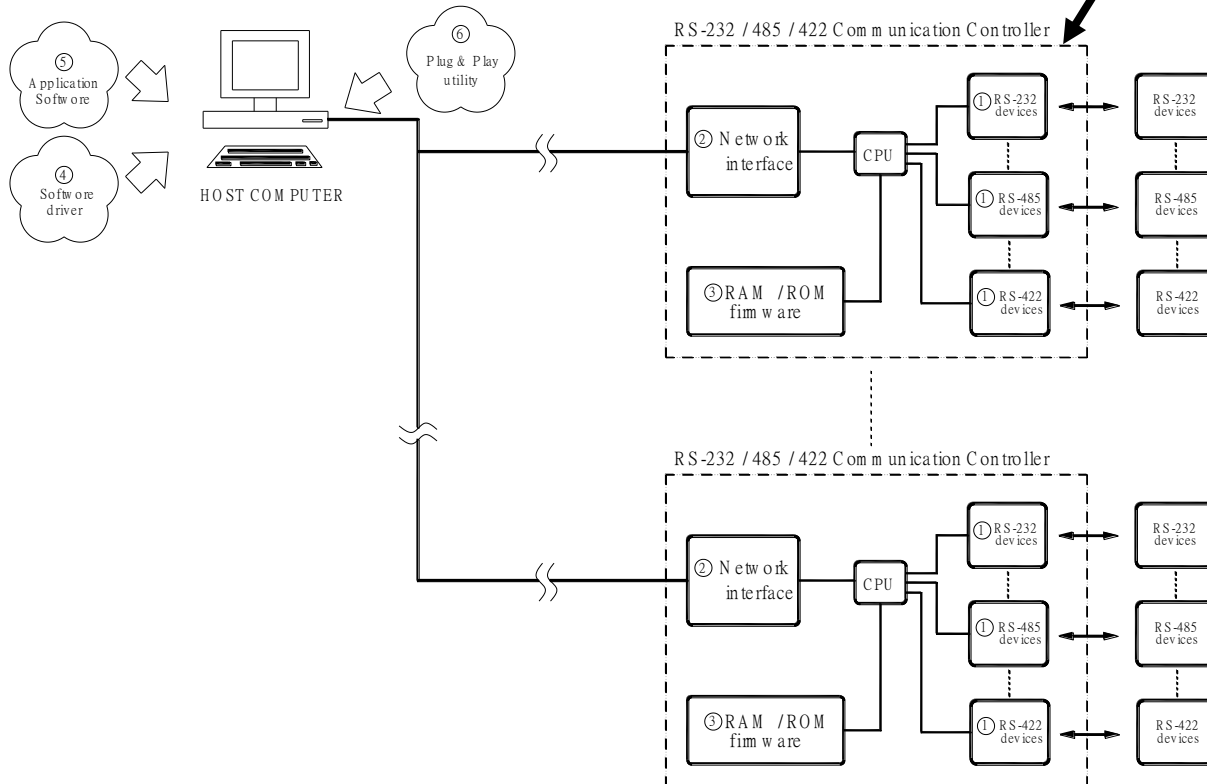
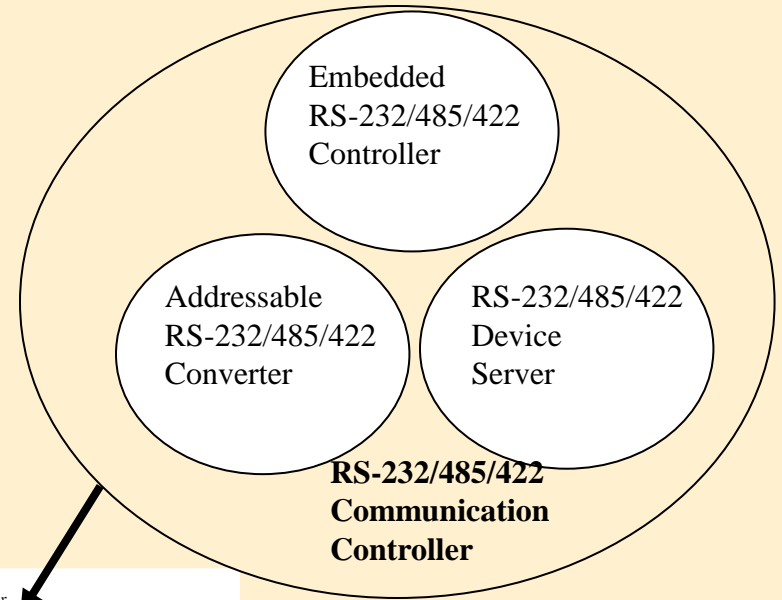
Specified Driver



Ethernet I/O Application



RS-232/485/422 Communication Controller

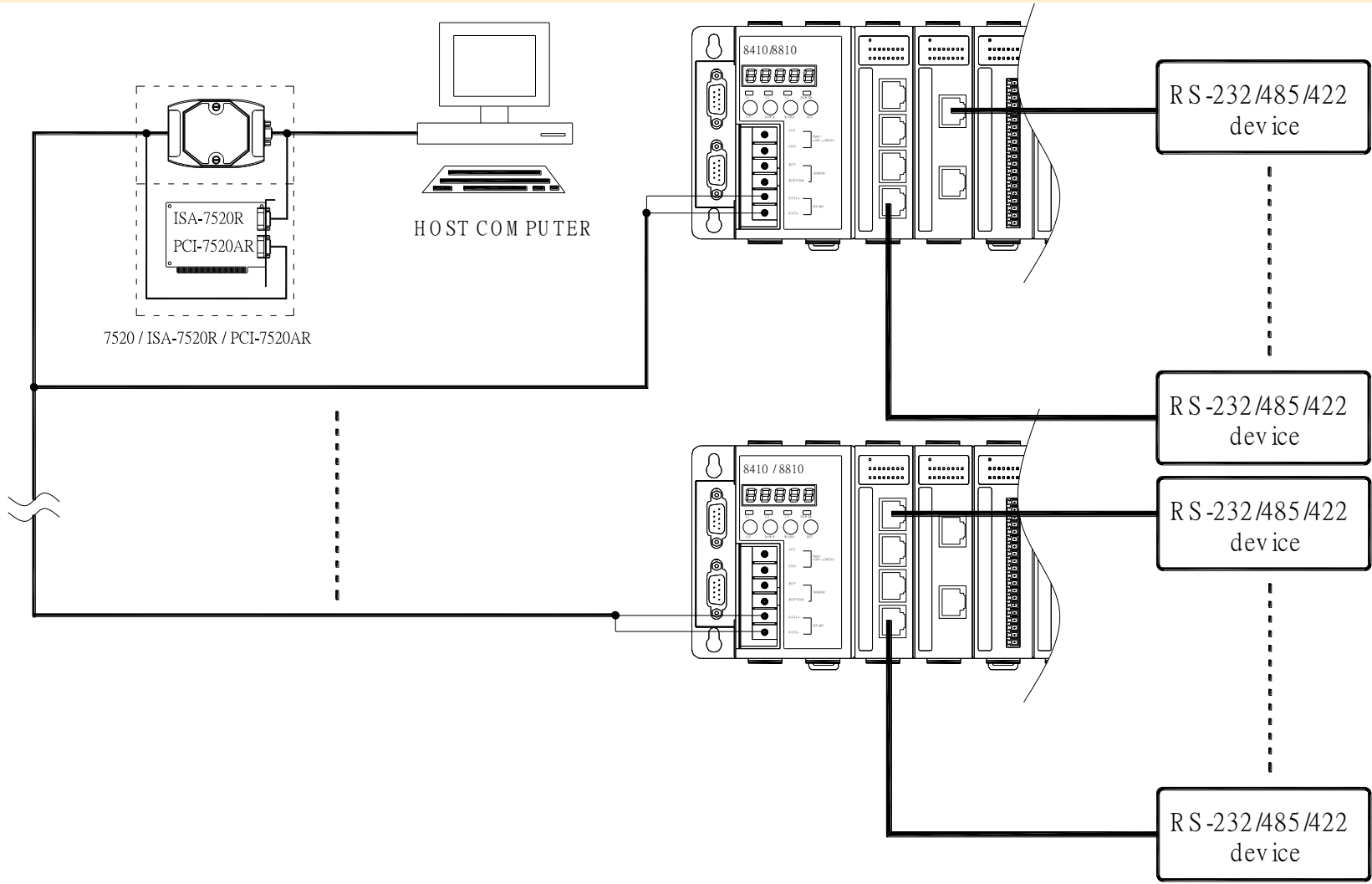


The comparison table of these 3 types of communication controller is given as follows:

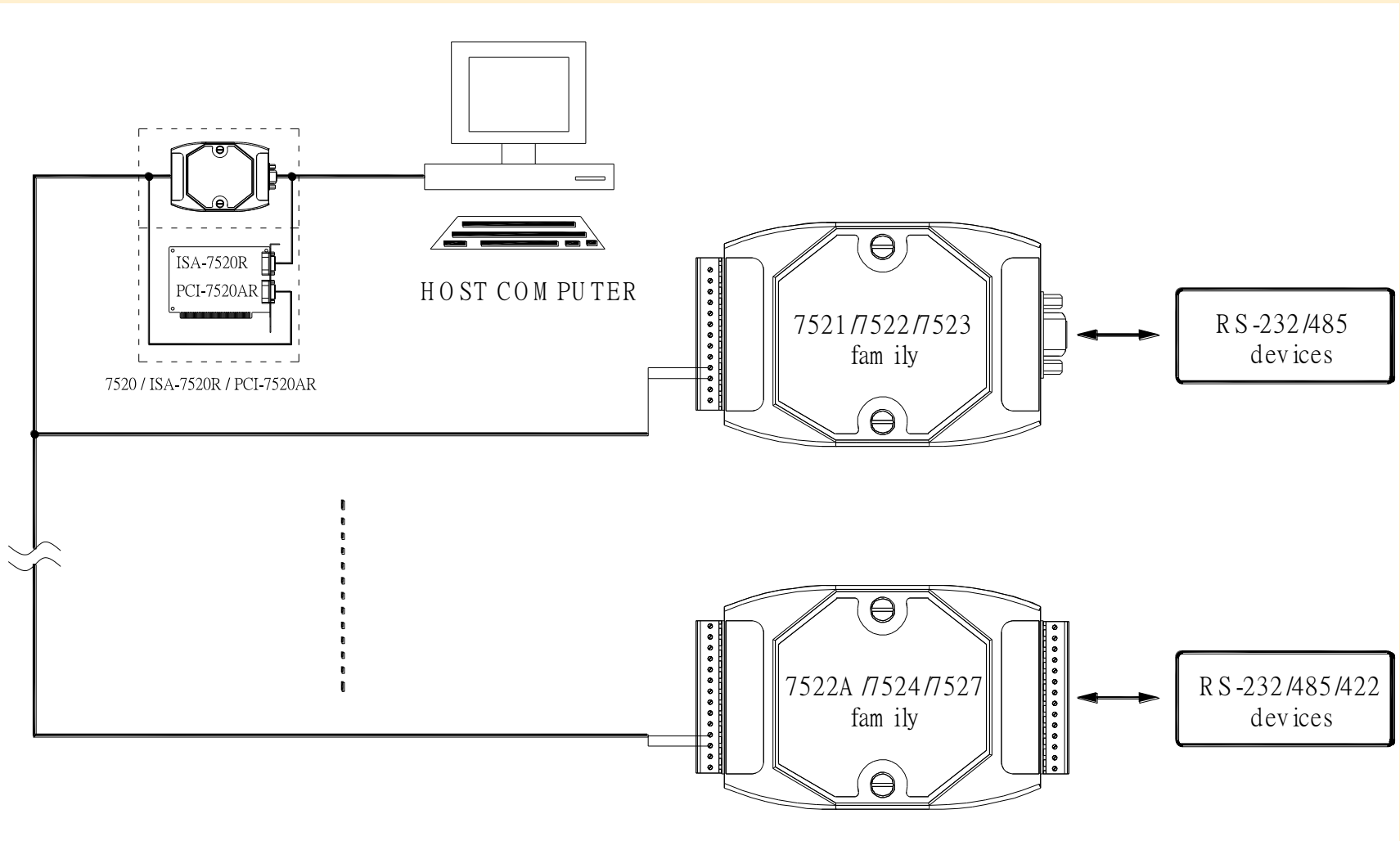
	Embedded RS-232/485/422 Controller	Addressable RS-232/485/422 Controller	RS-232/485/422 Device Server
1 RS-232/485/422 interface	RS-232 or RS-485 or RS-422	Same	same
2: network interface	RS-485 or Ethernet 10M or Ethernet 10/100M or Other network	same	same
3: RAM/ROM firmware	User must write his special application program	Well-design firmware & Source codes are open for user's modification to fit his special requirement.	Well-design firmware & Source codes are open for reference. It is very difficult to modify for user's special requirement.
4: software driver	DLLs	DLLs	DLLs
5: application software	User must write his special application program	User's old application program must be modified to fit the protocol defined by (3) firmware.	User's old application program can work without modification
6: Plug&Play Utility	No need	No need	MS-COMM standard
Easy Using	Low	Middle	Most
Flexibility	High	Middle	Most

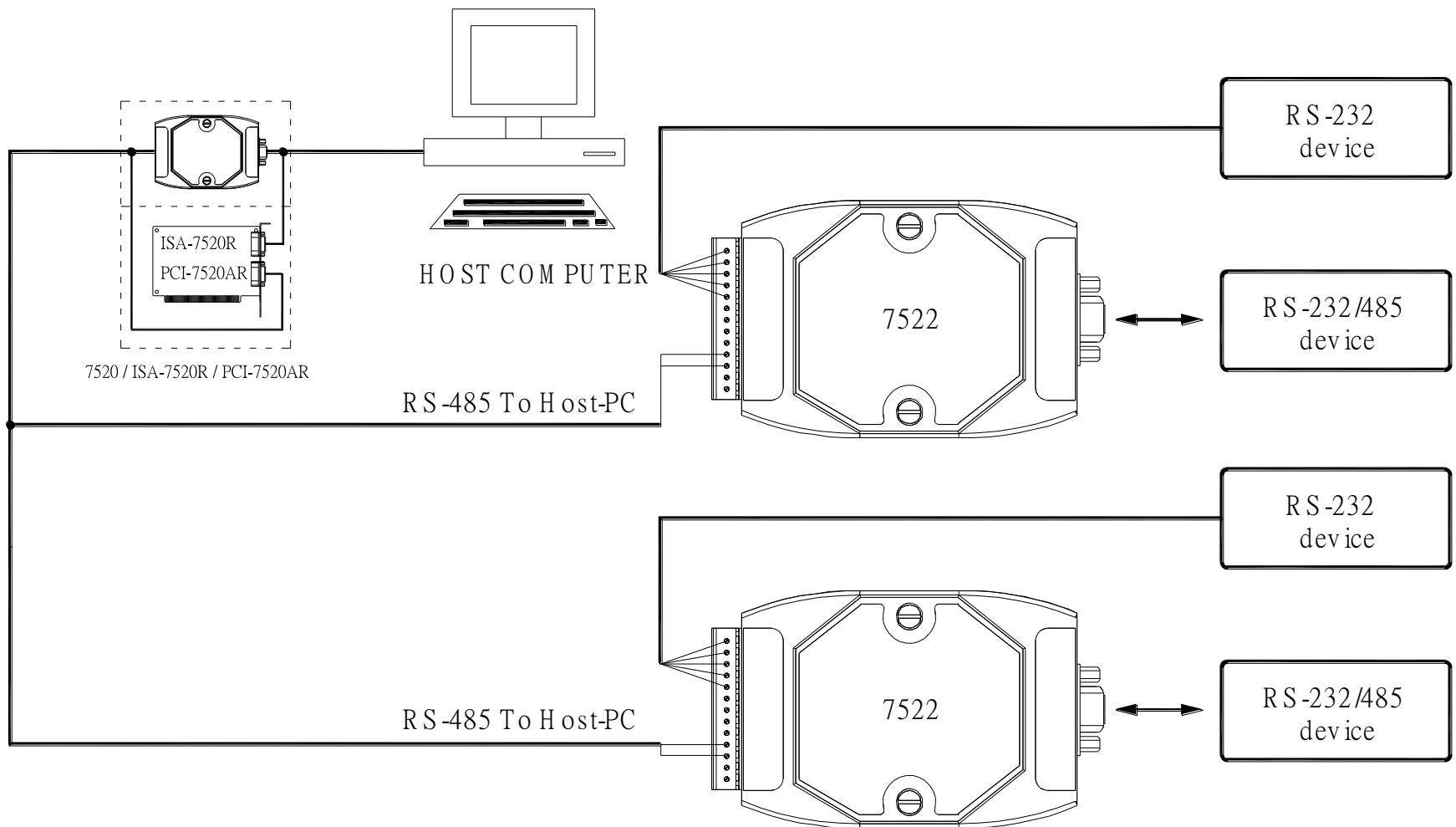
8000 Family

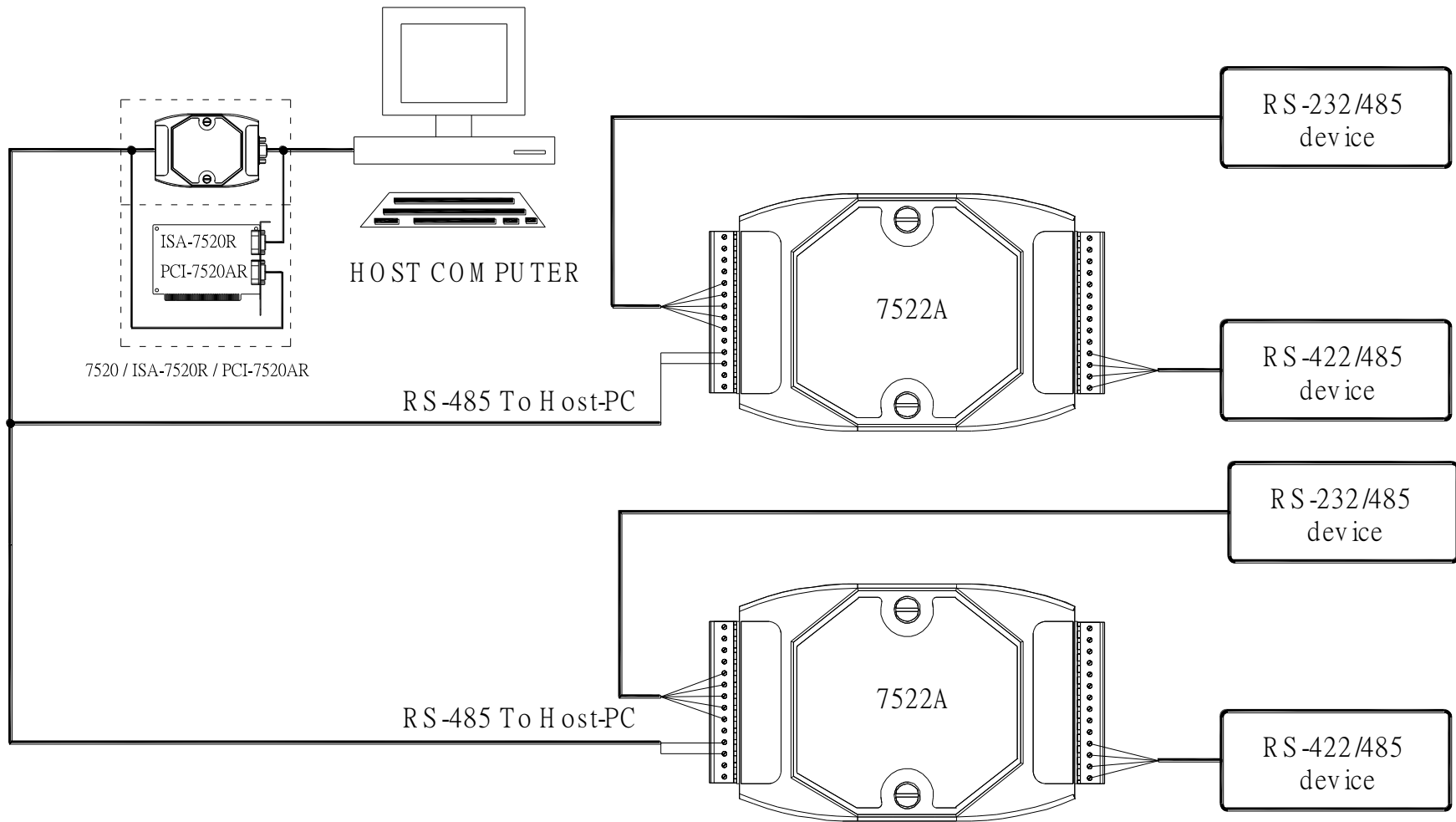
841X/881X + 8112/8114/8142/8144

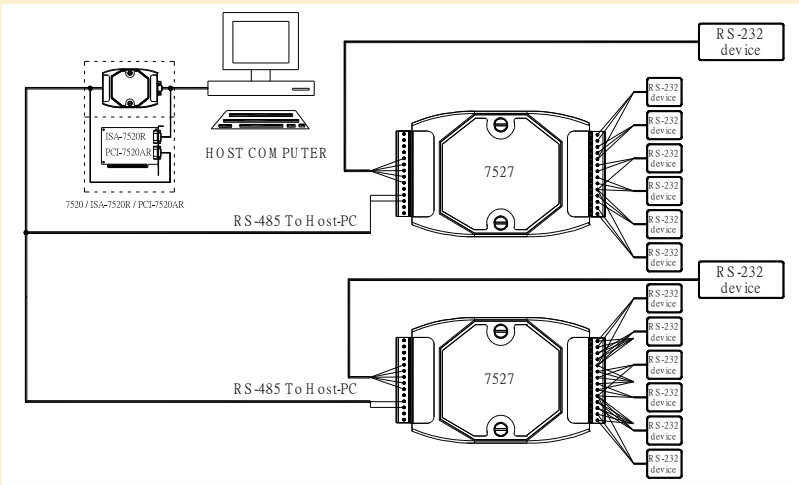
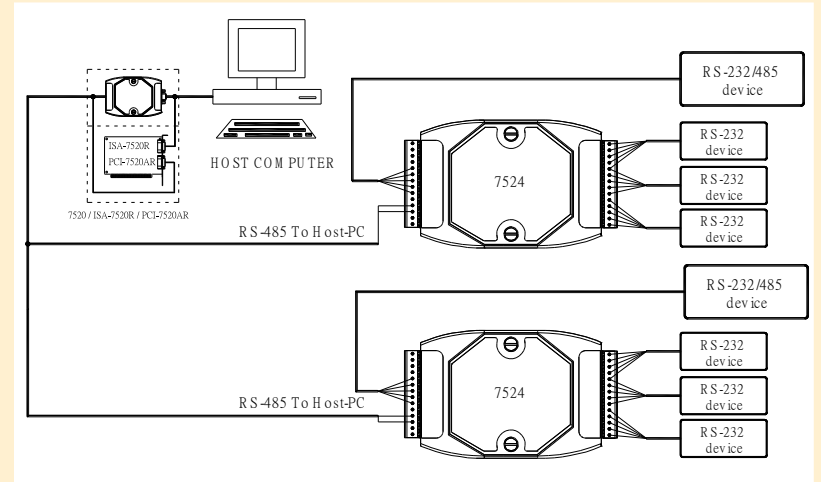
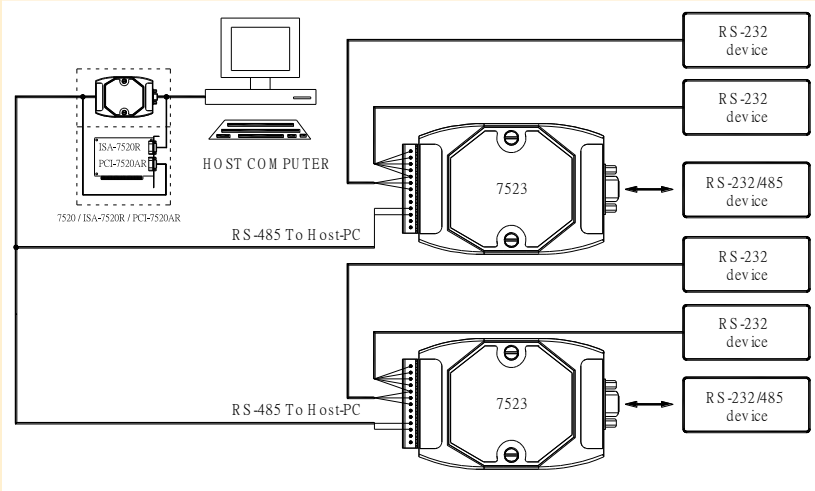


7521 family

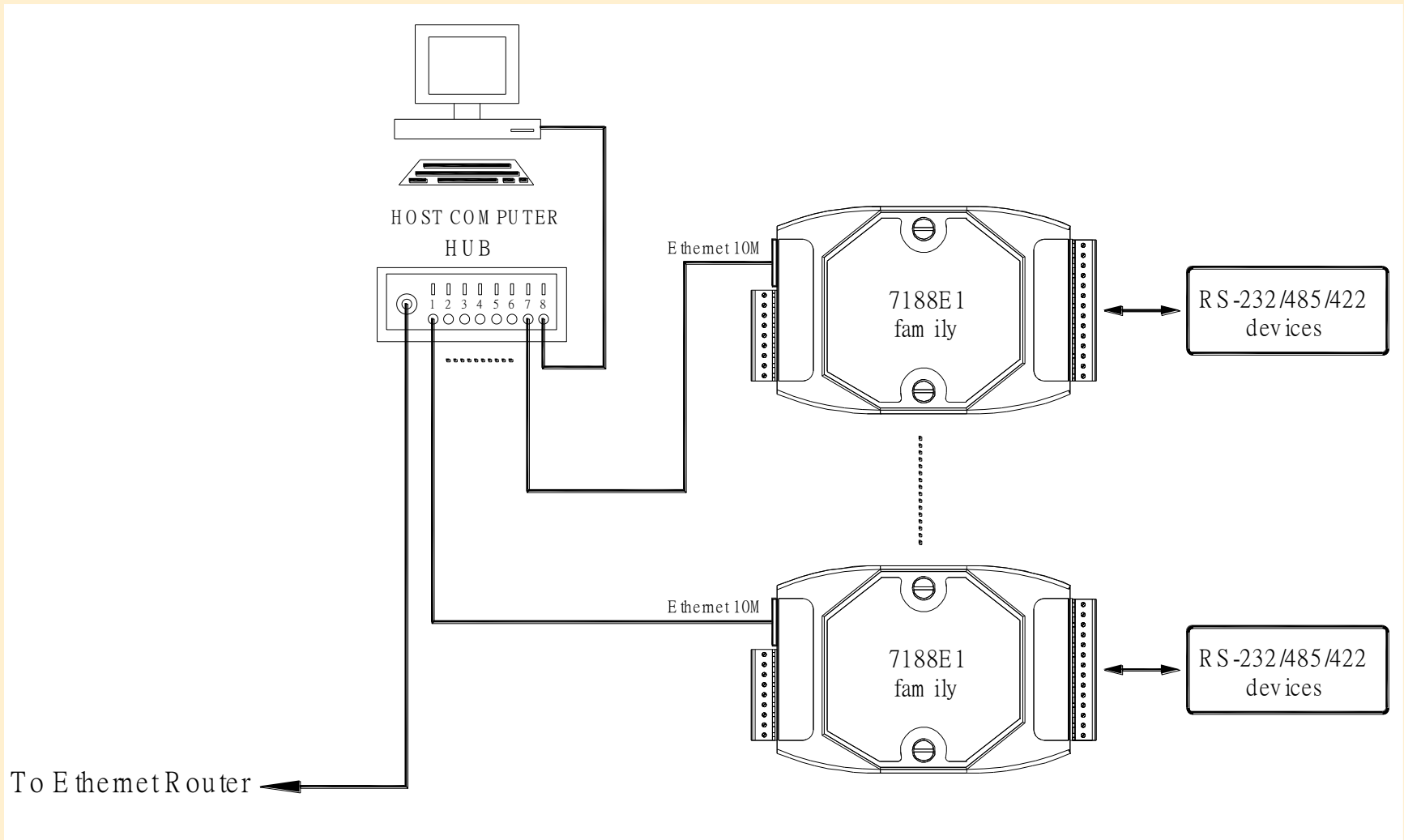


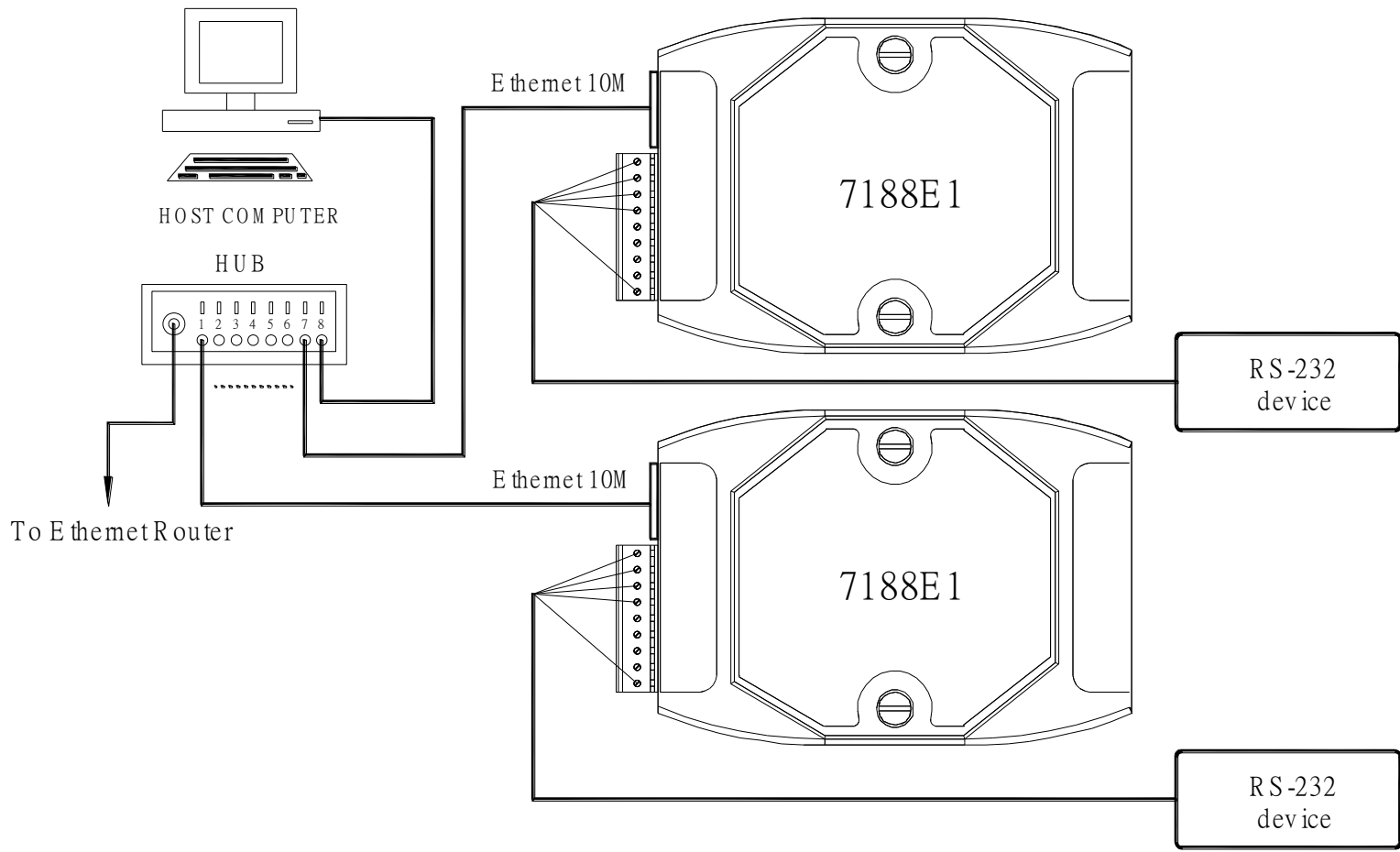


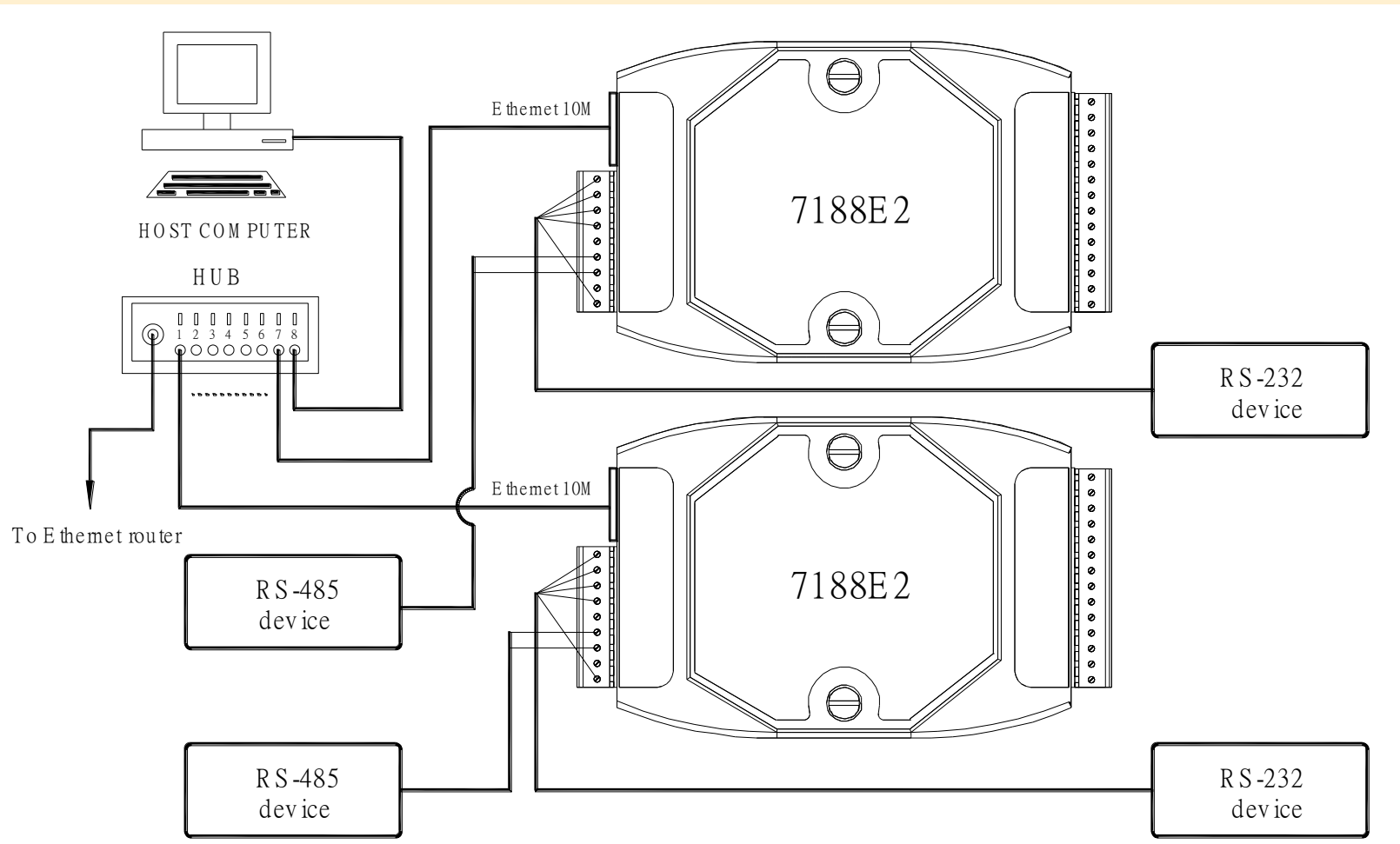


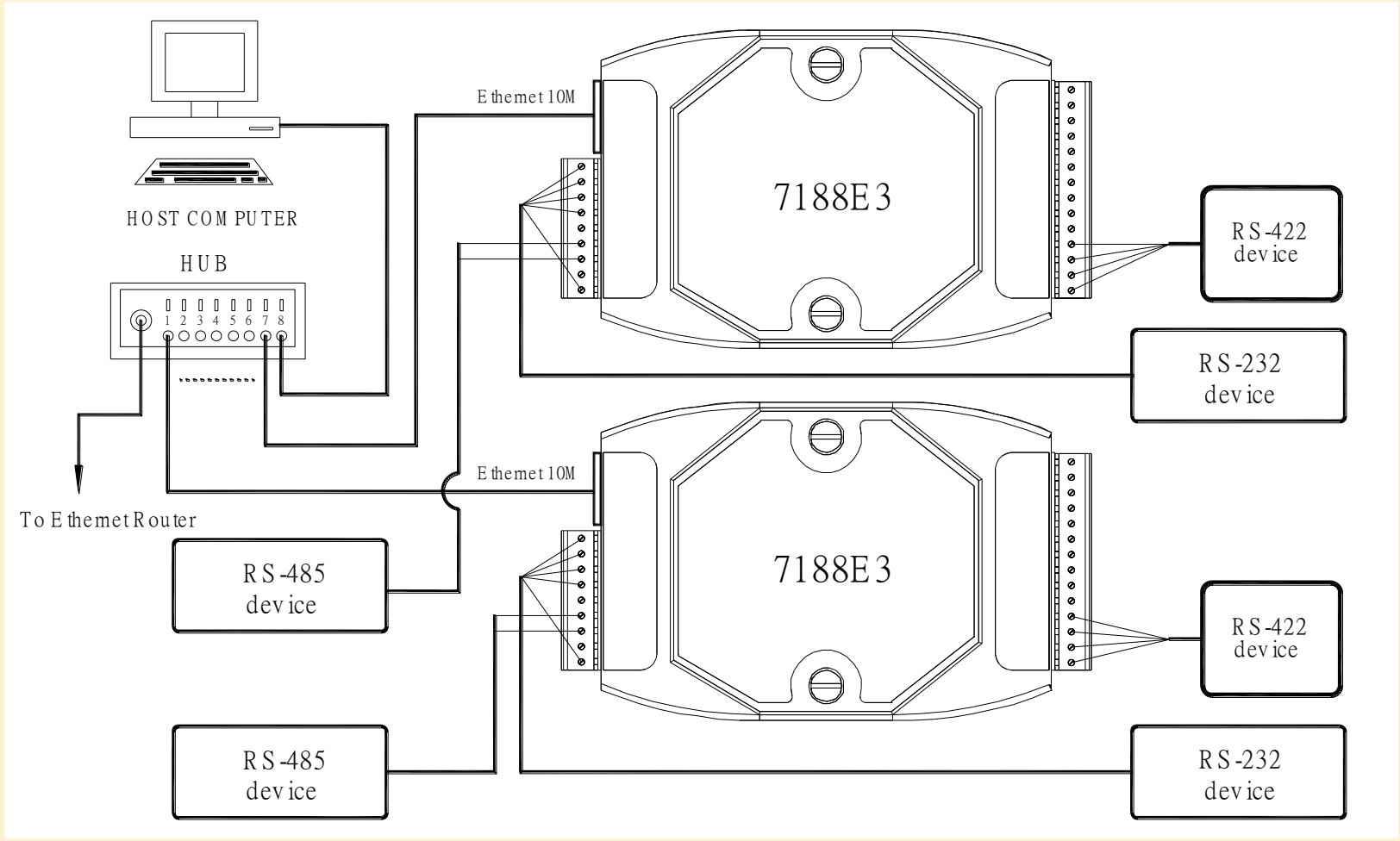


7188E1 family



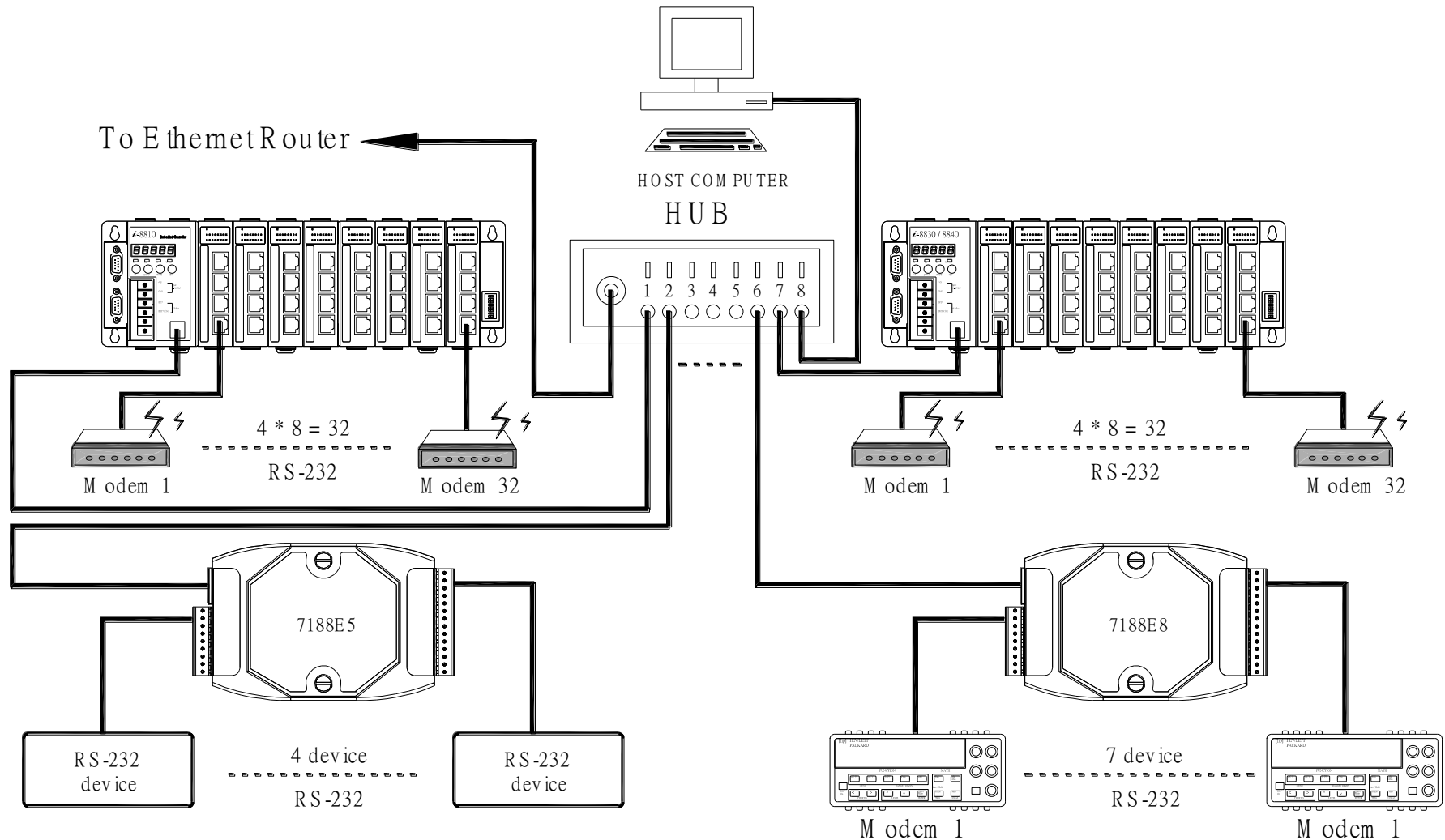






Typical Applications

N-port RS-232 to Ethernet Router

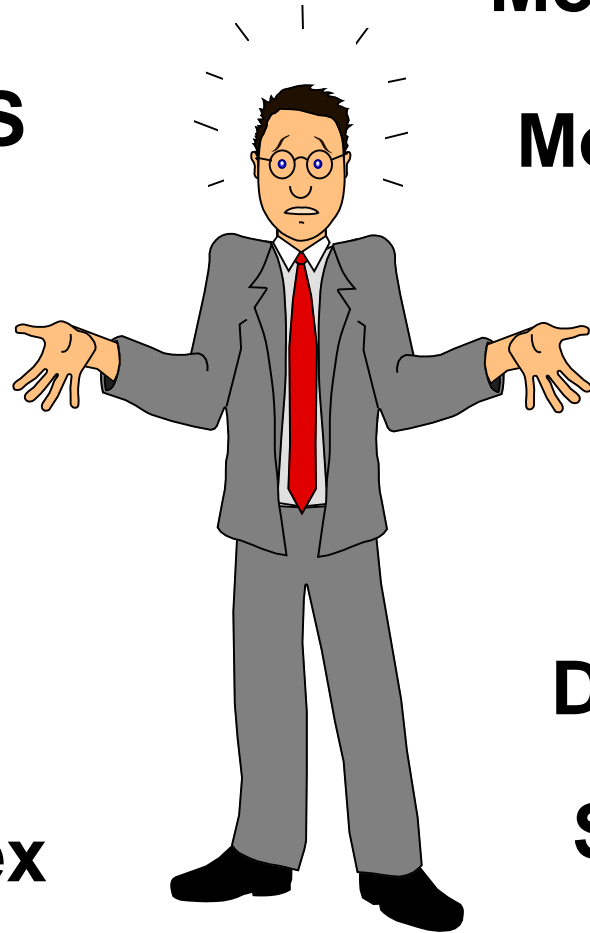




Supervisory Control & Data
Acquisition

Communication Technology

Which Network Do I Use?



Modbus

Modbus Plus

Profibus

CAN

Device Net

SDS

Interbus S

ASI

Echelon

Ethernet

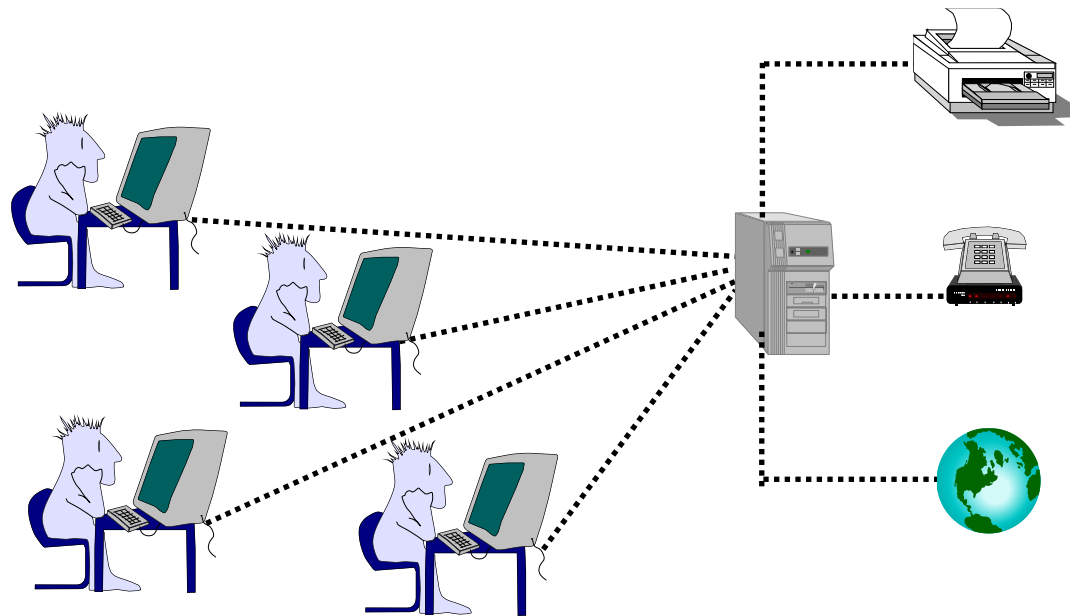
Seriplex

- [Seriplex FAQ](#)
- [CAN FAQ](#)
- [Interbus S FAQ](#)
- [IBS club](#)
- [WorldFIP](#)
- [LON works](#)
- [Modbus](#)

- [Fieldbus compare](#)
- [Honeywell](#)
- [Control competition](#)

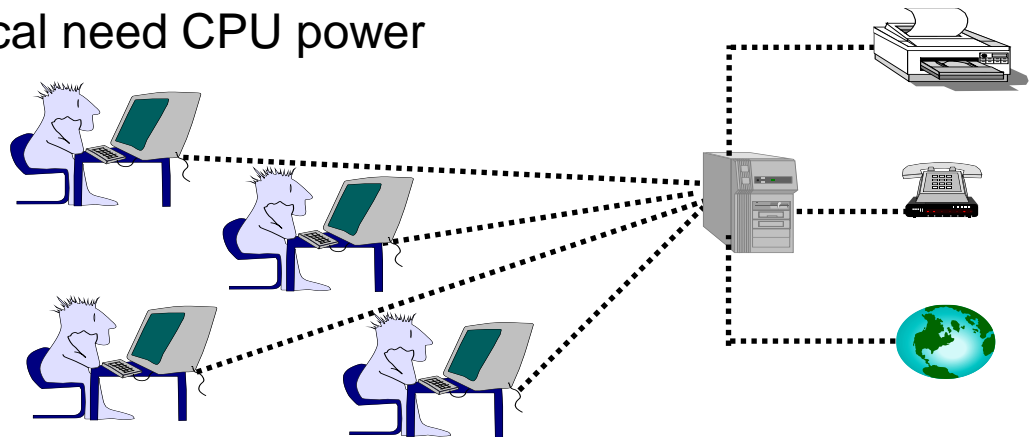
What is a Network ?

- A group of devices physically connected together for the purpose of sharing information, resources and a common transmission medium.



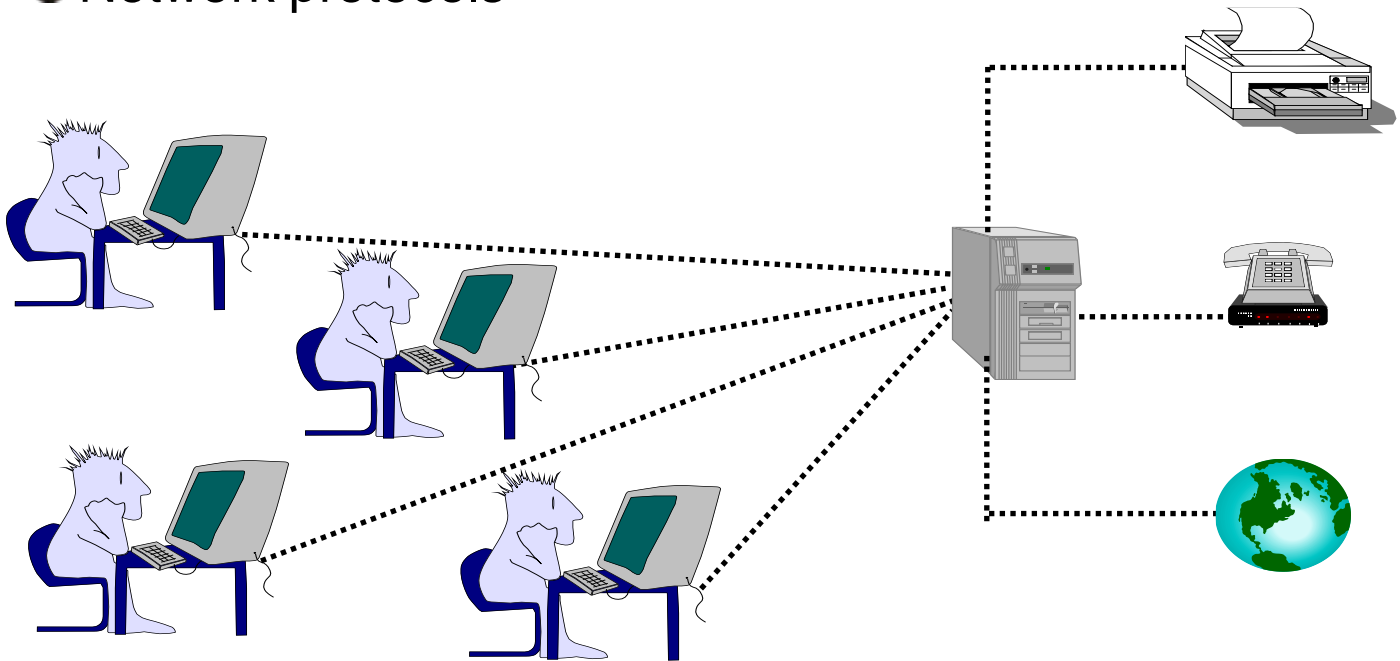
Why do you need a network?

- Control the flow of information
 - security, accessibility, data control
- Centralize resources
 - save cost on local work stations and peripherals
- Simplify data / software management
 - Backups, IS maintenance
 - Share information with multiple users
- Distribute tasks
 - reduce local need CPU power



What makes a network work ?

- Computers, printers
- Network interface cards
- Network adapter Software drivers
- Network Cabling
- Network protocols



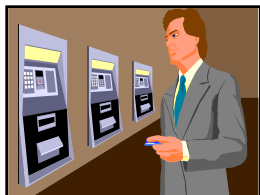
The Basic Types of Networks



- Master / Slave
(Similar to the teacher- student relationship)



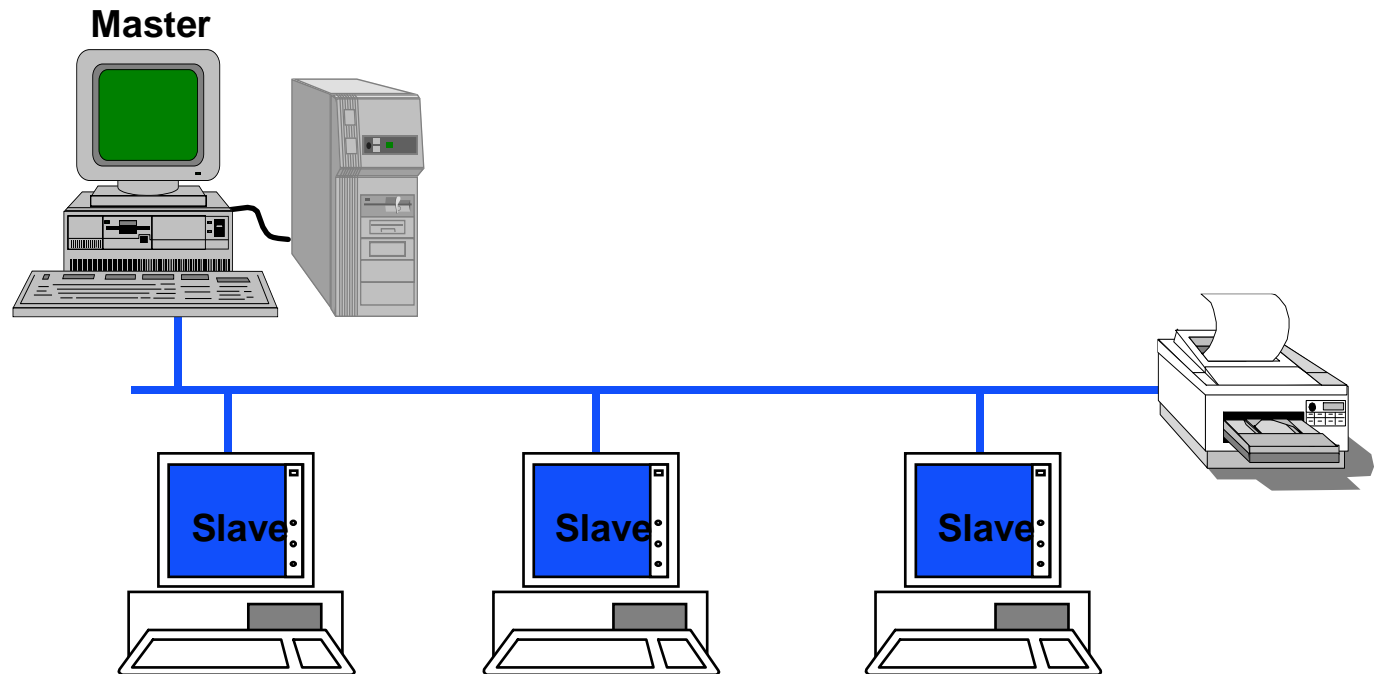
- Peer to Peer
(Similar to the workgroup concept)



- Client / Server
(Similar to an automated teller transaction)

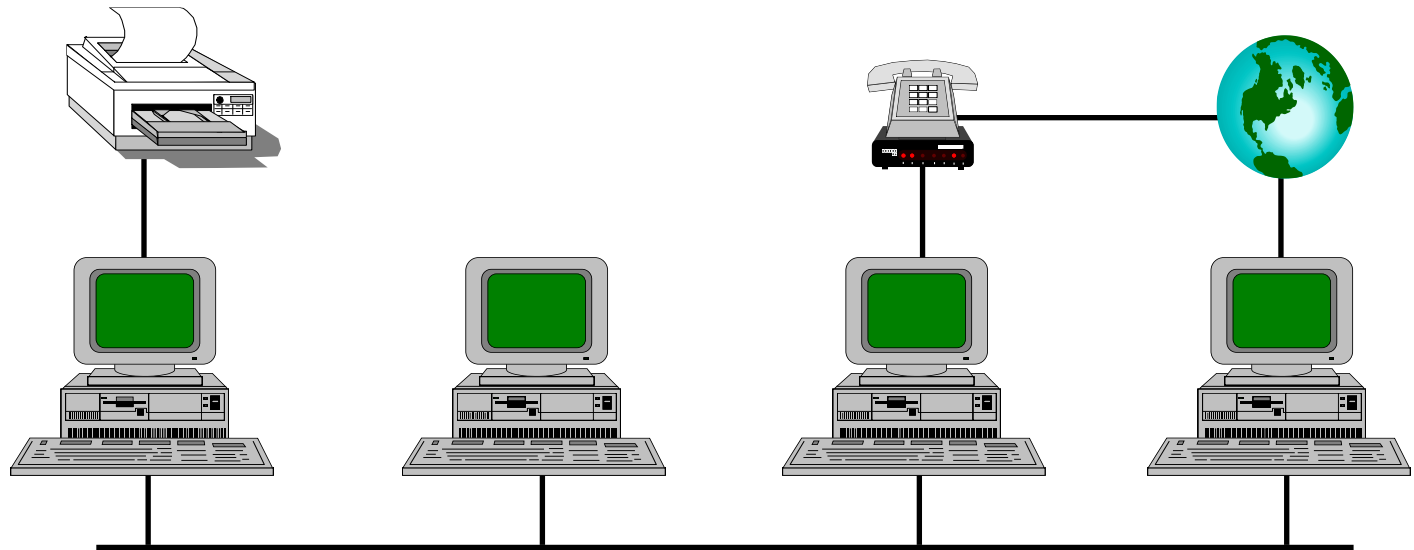
Master / Slave Network

- Central Control and Administration
- Central Processing of Information
- Polling of Stations



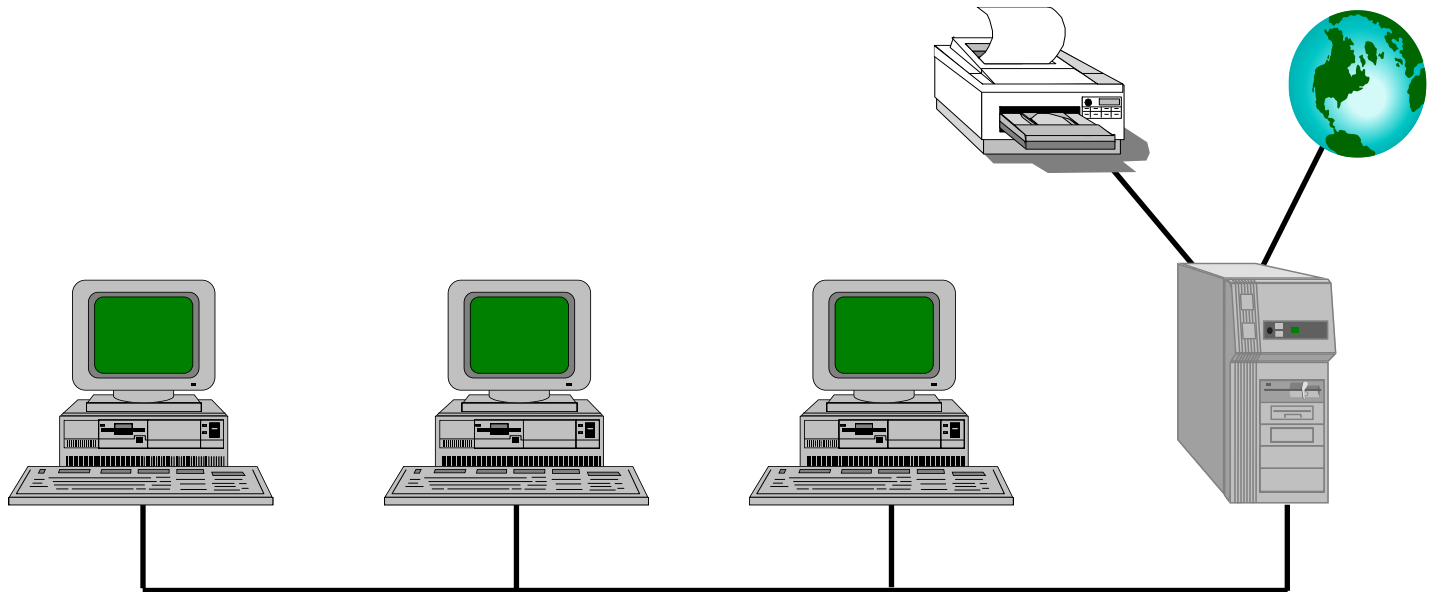
Peer To Peer Network

- Distributed Administration
- Independent Processing of Information
- Shared Transmission Medium



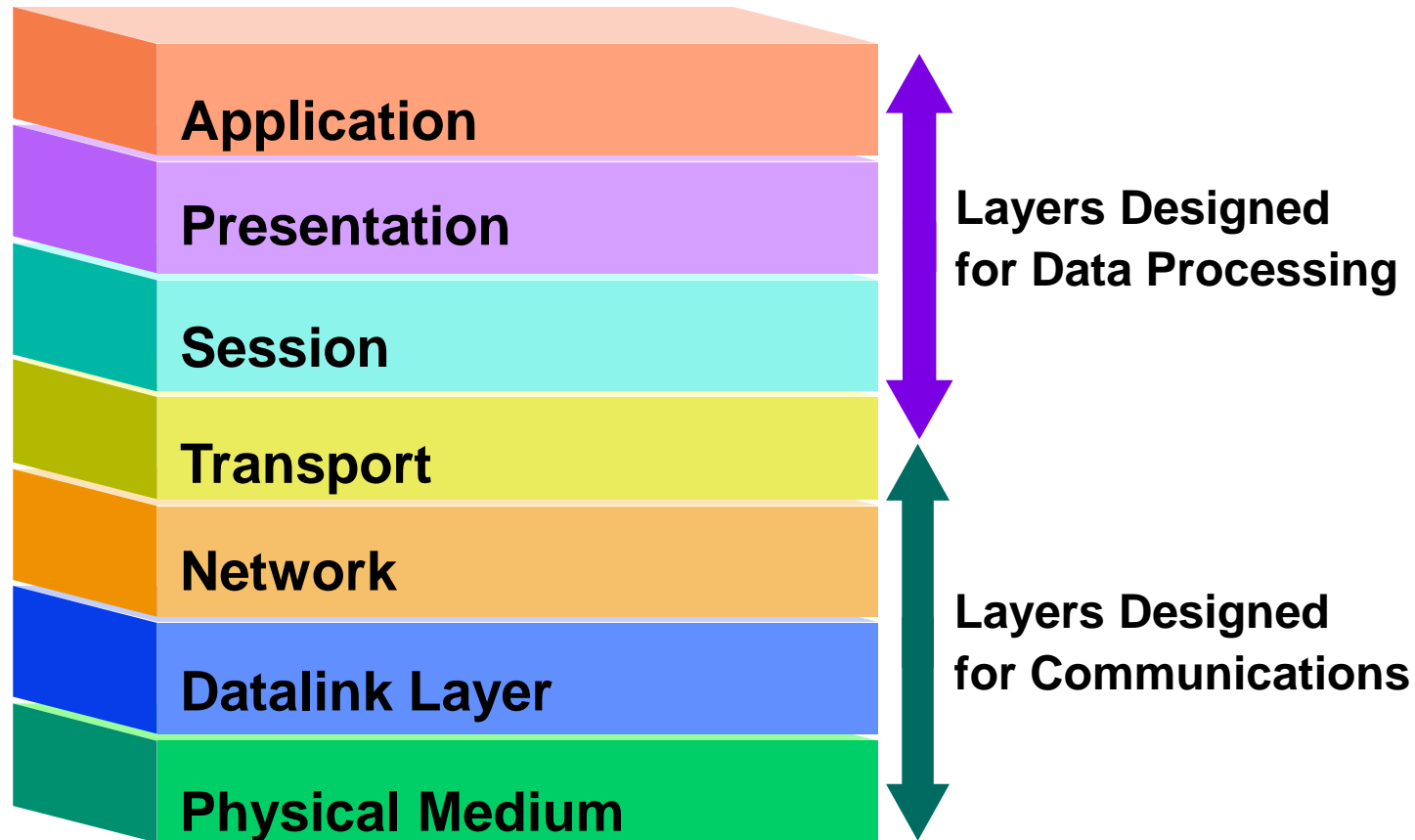
Client / Server Network

- Central Network Administrator
- Controlled Flow of Information
- Independent Processing of Information
- Shared Transmission Medium



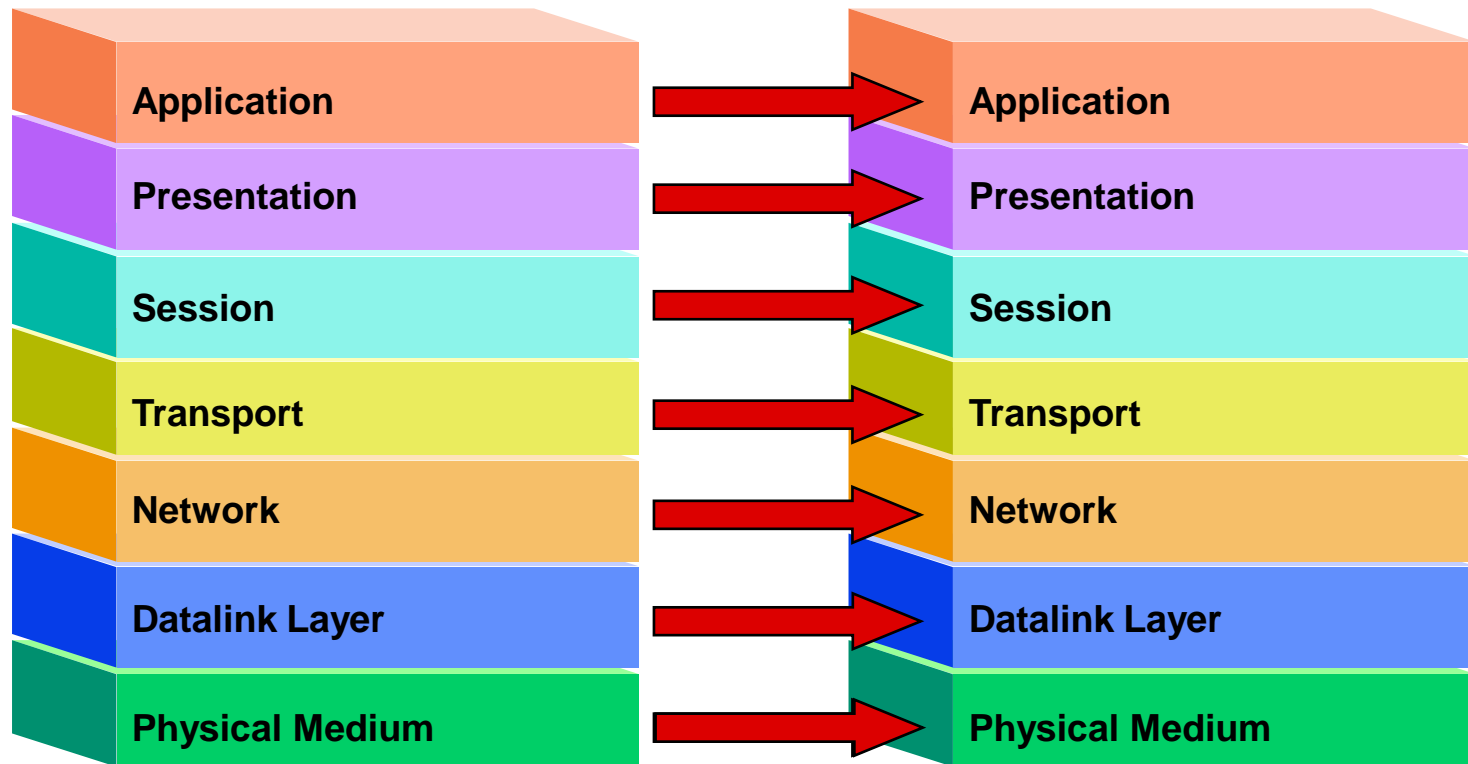
How is a network put together ?

- The OSI model uses 7 layers to categorized the main elements of a basic network



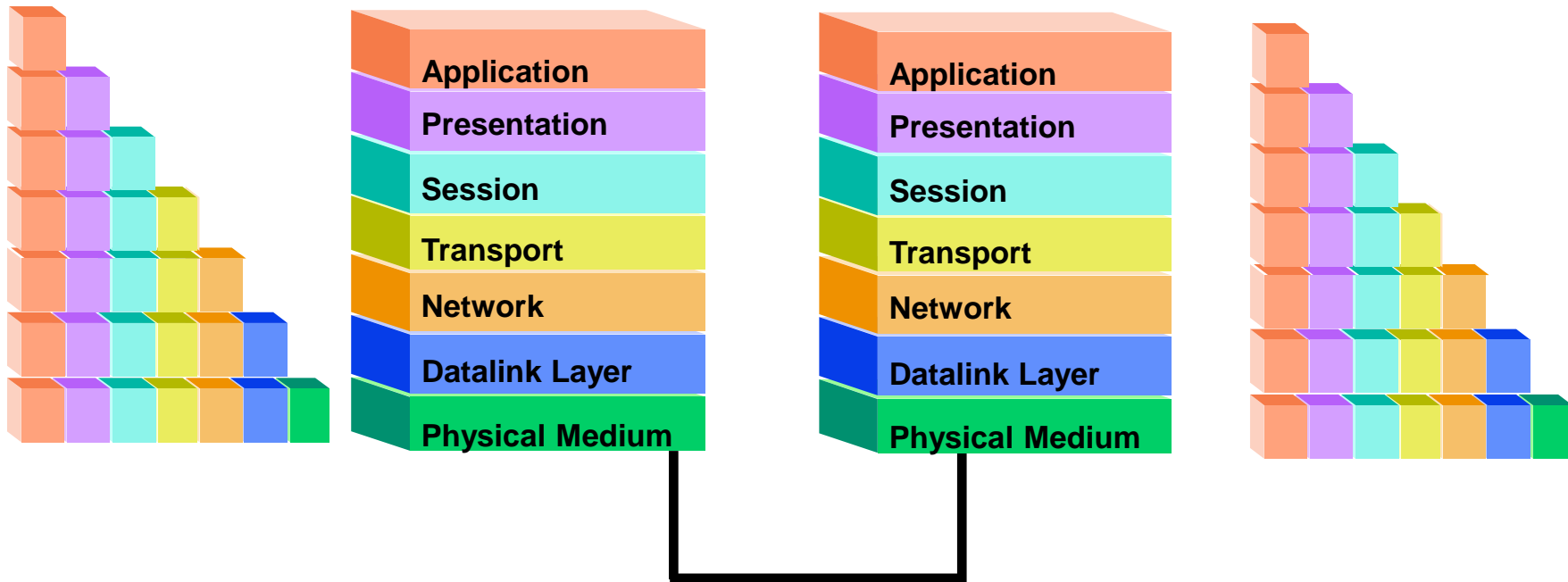
OSI Model Layers

- A Model is used to ensure Compatibility
- Each station utilizes the same Protocol Stack



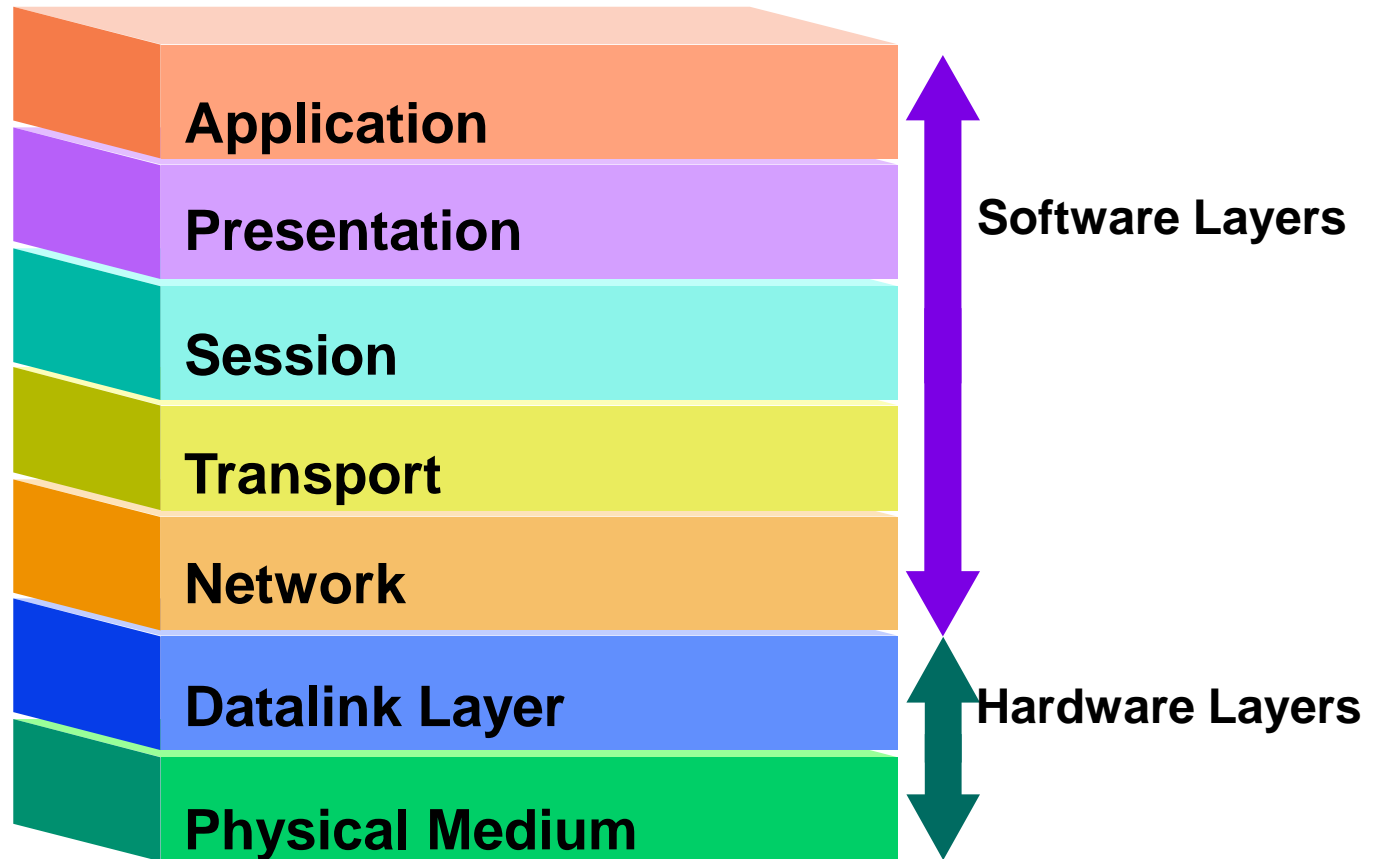
OSI Model: Flow of Data

- Logical Structure
- Data Multiplexing



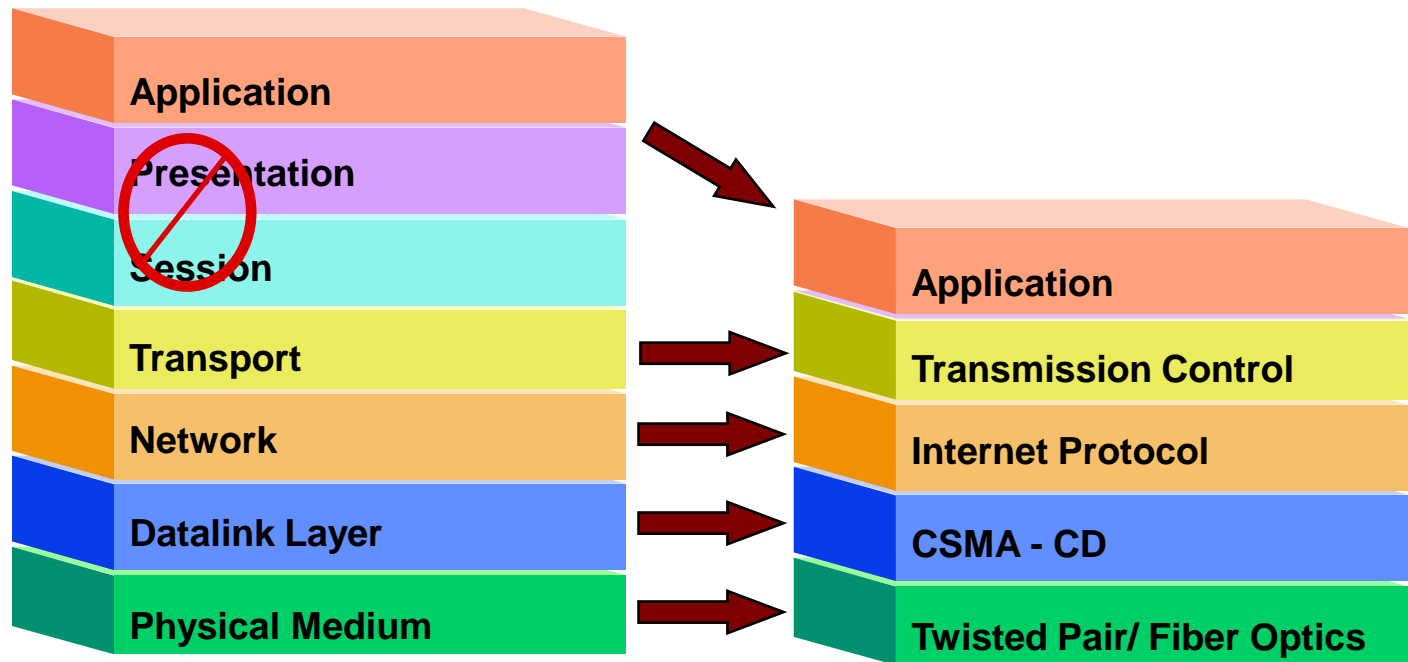
Hardware / Software integration

- The upper five layers are implemented with software
- The lower two layers are implemented with hardware



OSI Model vs. Ethernet -TCP/IP

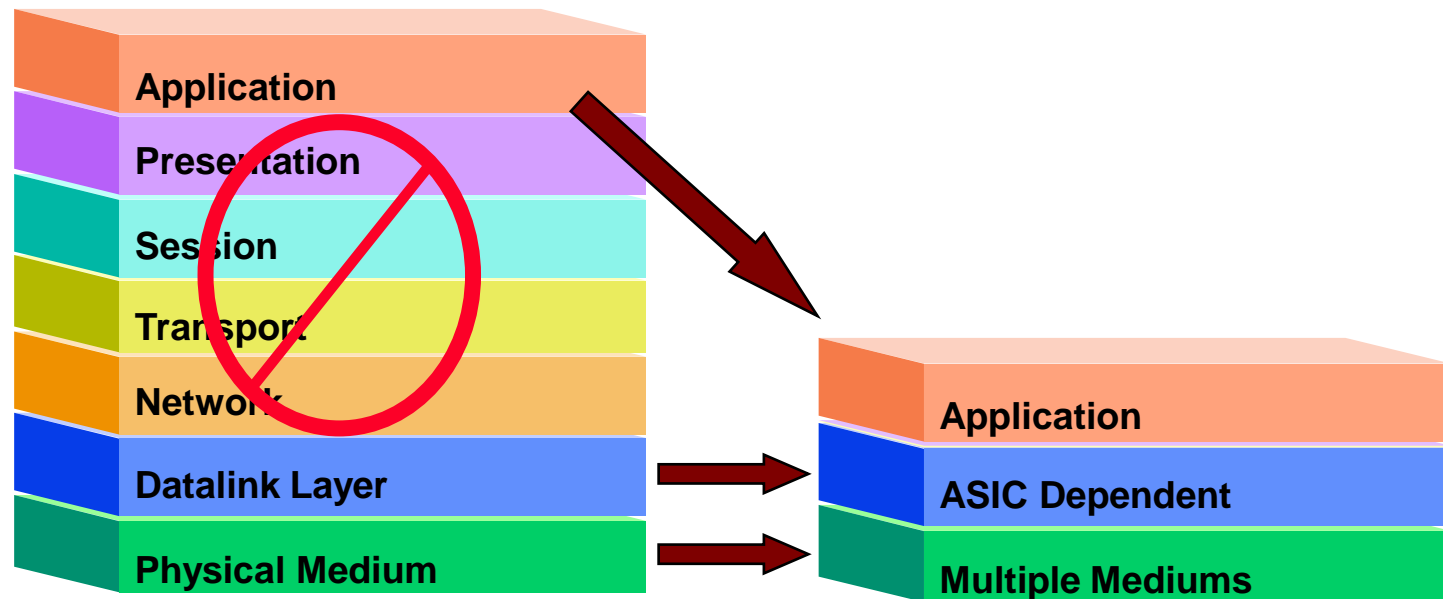
- Ethernet does not incorporate all 7 layers because it's development predated the OSI model



CDMA/CD – Carrier Sense Multiple Access with Collision Detection

OSI Model vs. Field Bus Model

- The typical field bus only utilizes 3 layers. The functions of the other layers are either built into the ASIC chip or the application layer



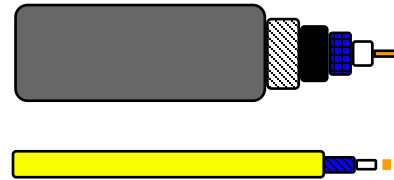
ASIC - Application-Specific Integrated Circuit

What is common to all networks ?

- Application message - dependent on software package
 - The **message** is created by the application package
- Media Access Control
 - The **procedure** for sharing the physical medium
- Logical Topology
 - The **associations** between the devices
- Physical Topology
 - The **layout** of all the device connections
- Transmission Mediums
 - The physical **path** used between the devices

Various Transmission Mediums

■ Coaxial Cable



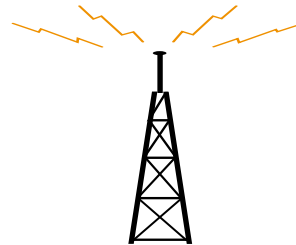
■ Twisted Pair



■ Optical Fiber

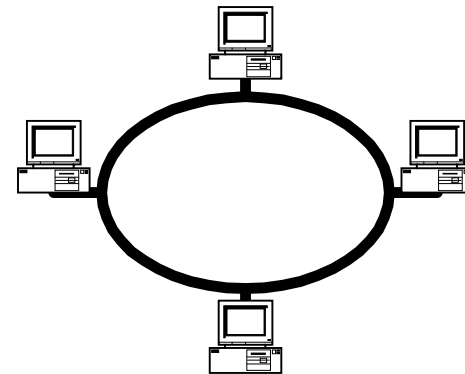
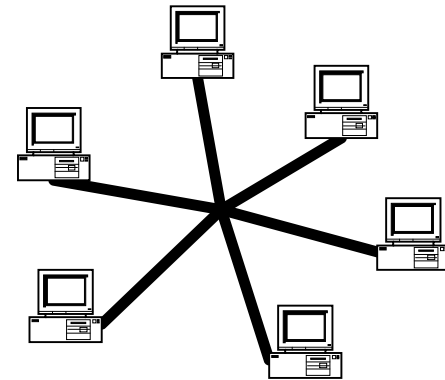
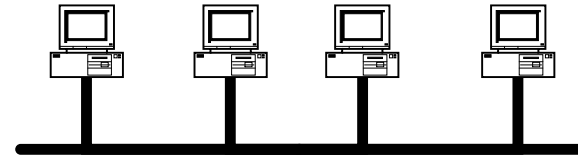


■ Wireless



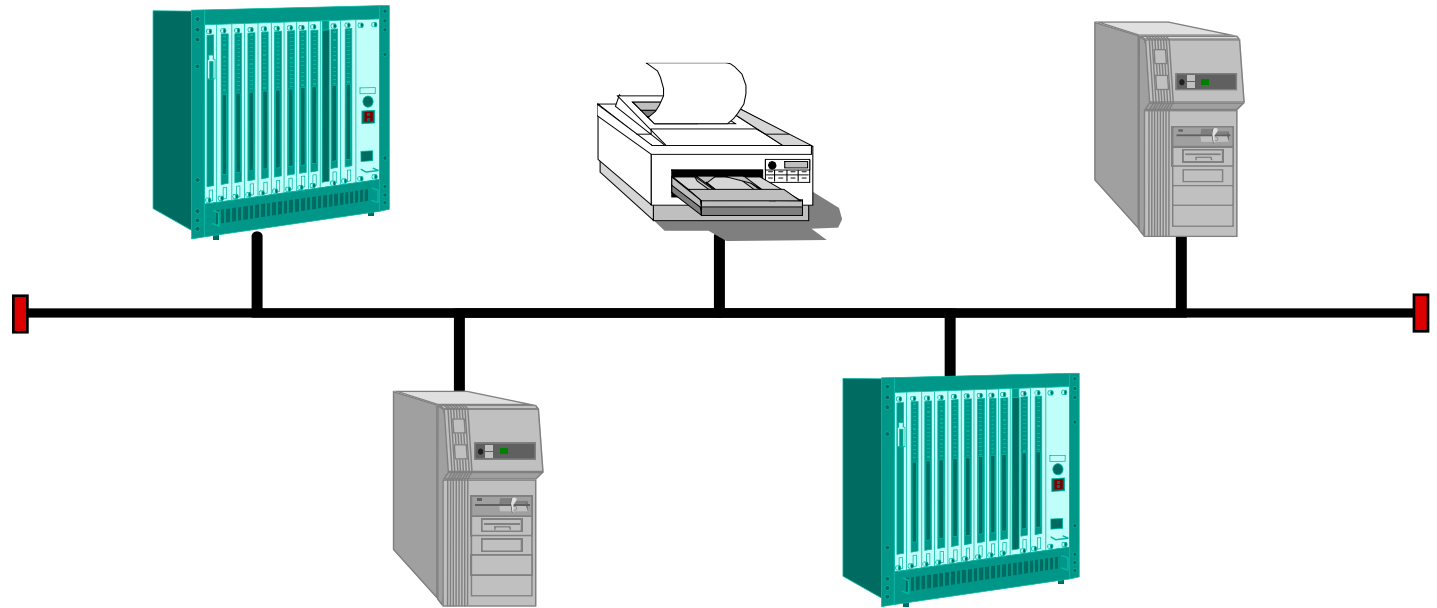
Basic Physical Topologies

- Bus Topology
- Star Topology
- Ring Topology
- Free Topology
- Hybrid Topology



Bus Architecture Topology

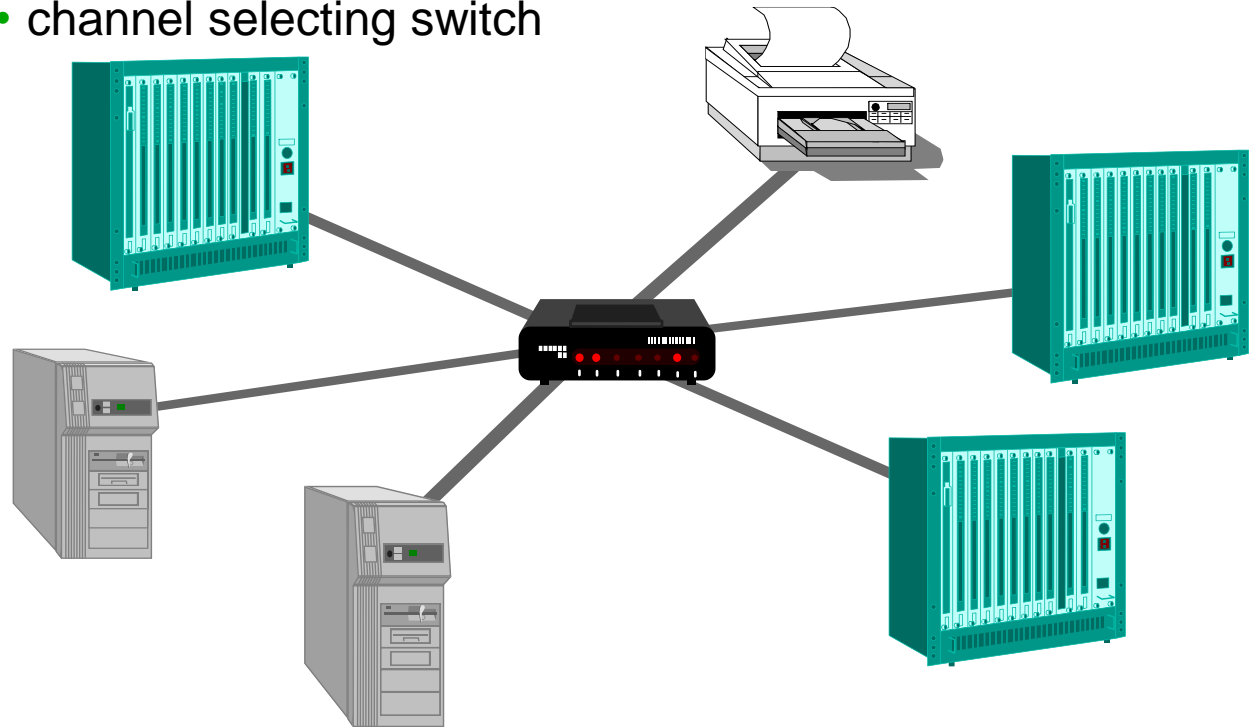
- Single Channel - Broadcast oriented
 - More delicate to maintain signal integrity based on cable selection, distance and drops/taps



Star Architecture Topology

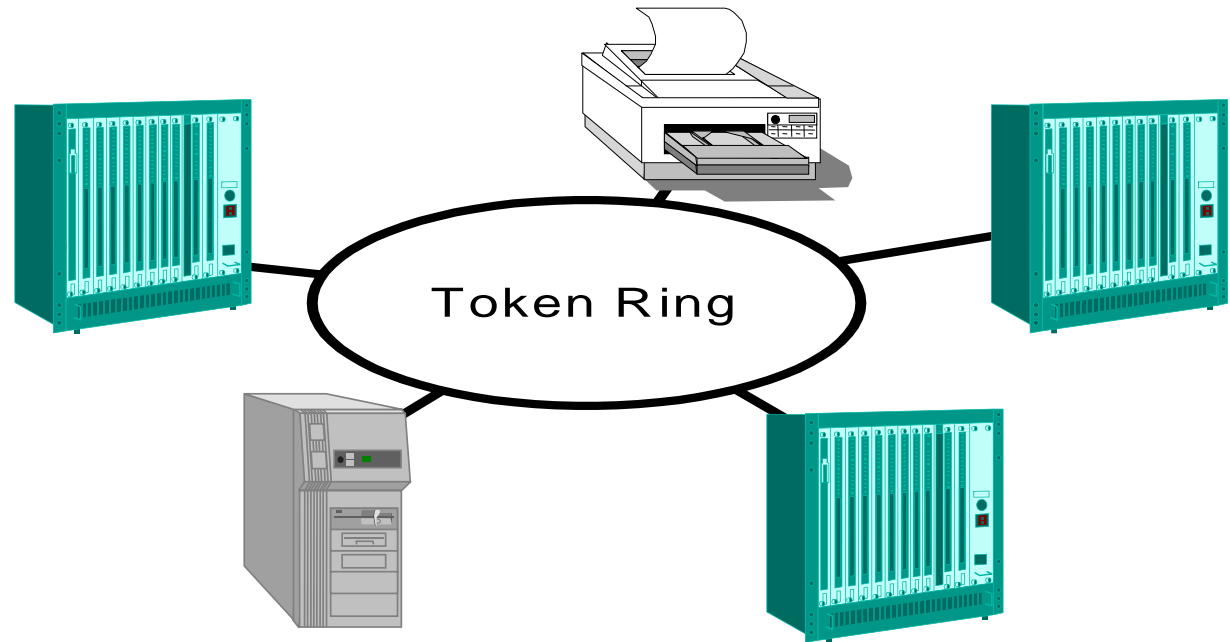
■ Central Hub

- multi-port repeater
 - Loose star = lost network
- channel selecting switch



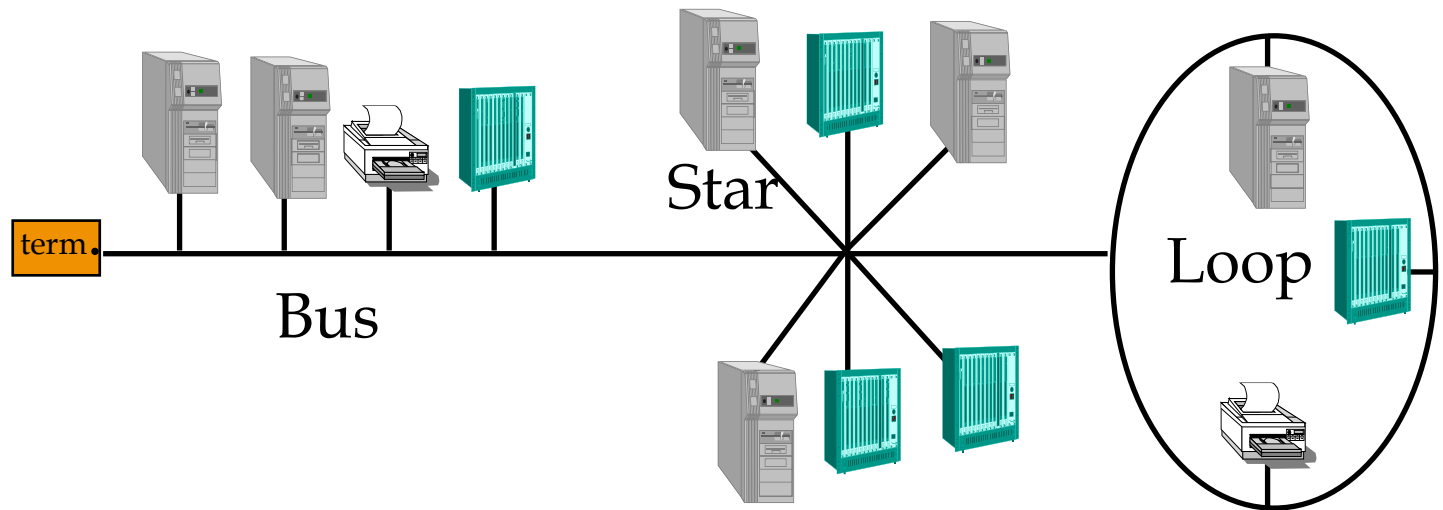
Ring Architecture Topology

- Each device is a repeater
 - Loose node= broken ring



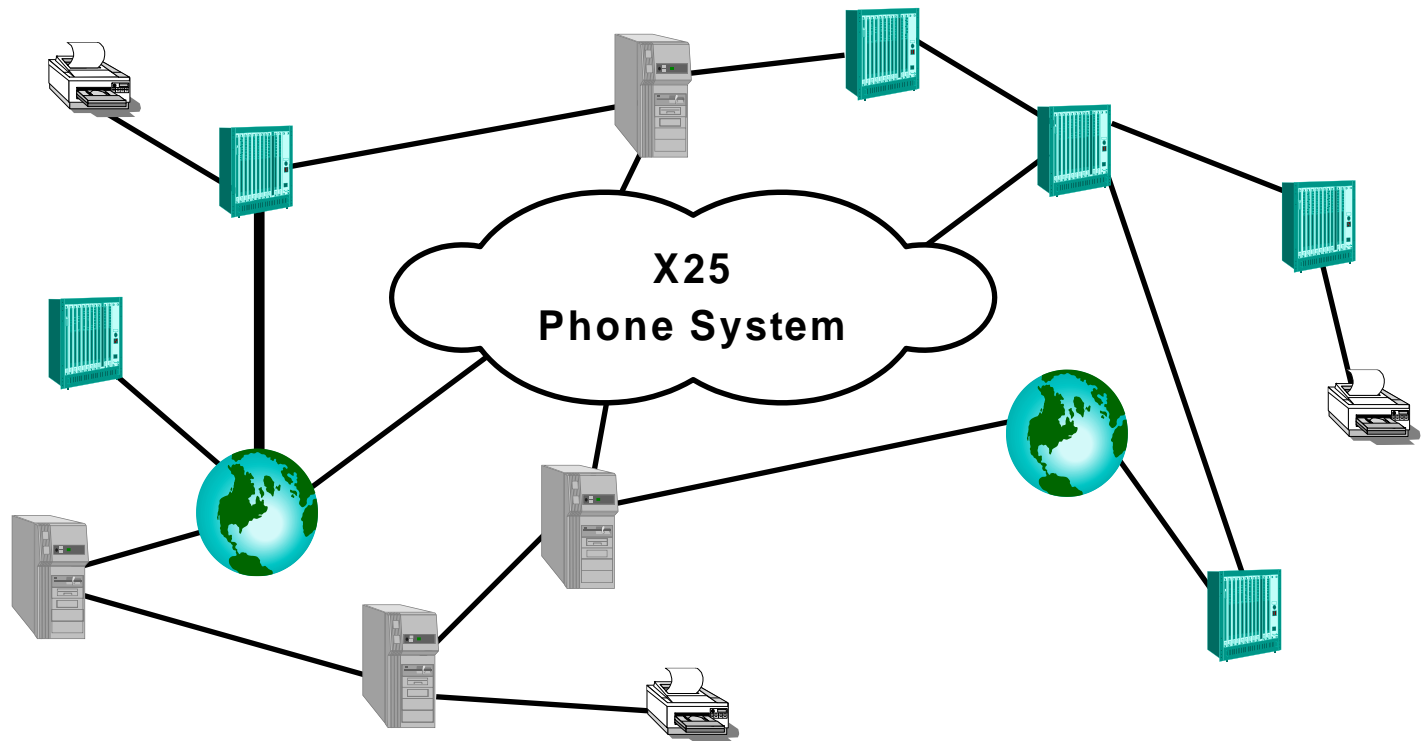
Free Topology

- Devices are wired in any combination of topologies
- Usually requires one terminator



Hybrid Architecture Topology

- Non-rooted Tree Architecture (ex.- Internet)



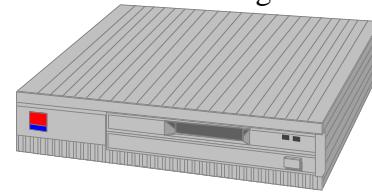
Network Extenders

- Repeater and Hub
- Bridge
- Router
- Gateway

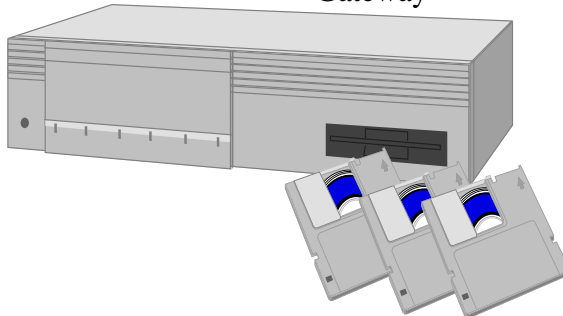
Repeater and Hub



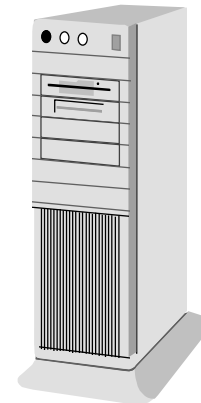
Bridge



Gateway



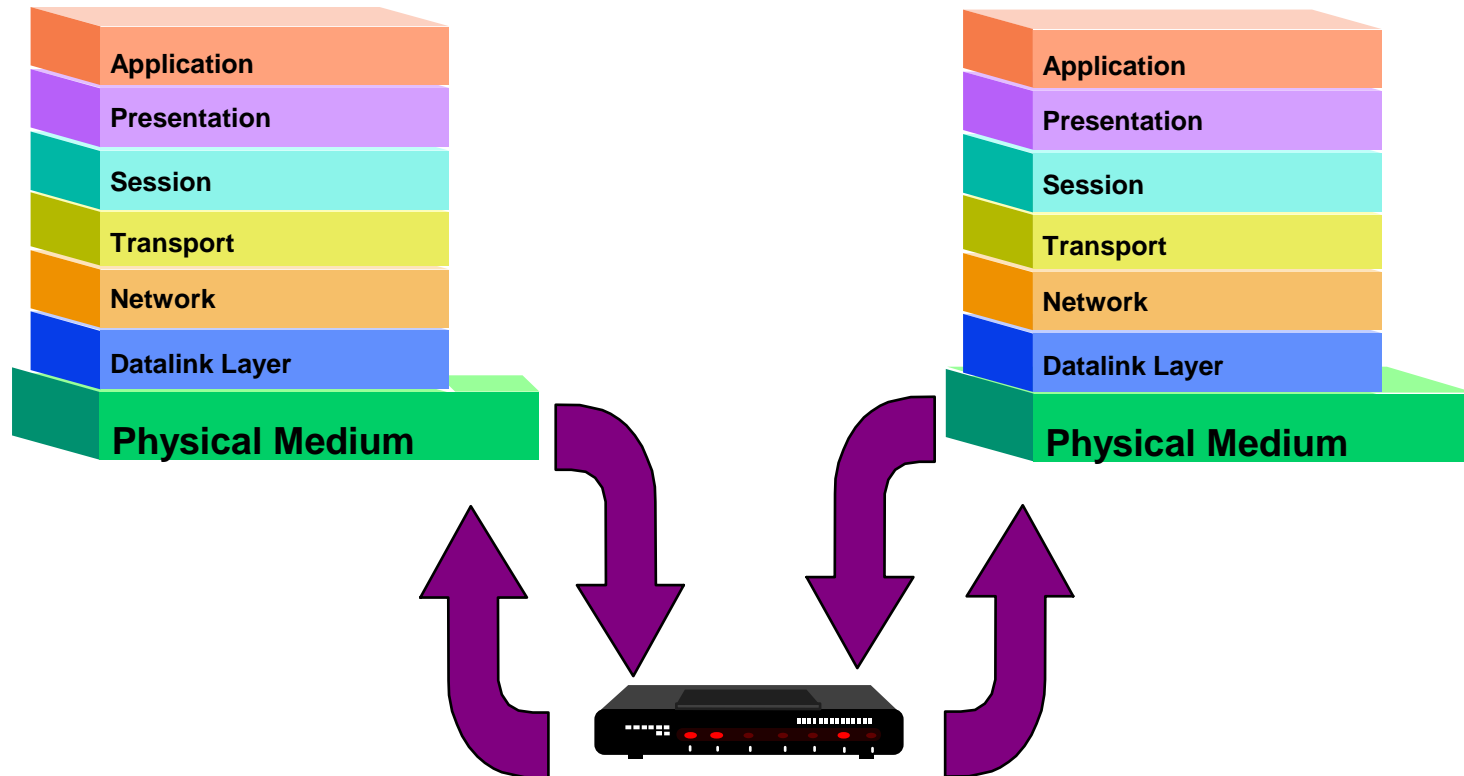
Router



Physical Layer Network Extenders

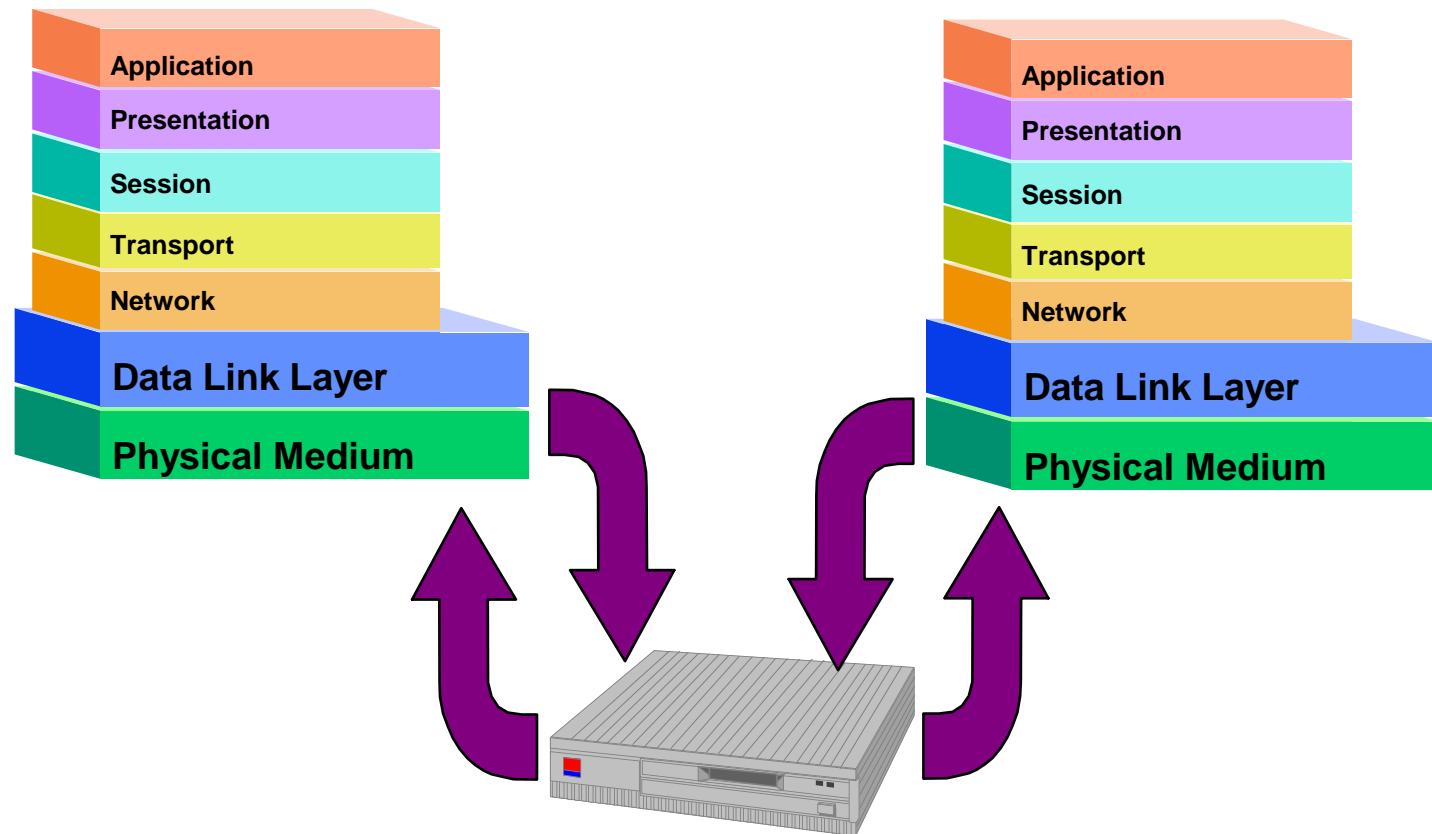
■ Repeater or Hub

- Active repeaters regenerate the signal (store and forward)
- Passive repeater amplifies the signal; adds no latency



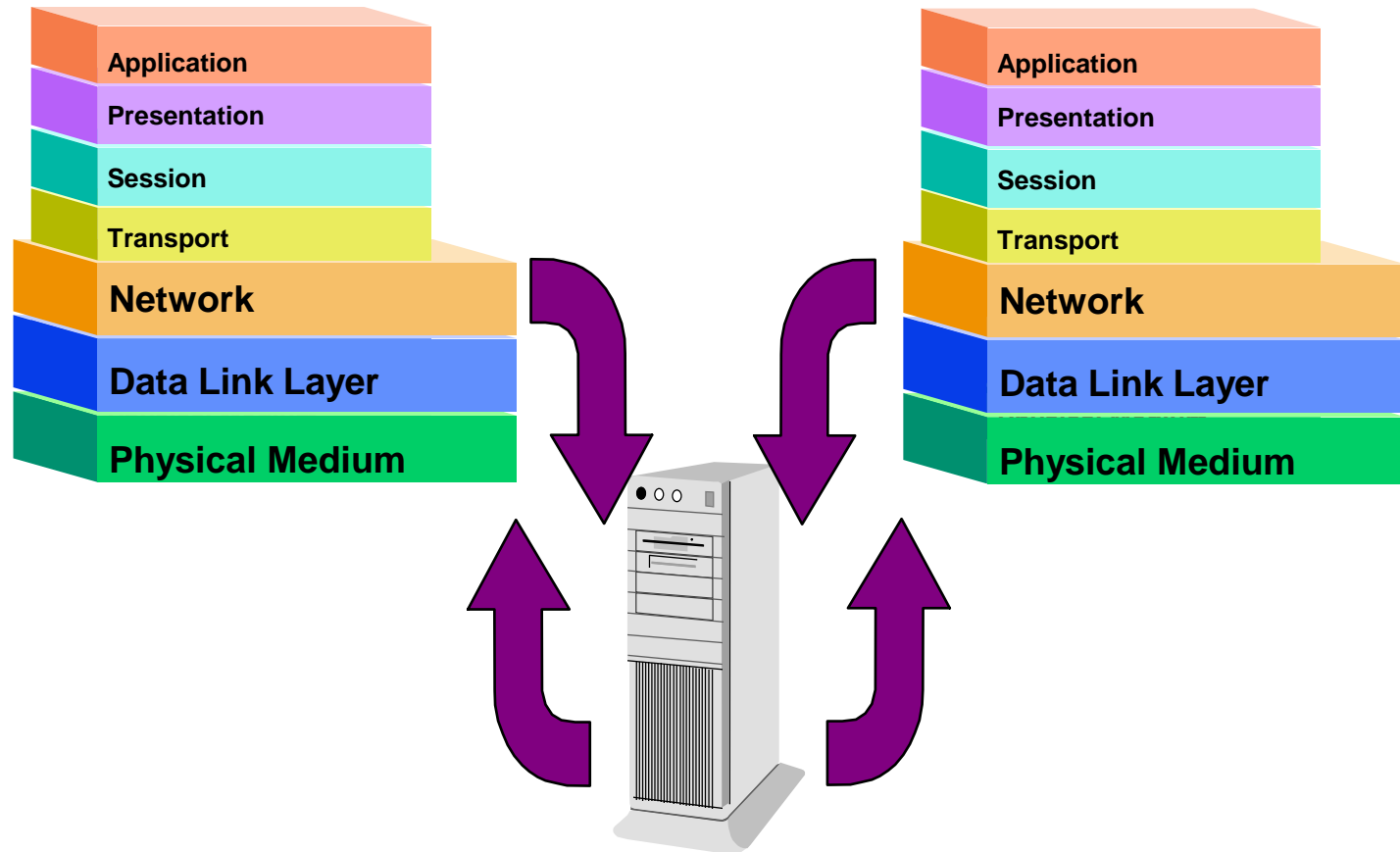
Data Link Layer Network Extenders

- Bridge or Level 2 Switch



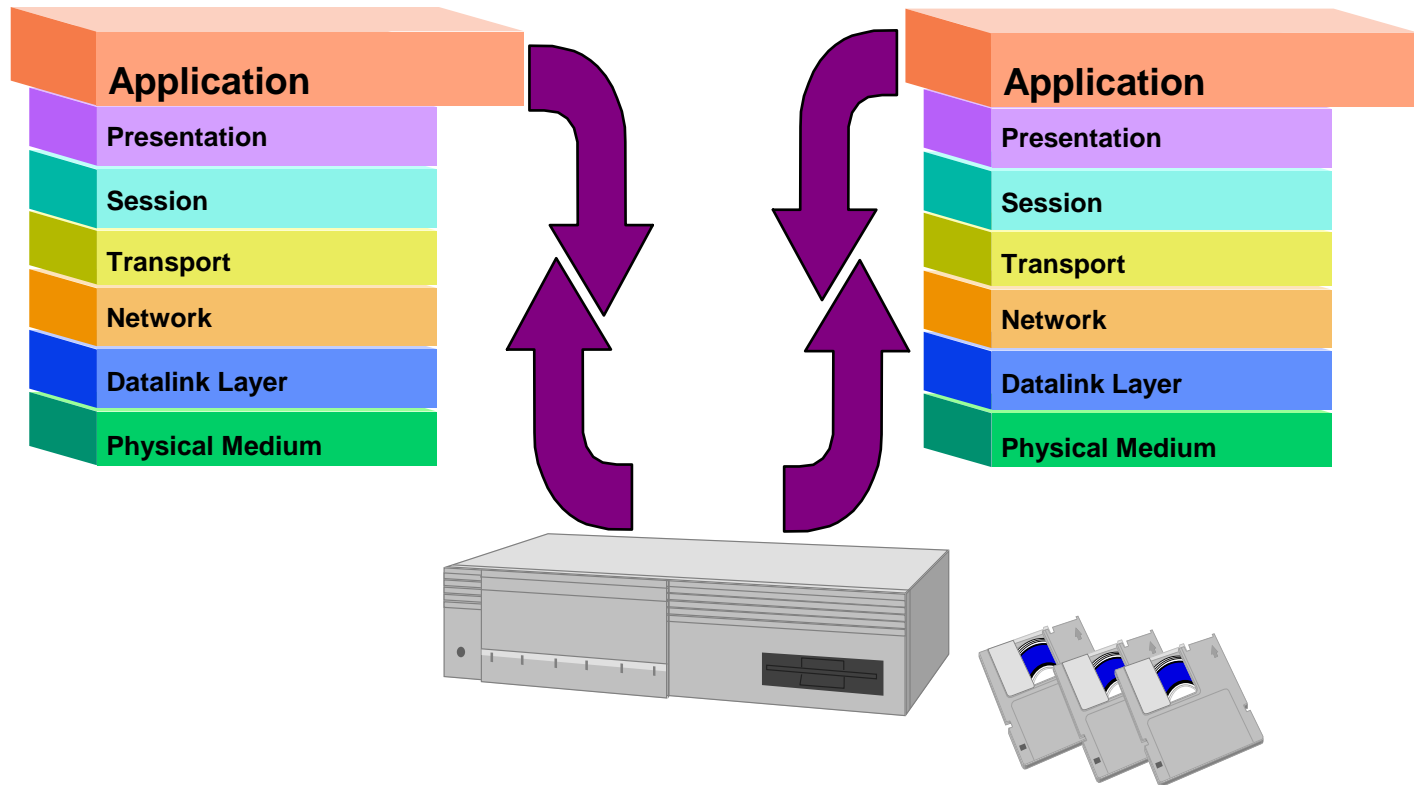
Network Layer Network Extenders

- Router or Level 3 Switch



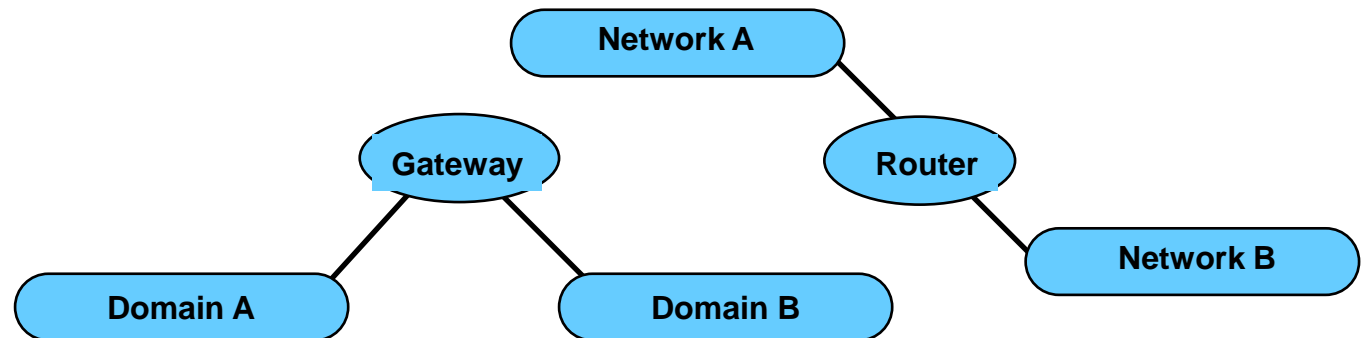
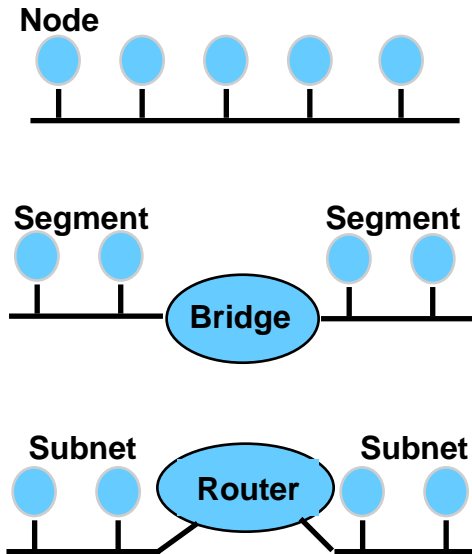
Application Layer Network Extenders

■ Gateway



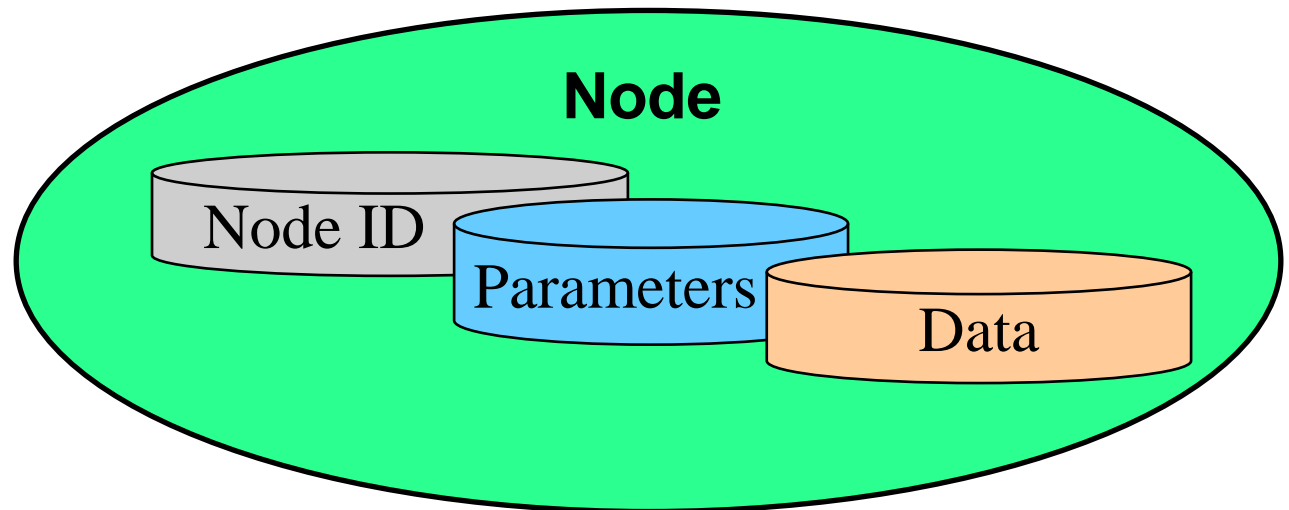
Logical Terminology & Topologies

- Node
- Segment / Channel
- Subnet
- Network
- Domain
- System



A Node

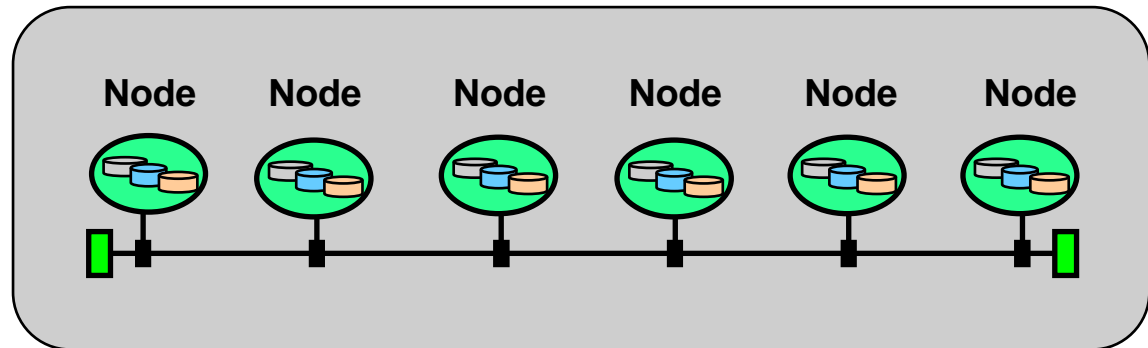
- A collection of SW objects that communicate over a network based upon a single hardware ID.
- A physical device may contain one or more nodes
 - Example - a router



Segment / Channel

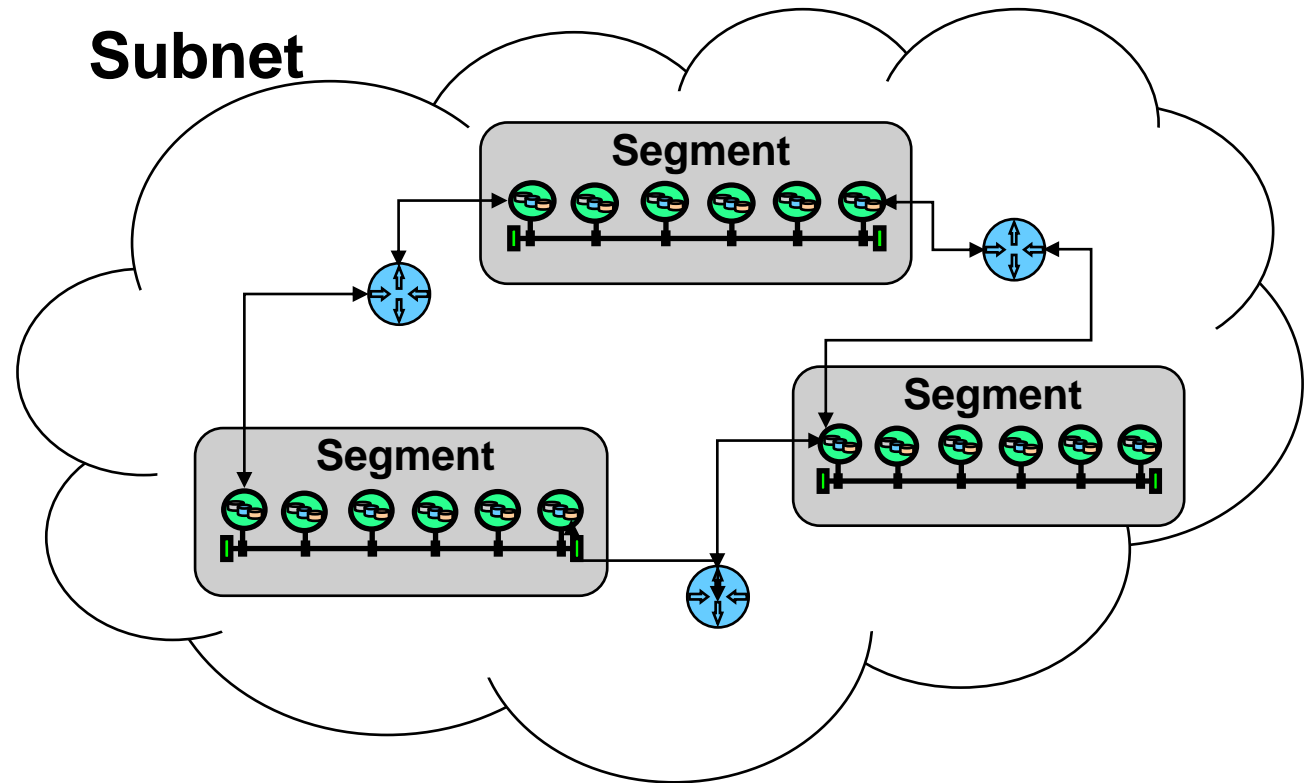
- One or more nodes electrically connected to a continuous section of a transmission media
- May include the trunk line, drop line, multi-port taps and other cable components.

Segment



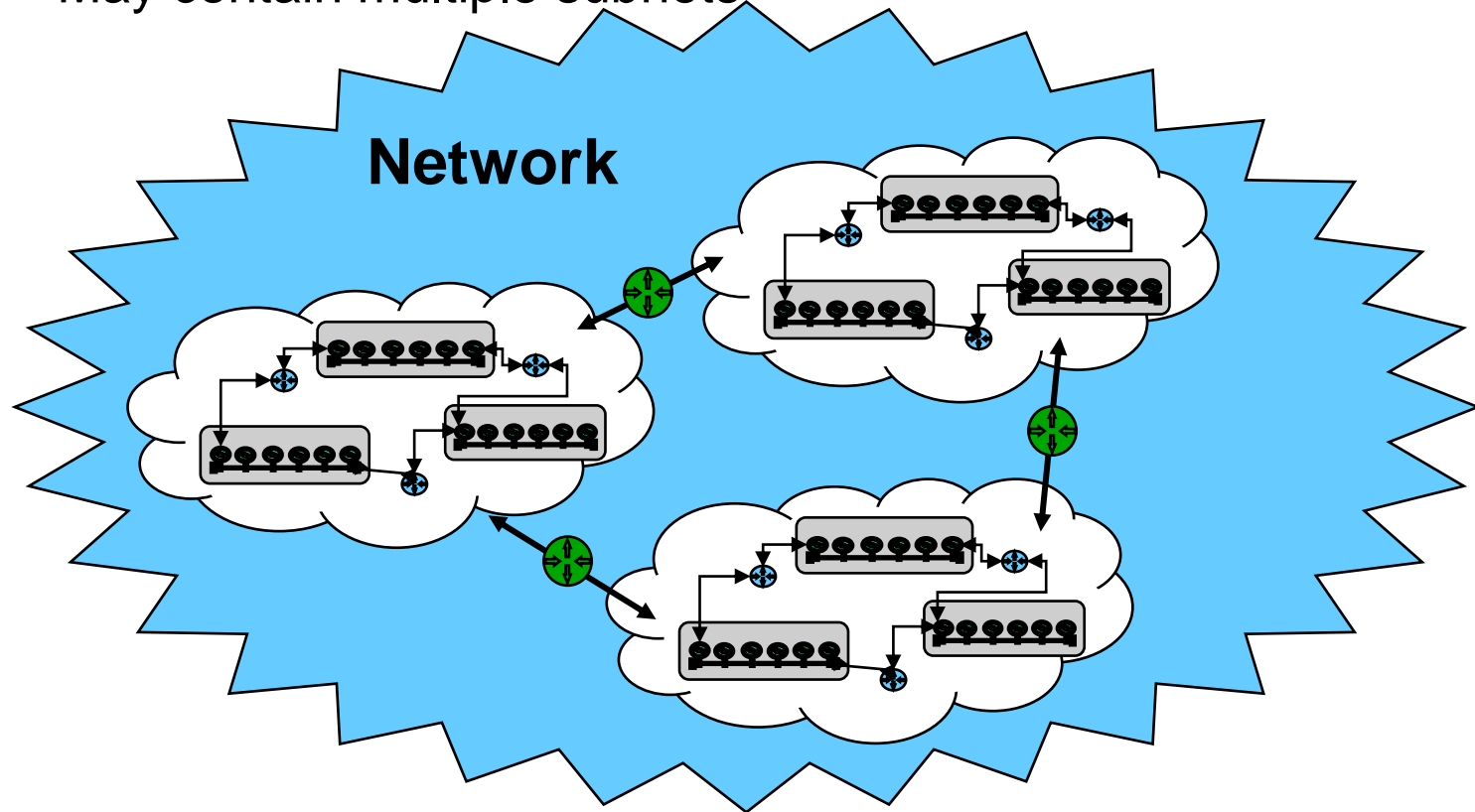
Subnetwork (Subnet)

- One or more nodes sharing a common protocol and a media access arbitration method
- May contain multiple segments



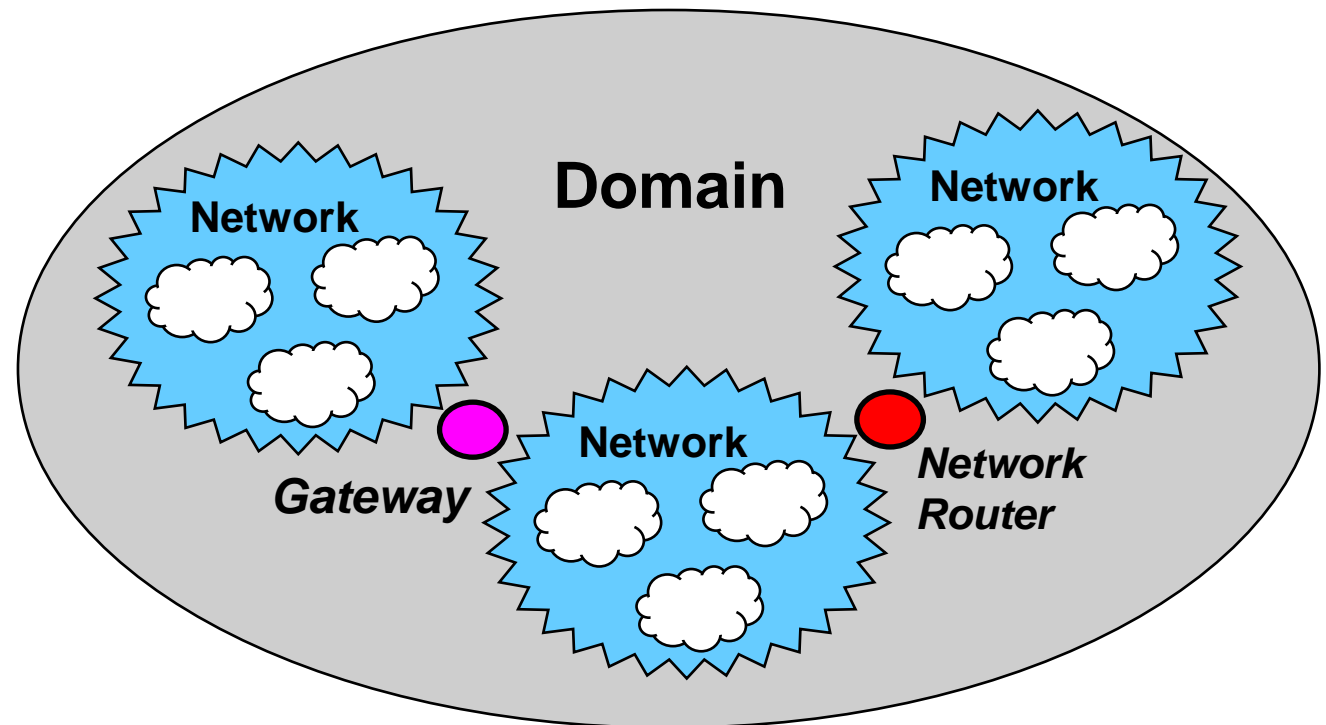
Network

- One or more subnets using a common network level protocol where each hardware ID is unique on the network
- May contain multiple subnets



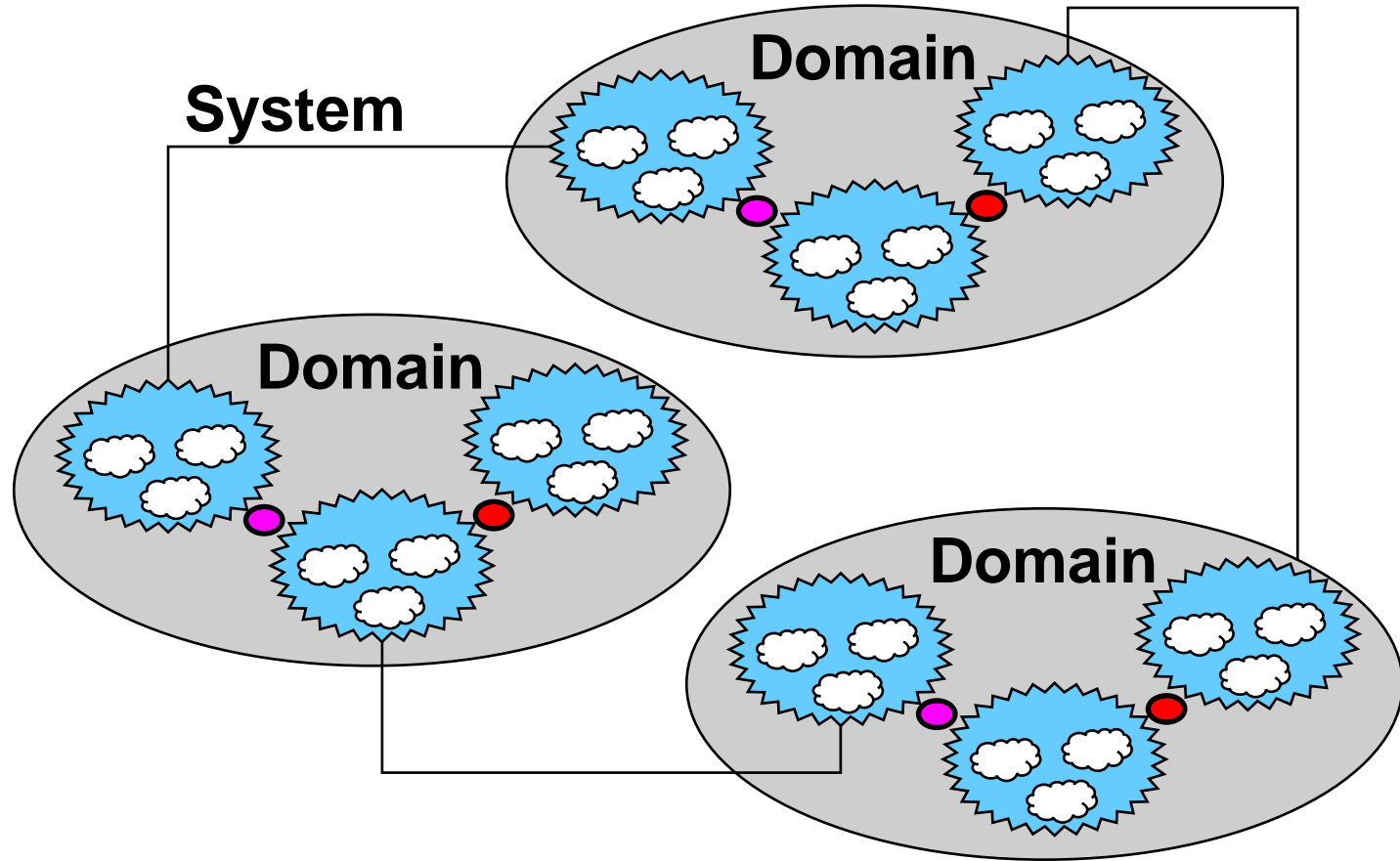
Domain

- One or more networks where each network has a unique ID within the domain
- May contain a variety of network types connected together with gateways and routers



System

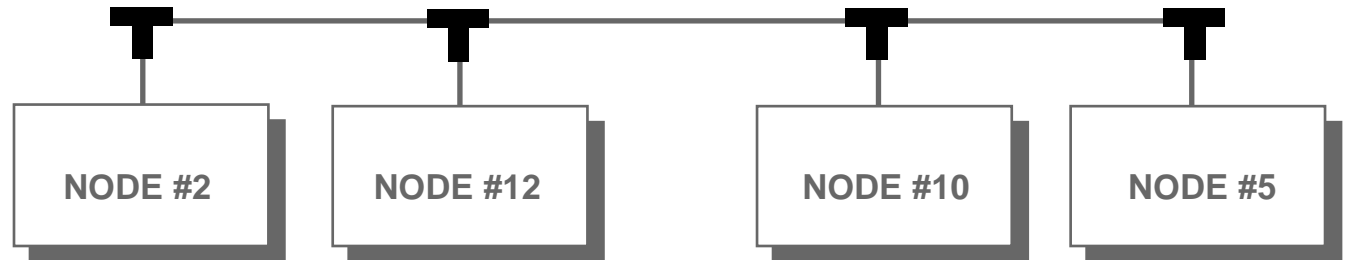
- One or more Domains



Basic Media Access Control Methods

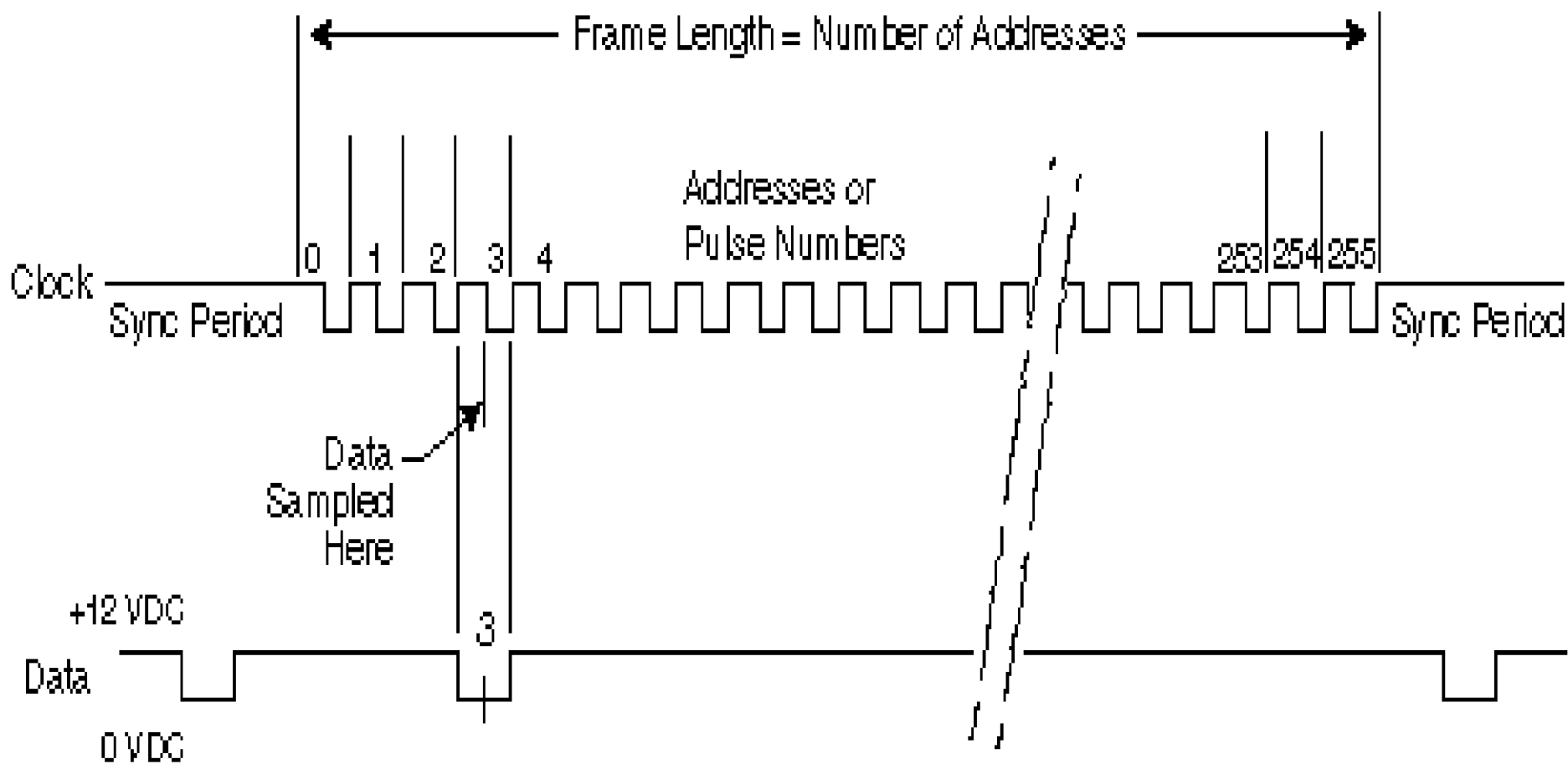
- Time Division Access
- Polled Access
- Token Access
- Carrier Sense Multiple Access

How will they share the common media ?



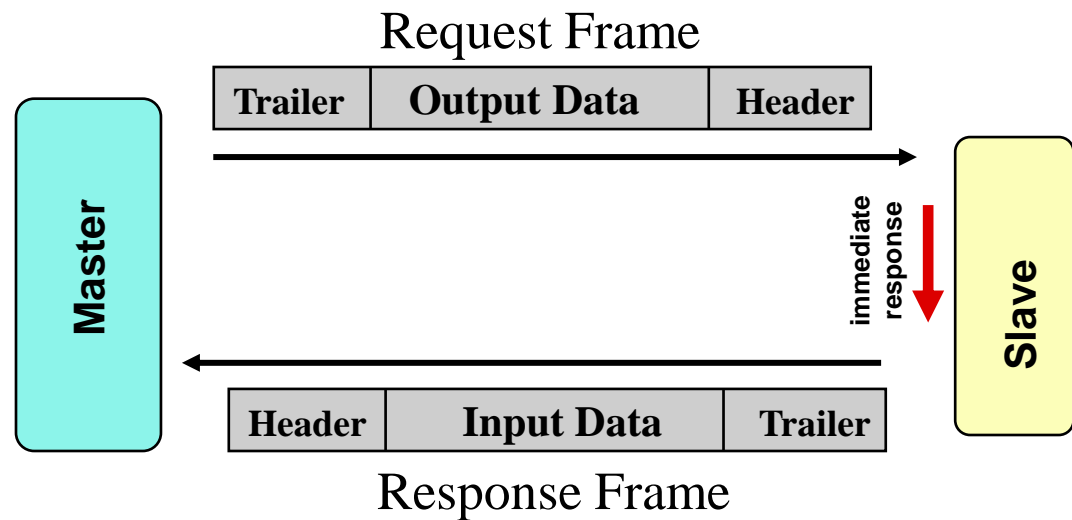
Time Division Access

- Utilizes a master clock
- Each device is assigned a slice of time



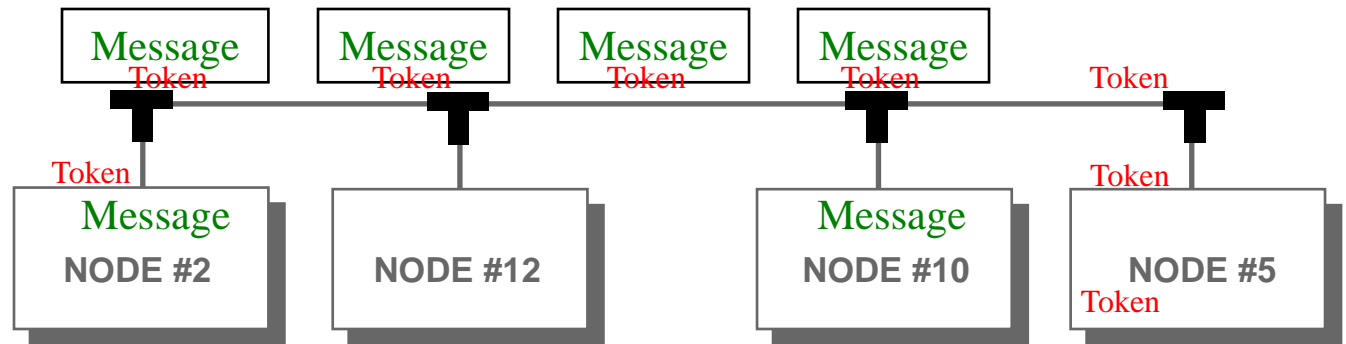
Polled Access

- Utilizes one device as a master
- Other devices are considered slave devices
- The master requests information from each slave
- The slave responds to the request



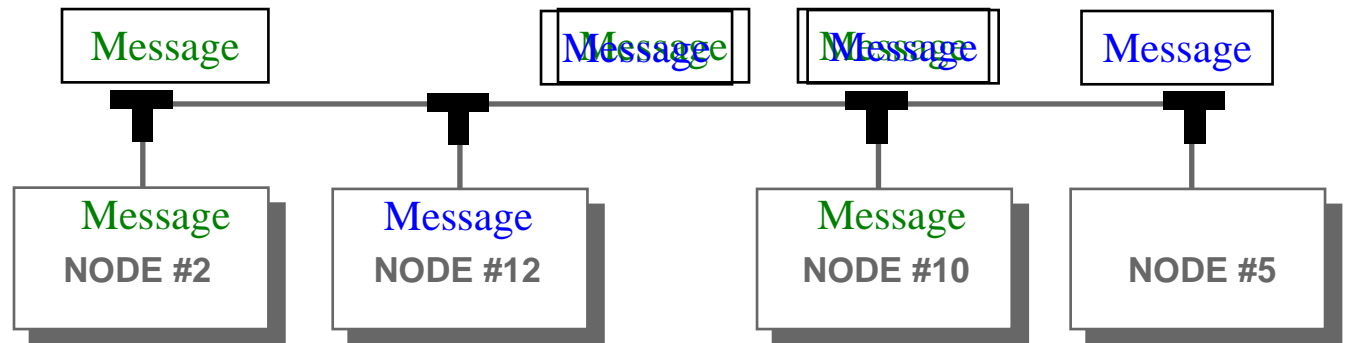
Token Access

- Every device is considered equal
- A token is used to administer access rights
- The token is passed logically among the devices
- Requests are initialized when a device possess the token



Carrier Sense Multiple Access

- No priority is assigned to the devices
- All the devices contend for access
- Messages are only sent if the shared media is not in use



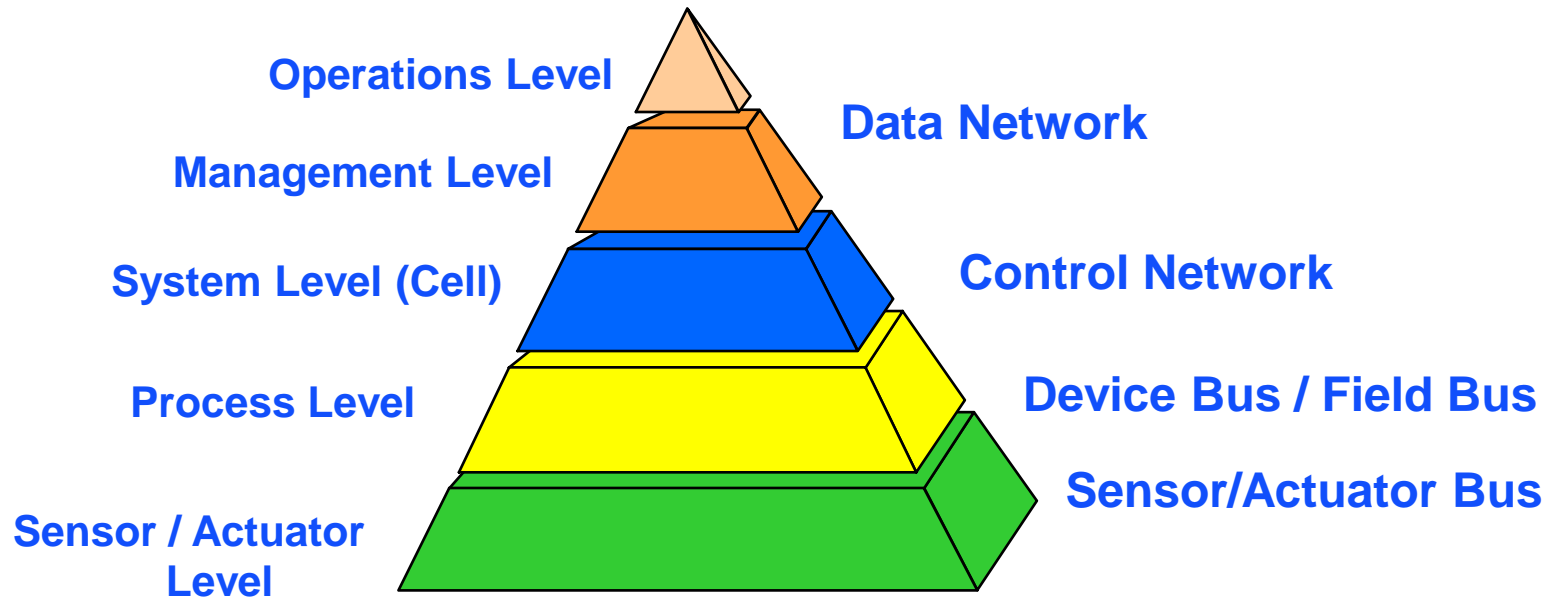
Typical Automation Field Bus Model

- The typical field bus only utilizes 3 layers. The functions of the other layers are either built into the ASIC chip or the application layer
- The application layer will characterize the overall function of the field bus or the network.



Basic Types of Automation Networks

- Data Network
- Control Network
- Device Bus / Field Bus
- Sensor / Actuator Bus



Network Levels - Typical Applications

■ Level 2: Hosts

- Recipe data downloads
- Production data uploads
- Application programming and management

■ Level 1: Control

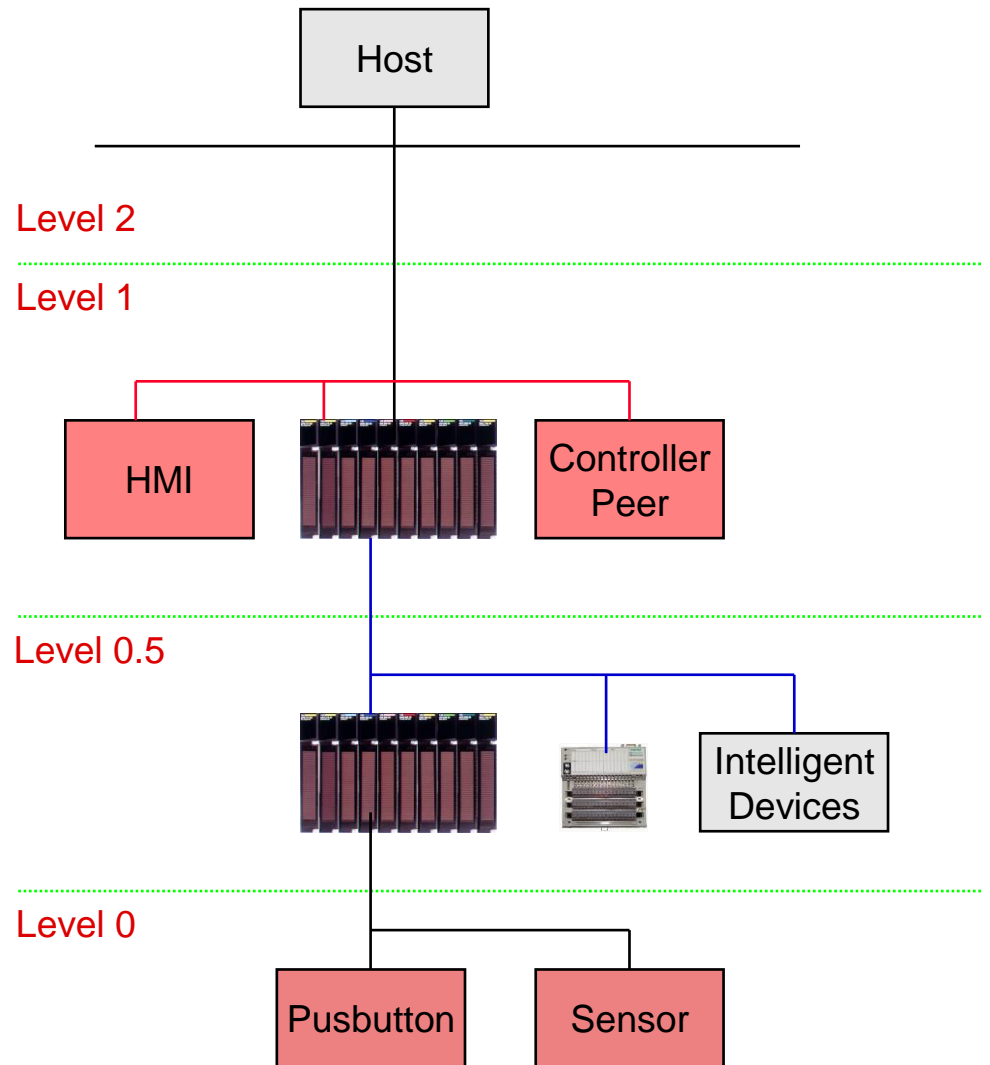
- Peer to peer data
- HMI displays

■ Level 0.5: I/O

- I/O racks and blocks
- Intelligent devices
 - Variable frequency drives
 - Process valves
 - Power meters
 - Complex Sensors

■ Level 0: Devices

- Pushbuttons / Simple sensors



Network Levels - Transmission Characteristics

■ Level 2: Hosts

- Data: blocks or files
- Frequency: seconds to minutes
- Distance: LAN / WAN / Internet

■ Level 1: Control

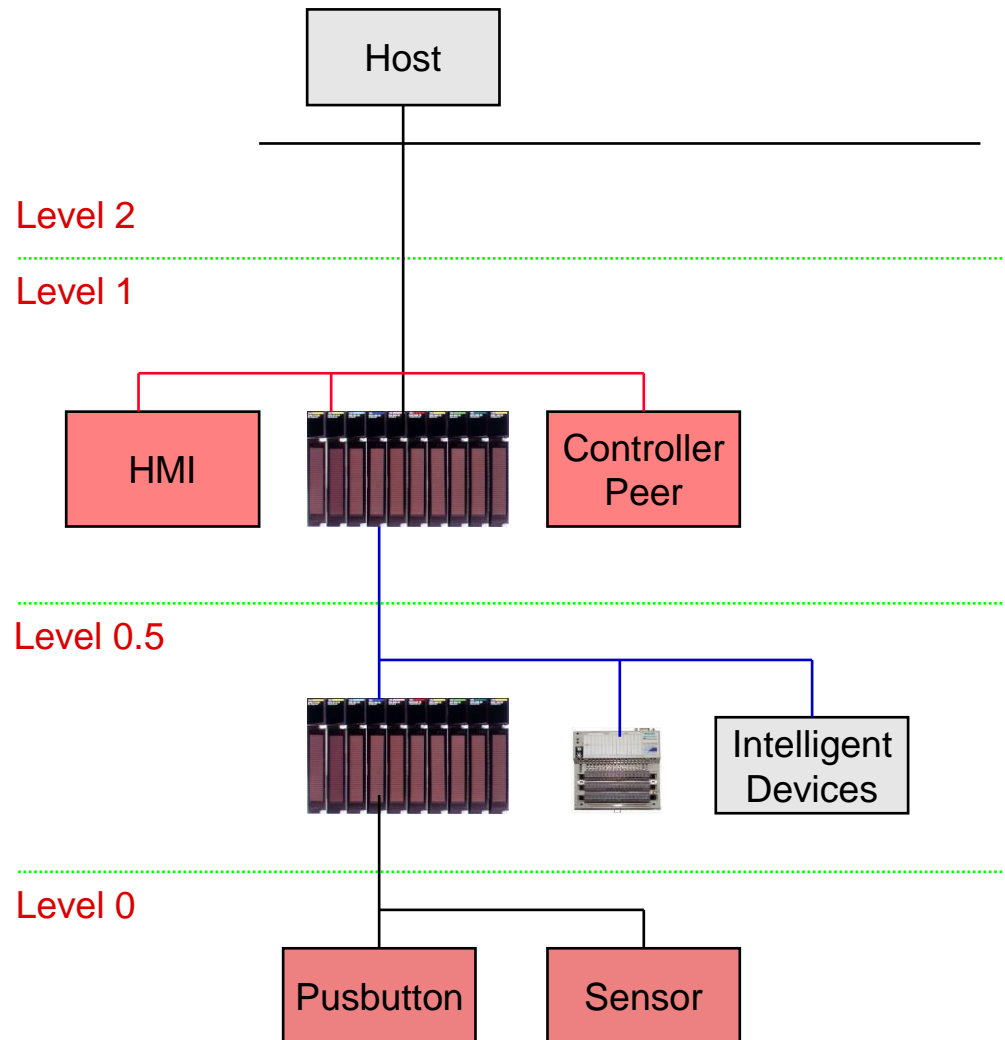
- Data: words or blocks
- Frequency: 50 to 500 Milliseconds
- Distance: 300m

■ Level 0.5: I/O

- Data: bytes or words
- Frequency: 20 to 100 Milliseconds
- Distance: 300m

■ Level 0: Devices

- Data: bits
- Frequency: 5 to 50 Milliseconds
- Distance: 30m



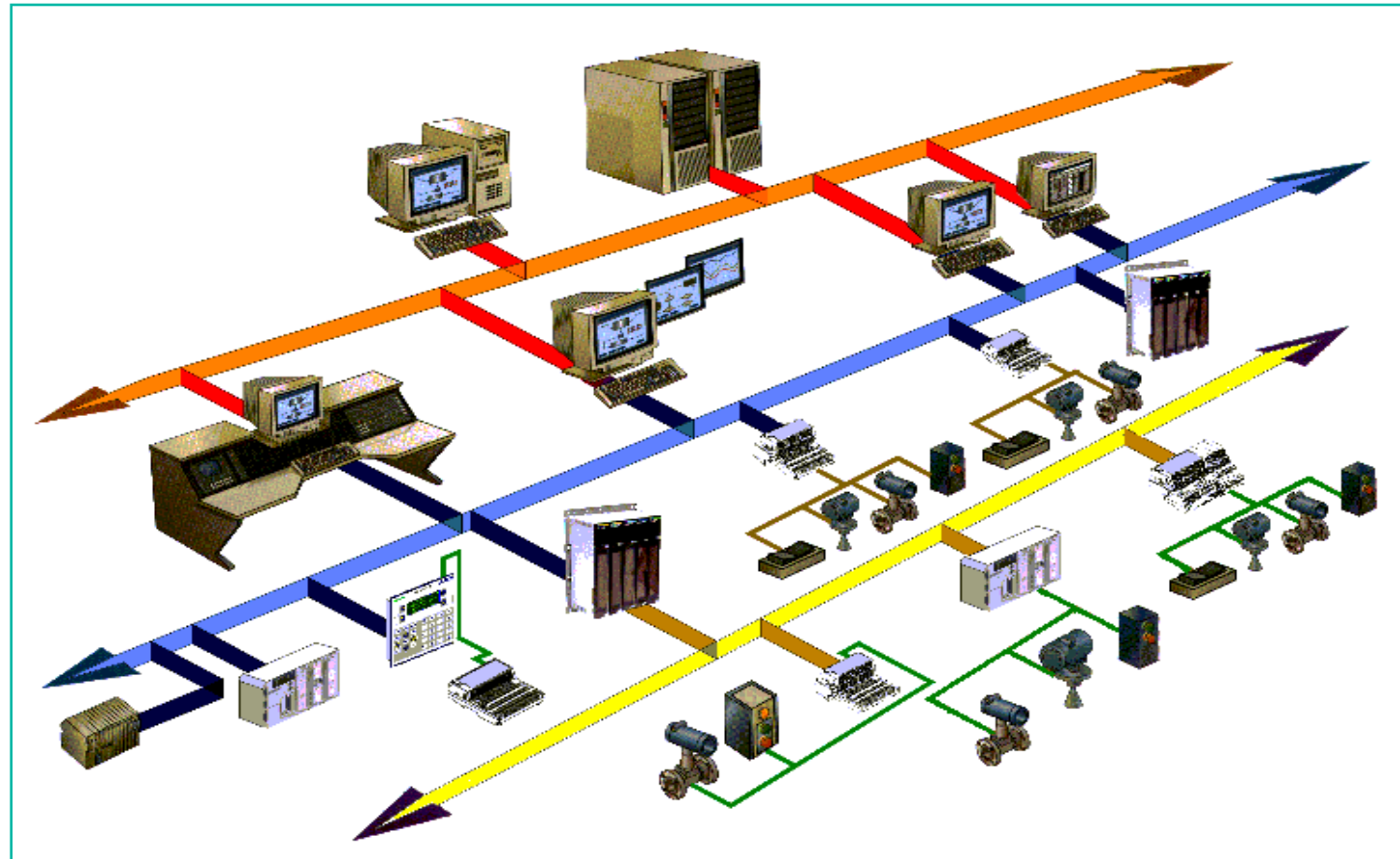
Communication Network Levels

Level	Network Name	Information Type
Data Network Information Level (Level 2)	Ethernet TCP/IP Token ring	File Transfers
Control Network Word Level (Level 1)	Modbus Plus FIPWay Device Net Ethernet TCP/IP	Data Transfer Program
Device / Field Bus Byte & Word Level (Level 0.5)	Remote I/O Interbus TM Profibus DP TM DeviceNet TM Lonworks TM Fip IO TM	Discrete Analog Set Points Words
Sensor/ Actuator Bus Bit Level (Level 0)	Lonworks TM Seriplex TM ASi TM	Discrete Limited Analog

* TM is a trade mark

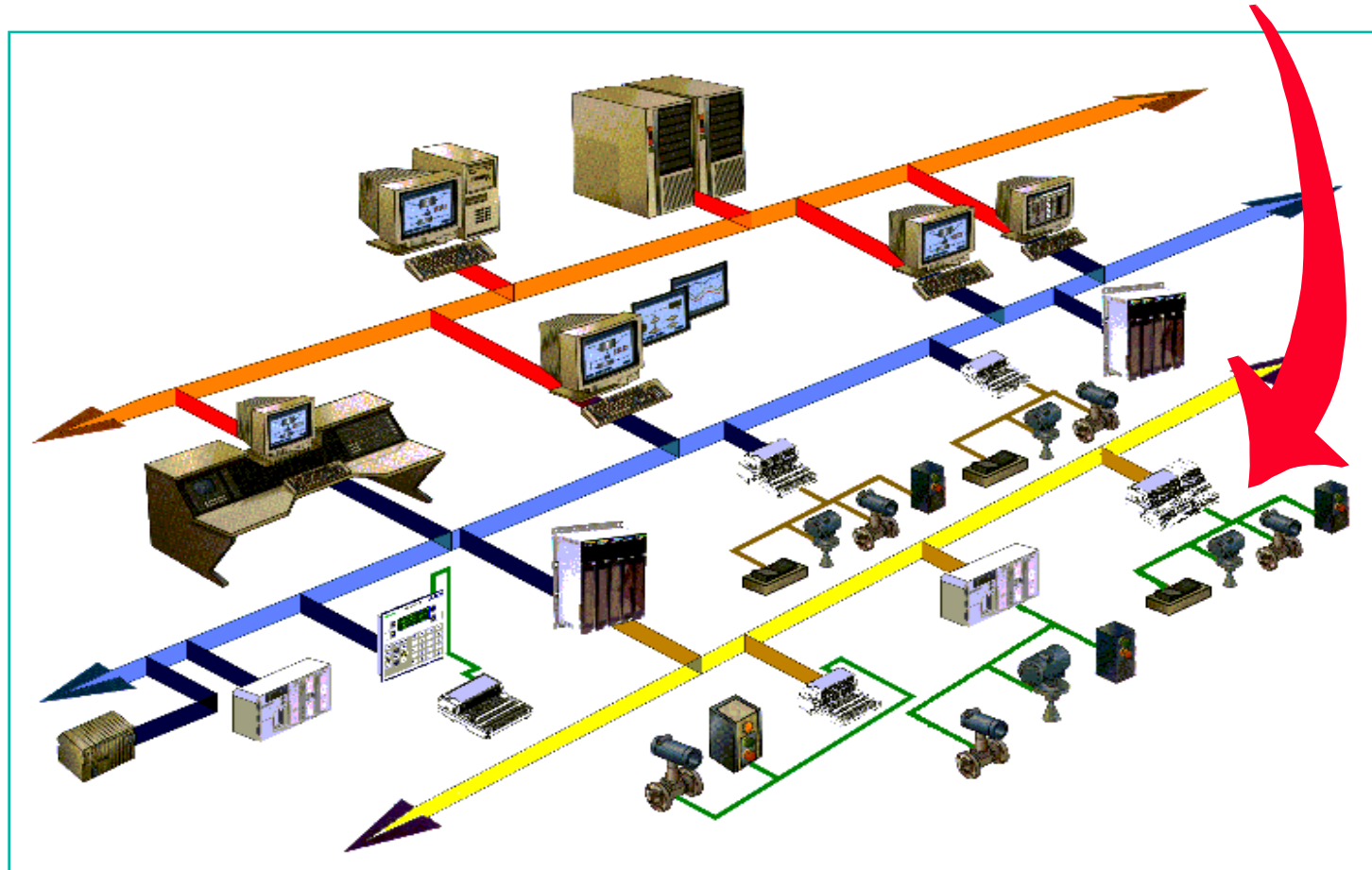
Automation & Control System Hierarchy

- Automation and control systems can incorporate them all





Sensor / Actuator Bus

- Sensor Actuator Buses are bit level oriented

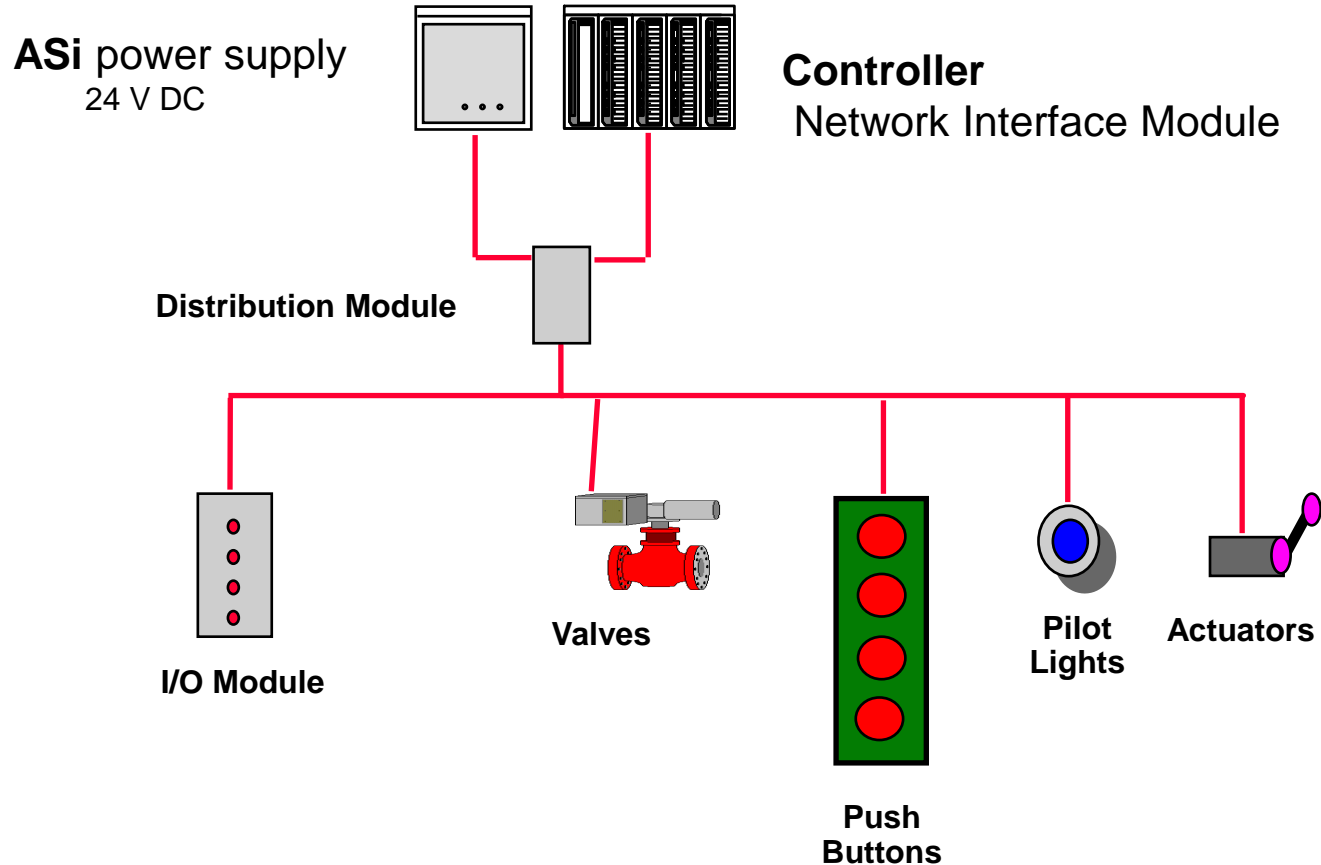


Sensor / Actuator Bus

	<u>ASi</u> 	 Seriplex	<u>LONworks</u>
Speed	5ms	0.7ms	16-100 ms
Nodes	31 nodes	64 nodes	64 nodes
	31	510	32385
Distance	100m	1500m	500m
Message Size	4 bits	1 bit	228 bytes
Node Cost	Low	Low	High
Installed Cost	Low	Low	High
Physical Size	Small	Small	Large

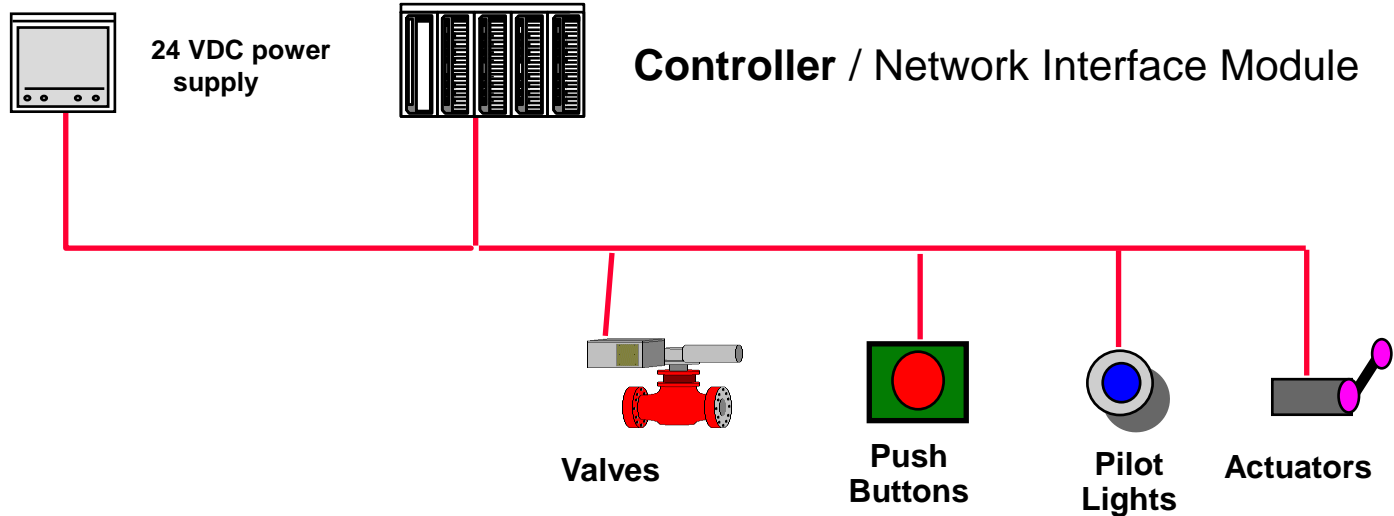
ASi Communication Network

- Master - Slave Network – polled media access method
- Utilizes a special 2 conductor cable for data and power
- Data size - 4 bits
- 31 maximum slave devices on the bus



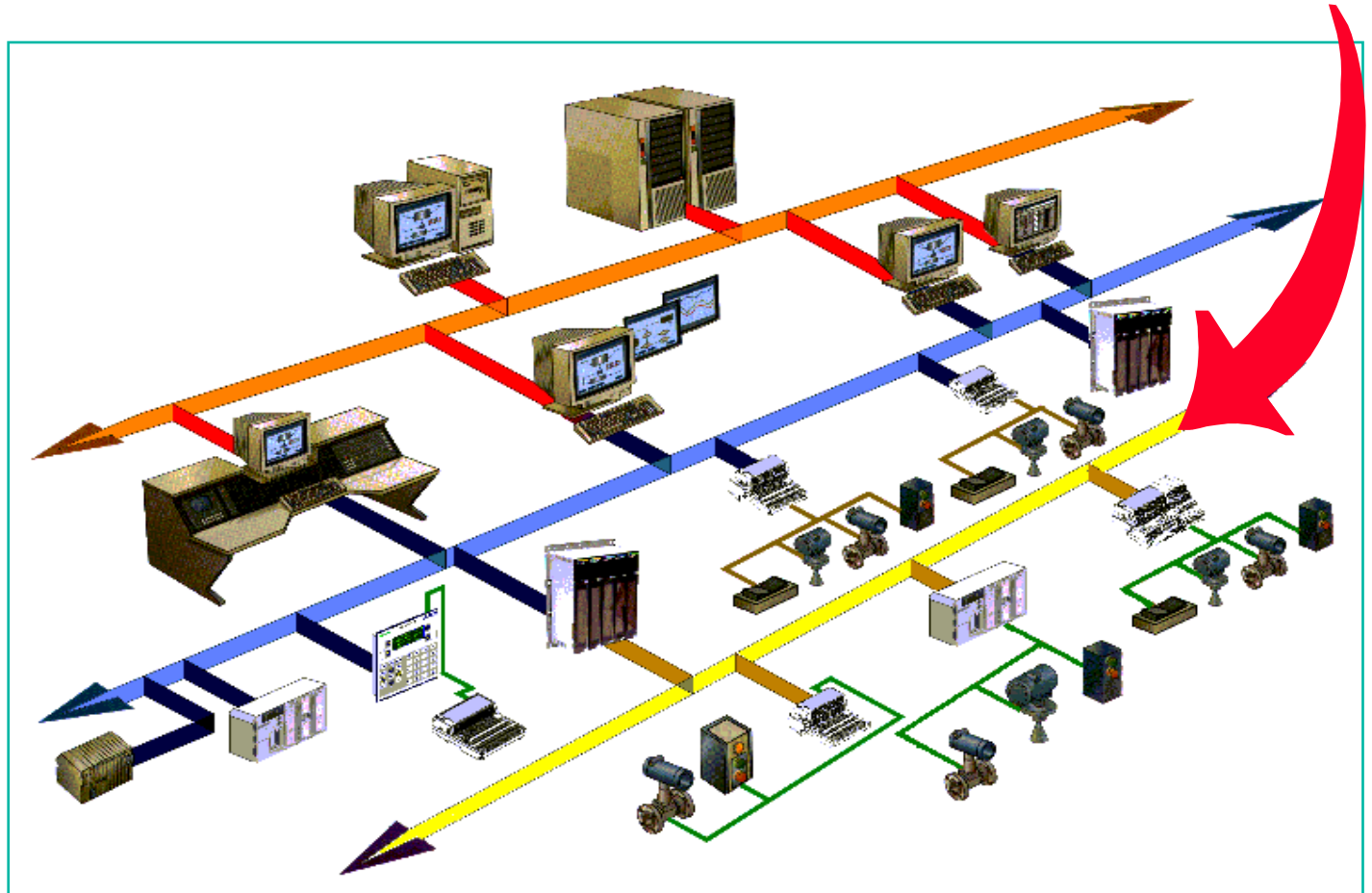
Serial Communication Network

- Master - slave or peer to peer network type
- Time division media access control - 100kHz clock
- Special 4 conductor cable - clock, power, data, common
- Clock pulse number equals address





Device / Field Bus

- Device Buses incorporate intelligent devices

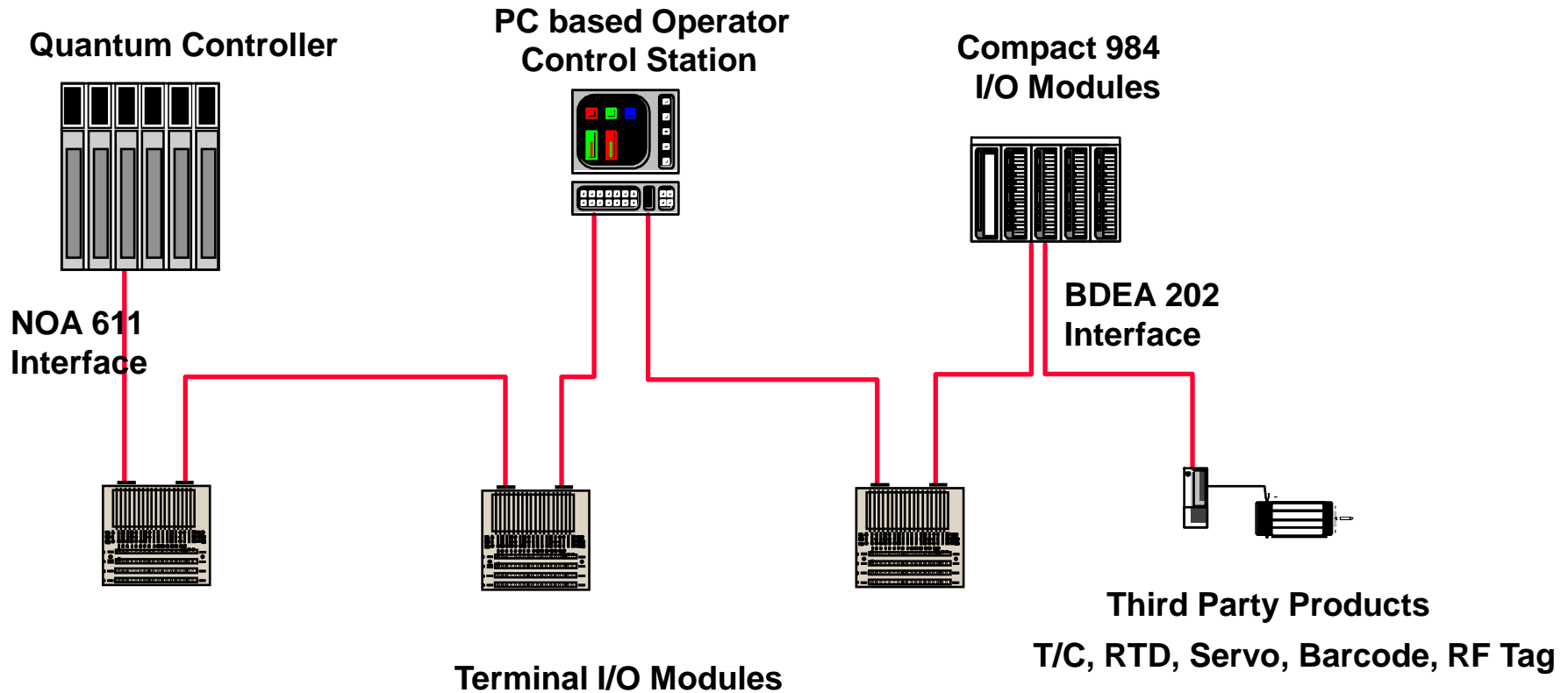


Device / Field Bus

	<u>Interbus S</u> 	Device Net (CAN)	<u>Profibus-DP</u> 	
Speed	4 ms	10-52 ms	10 ms	
Nodes	256 nodes	64 nodes	32 nodes	
Max nodes	N/A	64	32	
Distance	13km	250m	1200m	
Message Size		8 bytes	246 bytes	
Node Cost	Medium	Medium	Medium	
Installed Cost	Low	High	High	
Physical Size	Medium	Medium	Large	

Interbus Network

- Many third party suppliers
- Large physical topology 12km
- Price competitive
- Loss of device or connection disables complete system



InterBus-S

■ Strong points

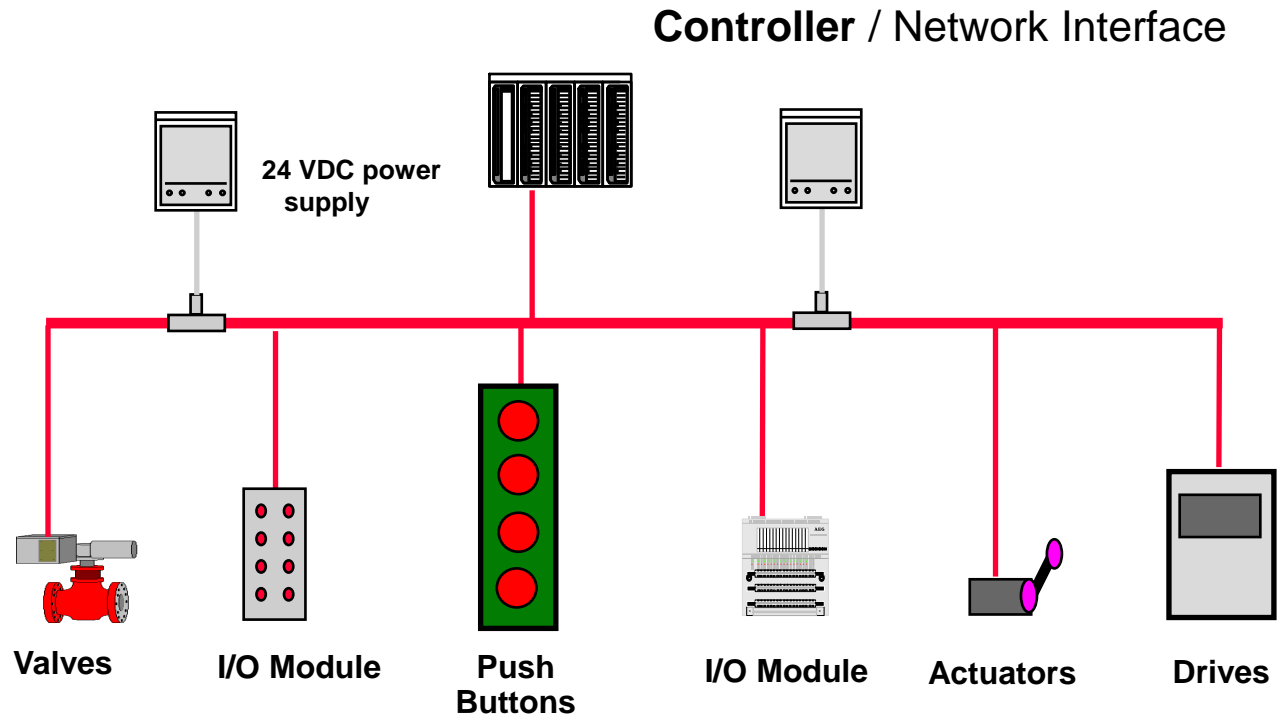
- High performance I/O exchanges, deterministic protocol
 - Approx. 7 ms for 128 devices with 32 I/O (2048 input bits, 2048 output bits)
 - Built-in reflex functions
- Standardized profiles , interoperability, unique installation tool : CMD tool, product catalogue
- Different physical layers (loop sensor)
- Activation / deactivation segment
- Fault localization

■ Weak points

- Low performance message handling
 - Adapted to parameter setting only. 1 to 8 words per device and per bus cycle
 - No slave to slave communication
- No diffusion (variable or message handling)
- One level of priority for cyclic variables
- No downgrade mode
 - The bus stops in the event of a fault on the device
 - When the bus restarts, the following devices cannot be accessed
- DIN standard
 - Similar to Phoenix Contact strategy (components, policies, etc)

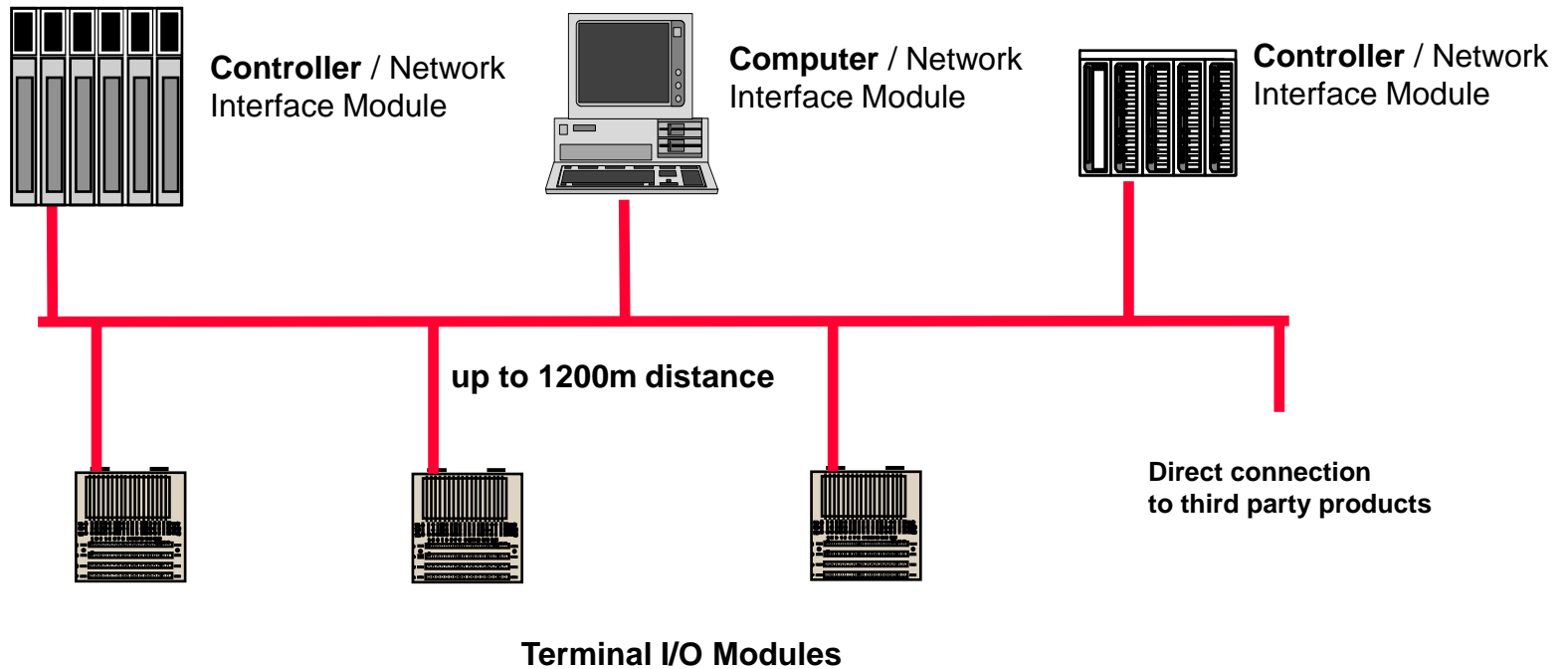
DeviceNet Communication Network

- Limited physical topology 500m



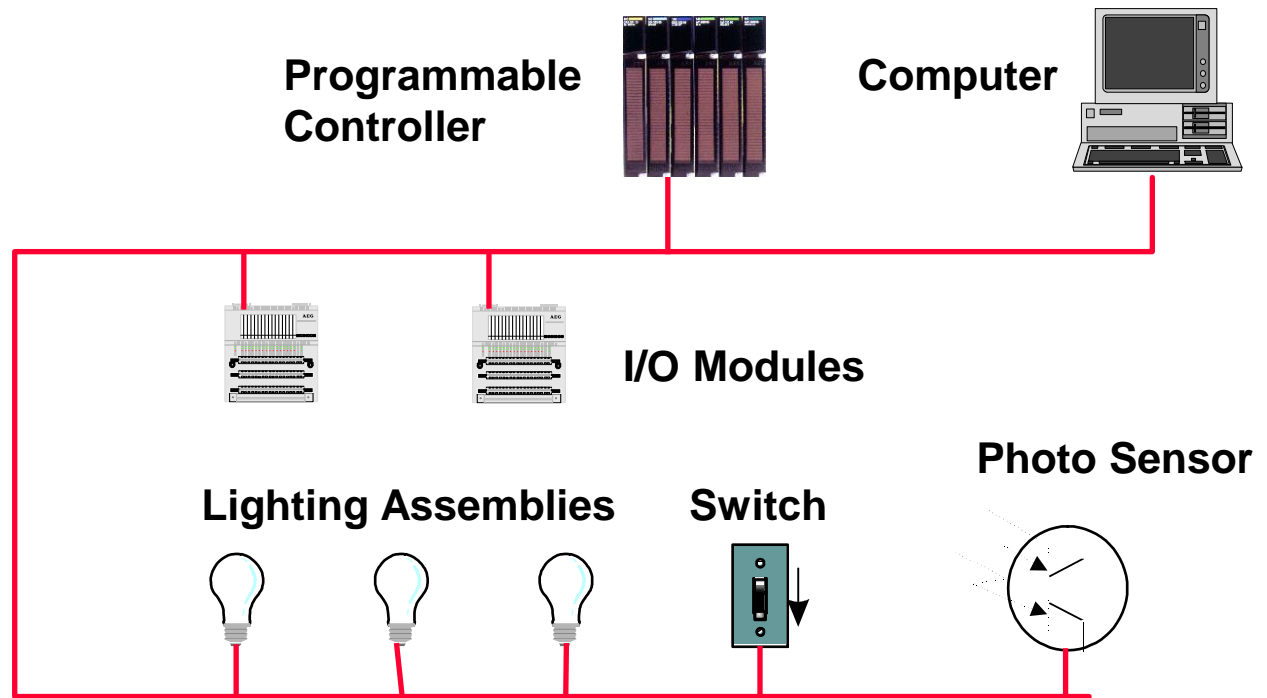
Profibus DP Network

- Large physical topology 24km (fiber)
- Difficult to configure and maintain; needs multiple software pieces



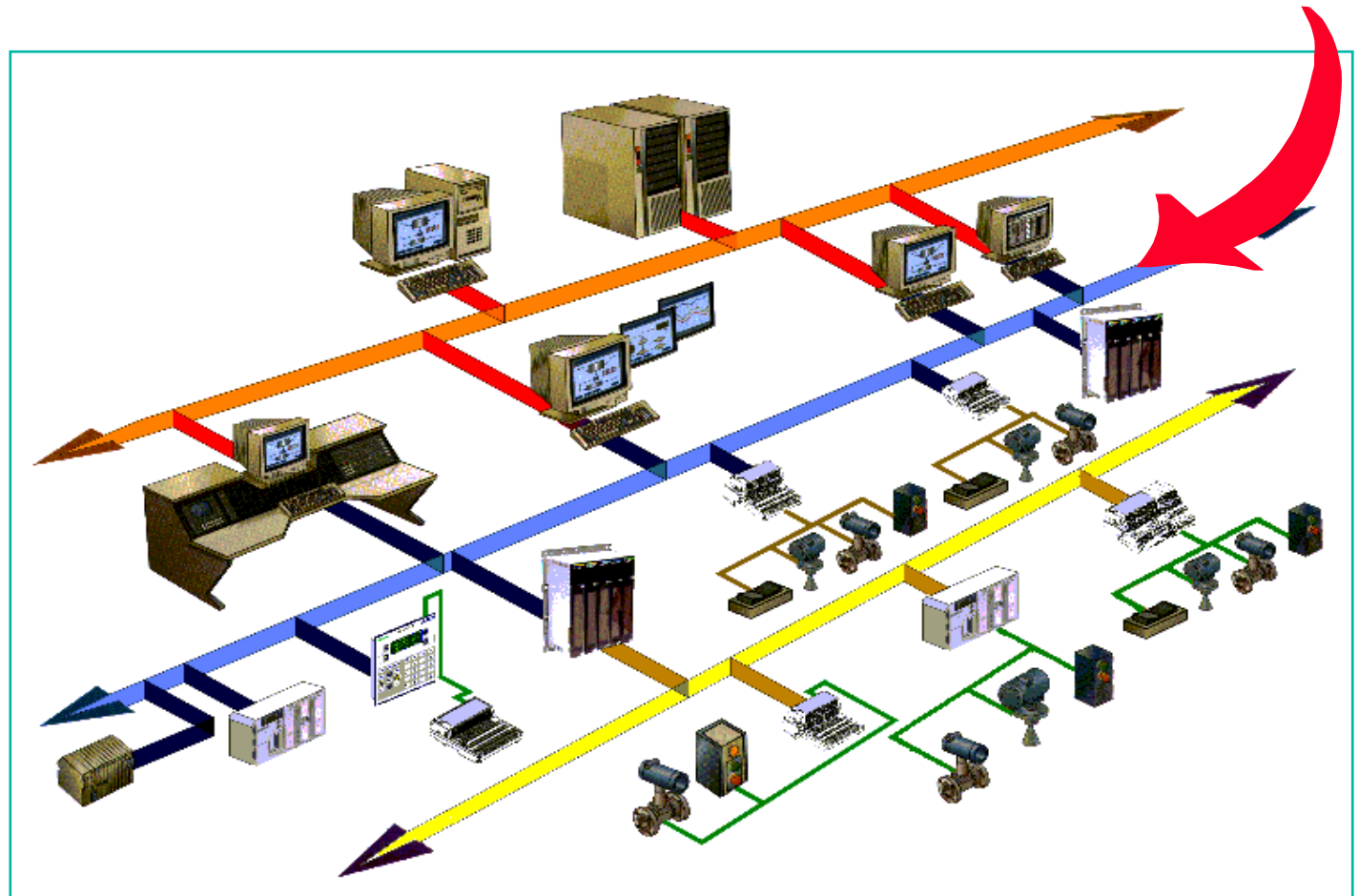
LONWorks Communication Network

- Free topology and highly flexible
- Large number of nodes 32000/domain
- Slower cycle times



Control Networks

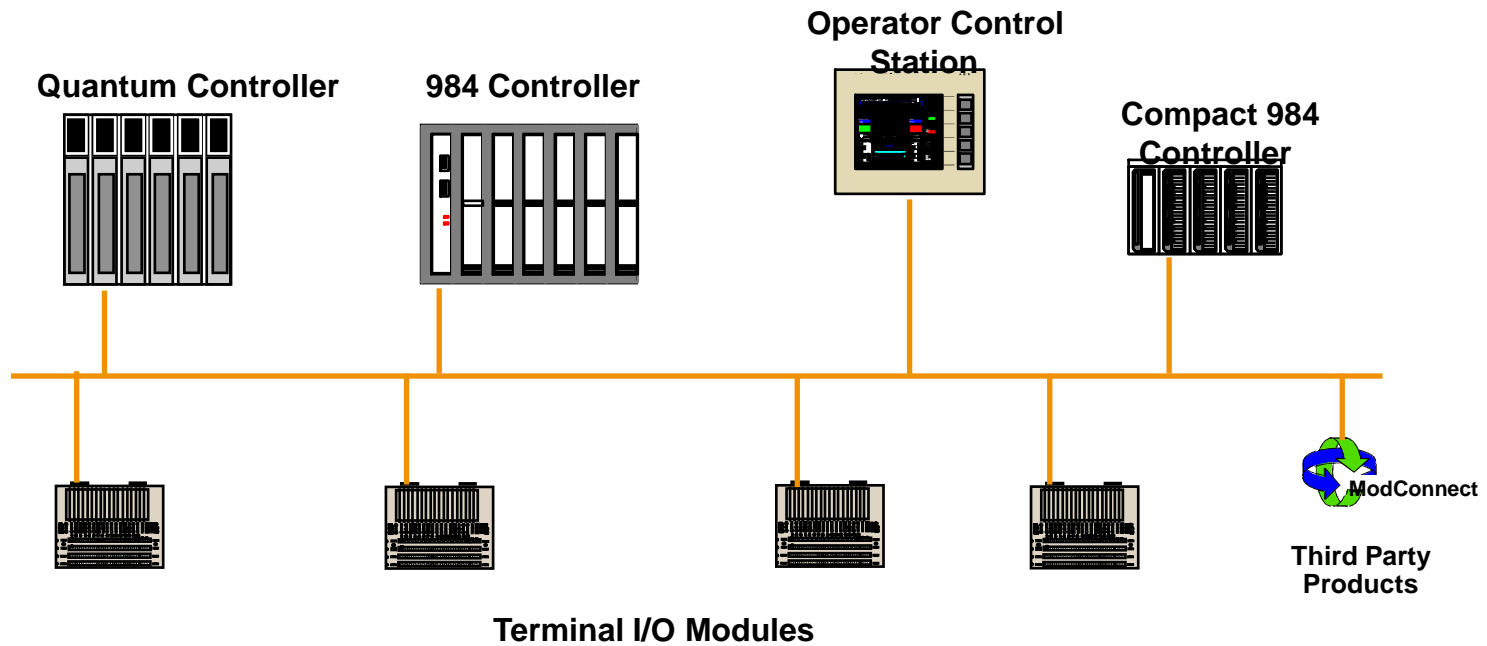
- Control Networks require critical performance



Control Networks

	<u>Modbus Plus</u>	<u>FIPWAY</u>
Speed	20k reg/sec	10-80 ms
Nodes	64	32
Distance	13km	1000m
Message Size	100 registers	128 bytes
Node Cost	Medium	Medium
Installed Cost	Low	High
Physical Size	Medium	Large

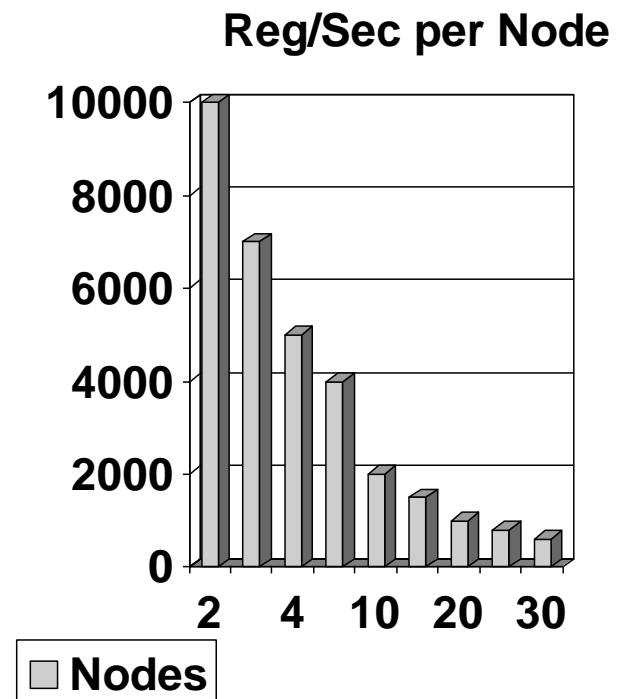
Modbus Plus Network



Modbus Plus excellent performance

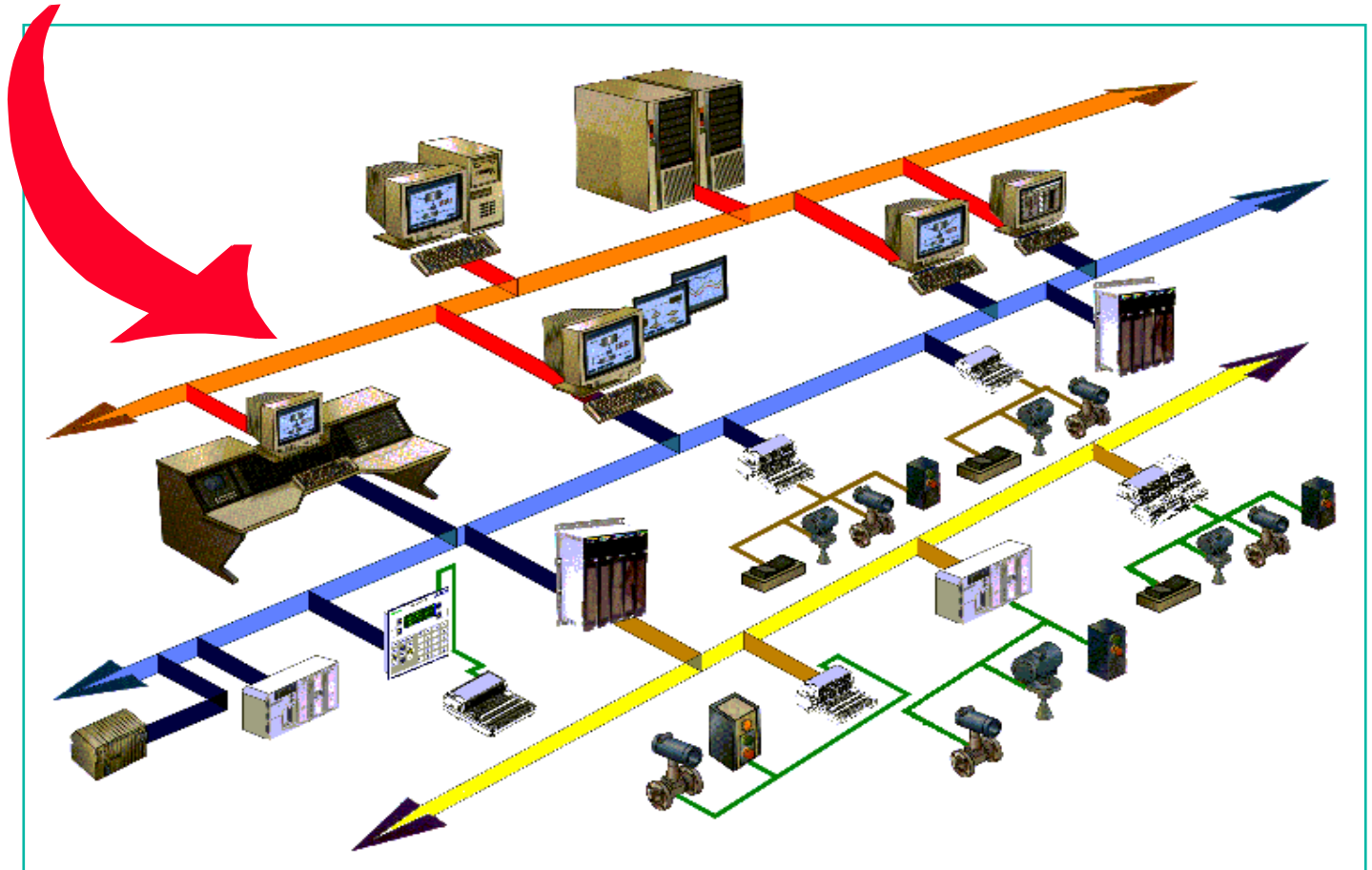
- Predictable performance
- As fast as hardwired I/O
 - Typical token rotation time of 2 milliseconds per node
 - Maximum network throughput of 20,000 registers/Sec

Network Throughput



Data Networks

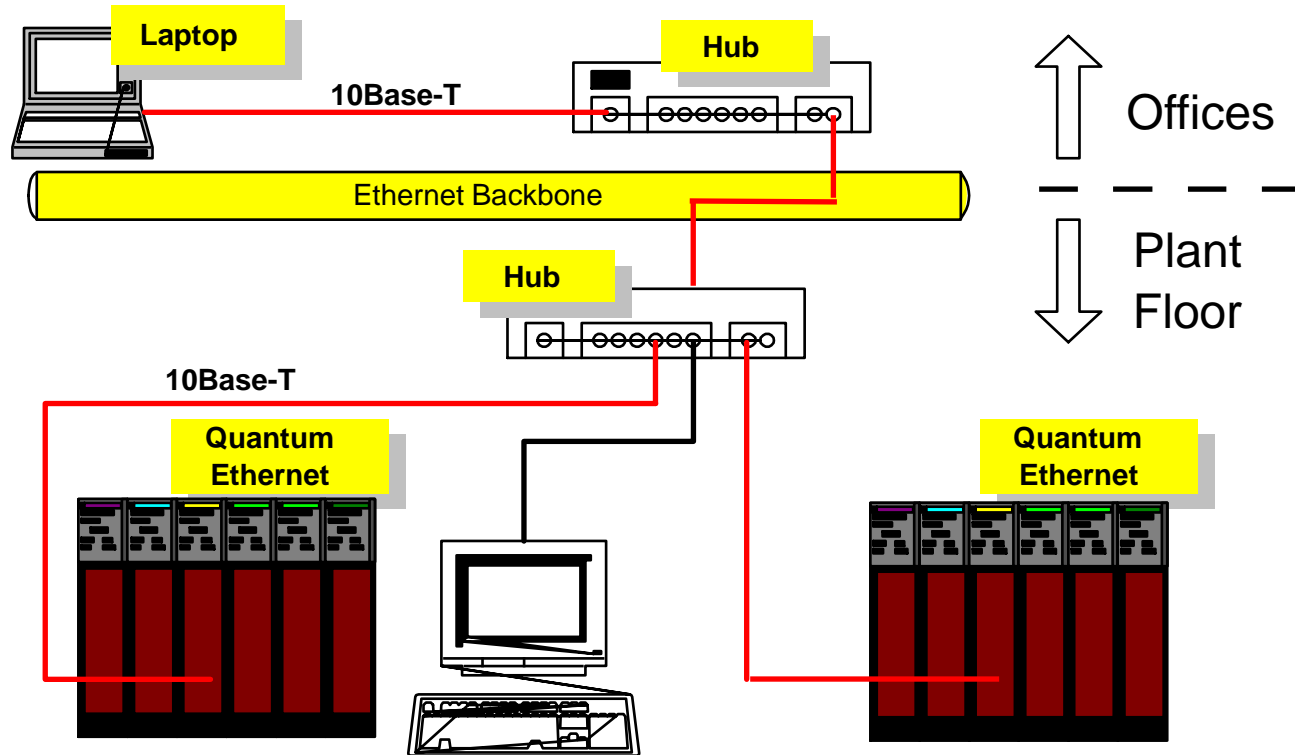
- Data Networks extend the information capabilities



Data Networks

	Ethernet
Nodes	8 - 24 nodes/hub
Max Nodes	Unlimited
Distance	100 m (node to hub)
Message Size	Unlimited
Node Cost	Low
Installed Cost	Depends on Extent of Network
Physical Size	World Wide

Ethernet Data Network



Important Application Information

- Number of nodes (I/O points) required.
- Distance between I/O points.
- Overall network distance.
- Data throughput required.
- Preferred PLC Controllers being utilized
- Bit level traffic quantity
- Word level traffic quantity
- Preferred Media (coax, fiber, twisted pair,.....)
- Maintenance preferences
- Greenfield installation
- Installed base
- Preferred vender list

Supervisory Control & Data
Acquisition

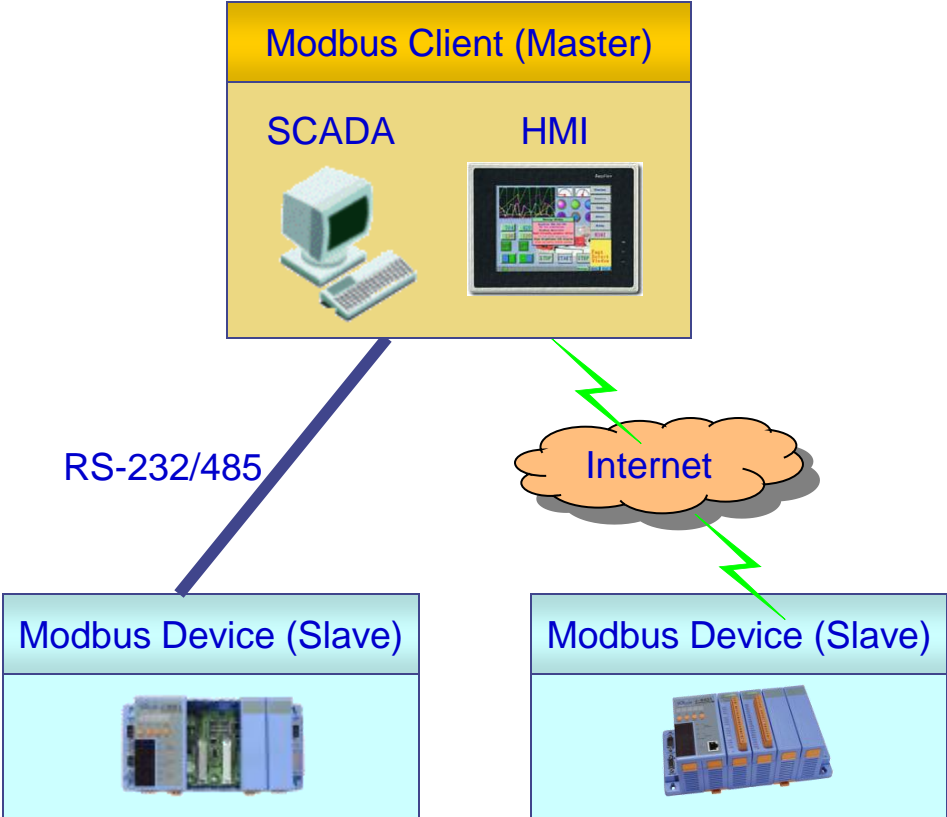
Communication Technology

Modbus Protocol

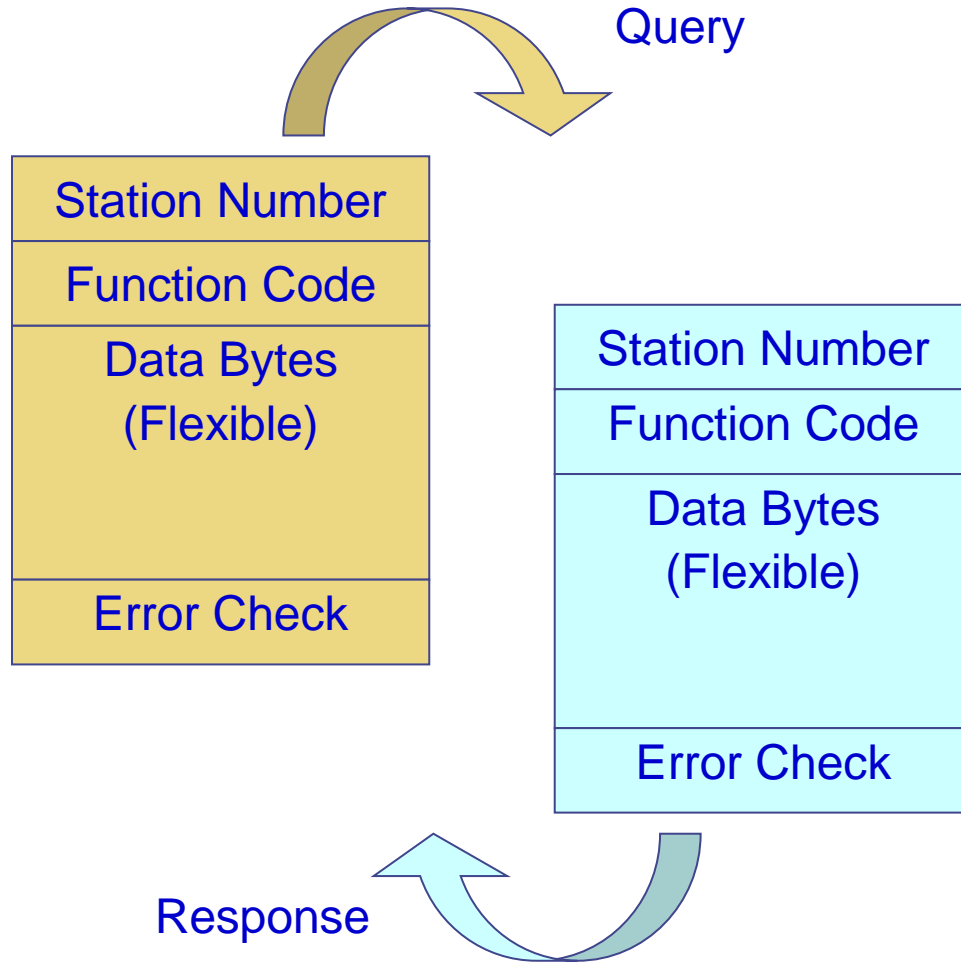
What is Modbus ?

- ◆ An open data communication protocol developed in 1979
- ◆ Published by Modicon
- ◆ Open structure
- ◆ Flexible
- ◆ Widely known
- ◆ Supplied by many SCADA and HMI software
- ◆ Have OPC Servers
- ◆ 2 serial transmission modes:
 - ASCII → 10 bits
 - RTU (Binary) → 11 bits
- ◆ Communication interface
 - RS-232/485
 - Ethernet (TCP/IP)

Application structure (general)



Query-Response Cycle



Hardware classification

- ◆ Station Device: 0 ~ 255
- ◆ Digital input
 - 1xxxx: 4 digits for hexadecimal address (0000 ~ FFFF)
 - 1xxxxx: 5 digital for decimal address (0 ~ 65535)
- ◆ Digital output
 - 0xxxx: 4 digits for hexadecimal address (0000 ~ FFFF)
 - 0xxxxx: 5 digital for decimal address (0 ~ 65535)
- ◆ Analog input
 - 4xxxx: 4 digits for hexadecimal address (0000 ~ FFFF)
 - 4xxxxx: 5 digital for decimal address (0 ~ 65535)
- ◆ Analog output
 - 3xxxx: 4 digits for hexadecimal address (0000 ~ FFFF)
 - 3xxxxx: 5 digital for decimal address (0 ~ 65535)

Two serial transmission modes

◆ ASCII Mode

- Data system
ASCII character, '0'~'9','A'~'F'
- Bits per data unit

1 Start Bit	7 Data Bits	1 Parity Bit (Even/Odd)	1 Stop Bit
1 Start Bit	7 Data Bits	2 Stop Bit	

- Error Check Field
Longitudinal Redundancy Check (LRC)

◆ RTU Mode

- Data system
8-bit Binary, 00~FF
- Bits per data unit

1 Start Bit	8 Data Bits	1 Parity Bit (Even/Odd)	1 Stop Bit
1 Start Bit	8 Data Bits	2 Stop Bit	

- Error Check Field
Cyclical Redundancy Check (CRC)

Modbus Message Packet

◆ ASCII Mode

Start	Station Number	Function Code	Data	Error Check	End
1 Char	2 Chars	2 Chars	n Chars	2 Chars	2 Chars
:				LRC	CR,LF

◆ RTU Mode

Start	Station Number	Function Code	Data	Error Check	End
3.5 Char	8 Bits	8 Bits	n x 8 Bits	2 x 8 Bits	3.5 Chars
Silence				CRC	Silence

◆ Modbus Plus network

Prefixed Data	Station Number	Function Code	Data
6 x 8 Bits			

Byte 0, 1: transaction ID – usually 0

Byte 2, 3: protocol ID = 0

Byte 4, 5: number of bytes following

Read/Write Modes

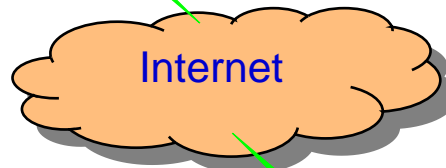
- ◆ Single Read/Write
- ◆ Multiple Read/Write

Basic application #1

Modbus/TCP Clients (Masters)

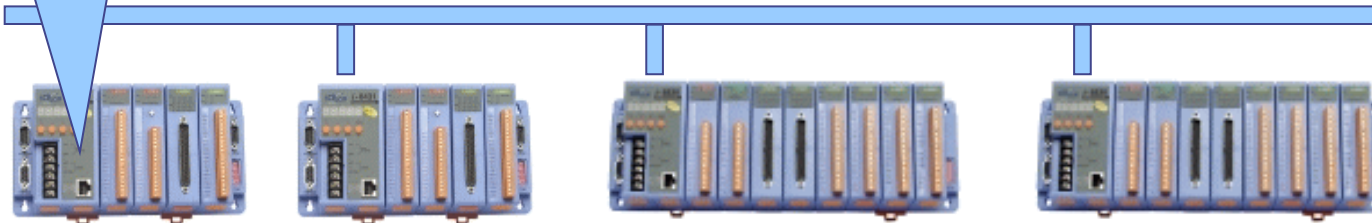


Modbus/TCP command protocol

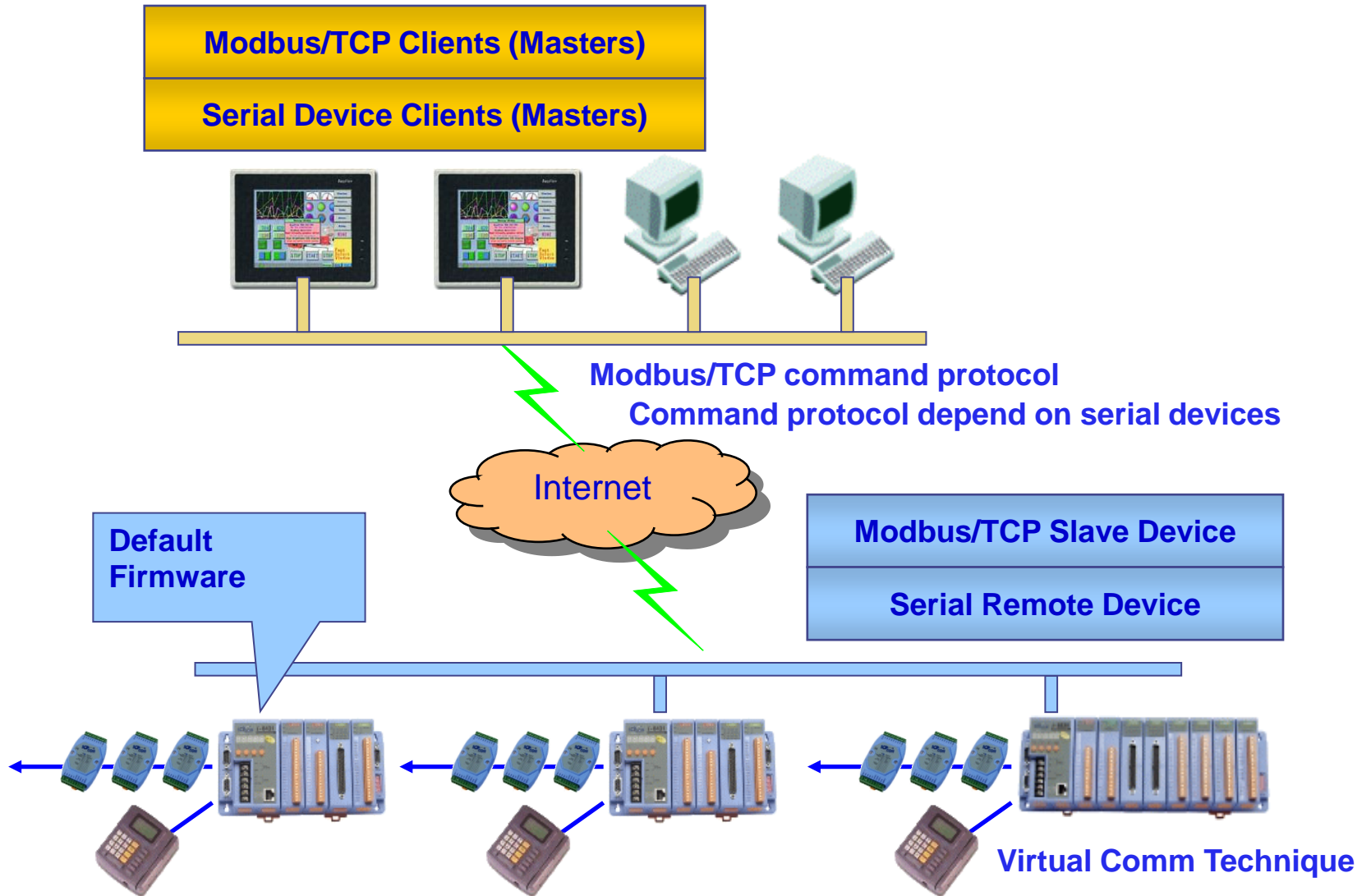


**Default
Firmware**

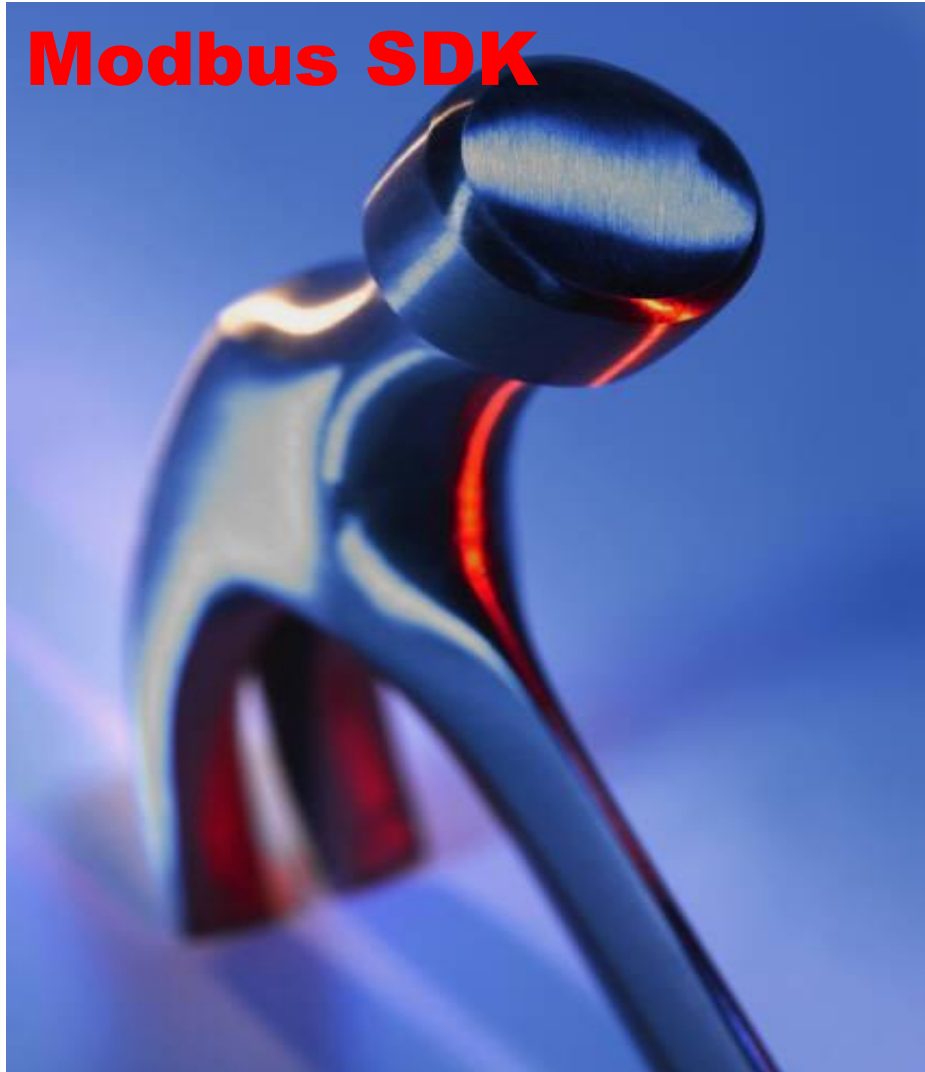
Modbus/TCP Slave Device



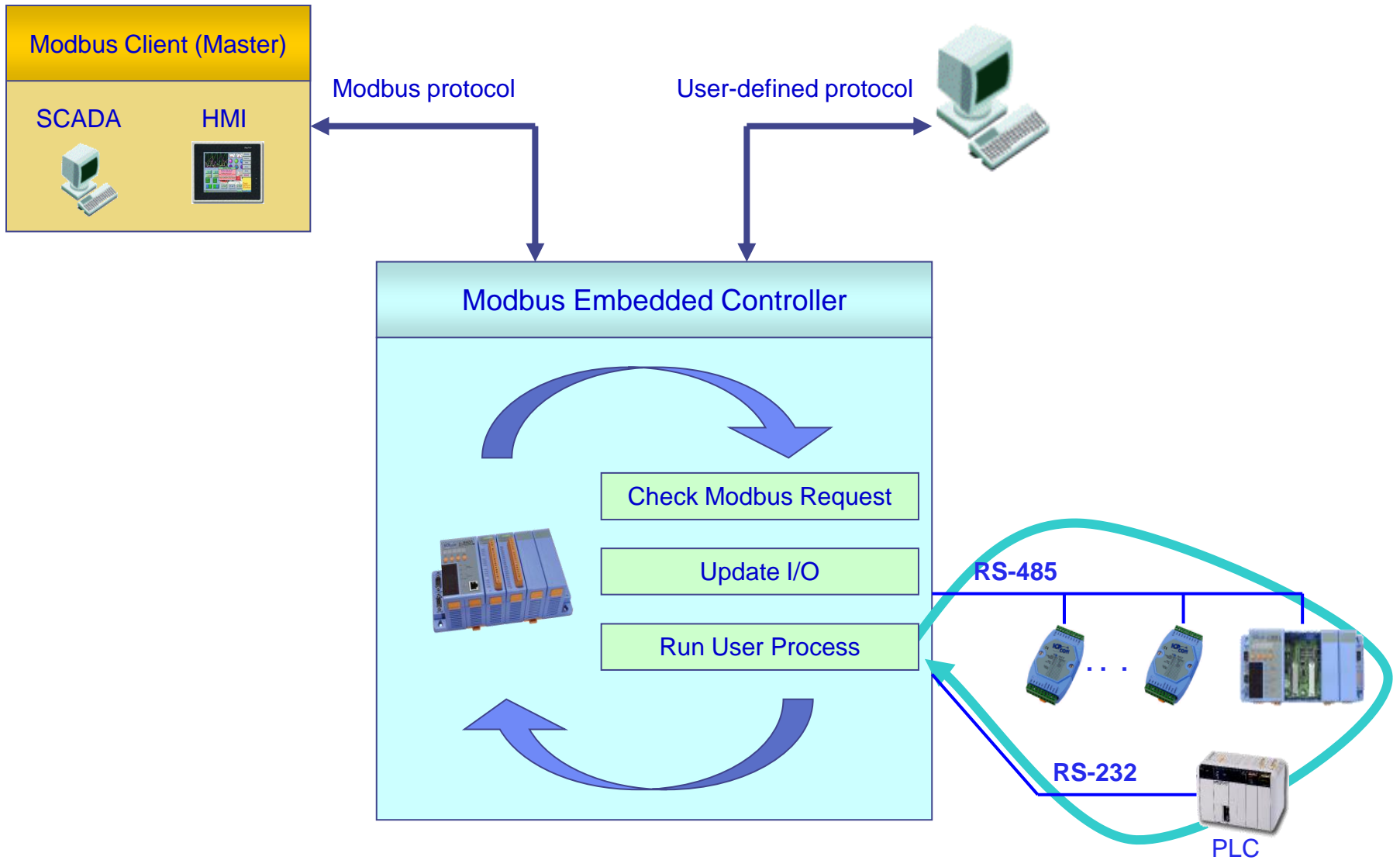
Basic application #2



Modbus SDK



Modbus Embedded Controller

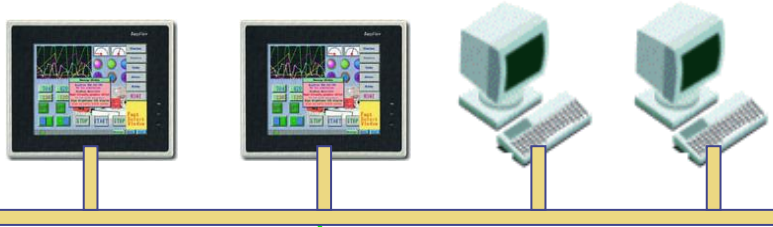


Advanced application

Modbus/TCP Clients (Masters)

General TCP/IP Client (Masters)

Serial Device Clients (Masters)



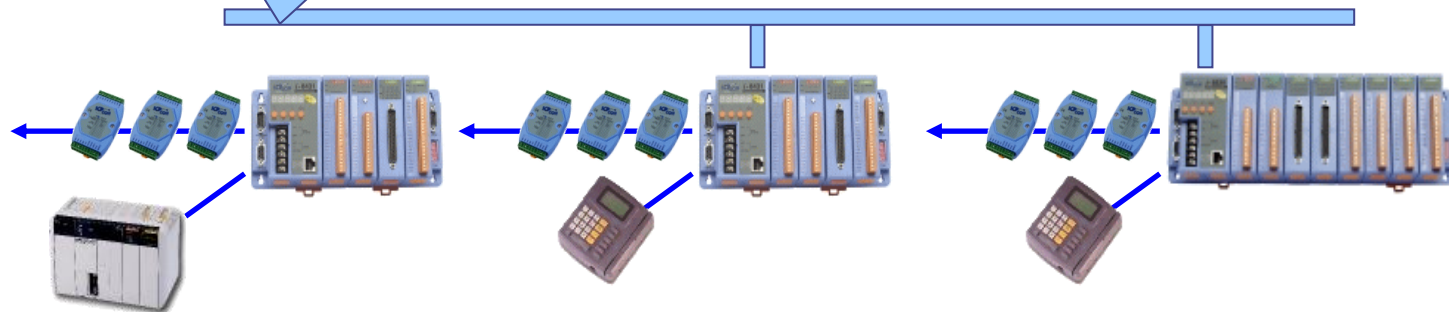
Modbus/TCP Slave

User-defined TCP/IP Slave

Modbus/RTU Master

Serial Device Master

Special Firmware
modified by users



Modbus Resources

<http://www.modicon.com>

Web page of original Modbus protocol inventor, MODICON Inc.

<http://www.modbg>

A community for Modbus users

<http://www.winsite.com/bin/Info?3088>

ActiveX freeware to communicate with equipments via
Modbus(RTU/ASCII) or Modbus/TCP

<http://www.pmcrae.freemove.co.uk/readregu.html>

Example C code for Linux Modbus/RTU communication

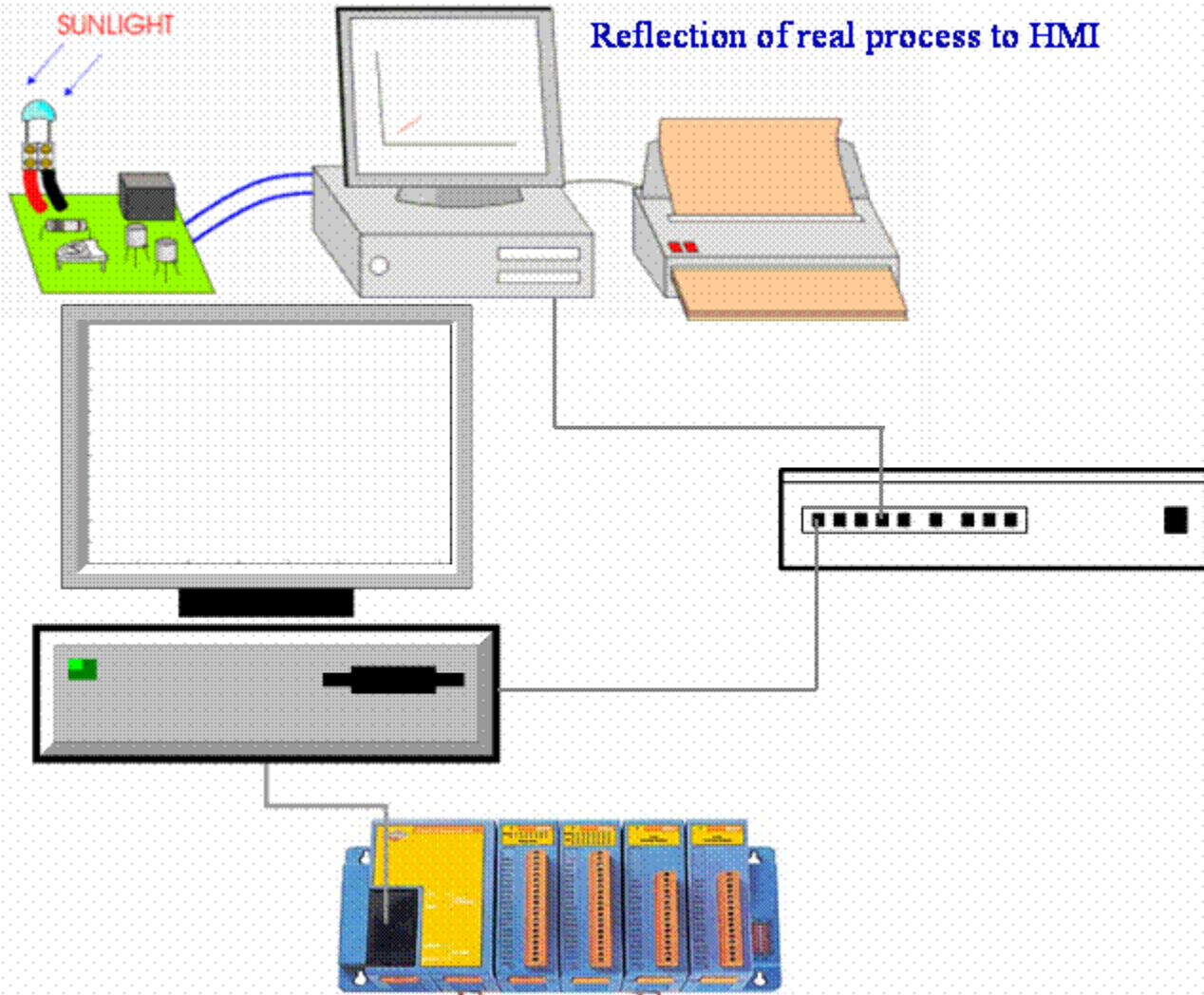


Supervisory Control & Data Acquisition

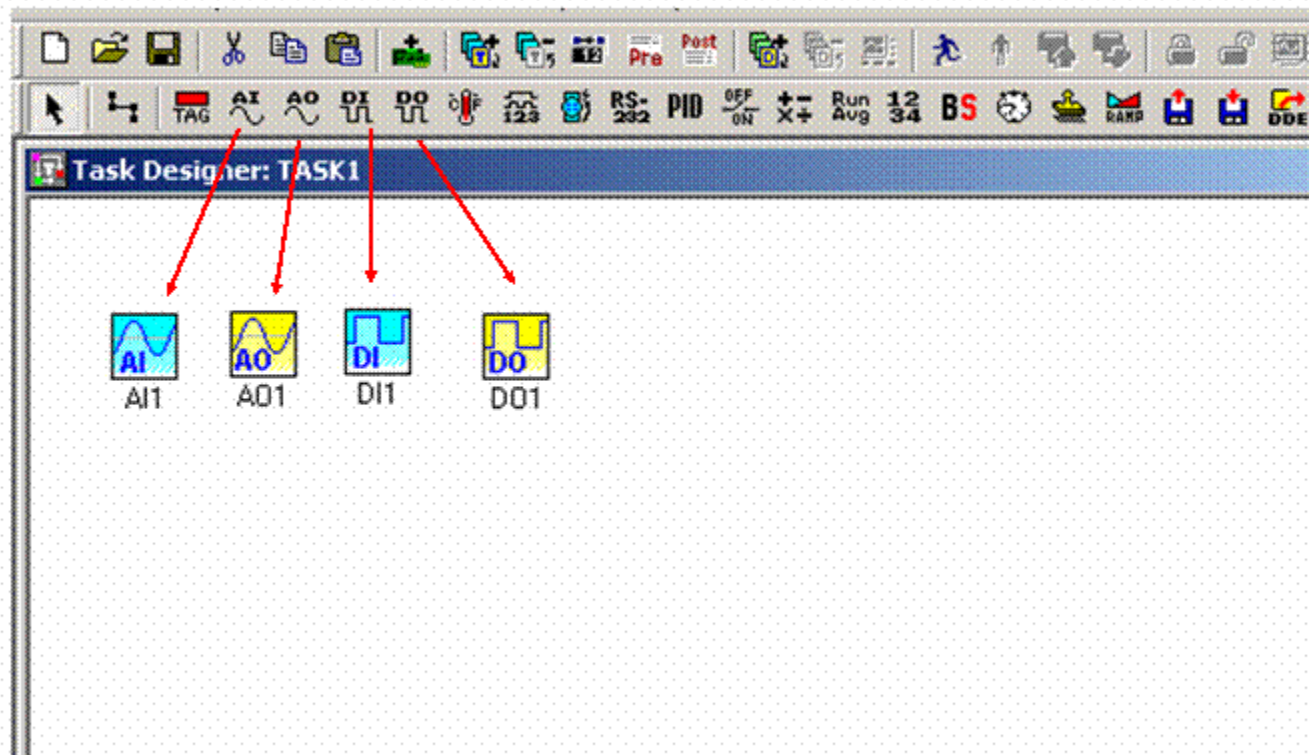
**Software + Algorithmization of
Control Problems**

Tag Concept





Tags in HMI



Abstracting of engineering process parameters in HMI (Intouch, Wonderware; RSWiew, Rockwell Automation)

Tag

Name: Type: Security:

Description:

Minimum: Scale: Units:

Maximum: Offset: Data Type:

Data Source

Type: Device Memory

Node Name: ... Scan Class:

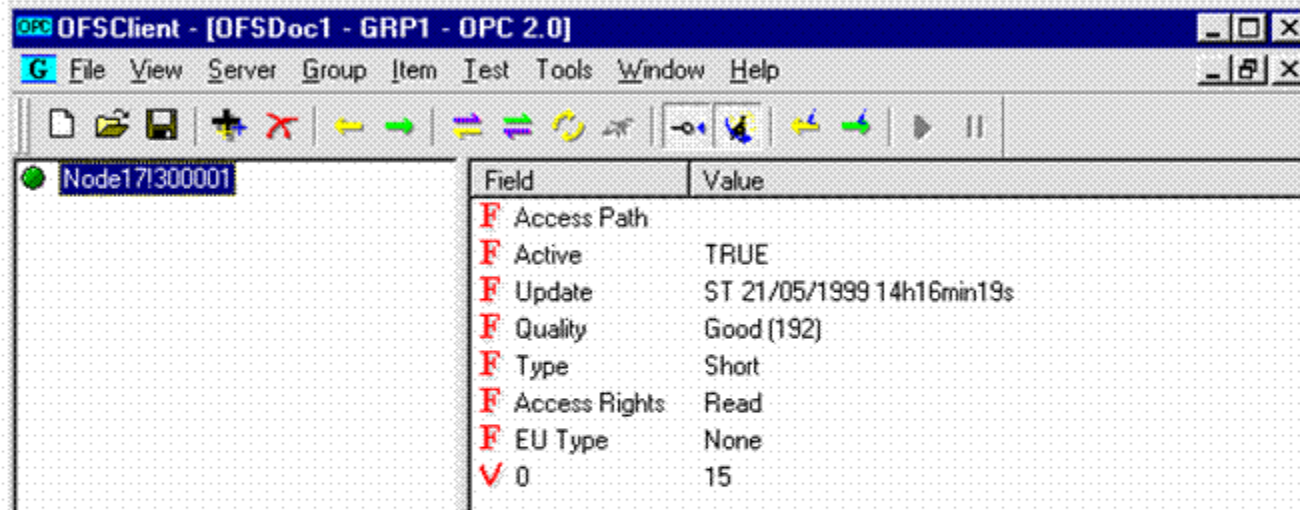
Address:

Abstracting of engineering process parameters in HMI (GeniDAQ, Advantech)

The screenshot displays the 'Analog Input Properties' dialog box. At the top, the 'Tag' is set to 'AI1' and the 'Description' is 'Pressure'. The 'Device/Item' is 'Advantech DEMO I/O=1H'. There is a checkbox for 'Connect to OPC server' which is currently unchecked, with a 'Select' button next to it. The 'I/O settings' section includes 'From channel' and 'To channel' both set to '0'. Below this is an 'Input range' section with 'Channel' and 'Range' dropdown menus. The 'Expansion channel' section has 'Exp. channel' and 'Board ID' dropdown menus. On the right side of the dialog, there are buttons for 'OK', 'Cancel', 'Help', and 'Scaling'. At the bottom right, there is an 'Update times' spinner box set to '1'. At the bottom left, there is a 'DDE' section with a checkbox for 'Establish DDE link (Channel 0 only)' which is unchecked.

Property	Value
Tag	AI1
Description	Pressure
Device/Item	Advantech DEMO I/O=1H
Connect to OPC server	<input type="checkbox"/>
From channel	0
To channel	0
Input range Channel	[Dropdown]
Input range Range	[Dropdown]
Expansion channel Exp. channel	[Dropdown]
Expansion channel Board ID	[Dropdown]
Update times	1
DDE Establish DDE link (Channel 0 only)	<input type="checkbox"/>

Reflection of engineering process parameters in HMI



The screenshot shows the OFSClient application window. The title bar reads "OFSClient - [OFSDoc1 - GRP1 - OPC 2.0]". The menu bar includes "File", "View", "Server", "Group", "Item", "Test", "Tools", "Window", and "Help". The toolbar contains various icons for file operations and navigation. The main area displays a tree view with "Node171300001" selected. To the right, a table lists the following parameters:

Field	Value
F Access Path	
F Active	TRUE
F Update	ST 21/05/1999 14h16min19s
F Quality	Good (192)
F Type	Short
F Access Rights	Read
F EU Type	None
✓ 0	15

Tag

- New concept in the industry automation.
- Abstracting of engineering process parameters of information sources.
- Real-time data from sensors: temperatures, pressures, flow, ... (primary parameters)
- State of actuators – open/close, start/stop, state of regulators (PID): automatic/manual/off.
- Value of counter, random number generator.
- Messages from measuring instruments.
- Secondary parameters.

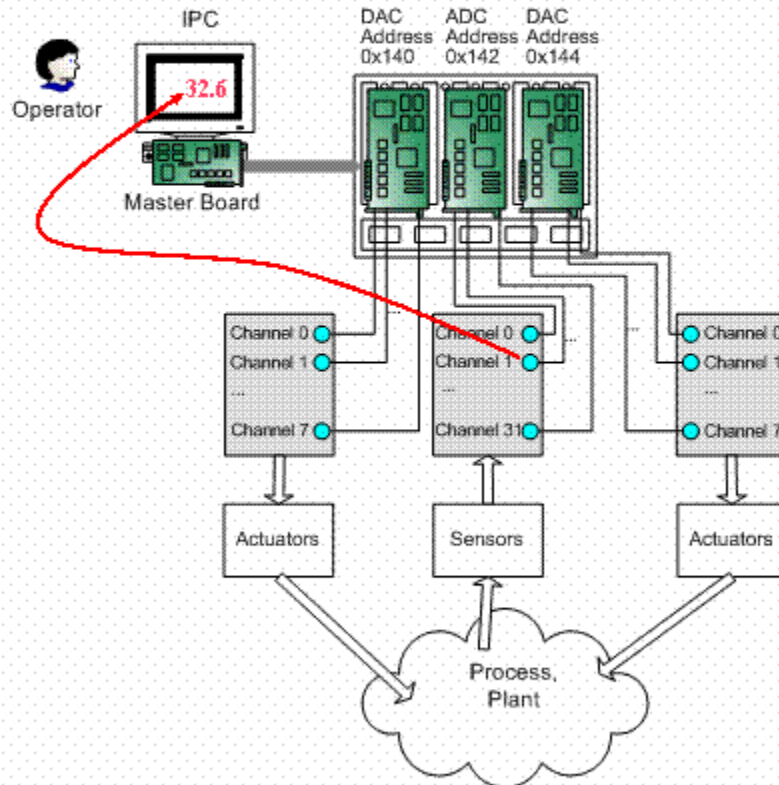
Tag

```
Tag =  
{  
    Characteristic 1,  
    Characteristic 2,  
    ...,  
    Characteristic N  
}
```

```
Tag =  
{  
    ID  
    Value,  
    EU,  
    Description,  
    Address,  
    Channel  
}
```

```
Tag =  
{  
    All  
    4.8,  
    "Pa"  
    "Pressure",  
    0x140,  
    5  
}
```

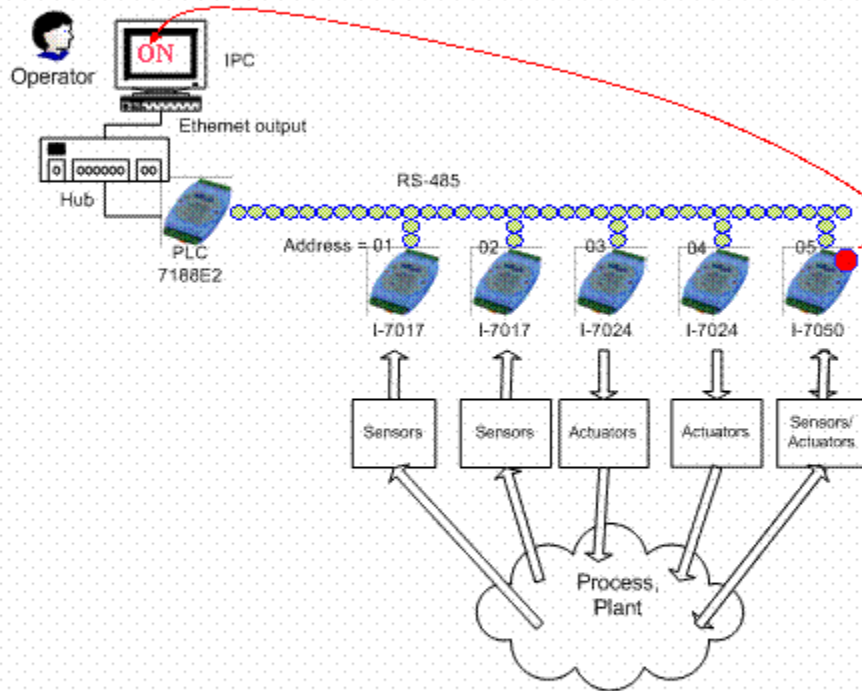
Tag abstracting by C programming language



```
struct Analog  
{  
    char          Name[N];  
    float        Value;  
    unsigned char Address;  
    unsigned char Channel;  
};
```

```
Analog  
{  
    "Temperature";  
    32.6;  
    0x142;  
    1;  
};
```

Tag abstracting by C programming language



```
struct Analog
{
    char          Name[N];
    bool         Value;
    unsigned char Address;
    unsigned char Channel;
};
```

```
Analog
{
    "Pump State";
    ON;
    5;
    12;
};
```

Signal form

- **Analog (temperature, pressure, ...)**
- **Digital (state of devices, input signal in the digital form)**
- **String (from measuring instruments, messages, warnings)**
- **Computing (average, power, profits, ...)**

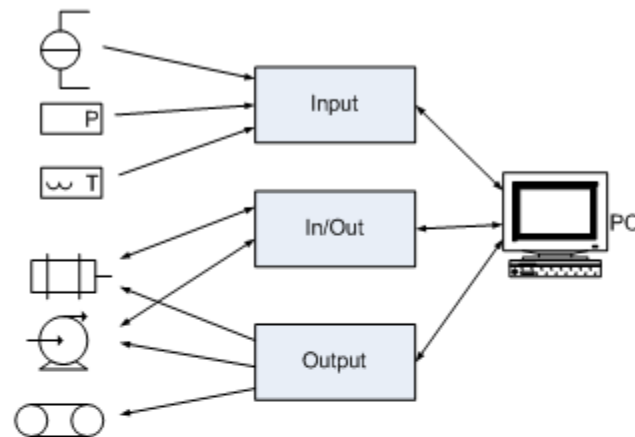
```
AnalogTag = { 101  
              55,  
              "C"  
              "Temp",  
              0x140,  
              6      };
```

```
DigitalTag = { 102  
              ON,  
              "Relay",  
              0x142,  
              2      };
```

```
StringTag = { 103  
              "Hello",  
              "SMS",  
              "COM1",  
              12      };
```

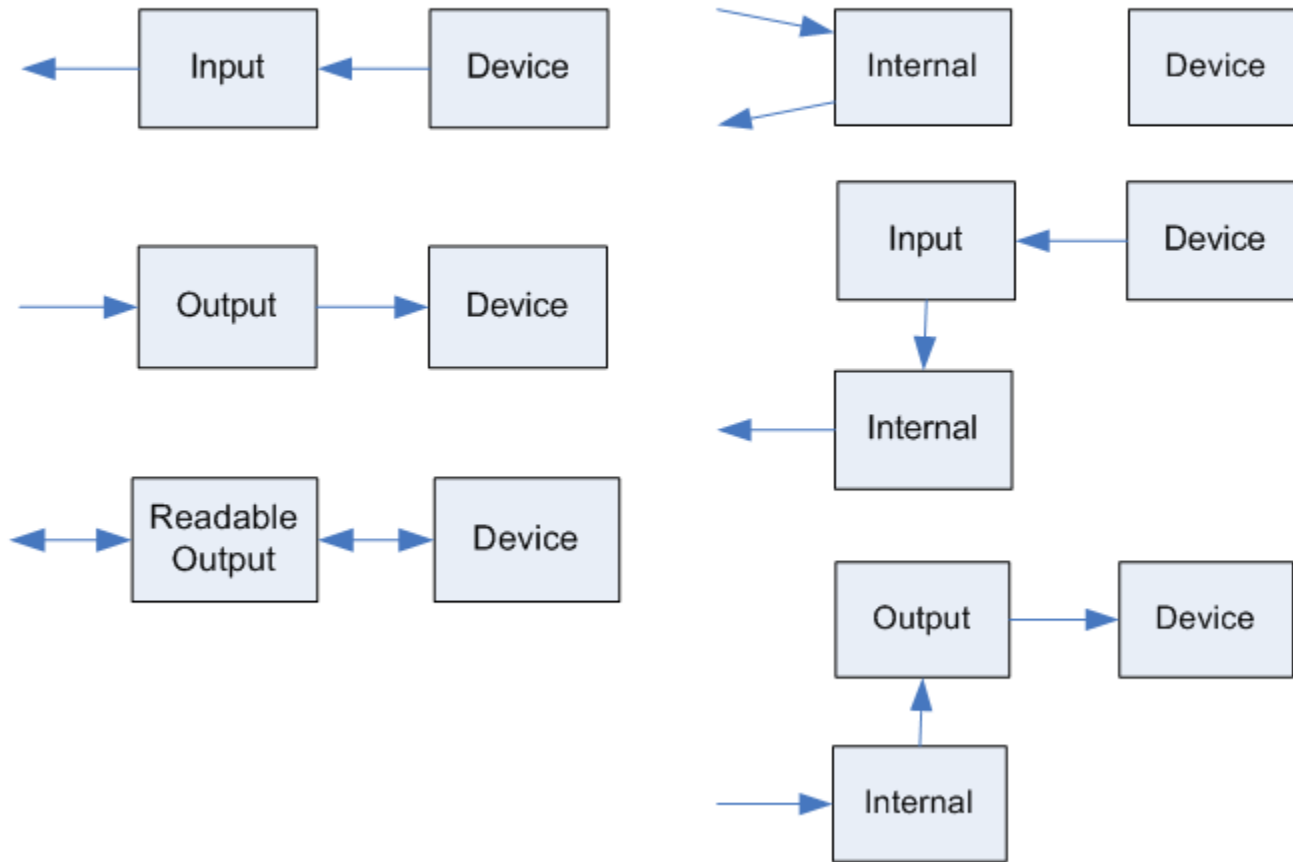
```
ComputingTag = { 104  
                 55.8,  
                 "T Average",  
                 101      };
```


Signal direction

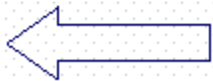
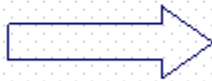
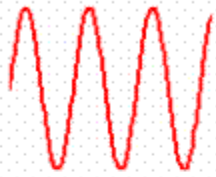


- **Input** – signal is resulted from physical devices.
- **Output** – signal is intended for write to physical devices.
- **Readable output** – Input/Output.
- **Internal** – resulted from other tags by specified algorithm, internal channels, buffers.

Signal direction

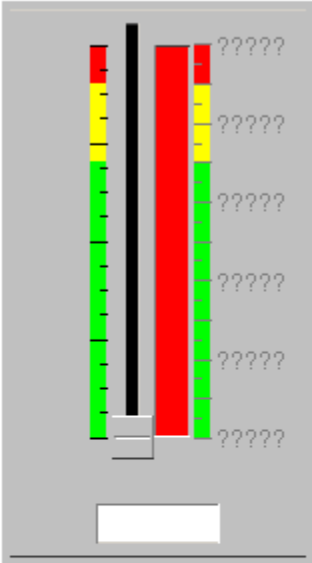
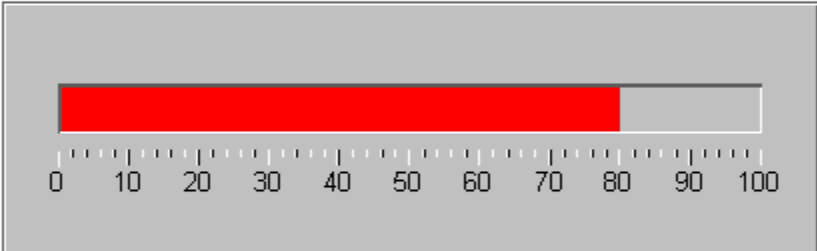
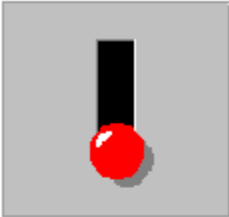


Popular tags



Analog Input;
Analog Output;
Digital Input;
Digital Output;

Tag abstracting by C programming language



Analog Input Tag

```
struct AnalogInput
{
    char           Name[N];
    float          Value;
    unsigned int   Address;
    unsigned char Channel;
    float          LowScale;
    float          HighScale;
    float          Lo;
    float          Hi;
    float          LoLo;
    float          HiHi;
    float          Deadband;
    float          MaxRateOfChange;
    float          SmoothFactor;
    int            AlarmAndEvent;
    float          Trend[M];
    int            TopIndex;
};
```

Tag basic characteristics

```
struct AnalogInput
{
    char           Name[N];
    float         Value;
    unsigned int  Address;
    unsigned char Channel;
    ...
}
```

- **Tag ID**
String: "Tag1", "Tag2", "Temperature", "Motor Speed";
Integer: 100, 101, ...
- **Physical address of tag' sources**
0x140, 0x144, ...
"COM1", "COM2", ...
1, 2, 3, ...
41000, 41001, ...
- **Current value and attendant characteristics.**

Tag basic characteristics

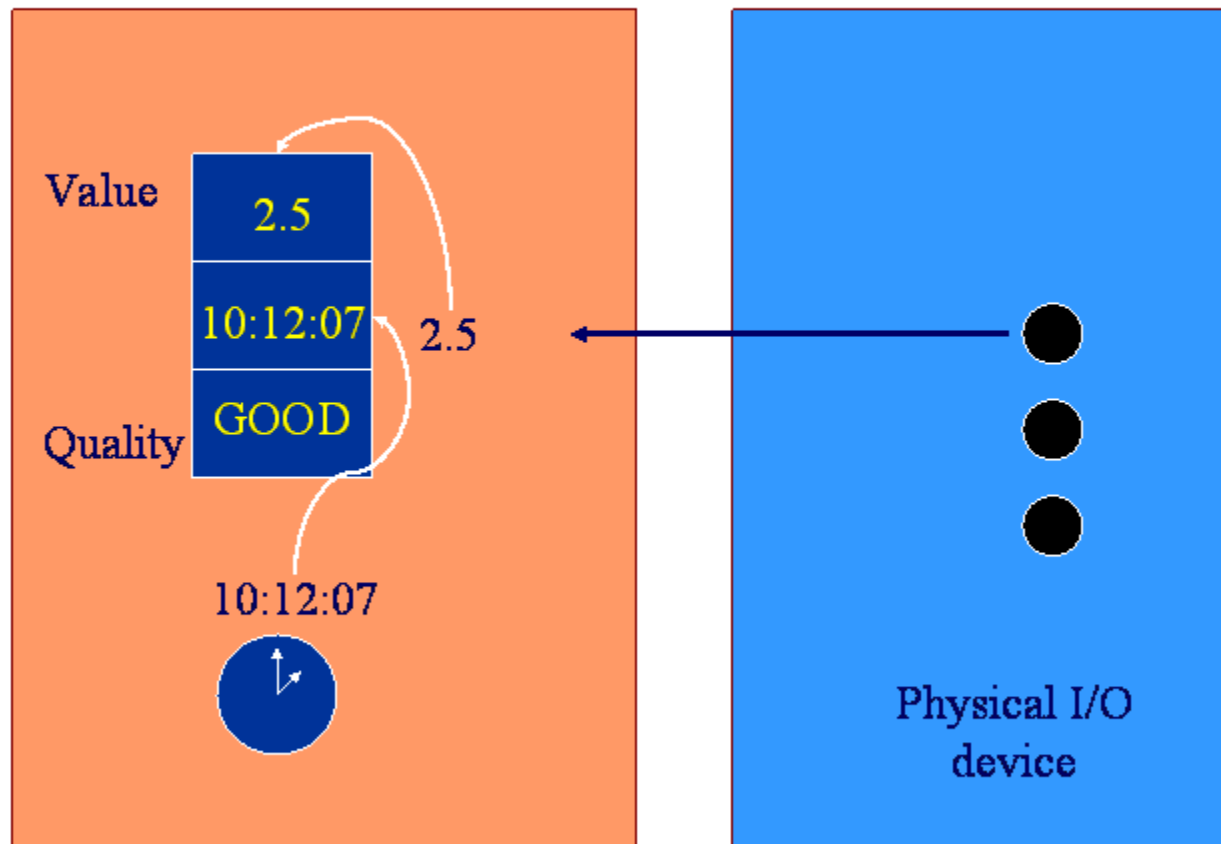
Current value and attendant characteristics:

Data value	VARIANT type
Time stamp	FILETIME
Quality	UNCERTAIN, GOOD, BAD

32.234, GOOD, 20 December, 2004 15:21:22:74

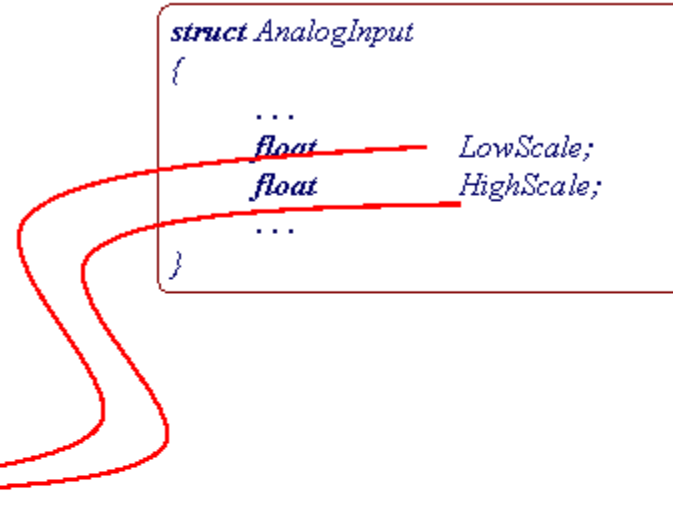
```
struct AnalogInput
{
    ...
    VARIANT           Value;
    WORD              Quality;
    FILETIME          TimeStamp;
    ...
};
```

Tag basic characteristics



Scaling range

```
struct AnalogInput  
{  
    ...  
    float LowScale;  
    float HighScale;  
    ...  
}
```



Input diapason: [n1, m1]

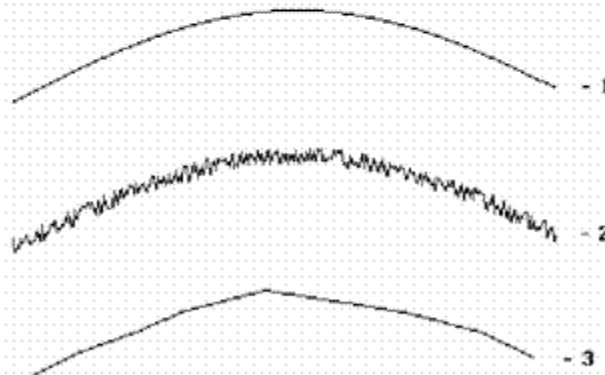
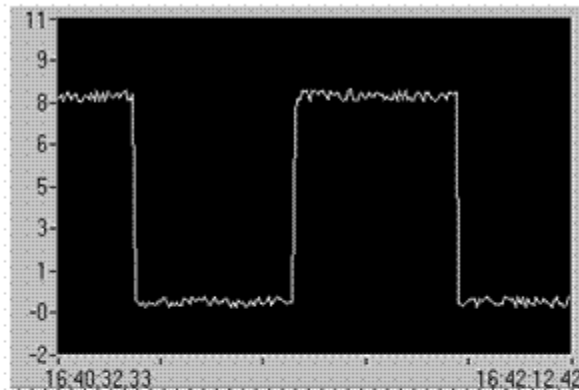
Output diapason: [n2, m2]

Linear scaling:
$$\text{Value} = n2 + (m2 - n2) \frac{(\text{Input} - n1)}{m1 - n1}$$

$$\text{Value} = -10 + (10 - (-10)) \frac{(\text{Code} - 0)}{4095 - 0}$$

Quadratic scaling:
$$\text{Value} = n2 + (m2 - n2) \sqrt{\frac{(\text{Input} - n1)}{m1 - n1}}$$

Deadband



```
struct AnalogInput  
{  
    ...  
    float      Deadband;  
    ...  
};
```

$$|\text{CurrentValue} - \text{PrevChange}| > (\text{HighScale} - \text{LowScale}) \frac{\text{Deadband}}{100.0}$$

Deadband

Deadband = 0; 500ms

Timestamp 15:5:0:177	Value = 8.4834
Timestamp 15:5:0:678	Value = 8.4901
Timestamp 15:5:1:179	Value = 8.4305
Timestamp 15:5:1:679	Value = 8.4454
Timestamp 15:5:2:180	Value = 8.4492
Timestamp 15:5:2:681	Value = 8.4240


Deadband = 0.1V; 500ms

Timestamp 15:8:17:140	Value = 7.6986
Timestamp 15:8:48:686	Value = 0.8023
Timestamp 15:9:20:231	Value = 7.6478
Timestamp 15:9:51:776	Value = 0.7230
Timestamp 15:10:23:322	Value = 7.6375
Timestamp 15:10:54:867	Value = 0.7939

T = 31.5

Smooth factor

```
struct AnalogInput  
{  
    ...  
    float SmoothFactor;  
    ...  
};
```



$$s[i] = k * s[i-1] + (1 - k) * s[i]$$

$$0 \leq k \leq 1$$

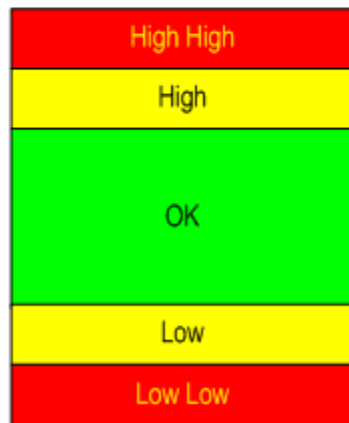
$$k = 0$$

$$k = 1$$

Alarms & Events

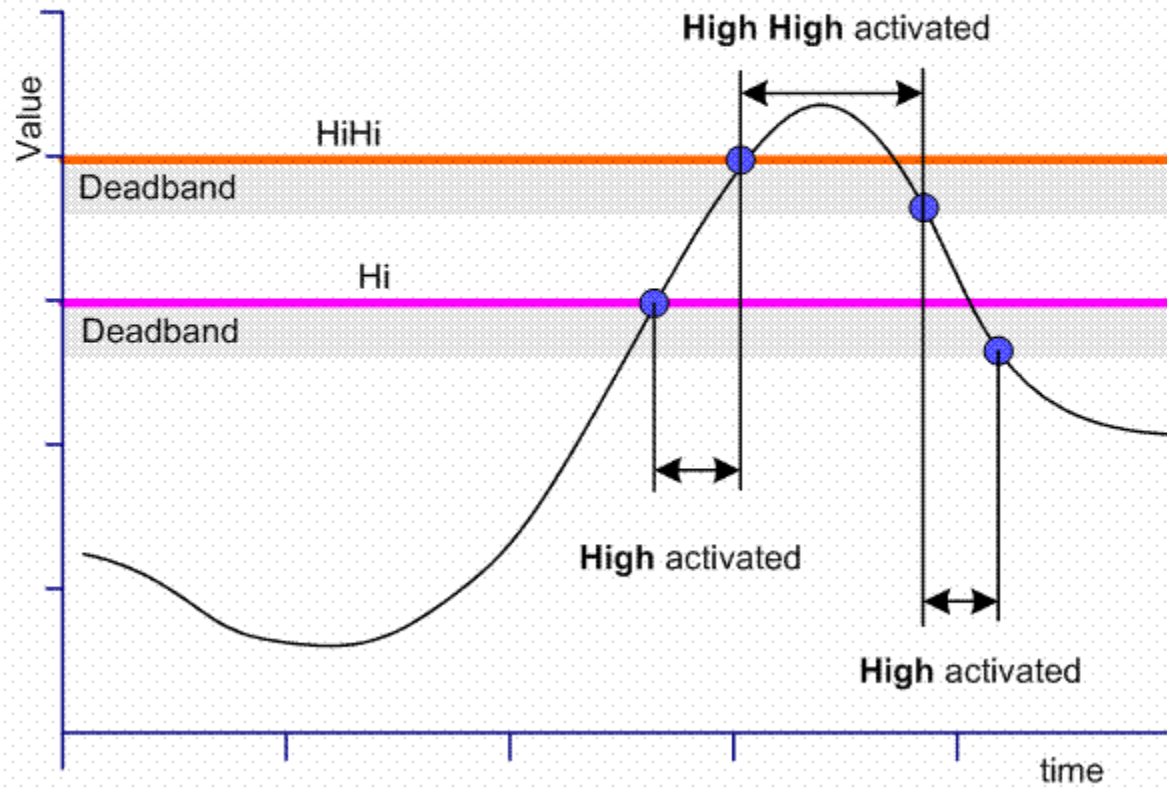
Analog alarm (HI, LO, HI HI, LO LO)

```
struct AnalogInput  
{  
    ...  
    float Lo;  
    float Hi;  
    float LoLo;  
    float HiHi;  
    ...  
};
```



Alarms & Events

Analog alarm (HI, LO, HI HI, LO LO)



Alarms & Events

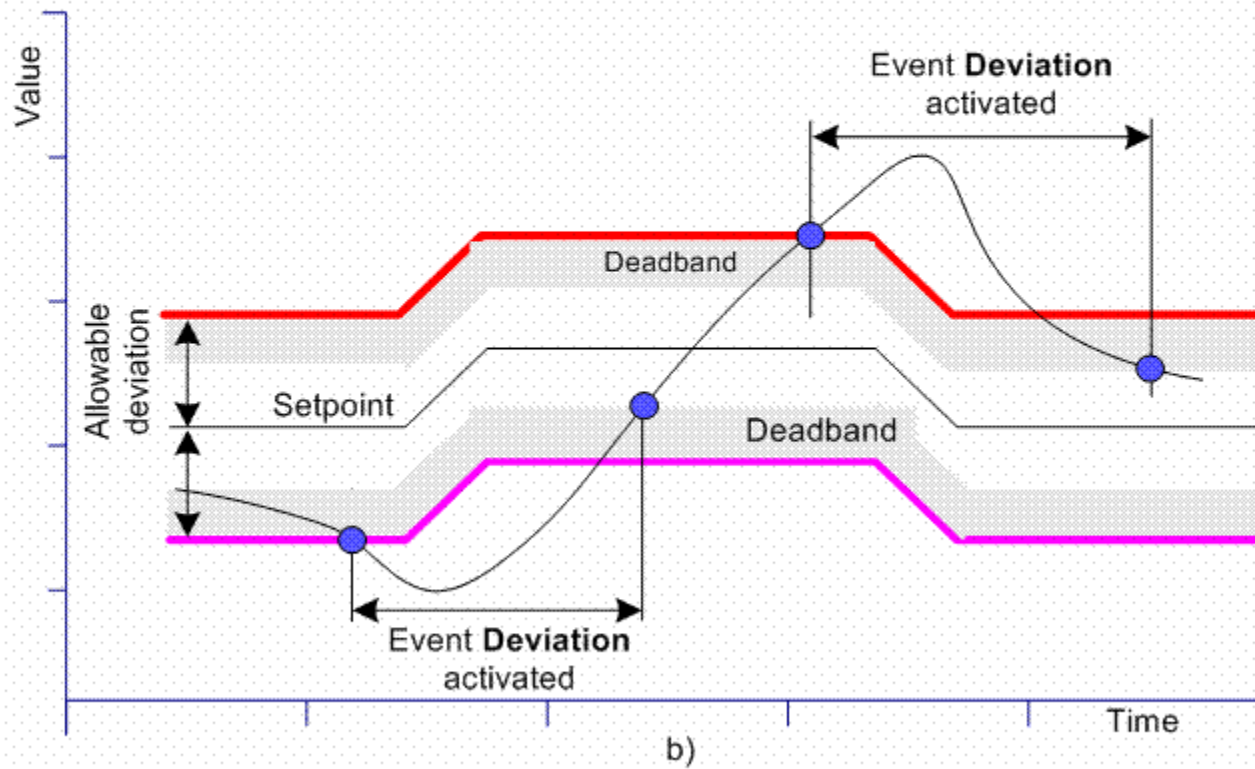
Analog alarm (HI, LO, HI HI, LO LO)

LO_LO = 0V, LO = 2V, HI = 7V, HI_HI = 9V, deadband = 0.5V

Signal	Alarm	Signal	Alarm
9,1453	HI_HI	2,7378	OK
8,9519	HI_HI	2,3881	OK
8,7391	HI_HI	2,0454	OK
8,5113	HI_HI	1,7078	LO
8,2665	HI	1,3796	LO
8,0047	HI	1,0610	LO
6,8307	HI	0,7523	LO
6,5072	HI	0,1730	LO
6,1760	OK	-0,0948	LO_LO
5,4914	OK	-1,0198	LO_LO
4,7844	OK	-0,6591	LO_LO
4,0670	OK	-0,1743	LO_LO
3,3488	OK	0,0903	LO_LO
2,6399	OK	0,3695	LO_LO
2,2926	OK	0,6611	LO
1,9501	LO	0,9672	LO
0,9729	LO	1,6075	LO
0,3746	LO	1,9432	LO
0,0946	LO	2,2844	LO
-0,1686	LO_LO	2,6328	OK
-0,4191	LO_LO	2,9842	OK

Alarms & Events

Analog alarm (Deviation)



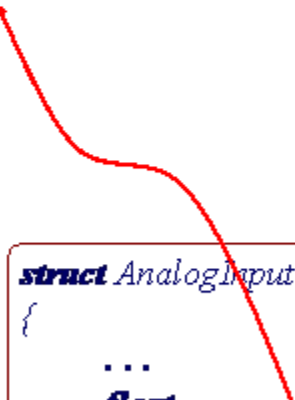
Alarms & Events

Analog alarm (Rate of Change - ROC)

$$Rate = \frac{s[i] - s[i - 1]}{t_i - t_{i-1}}$$

$$Rate \geq MaxRateOfChange \Rightarrow ROC$$

```
struct AnalogInput  
{  
    ...  
    float MaxRateOfChange;  
    ...  
};
```



Trend

```
struct AnalogInput
```

```
{
```

```
...
```

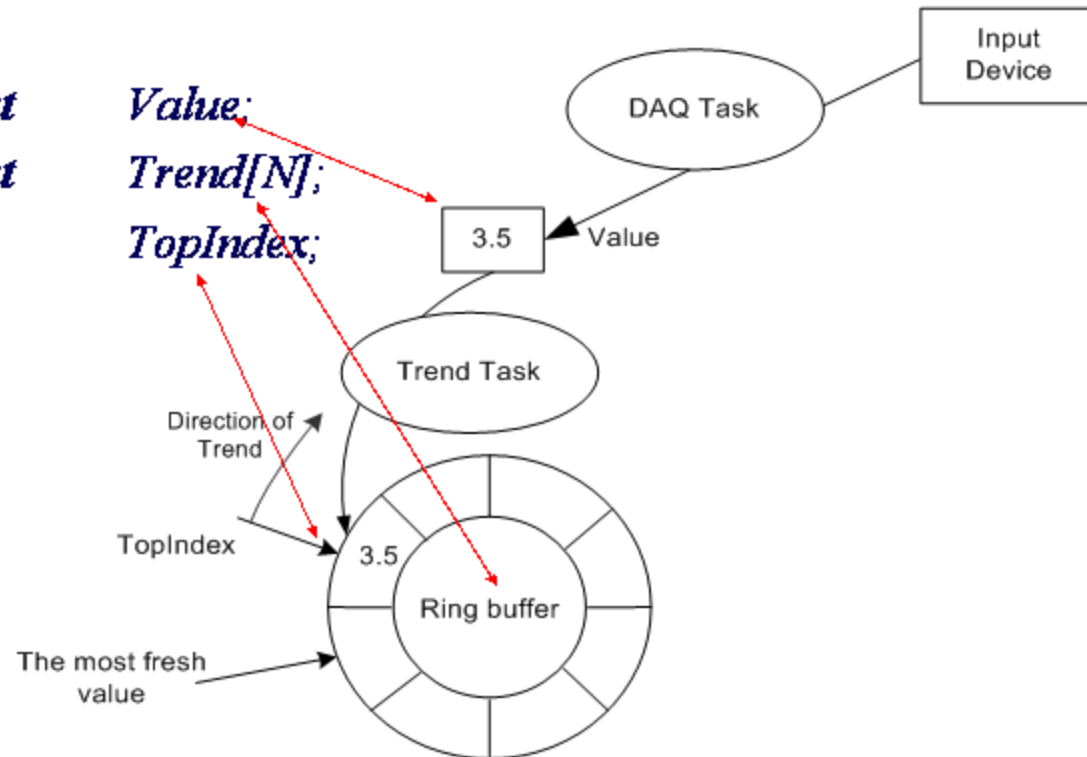
```
float Value;
```

```
float Trend[N];
```

```
int TopIndex;
```

```
...
```

```
};
```



Device & Connection Concept

New Concepts in Modern Automated Control System

Tag

Device = { Tag₁, Tag₂, ..., Tag_N }

Connection = { Device₁, Device₂, ..., Device_N }

Setup Connection ✕

Main | Details

Communications Port: COM1 ▾

Baud Rate: 9600 ▾

Custom Baud Rate: 9600

Data Bits: 8 ▾

Parity: Even ▾

Stop Bits: 1 ▾

OK Отмена

Advantech PCL-818L/H/HD/HG Data Acquisition Card Setup

Board Type

PCL-818L
 PCL-818H
 PCL-818HD
 PCL-818HG

Base Address

Hex

Interrupt Channel

▾

A/D Channels Configuration

8 Differential
 16 Single-ended

Range: ▾

Clock Selection

1 MHz
 10 MHz

DMA Selection

Level 1
 Level 3

D/A Voltage Ref - Channel 1

External Internal

Voltage: ▾

OK Cancel Options... Help About

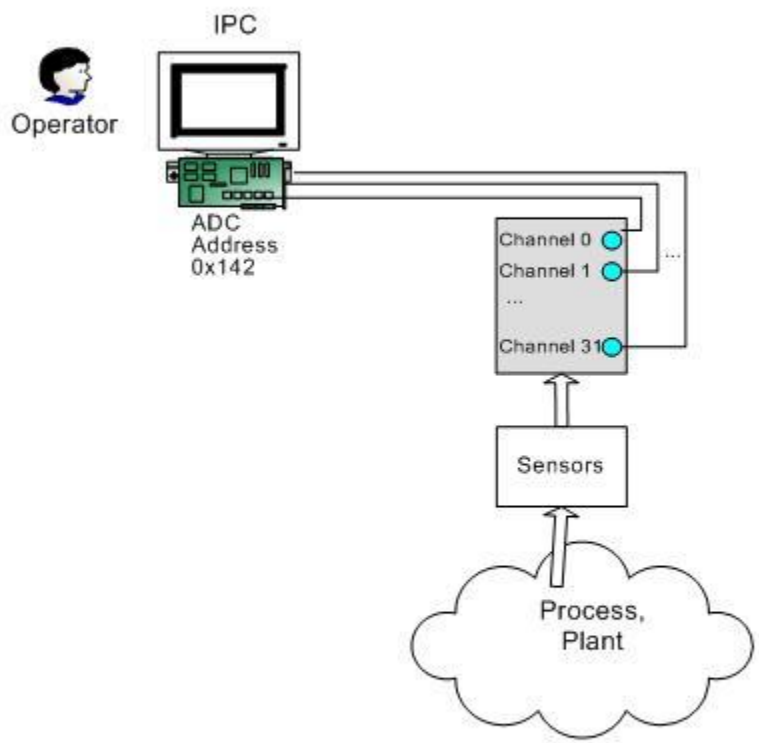
Add PLC ✕

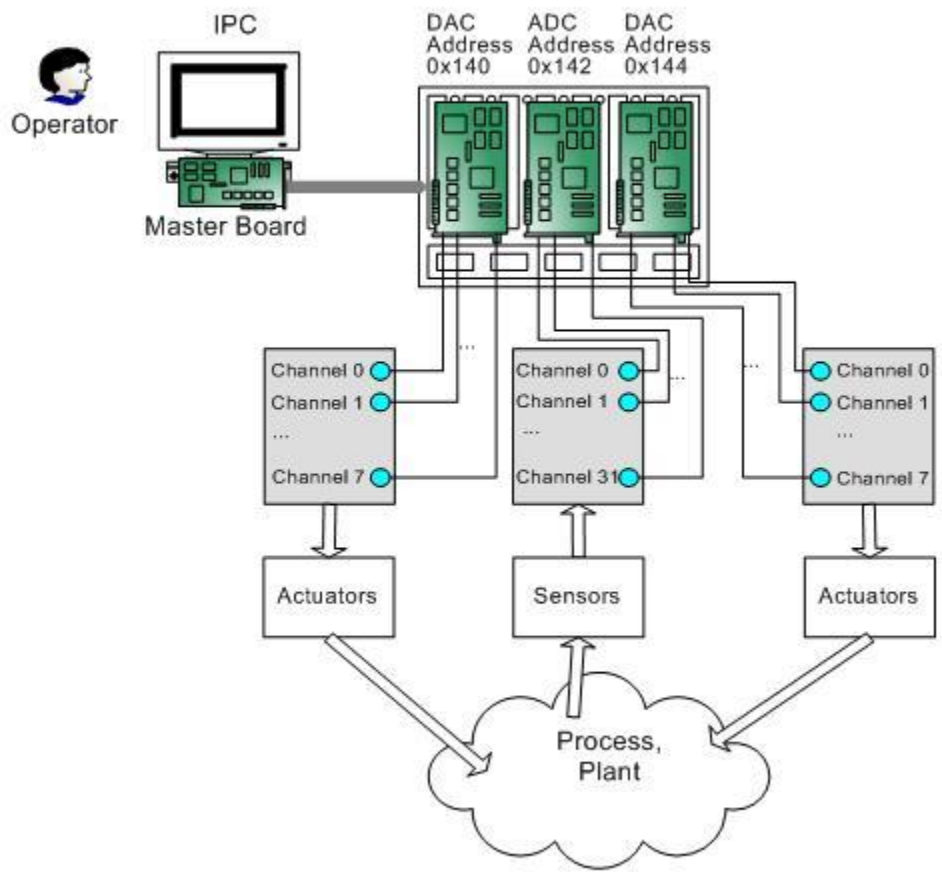
Device Name

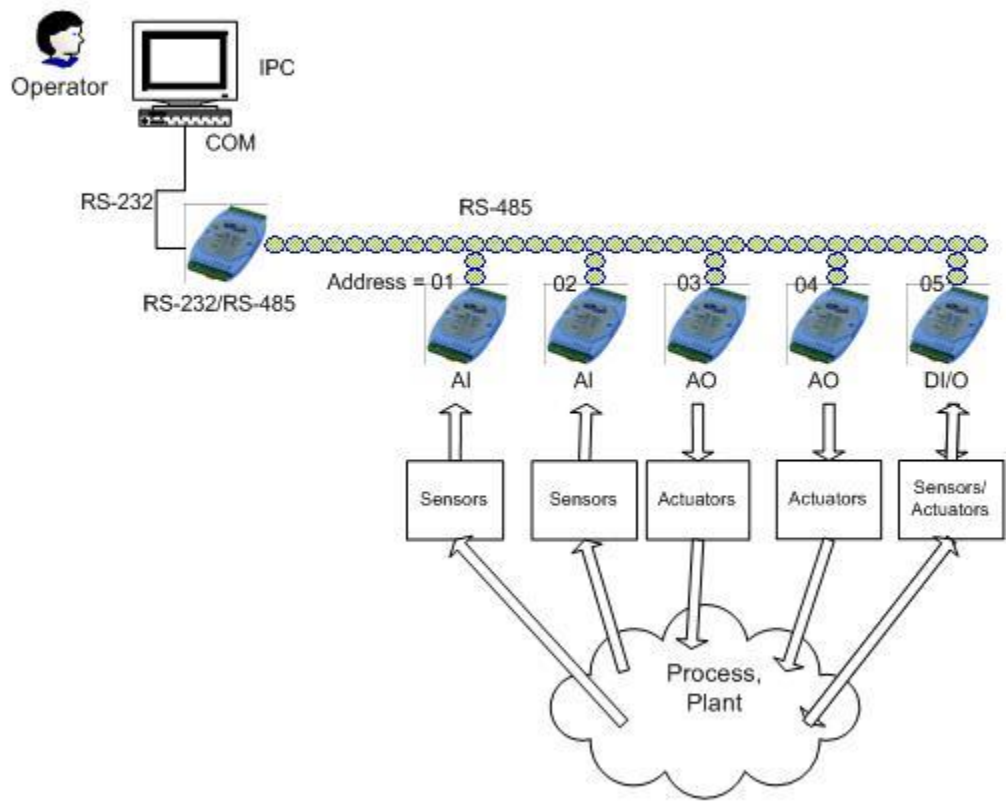
Device Type
 ▼

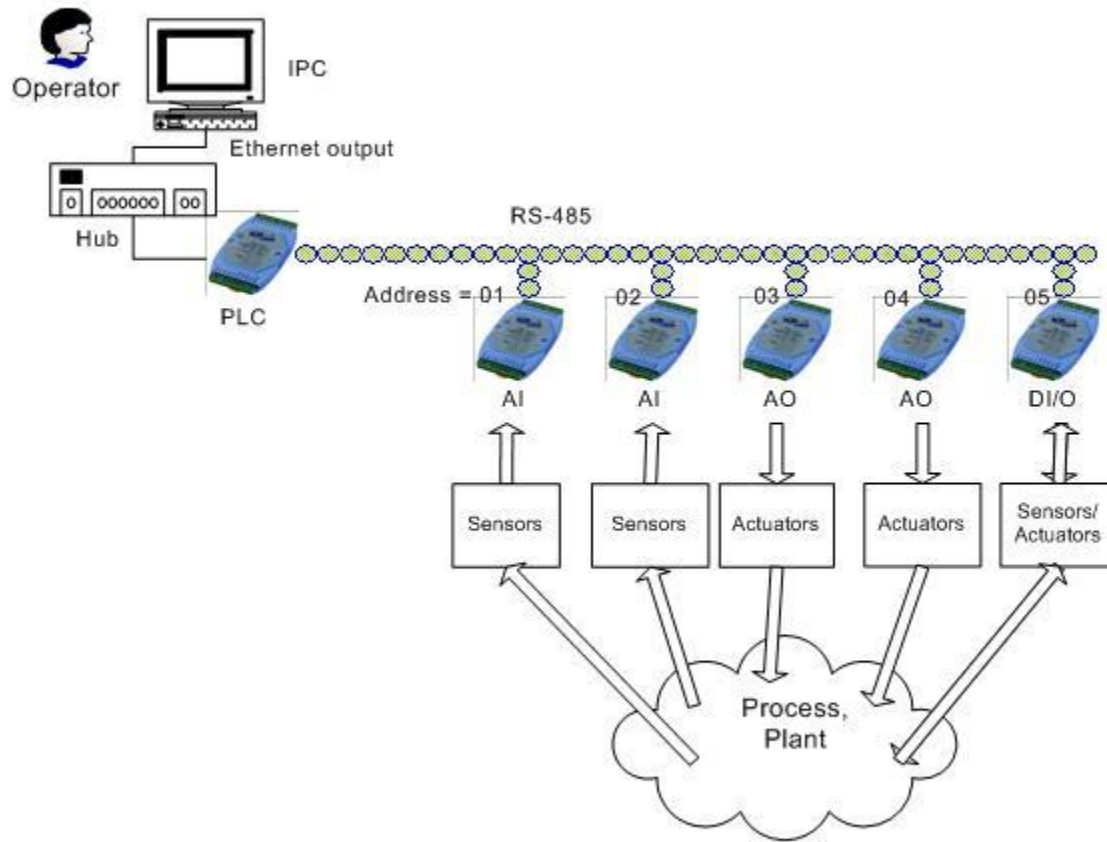
Network Type
 ▼

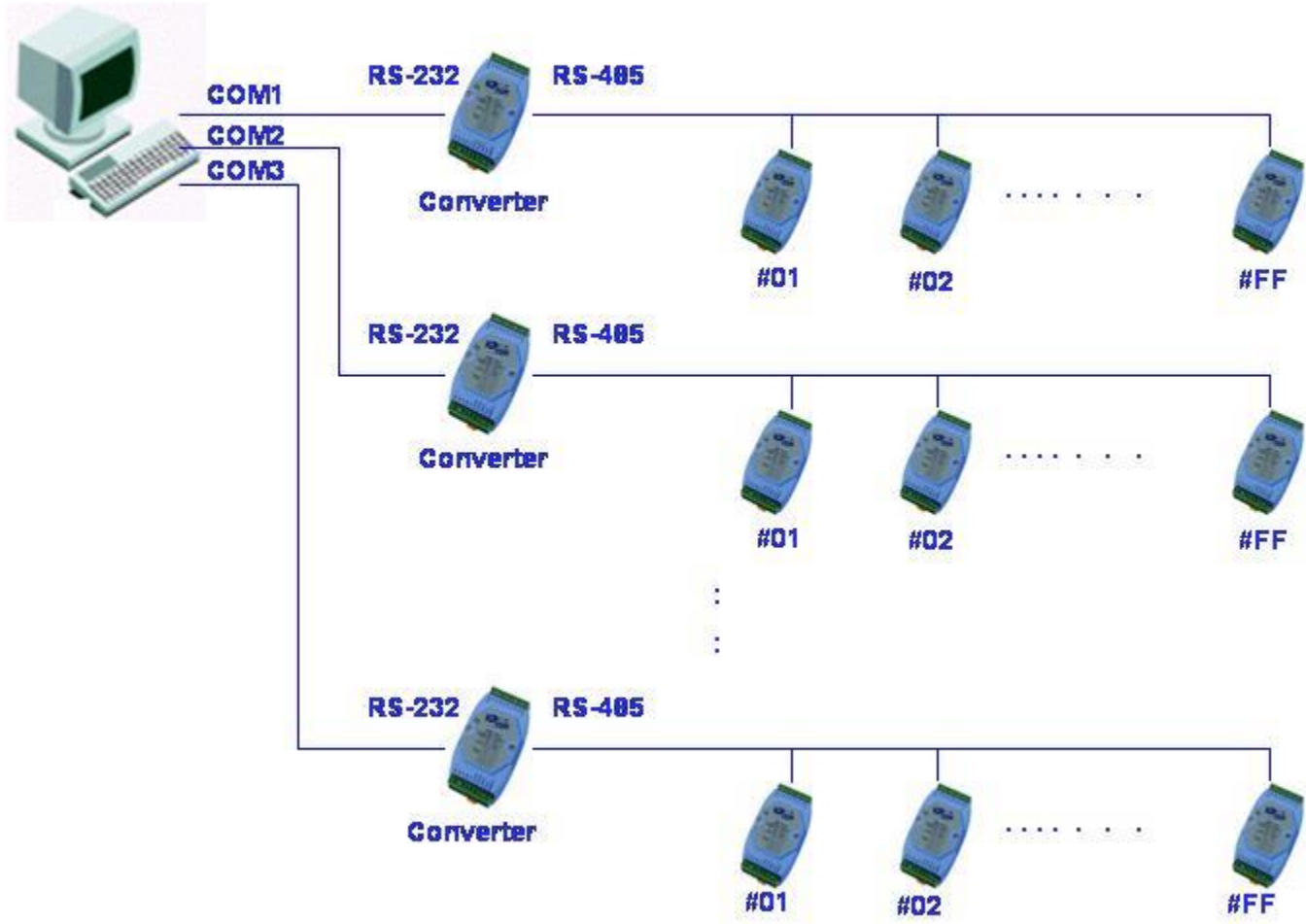
Comment

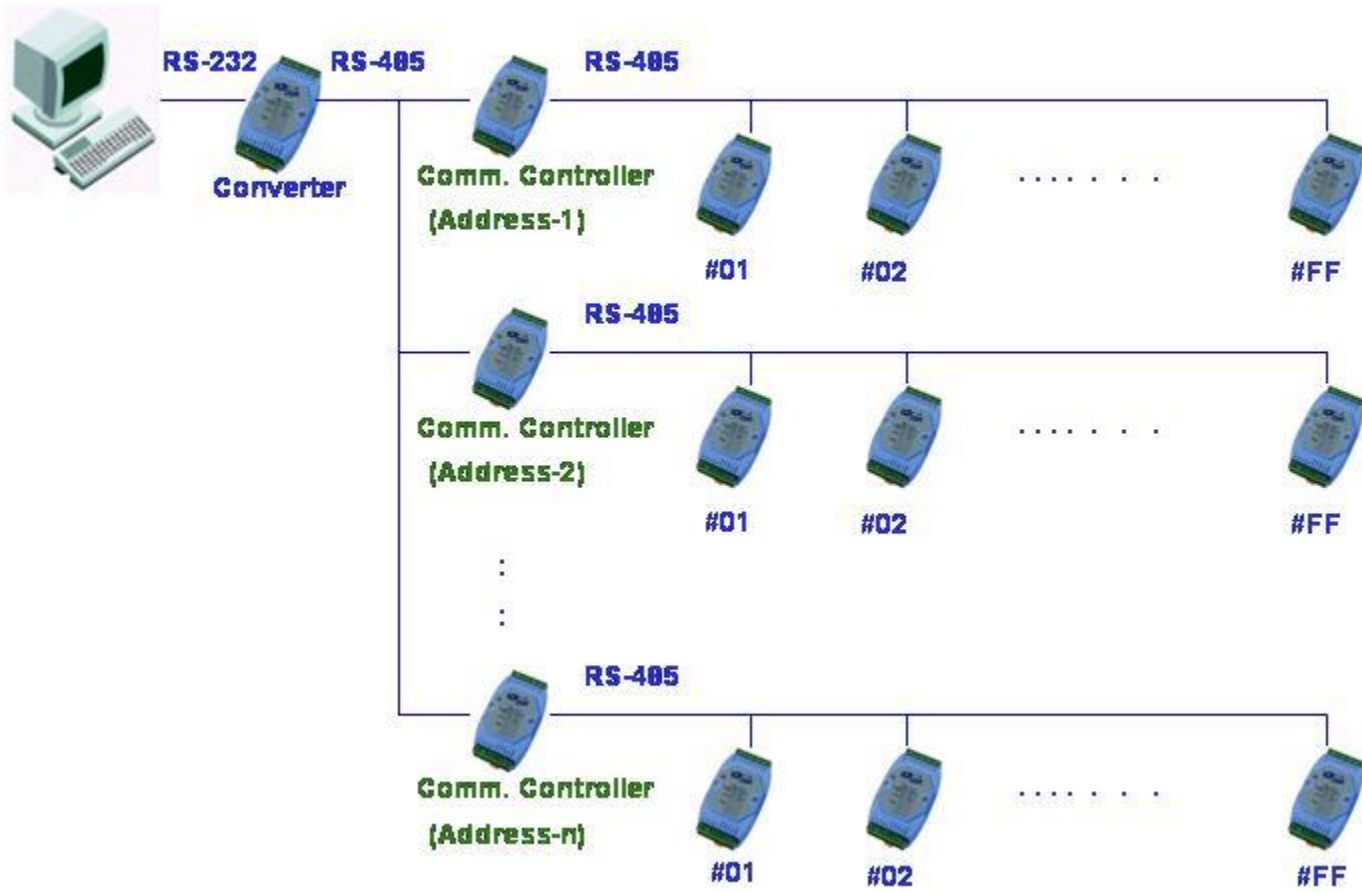


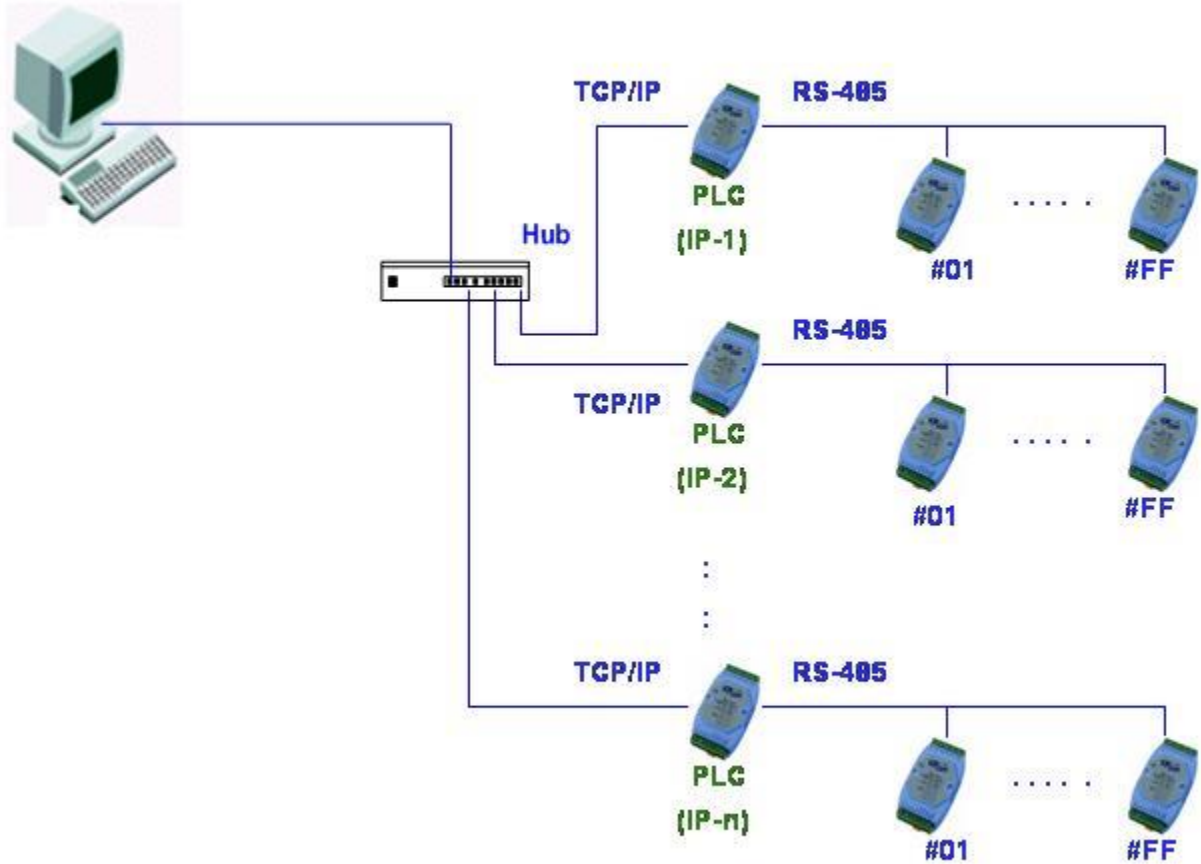


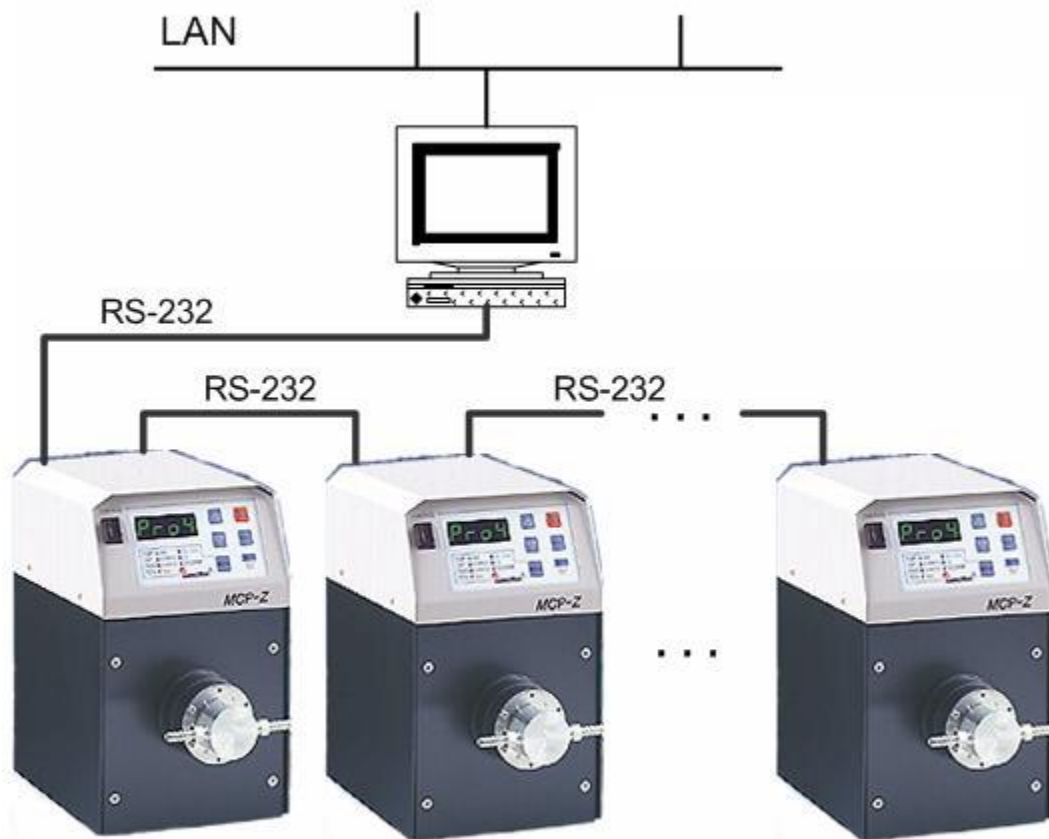






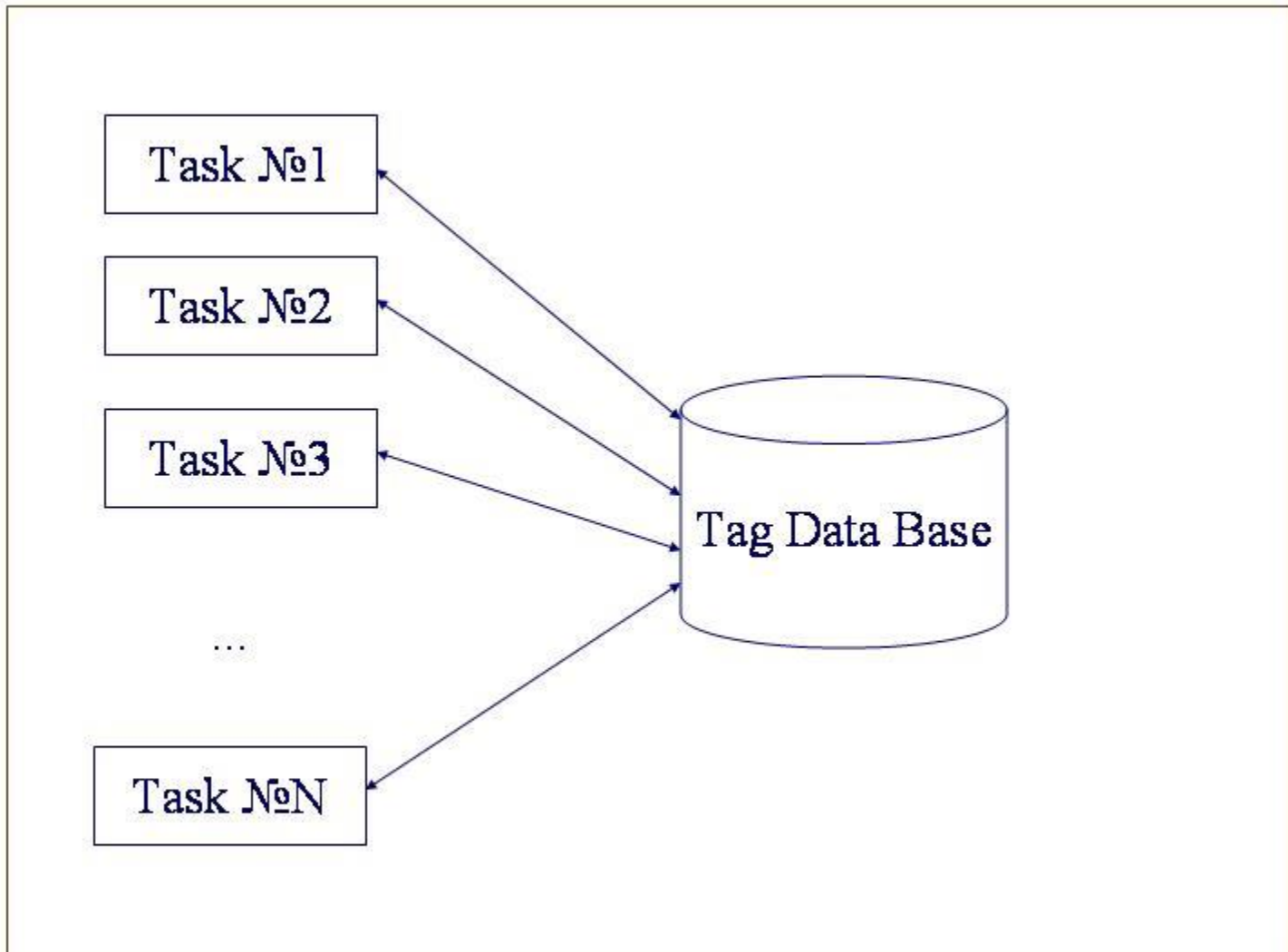


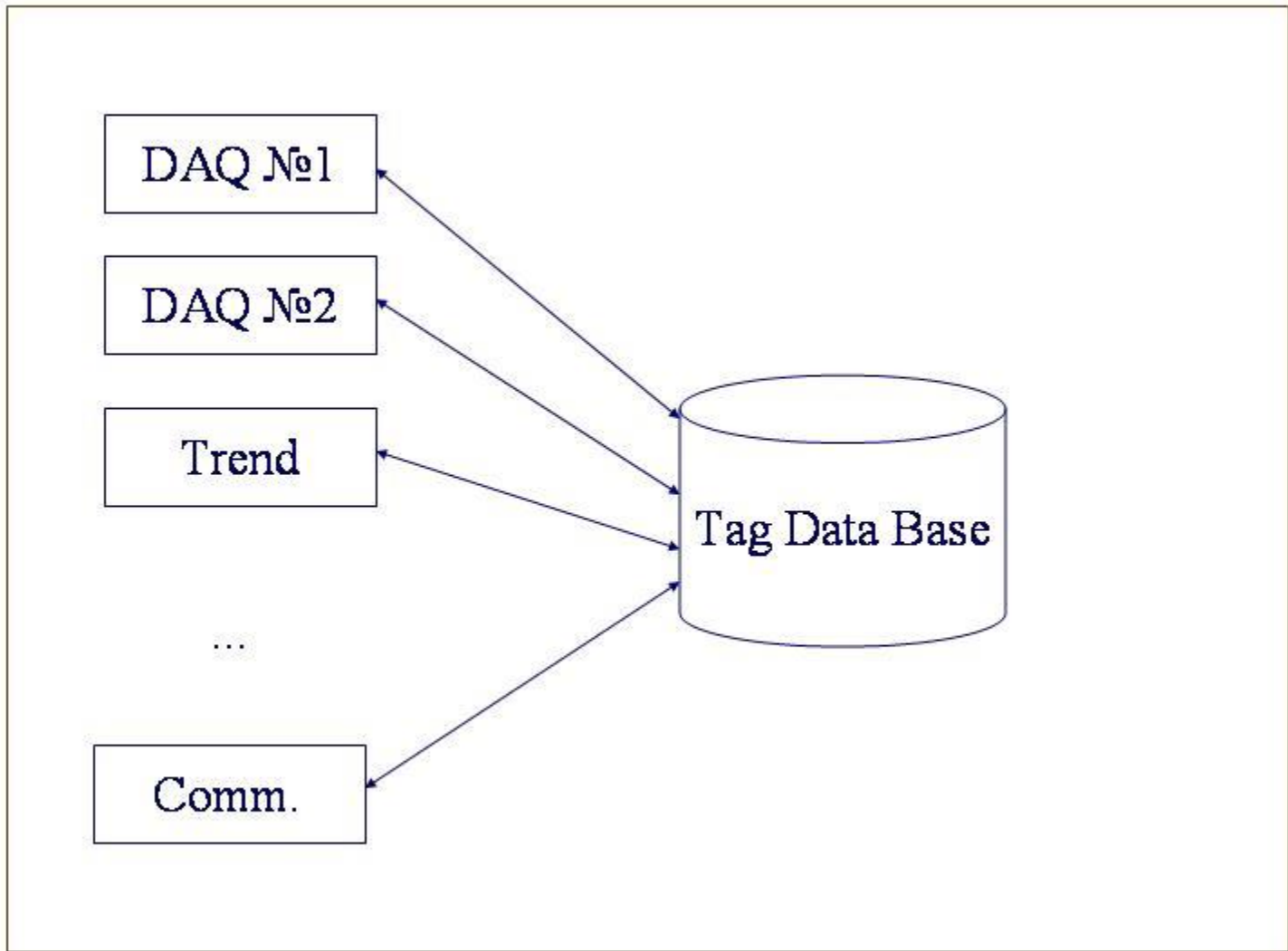




Task Concept

$$\text{Task} = \langle \text{Tag}_1, \text{Tag}_2, \dots, \text{Tag}_N \rangle$$



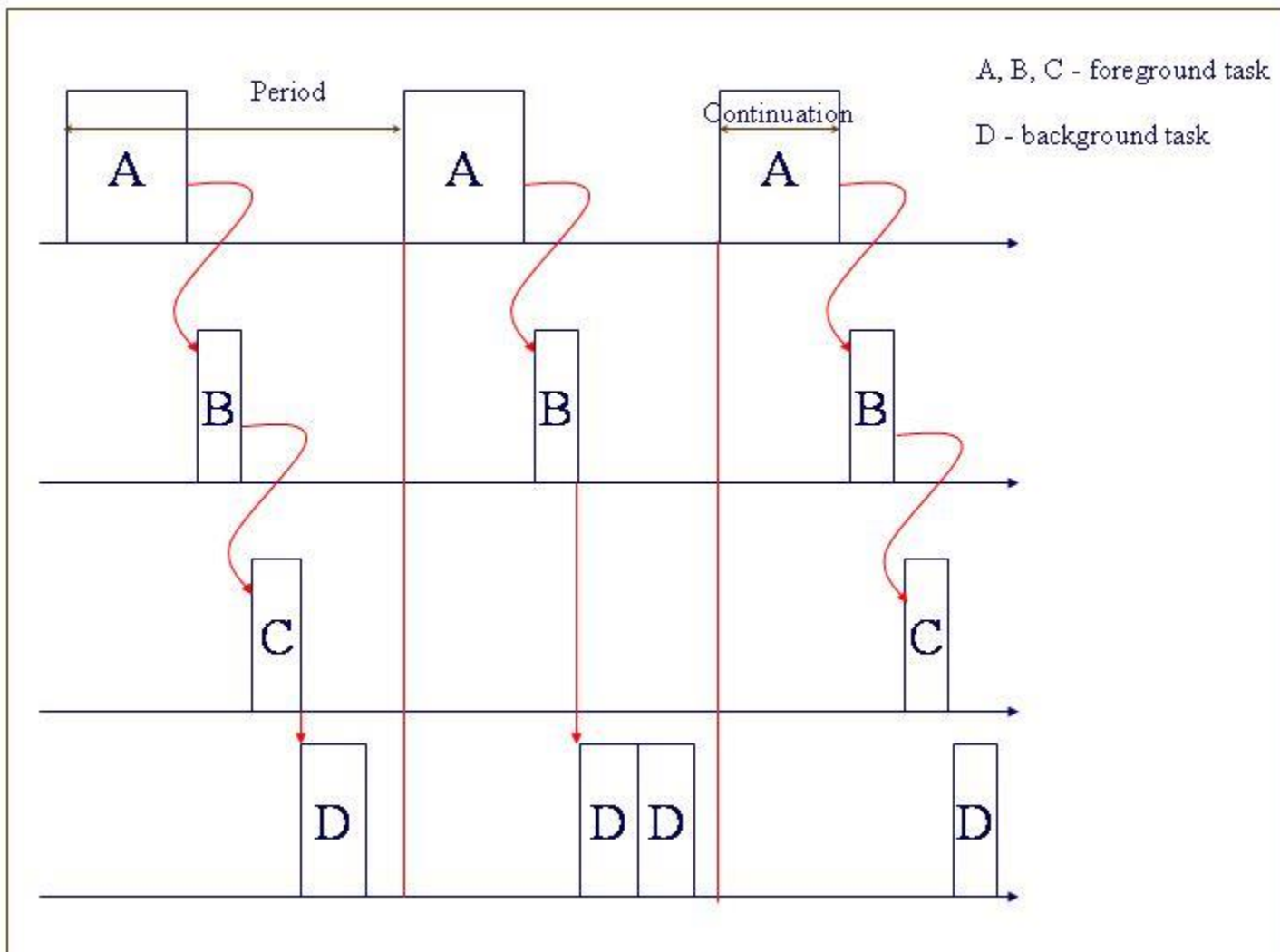


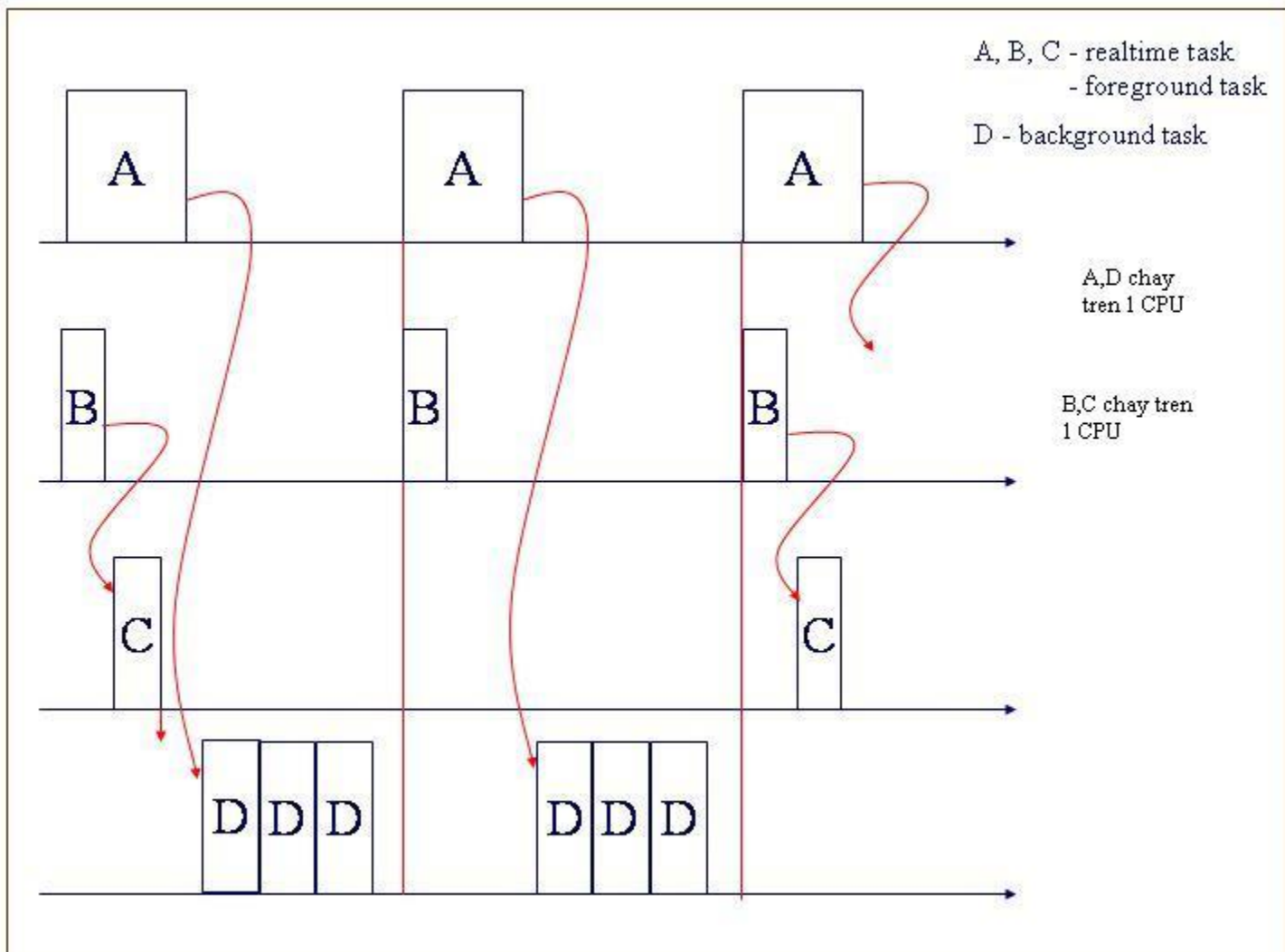
Realtime task (foreground task):

**Period,
Continuation,
Priority**

Background task:

**Continuation,
Priority**





Priority:

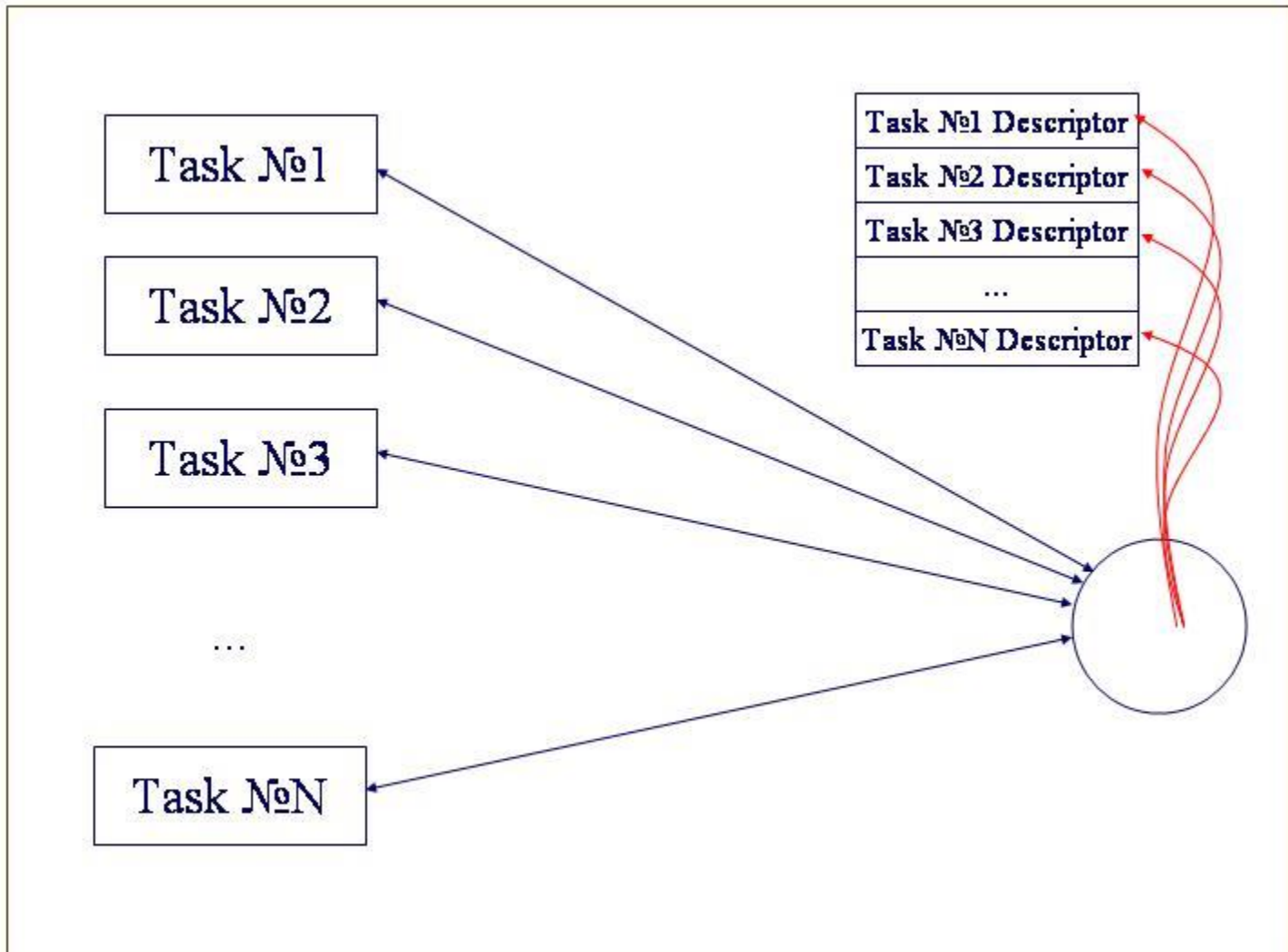
- + Absolute priority.
- + Relative priority - the task has when using the operating system's multitasking features.
- + The priorities are (from lowest to highest):
 - Lowest,
 - Below normal,
 - Normal,
 - Above normal,
 - Time critical.



Realtime Task

- **Scan Period:** the time interval between successive retrievals of information used by the task
- **Duration** is the time when you want the task to run.
 - + **Free Run** will enable the task to run continuously,
 - + **Time-based** will run the task for a given period of time,
 - + **Scan-based** will run the task for a specified number of scans (i.e., retrievals of data).
- **Starting method** controls when the task starts to run.
 - + **Immediate** automatically runs the task when the runtime is enabled,
 - + **Inactive** keeps the task disabled until invoked by command,
 - + **Delayed** enables the task after a period of time has elapsed from starting the runtime,
 - + **System Time** uses the computer's real-time clock to enable the task.

- Pre-task
- Post-task



```
AfxBeginThread(ThreadProc, NULL, 0);  
    // _beginthread( ThreadProc, 0, NULL );
```

```
AfxBeginThread(ThreadProc1, NULL, 0);  
AfxBeginThread(ThreadProc2, NULL, 1);
```

```
void ThreadProc1(void * pParam)  
{  
    while(1)  
    {  
        ...  
        Sleep(100);  
    }  
}
```

```
void ThreadProc2(void * pParam)  
{  
    while(1)  
    {  
        ...  
        Sleep(100);  
    }  
}
```

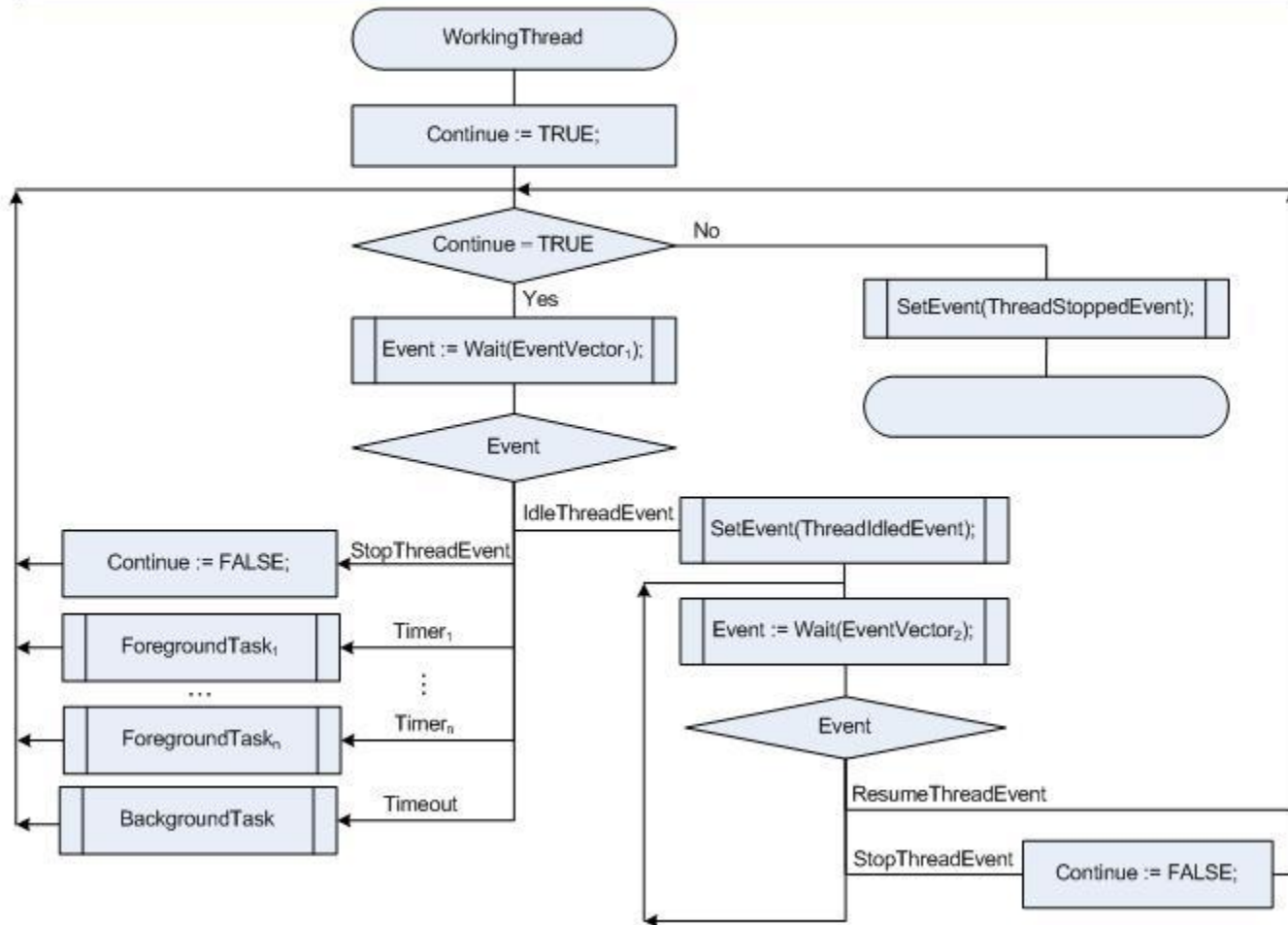
```

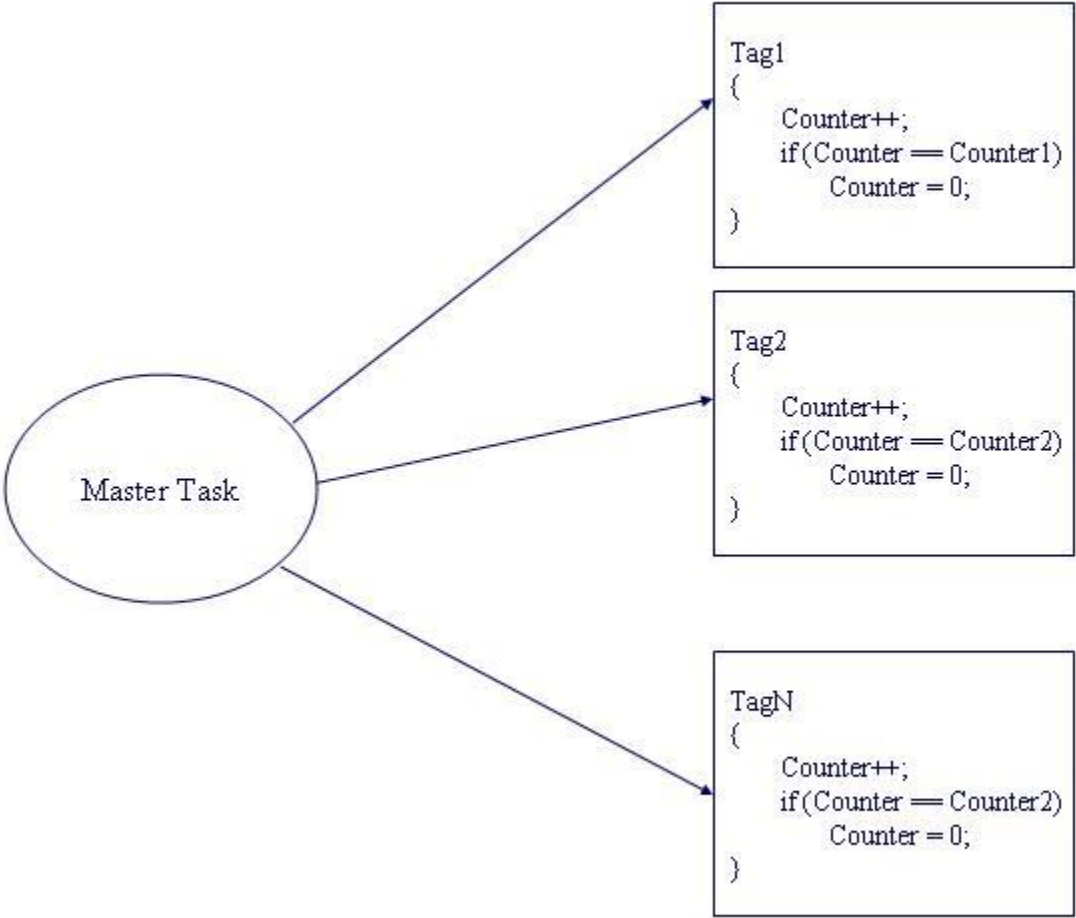
float          Volts[2];
void WorkingThread(void * pParam)
{
    float  dt = (float)0.1;
    float  t = 0;
    int    i;
    float  Volt;
    while(1)
    {
        Sleep(1000);
        t = t + dt;
        Volts[0] = (float)5* sin(1*t);

        if(Volts[0] >= 0.0)
            Volts[1] = 5.0;
        else
            Volts[1] = -5.0;
    }
}
main()
{
    _beginthread( WorkingThread, 0, NULL );
    while( 1 )
    {
        Sleep(1000);
        printf("%20f%20fn", Volts[0], Volts[1]);
    }
}

```

$EventVector1 = \{IdleThreadEvent, StopThreadEvent, Timer1, Timer1, \dots, Timern\}$
 $EventVector2 = \{ResumeThreadEvent, StopThreadEvent\}$

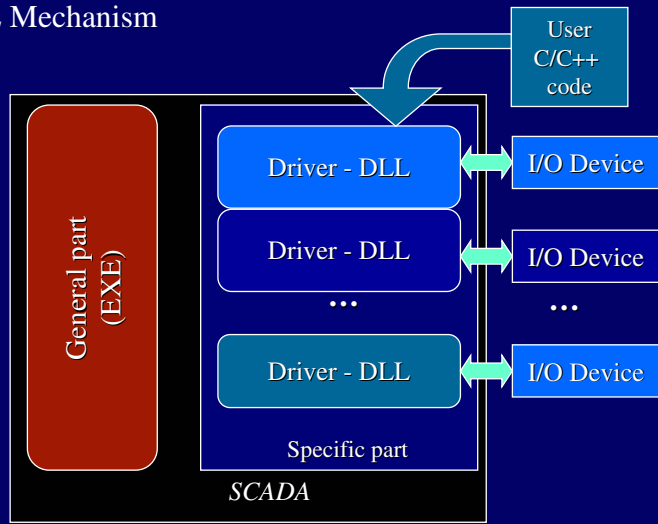




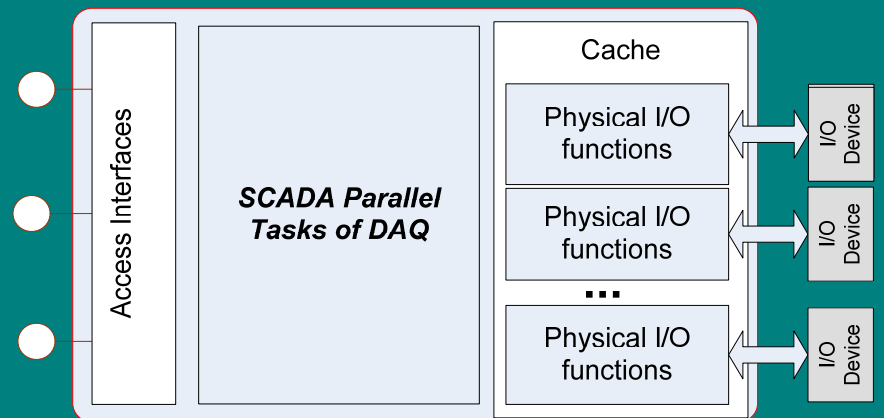
Device Connecting Mechanisms

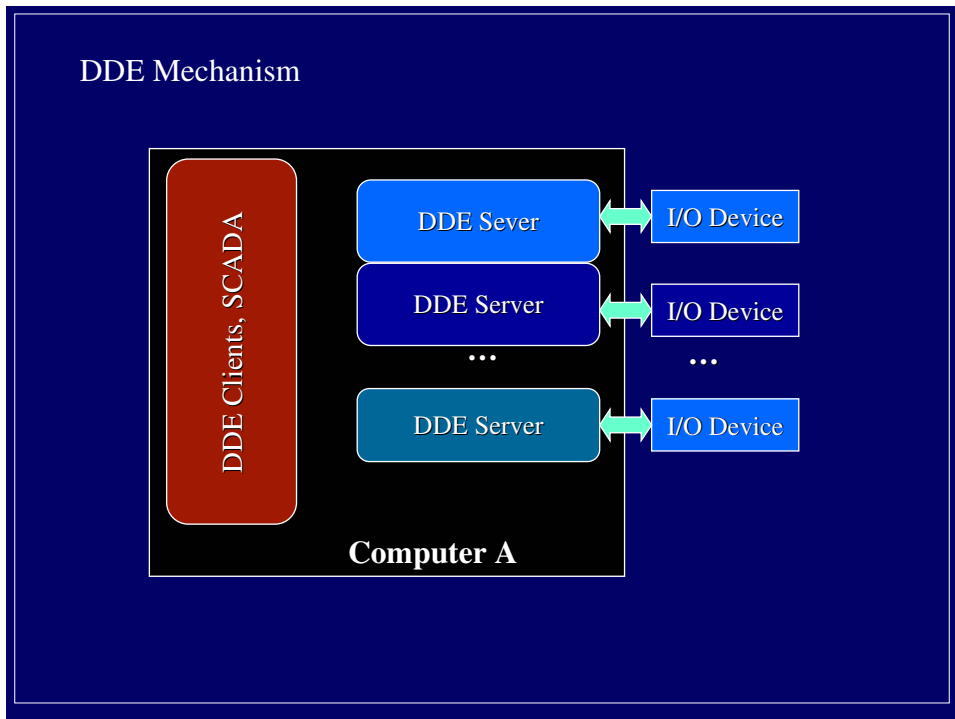
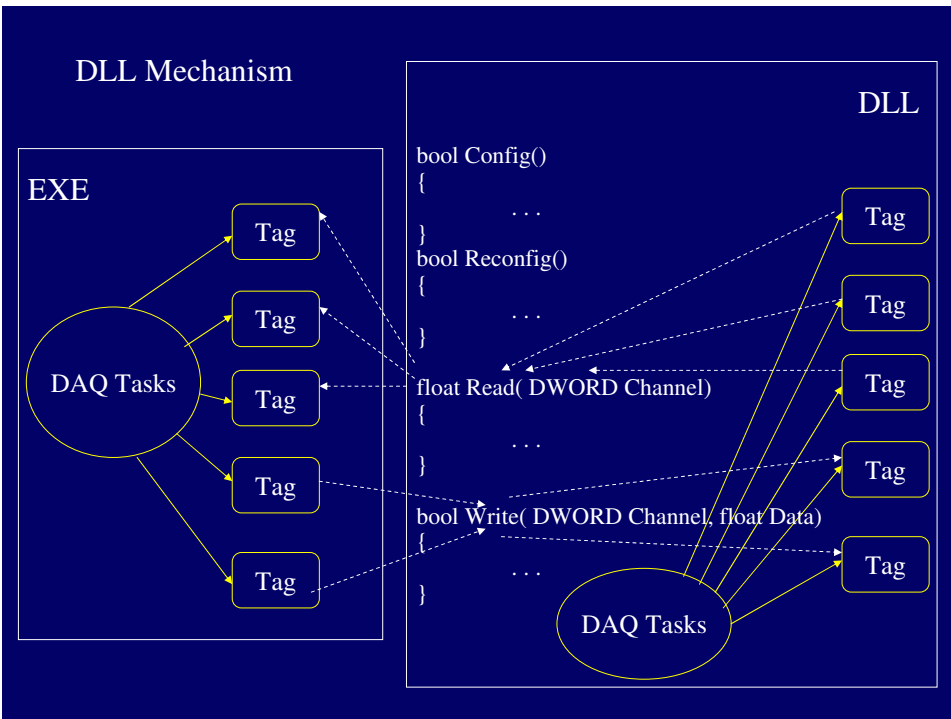
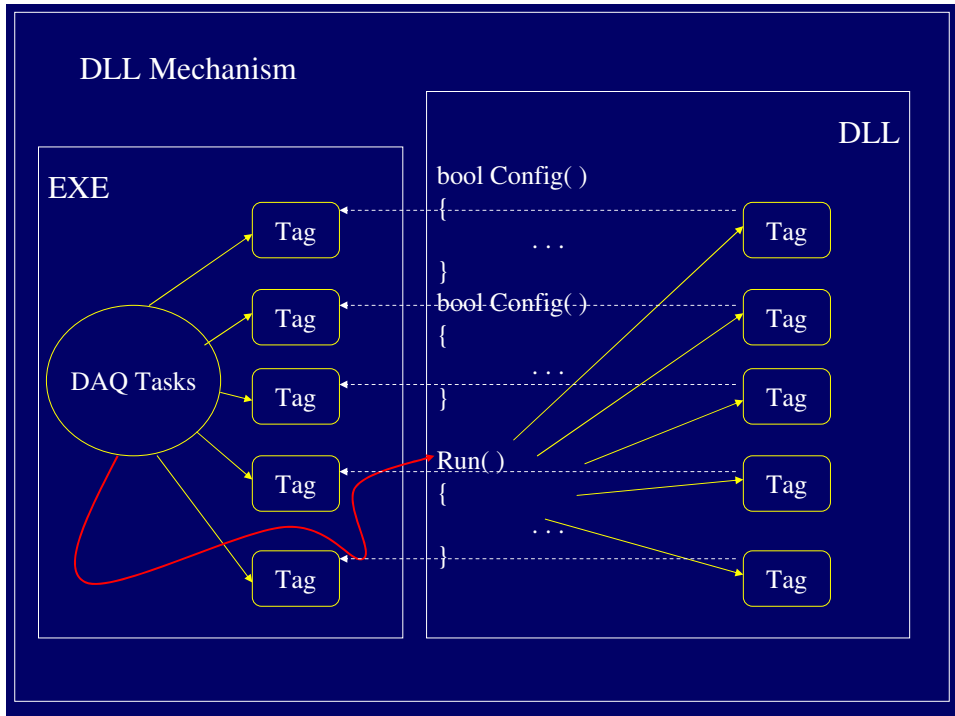
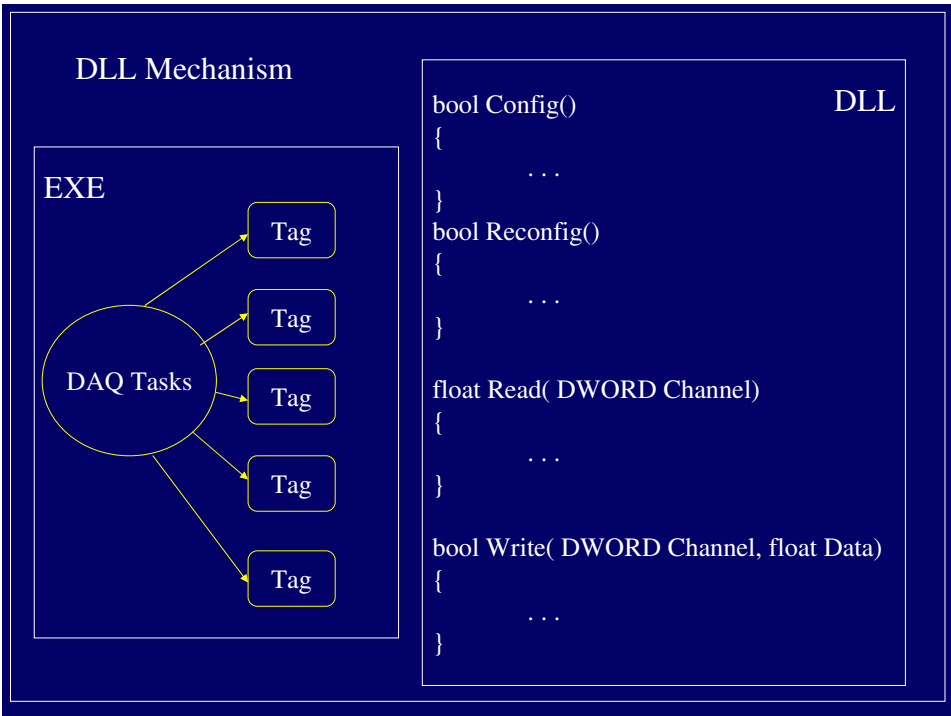
1. DLL - Dynamic-link Library
2. DDE - Dynamic Data Exchange
3. OLE - Object Linking & Embedding
4. OPC - OLE for Process Control

DLL Mechanism

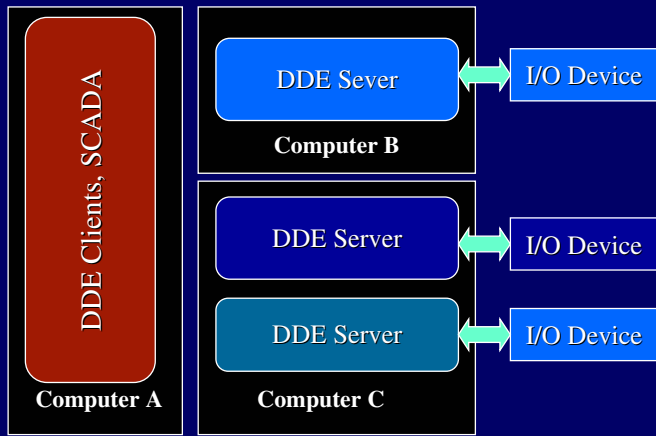


DLL Mechanism

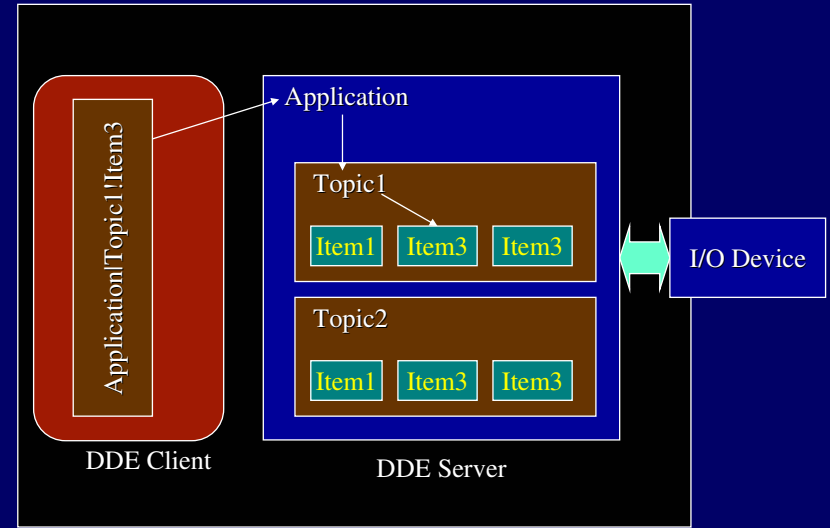




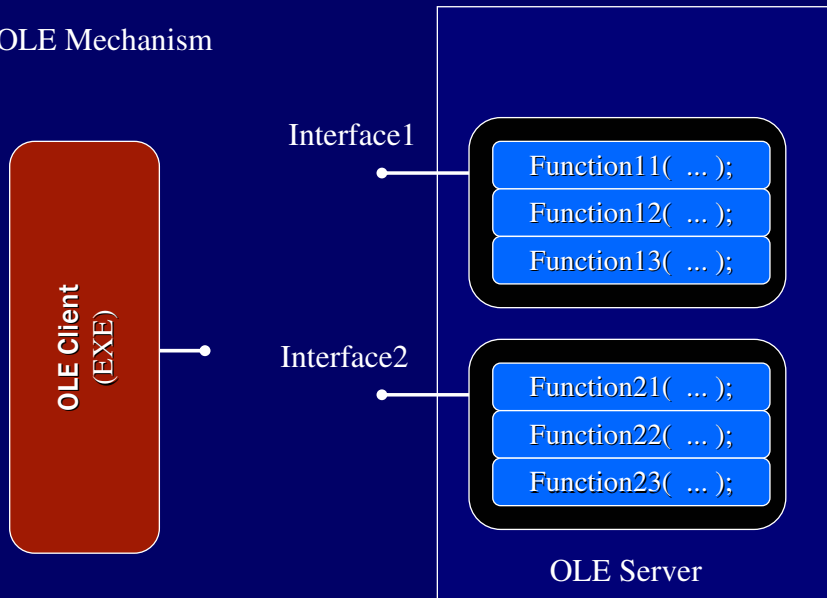
DDE Mechanism



DDE Mechanism

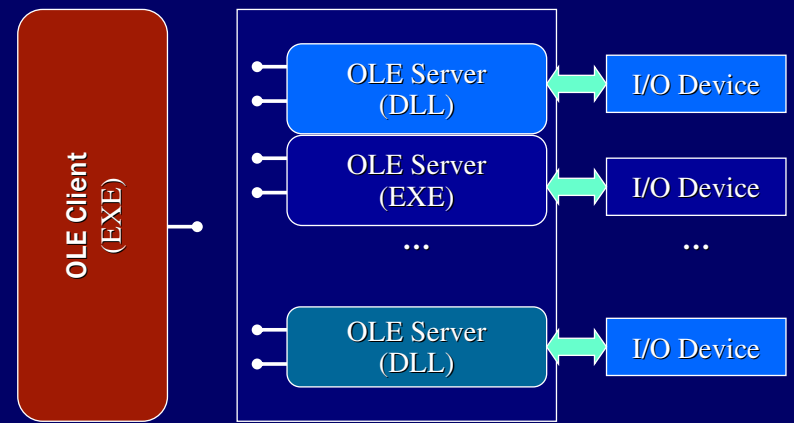


OLE Mechanism

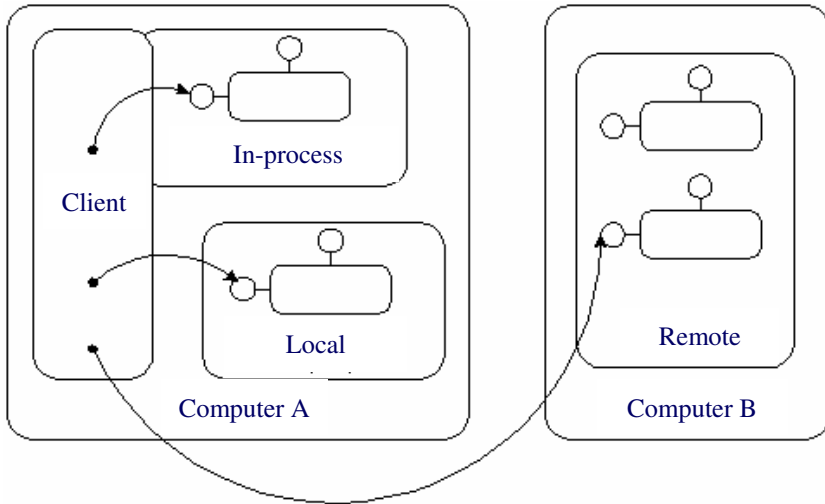


COM - Component Object Model

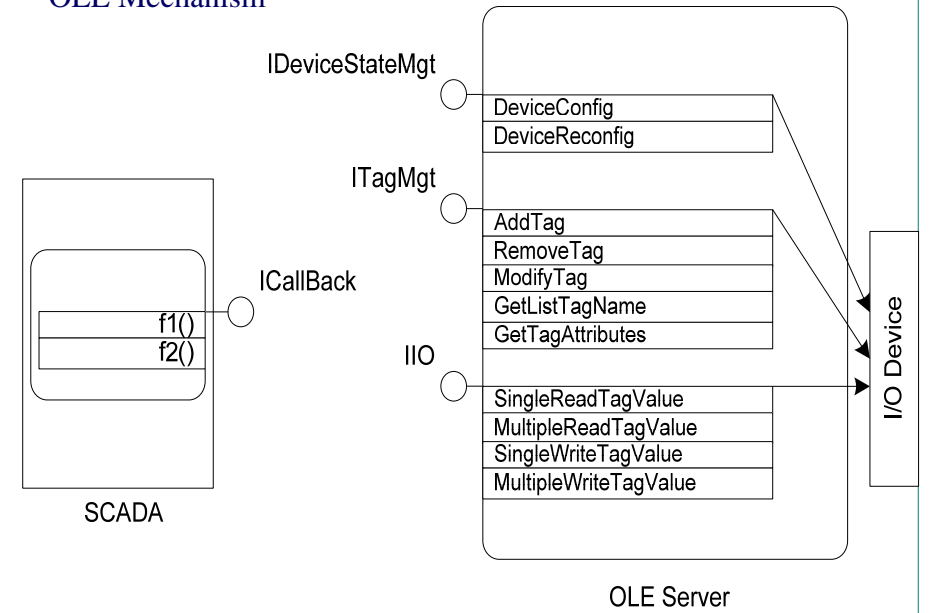
OLE Mechanism



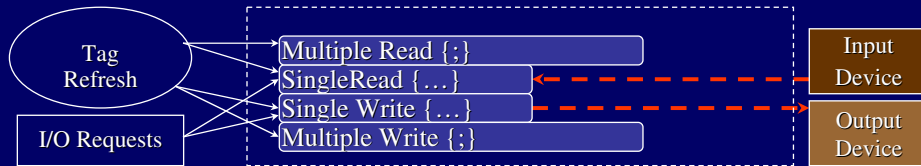
OLE Mechanism



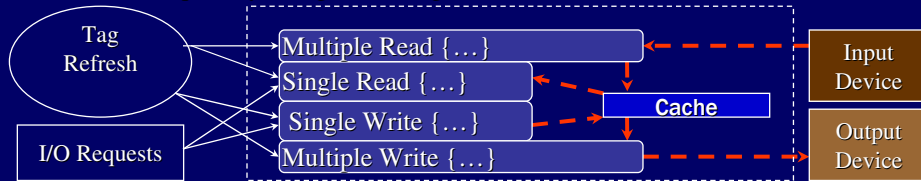
OLE Mechanism



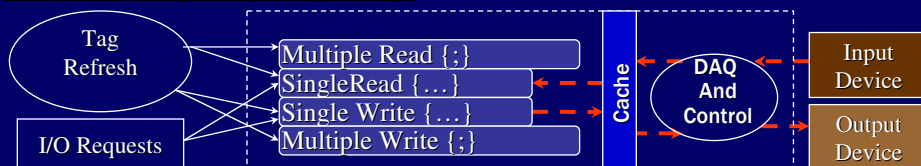
a) Fast device, quantity of tags is not high



b) Slow device, quantity of tags is not high



c) Slow device, quantity of tags is high



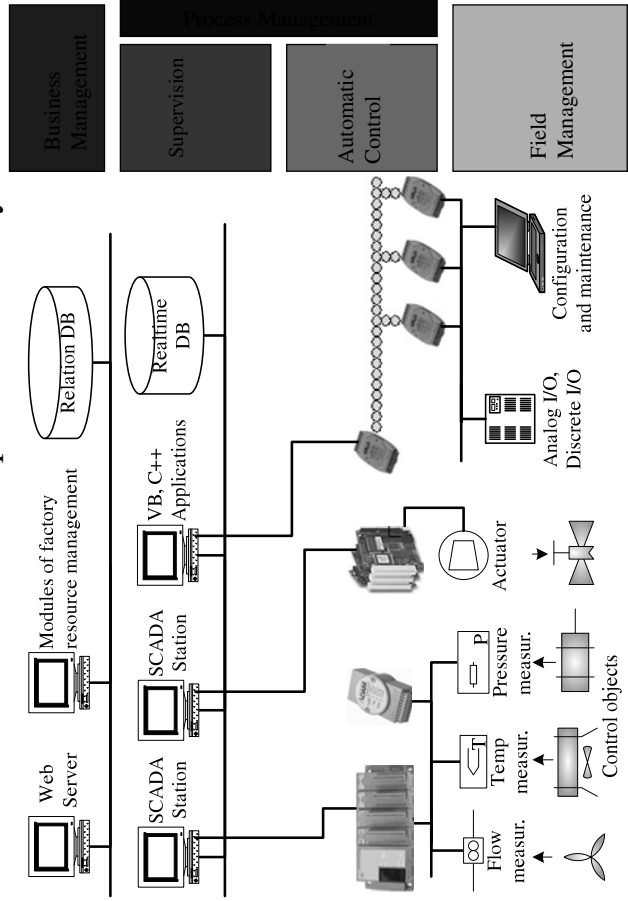
○ Task {;} - Empty Meth. {...} - Not empty Meth. - -> Data direction - -> Call



Environment for Industrial System Integration

St. Petersburg, December 3, 2002

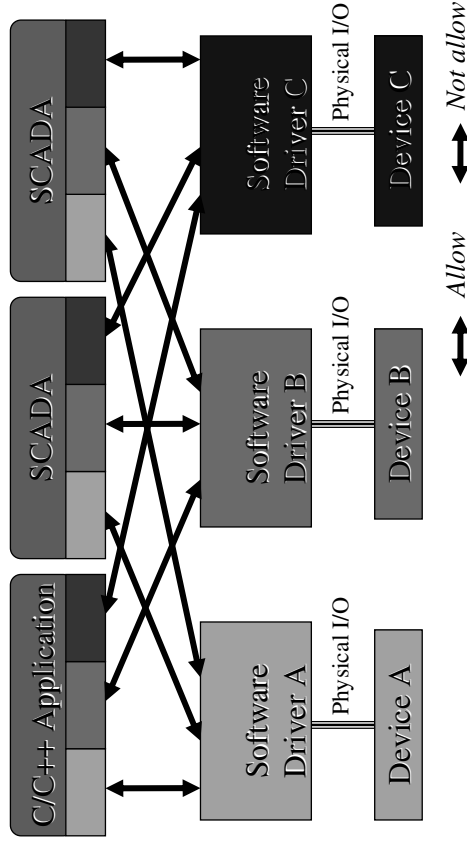
Architecture of modern process control system



Topics

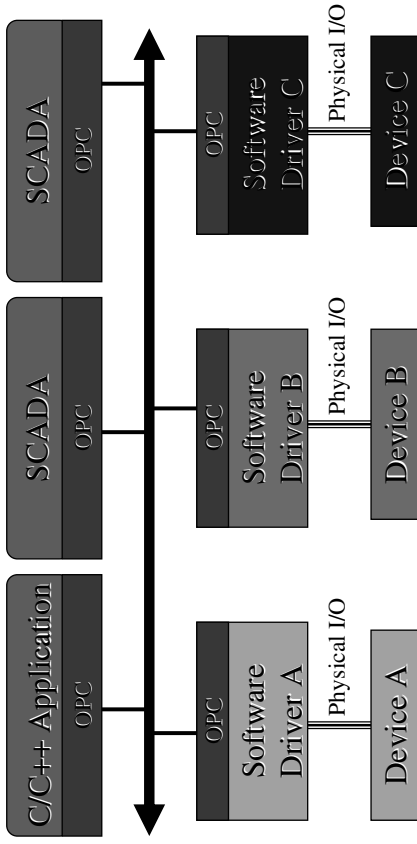
- Why OPC?
- What is OPC?
- Technical features of OPC Data Access (DA)
 - Logical Object Model of OPC Applications (Client/Server)
 - Interfaces of OPC Data Access
 - Data source in OPC Server
 - Mechanism of data exchange between OPC Client/Server
- Structure of modern SCADA systems
- Implementation
- Application

The I/O driver problem

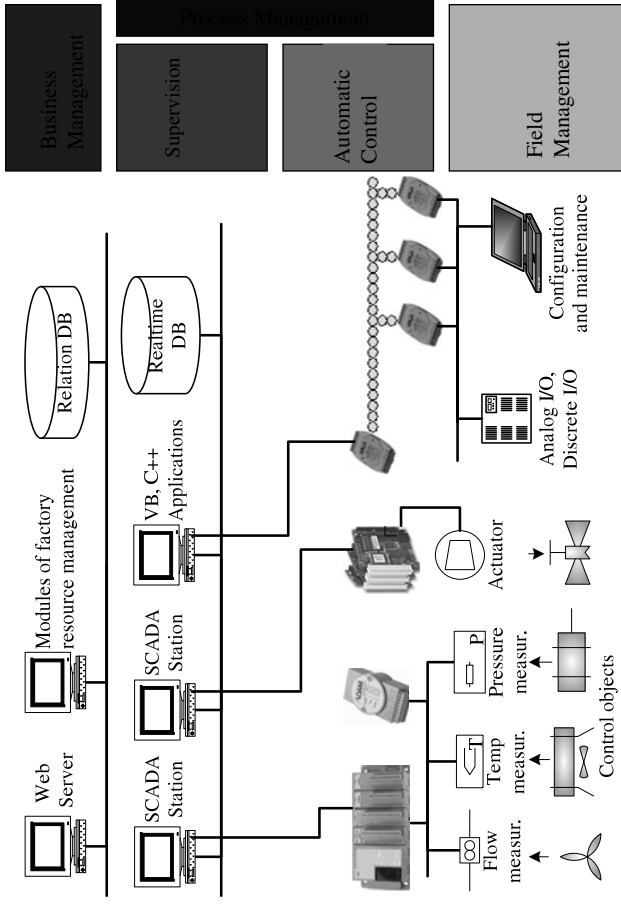


- Each SCADA system \leftrightarrow driver for each equipment.
- Two applications can not have access to one driver in the same time.
- Remote access is not supported

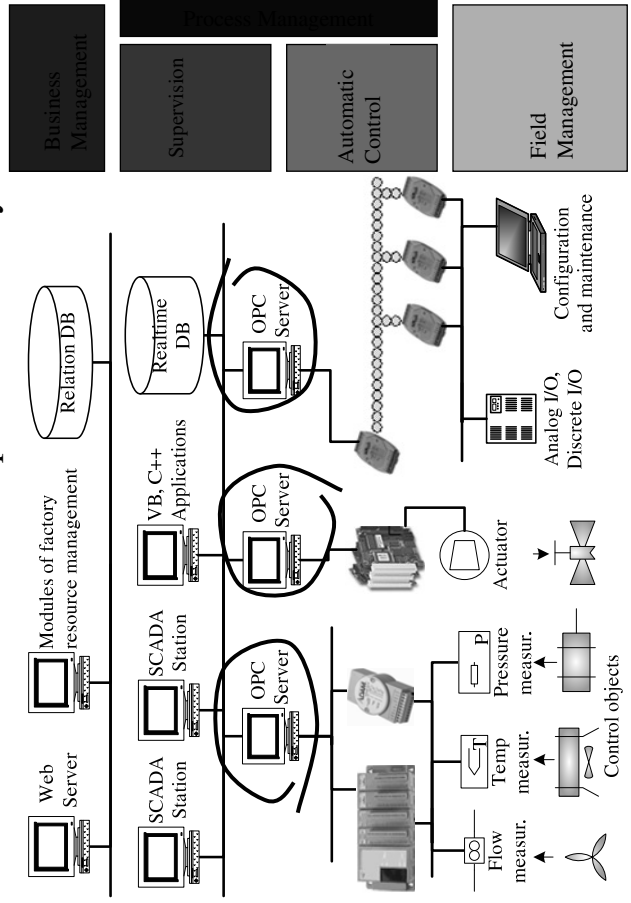
How does OPC solve the problem?



Architecture of modern process control system



Architecture of modern process control system



What are OPCs

Company	Country
InTouch	USA
GenieDAQ	USA
Genesis32	USA
Trace Mode	Russia
Vijeo Look	France
Citect	Australia
Factory Link	USA
RSView	USA
Lab View	USA
iFIX	USA
WinCC	Germany
Master SCADA	Russia
COMPLIcity	USA
Contour	Ukraine
Wizcon	USA
Crug-2000	Russia
Ellipse SCADA	USA
Wonderware	USA
Advantech	USA
Iconics	USA
Ad Astra	Russia
Schneider Electric	France
CI Technologies	Australia
United States DATA Co.	USA
Rockwell Software Inc.	USA
National Instruments	USA
Intellution	USA
Siemens	Germany
InSAT	Russia
GE Fanuc	USA
Obedinente Ujig	Ukraine
Axeda	USA
Crug	Russia
Ellipse Software	USA

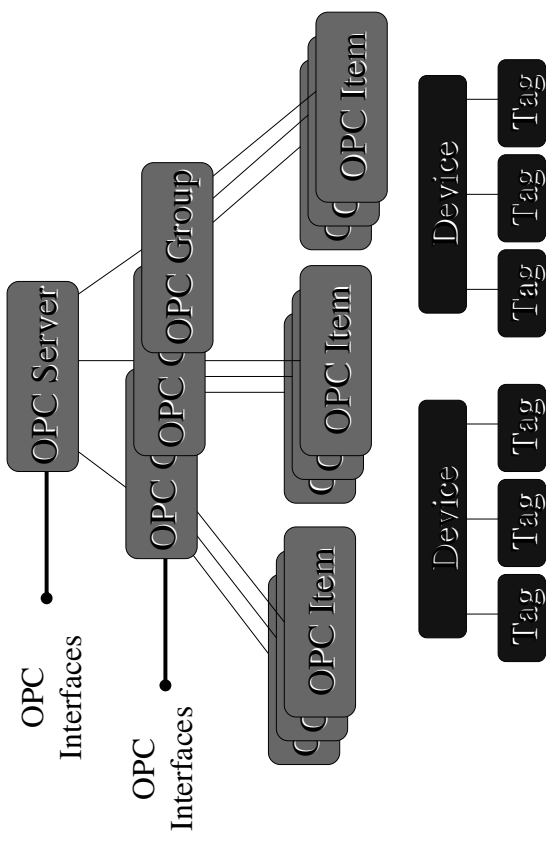
What is OPC?

Object Linking & Embedding

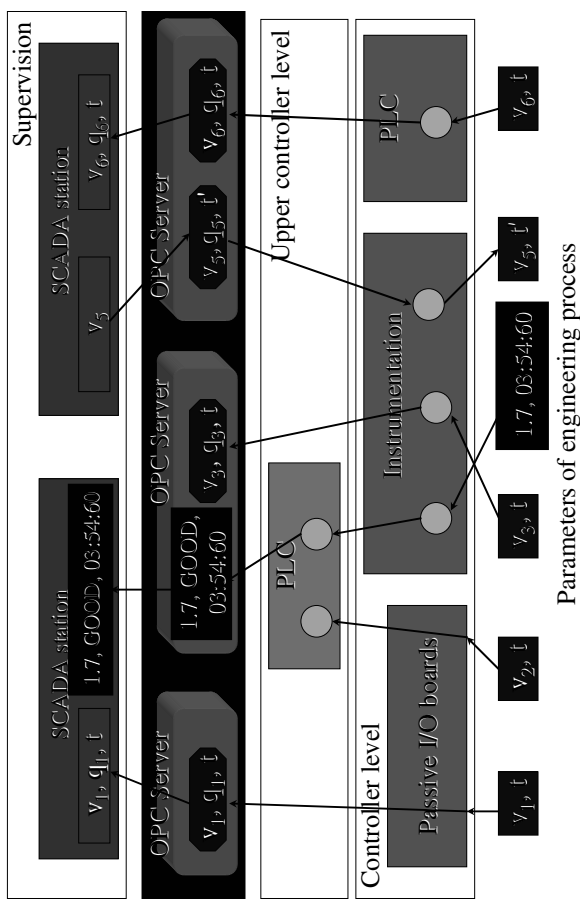
- OPC – OLE for Process Control
- OPC – OLE for Production Control
- OPC – Openness, Productivity and Connectivity
- OPC – ~~O~~LE COM for Process Control

COM – “Component Object Model” provides interfaces and inter-component communication. Through COM, an application may use features of any other application object. COM is the core of DCOM (Distributed COM), ActiveX.

The logical object model OPC Data Access Server

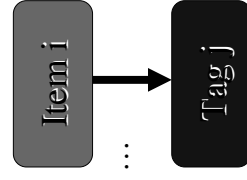


The path of real-time data in the integrated process control system

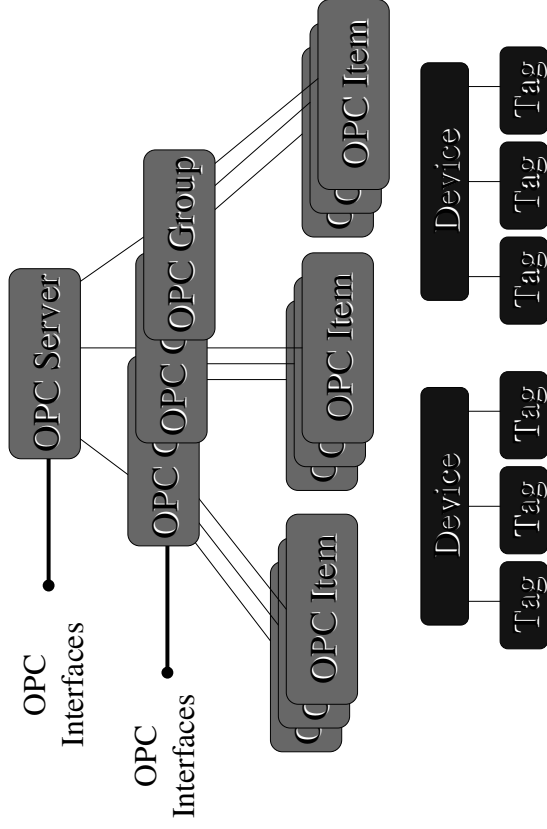


OPC Item Object

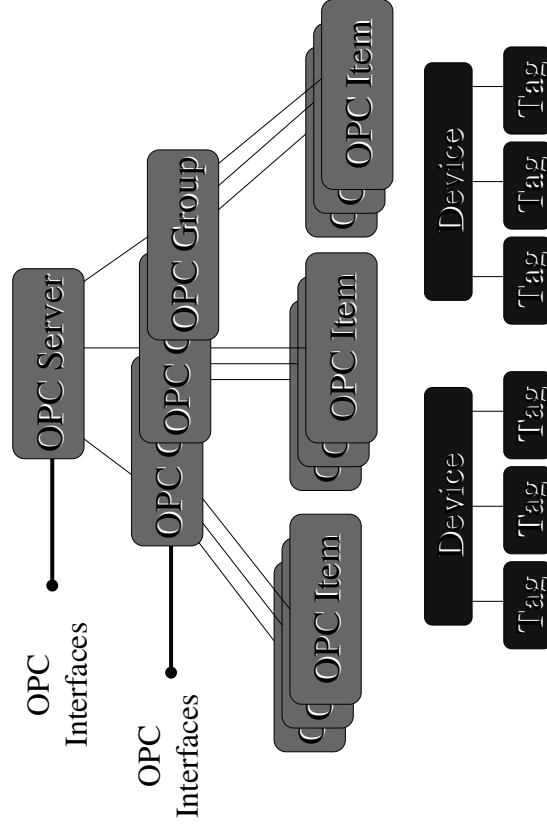
- ❖ Connection to a data source.
- ❖ Properties:
 - Data value (v)
VARIANT type: boolean, integer, float, string, ...
 - Quality (q)
UNCERTAIN, GOOD, BAD, ...
 - Time stamp (t)
 - FILETIME: {yyyy:mm:dd:hh:mm:ss:...}
 - The Timestamp should indicate the time that the value and quality was obtained by the device.
 - ...



The logical object model OPC Data Access Server

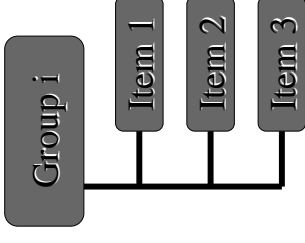


The logical object model OPC Data Access Server



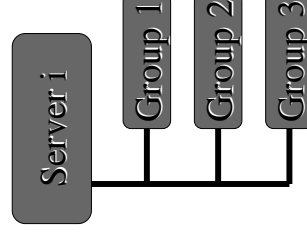
OPC Group Object

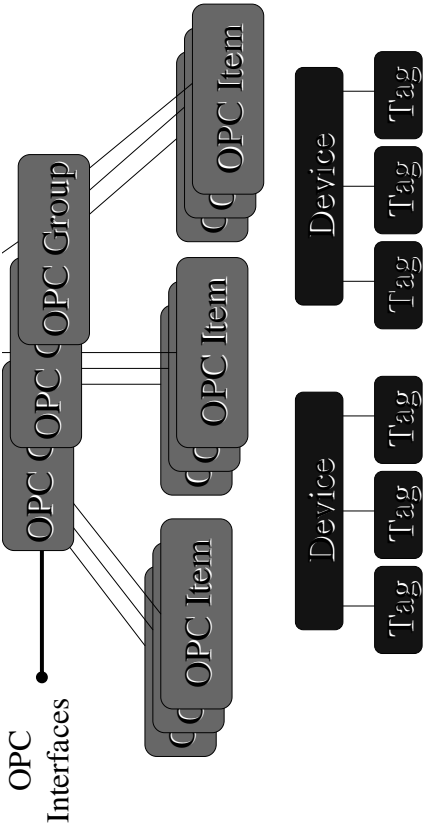
- ❖ Collection of items.
- ❖ Clients must use group to access items.
- ❖ Properties:
 - Requested update rate;
 - Percent dead band;
 - Time bias;



OPC Server Object

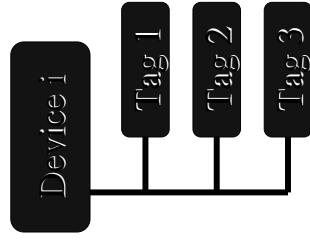
- ❖ It is a COM object.
- ❖ Collection of groups.





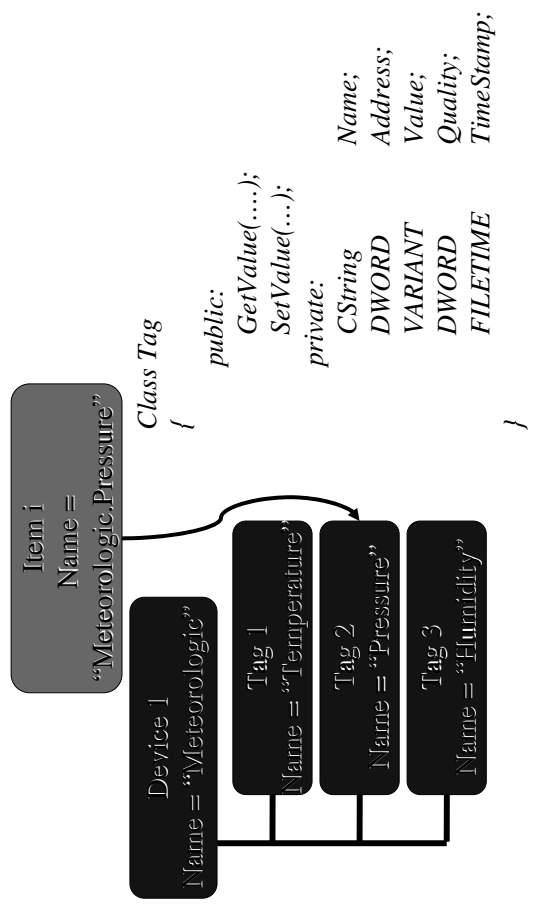
Device Object

- ❖ Physical device abstraction, hardware connection management.
- ❖ Direct communication with I/O devices or user designed DLL.
- ❖ Collection of tags.
- ❖ Tags are updated regularly.

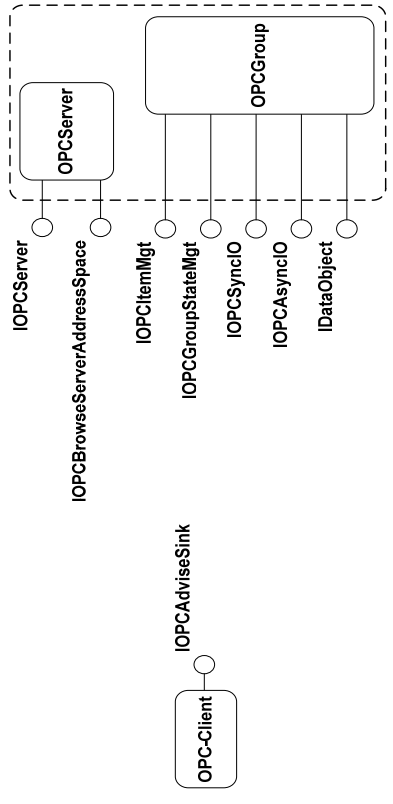


Tag

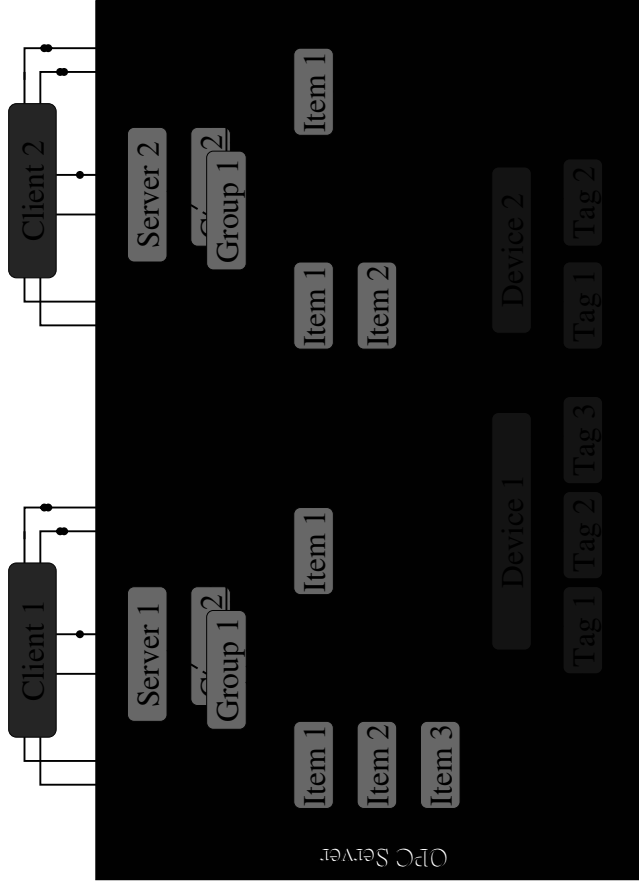
- ❖ Parameter of engineering process.
- ❖ Analog, digital, numeric, string, etc.
- ❖ Item Name = Full tag name



OPC DA 1.0



Overall picture of the client/server interaction



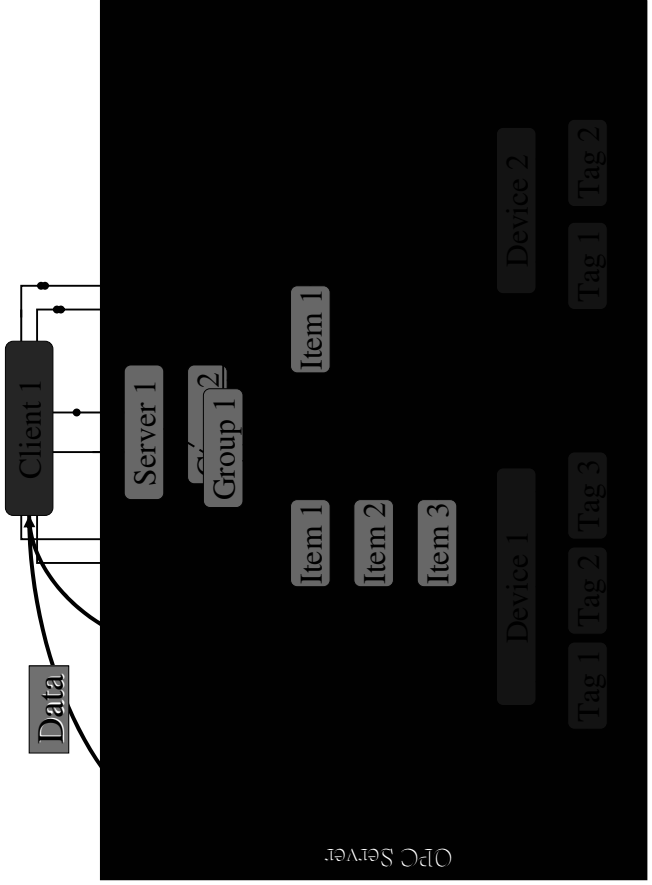
Data source in OPC Servers

- Read from server's internal cache.
- Read directly from device.
- Write directly to the device.

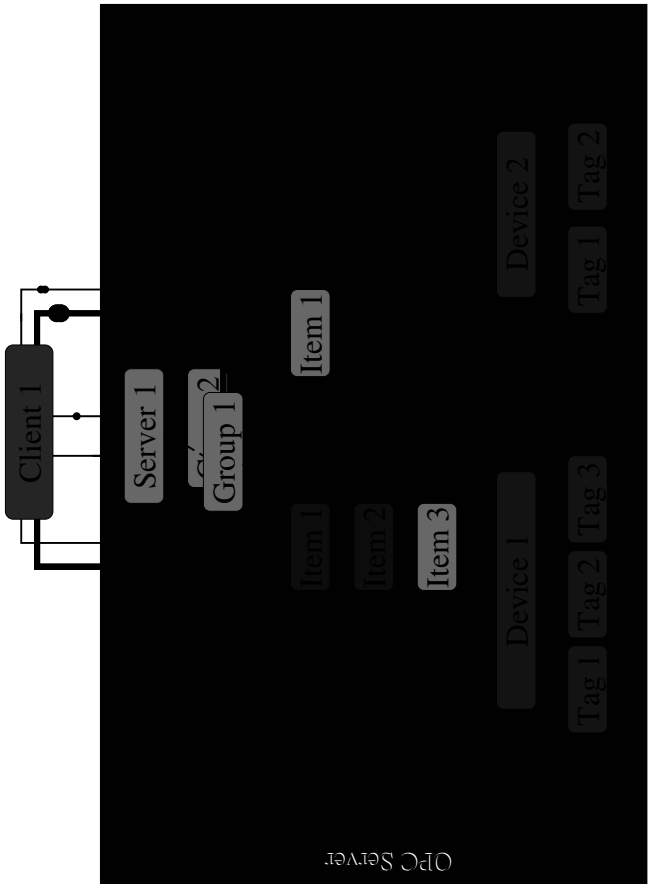
Modes of data exchange in OPC Servers

- Synchronous read/write.
- Asynchronous read/write.
- Subscription (for read).
- Refresh (for read).

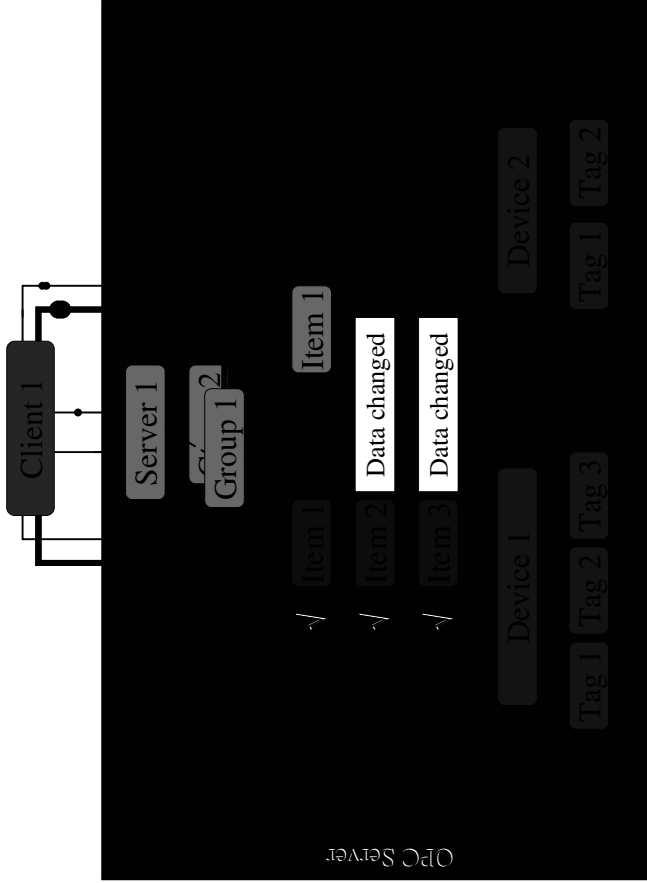
Synchronous mode



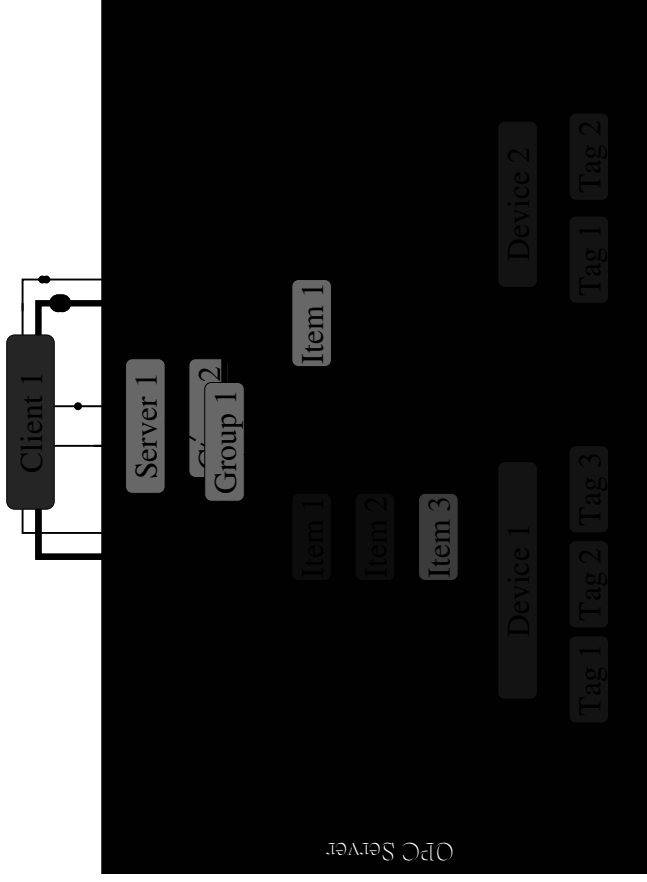
Asynchronous mode



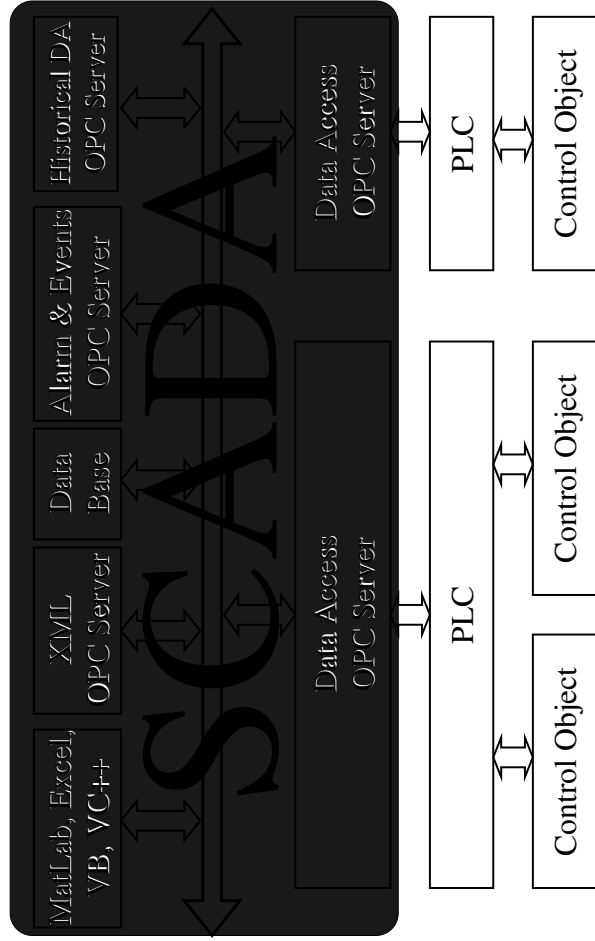
Subscription



Refresh = Asynchronous mode for all items in a group



Structure of modern SCADA systems



Implementation

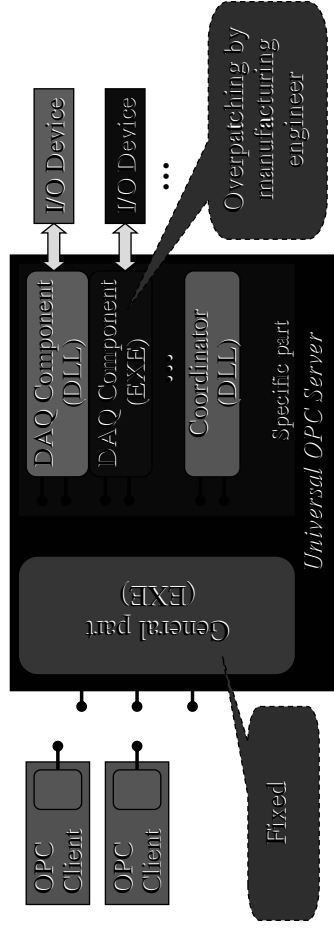
From zero

- OPC Specification;
- Protocol, Driver or Server of I/O devices;
- COM Programmers, Visual studio C++.

OPC Toolkits

- Very high price;
- Restriction.

Universal OPC Server





Summary

- Very interesting concept
- Need for industrial system integration (industrial devices, SCADA systems)
- Standard across vendors, no longer separate drivers for SCADA.
- Allow applications to easily access subsystem data
- Are optimized for use over a network

References

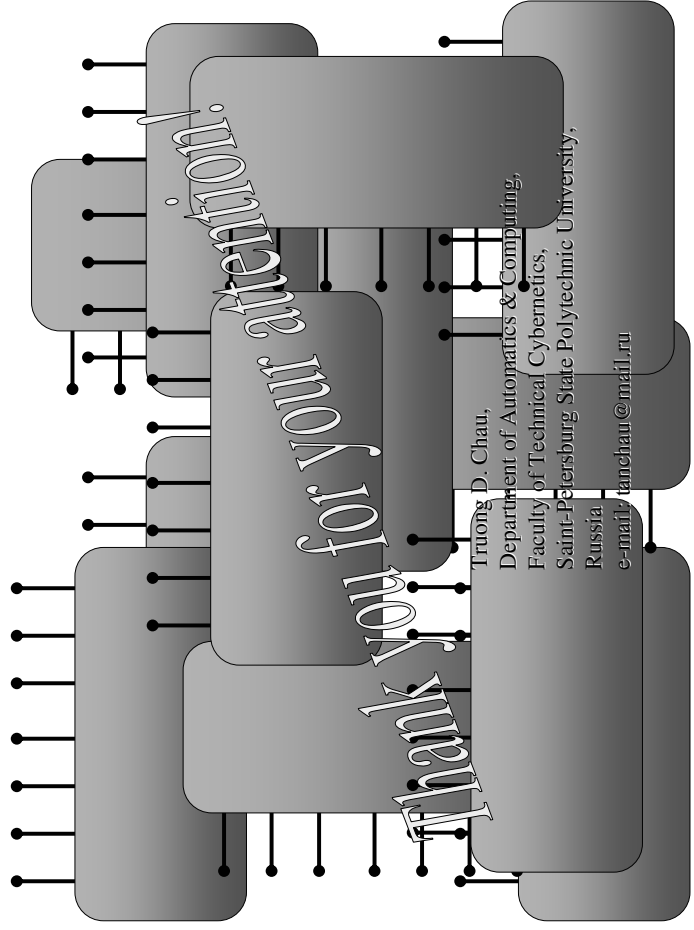
1. Frank Iwanitz, Jürgen Lange. OPC: fundamentals, implementation, and application / [Softing]. Heidelberg: Huthig, 2002. 221 p.
2. František Plášil, Michael Stal. An architecture view of distributed objects and components in CORBA, Java RMI and COM/DCOM // Software – Concepts & Tools. 1998. №19. P. 14-28.
3. Mike Santory. OPC: OLE for Process Control // Real-Time Magazine. 1997. №4. P. 78-81.
4. OPC Alarms and Events Custom Interface Standard Version 1.02. OPC Foundation. Austin, Texas, November 1999.
5. OPC Data Access Custom Interface Standard Version 2.05. OPC Foundation. Austin, Texas, December 2001.
6. OPC Historical Data Access Custom Interface Standard Version 1.1. OPC Foundation. Austin, Texas, January 2001.

MODBUS RS-232 PLC - Simulator (port: COM1.115200.B,N,I)

Connected (1): (received/sent) (794/794) Serv. read data.

Addresses: Hex Dec Bin Ascii Ascii2 AsciiZ AsciiX AsciiY AsciiZ2 AsciiX2 AsciiY2 AsciiZ3 AsciiX3 AsciiY3 AsciiZ4 AsciiX4 AsciiY4 AsciiZ5 AsciiX5 AsciiY5 AsciiZ6 AsciiX6 AsciiY6 AsciiZ7 AsciiX7 AsciiY7 AsciiZ8 AsciiX8 AsciiY8 AsciiZ9 AsciiX9 AsciiY9 AsciiZ10 AsciiX10 AsciiY10 AsciiZ11 AsciiX11 AsciiY11 AsciiZ12 AsciiX12 AsciiY12 AsciiZ13 AsciiX13 AsciiY13 AsciiZ14 AsciiX14 AsciiY14 AsciiZ15 AsciiX15 AsciiY15 AsciiZ16 AsciiX16 AsciiY16 AsciiZ17 AsciiX17 AsciiY17 AsciiZ18 AsciiX18 AsciiY18 AsciiZ19 AsciiX19 AsciiY19 AsciiZ20 AsciiX20 AsciiY20 AsciiZ21 AsciiX21 AsciiY21 AsciiZ22 AsciiX22 AsciiY22 AsciiZ23 AsciiX23 AsciiY23 AsciiZ24 AsciiX24 AsciiY24 AsciiZ25 AsciiX25 AsciiY25 AsciiZ26 AsciiX26 AsciiY26 AsciiZ27 AsciiX27 AsciiY27 AsciiZ28 AsciiX28 AsciiY28 AsciiZ29 AsciiX29 AsciiY29 AsciiZ30 AsciiX30 AsciiY30 AsciiZ31 AsciiX31 AsciiY31 AsciiZ32 AsciiX32 AsciiY32 AsciiZ33 AsciiX33 AsciiY33 AsciiZ34 AsciiX34 AsciiY34 AsciiZ35 AsciiX35 AsciiY35 AsciiZ36 AsciiX36 AsciiY36 AsciiZ37 AsciiX37 AsciiY37 AsciiZ38 AsciiX38 AsciiY38 AsciiZ39 AsciiX39 AsciiY39 AsciiZ40 AsciiX40 AsciiY40 AsciiZ41 AsciiX41 AsciiY41 AsciiZ42 AsciiX42 AsciiY42 AsciiZ43 AsciiX43 AsciiY43 AsciiZ44 AsciiX44 AsciiY44 AsciiZ45 AsciiX45 AsciiY45 AsciiZ46 AsciiX46 AsciiY46 AsciiZ47 AsciiX47 AsciiY47 AsciiZ48 AsciiX48 AsciiY48 AsciiZ49 AsciiX49 AsciiY49 AsciiZ50 AsciiX50 AsciiY50 AsciiZ51 AsciiX51 AsciiY51 AsciiZ52 AsciiX52 AsciiY52 AsciiZ53 AsciiX53 AsciiY53 AsciiZ54 AsciiX54 AsciiY54 AsciiZ55 AsciiX55 AsciiY55 AsciiZ56 AsciiX56 AsciiY56 AsciiZ57 AsciiX57 AsciiY57 AsciiZ58 AsciiX58 AsciiY58 AsciiZ59 AsciiX59 AsciiY59 AsciiZ60 AsciiX60 AsciiY60 AsciiZ61 AsciiX61 AsciiY61 AsciiZ62 AsciiX62 AsciiY62 AsciiZ63 AsciiX63 AsciiY63 AsciiZ64 AsciiX64 AsciiY64 AsciiZ65 AsciiX65 AsciiY65 AsciiZ66 AsciiX66 AsciiY66 AsciiZ67 AsciiX67 AsciiY67 AsciiZ68 AsciiX68 AsciiY68 AsciiZ69 AsciiX69 AsciiY69 AsciiZ70 AsciiX70 AsciiY70 AsciiZ71 AsciiX71 AsciiY71 AsciiZ72 AsciiX72 AsciiY72 AsciiZ73 AsciiX73 AsciiY73 AsciiZ74 AsciiX74 AsciiY74 AsciiZ75 AsciiX75 AsciiY75 AsciiZ76 AsciiX76 AsciiY76 AsciiZ77 AsciiX77 AsciiY77 AsciiZ78 AsciiX78 AsciiY78 AsciiZ79 AsciiX79 AsciiY79 AsciiZ80 AsciiX80 AsciiY80 AsciiZ81 AsciiX81 AsciiY81 AsciiZ82 AsciiX82 AsciiY82 AsciiZ83 AsciiX83 AsciiY83 AsciiZ84 AsciiX84 AsciiY84 AsciiZ85 AsciiX85 AsciiY85 AsciiZ86 AsciiX86 AsciiY86 AsciiZ87 AsciiX87 AsciiY87 AsciiZ88 AsciiX88 AsciiY88 AsciiZ89 AsciiX89 AsciiY89 AsciiZ90 AsciiX90 AsciiY90 AsciiZ91 AsciiX91 AsciiY91 AsciiZ92 AsciiX92 AsciiY92 AsciiZ93 AsciiX93 AsciiY93 AsciiZ94 AsciiX94 AsciiY94 AsciiZ95 AsciiX95 AsciiY95 AsciiZ96 AsciiX96 AsciiY96 AsciiZ97 AsciiX97 AsciiY97 AsciiZ98 AsciiX98 AsciiY98 AsciiZ99 AsciiX99 AsciiY99 AsciiZ100 AsciiX100 AsciiY100

Address	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
40001-40010	134	56	27	18	32	23	545	12	0	0
40011-40020	12	0	0	0	0	0	0	0	0	0
40021-40030	0	0	0	0	0	0	0	0	0	0
40031-40040	0	0	0	0	0	0	0	0	0	0
40041-40050	0	0	0	0	0	0	0	0	0	0
40051-40060	0	0	0	0	0	0	0	0	0	0
40061-40070	0	0	0	0	0	0	0	0	0	0
40071-40080	0	0	0	0	0	0	0	0	0	0
40081-40090	0	0	0	0	0	0	0	0	0	0
40091-40100	0	0	0	0	0	0	0	0	0	0
40101-40110	0	847	0	695	0	825	0	575	3409	3340
40111-40120	2842	0	2369	0	1164	809	1116	0	0	0
40121-40130	0	0	0	0	0	0	0	0	0	0
40131-40140	0	0	0	0	0	0	0	0	0	0
40141-40150	0	0	0	0	0	0	0	0	0	0
40151-40160	0	0	0	0	0	0	0	0	0	0
40161-40170	0	0	0	0	0	0	0	0	0	0
40171-40180	0	0	0	0	0	0	0	0	0	0
40181-40190	0	0	0	0	0	0	0	0	0	0
40191-40200	0	0	0	0	0	0	0	0	0	0



DDE (Dynamic Data Exchange): là giao thức kết nối thiết kế bởi Microsoft cho phép những ứng dụng trong môi trường Windows gửi hay nhận dữ liệu qua lại. Nó bổ sung một mối quan hệ server – client giữa hai ứng dụng đang chạy. Ứng dụng server cung cấp dữ liệu và chấp nhận những yêu cầu của những ứng dụng khác (client) .

FastDDE: cung cấp một phương tiện cho việc gom tất cả những DDE message của Wonderware vào một DDE message duy nhất của Microsoft. Việc gom những message này phát huy được tính hiệu quả và hiệu suất cao bằng việc giảm số lượng tổng cộng của những trao đổi DDE được yêu cầu giữa một server và một client.

NetDDE: mở rộng chuẩn Windows DDE để có thể hoạt động qua mạng nội bộ hoặc thông qua các cổng nối tiếp. NetDDE phải được cài trên tất cả các máy tính trong mạng mà dữ liệu DDE cần trao đổi.

SuiteLink : là giao thức dựa trên nền tảng TCP/IP. SuiteLink được thiết kế để đáp ứng những nhu cầu trong công nghiệp, như là việc bảo quản dữ liệu, tốc độ truyền nhanh và dễ kiểm tra sự cố truyền dữ liệu. SuiteLink không phải là sự thay thế cho DDE hay NetDDE. Mỗi kết nối giữa client và server dựa vào tình trạng của mạng. SuiteLink được thiết kế đặc biệt cho những ứng dụng tốc độ cao và cung cấp những đặc tính sau :

- Những loại dữ liệu phù hợp có thể được giữ giữa những ứng dụng bất chấp những ứng dụng này đang chạy trên một máy tính hay trên nhiều máy trong một mạng.

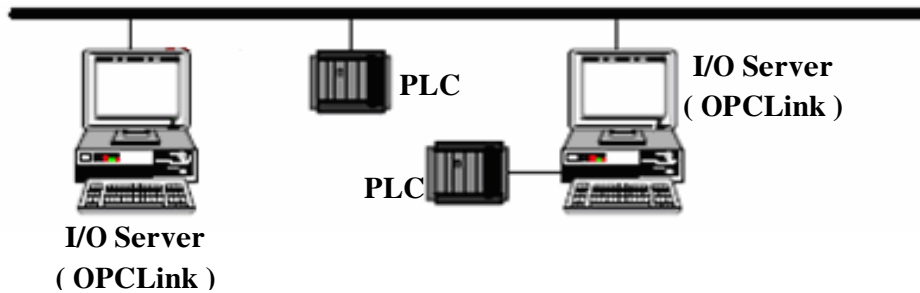
- Giao thức truyền trong mạng là TCP/IP sử dụng chuẩn giao tiếp của Microsoft là Winsock.

OPC: Những bổ sung của OPC cho Wonderware FactorySuite bao gồm hai phần:

- Wonderware OPCLink Server: là Server giao tiếp giữa những ứng dụng SuiteLink/DDE với OPC Server theo chuẩn OPC.

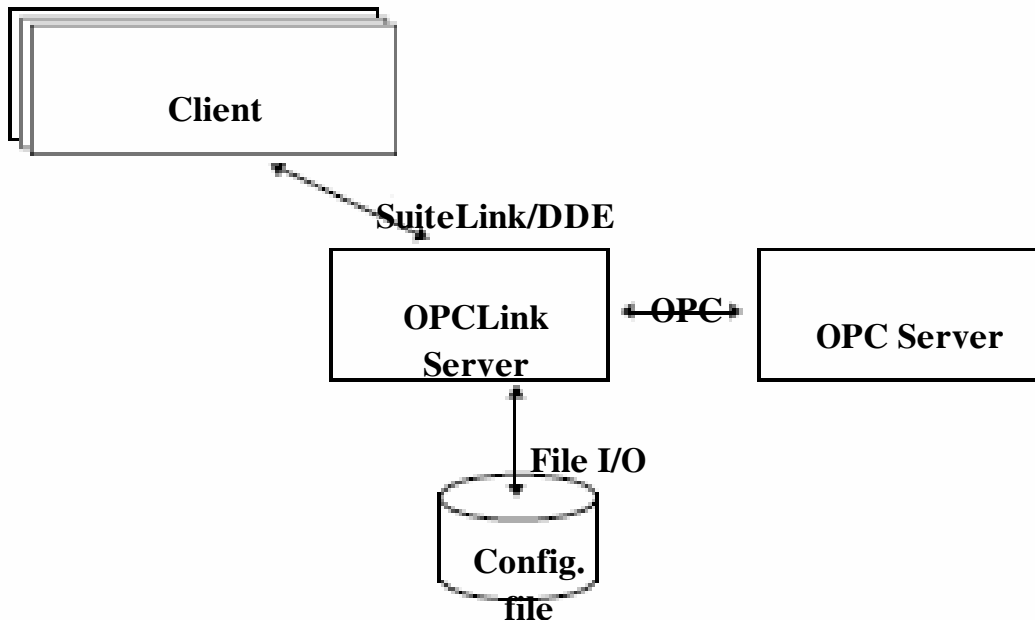
- Wonderware OPC Browse : được sử dụng bởi WindowMaker để quan sát những OPC Server đã được đăng ký trong một mạng và có thể tự động tạo ra những InTouch tag kết nối đến những OPC item. Để OPC Browse thấy được một OPC Server thì OPC Server đó phải được đăng ký ở trên máy đó.

OPCLink: Wonderware cung cấp khá nhiều các I/O Server của các hãng sản xuất thiết bị trên thế giới. Một trong những I/O Server phổ biến nhất là OPCLink. OPCLink cho ta liên kết các ứng dụng trong Wonderware với các thiết bị phân cứng. Tuy nhiên để có thể liên kết được ta cần phải có một OPC Server của thiết bị đó.



Mô hình kết nối giữa I/O Server và PLC.

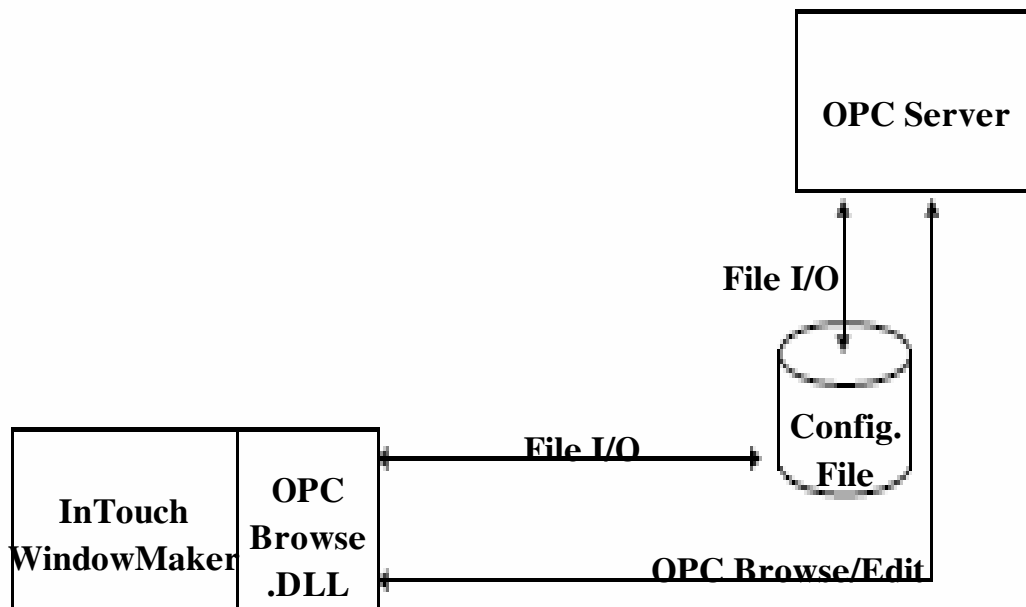
OPCLink Server: kết nối tất cả những sản phẩm của Wonderware với OPC. Nó có thể kết nối cục bộ trong một máy hay từ xa qua mạng tới bất kỳ OPC Server nào và truyền dữ liệu thông qua giao thức DDE, và SuiteLink tới bất kỳ client nào. OPCLink Server cho phép quan sát không gian địa chỉ của OPC Server. OPCLink Server có một Topic Editor cho phép thiết lập các thông số kết nối từ những thành phần của Wonderware đến thiết bị phân cứng. Sau đây là đồ thị kết nối của OPCLink Server :



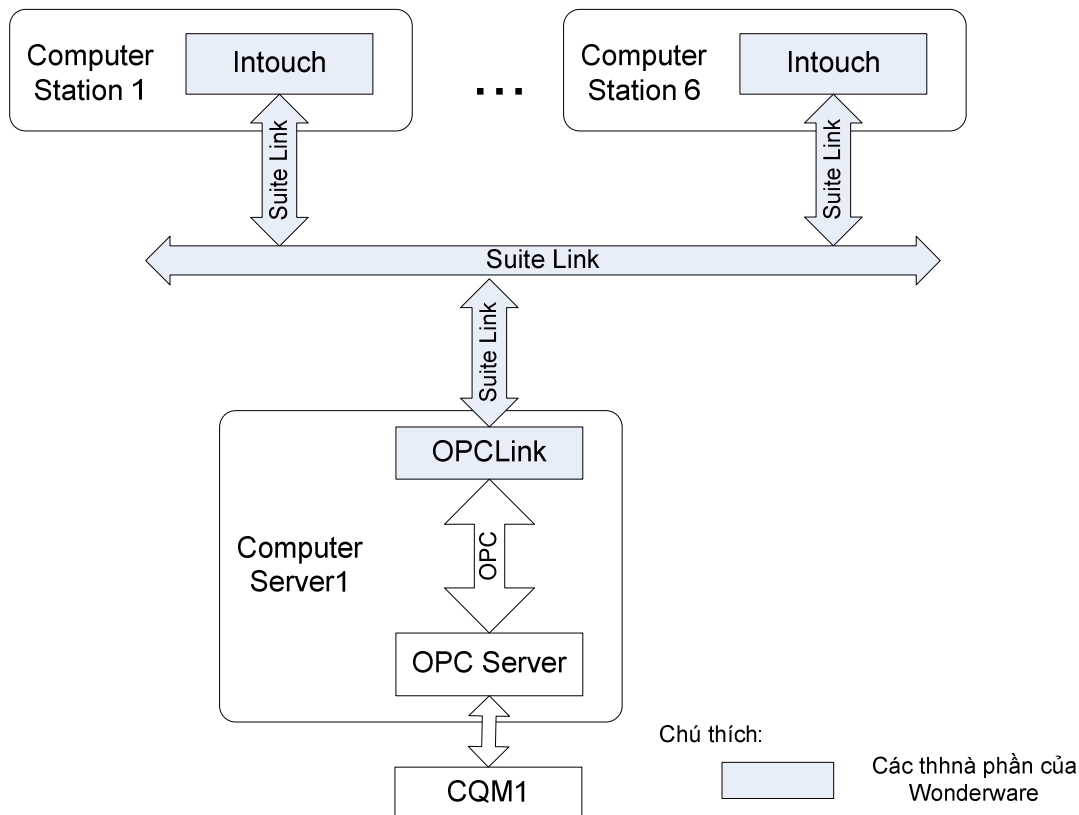
Hình 2.2 Giao thức kết nối giữa OPCLink với OPC Server và Client

OPC Browse: là một giao tiếp của WindowMaker cho phép :

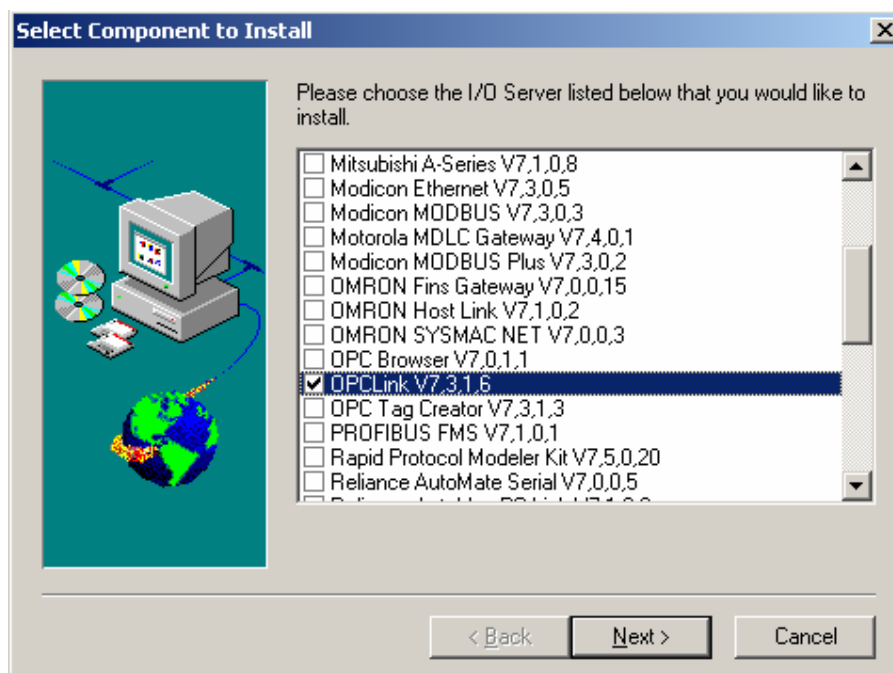
- + Chọn một OPC Server trong cùng một máy hay từ máy khác trong mạng.
- + Quan sát không gian địa chỉ của những OPC Server bao gồm các OPC item.
- + Tạo tag và access name.
- + Cấu hình các OPC Server trong 1 máy hay từ máy khác trong một mạng.
- Dưới đây là mô hình kết nối mạng cho việc sử dụng trong InTouch:



Mô hình kết nối giữa InTouch và OPC Server thông qua OPCLink.

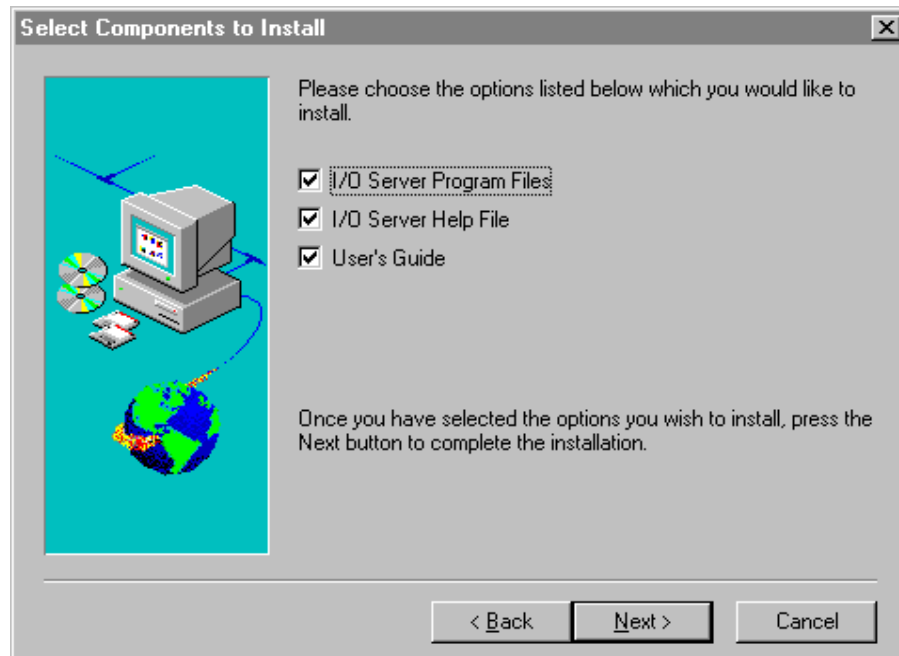


Cài đặt và cấu hình OPCLink: Trước tiên ta cài đặt OPCLink vào máy tính:



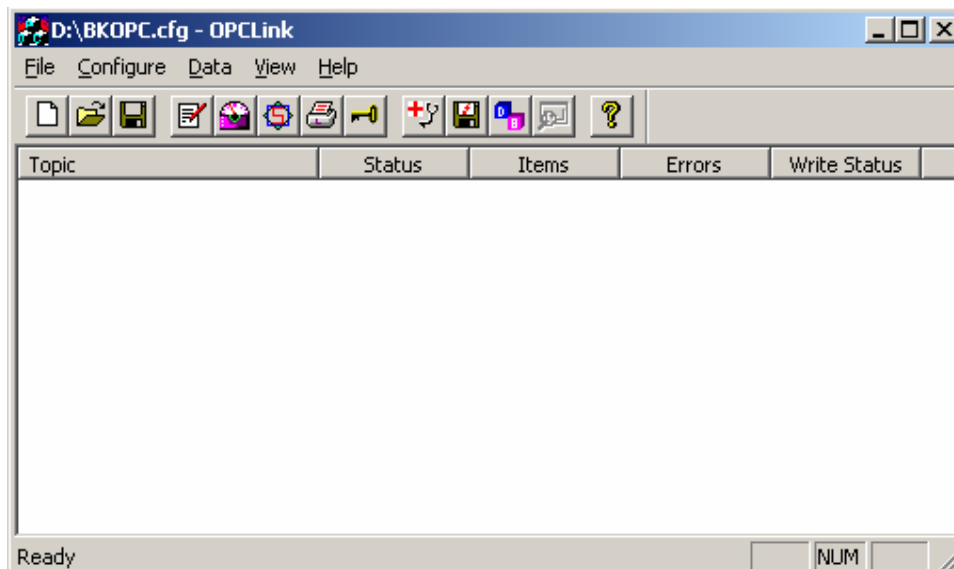
Cửa sổ chọn lựa I/O Server cần cài đặt.

- Chọn OPCLink, chọn Next:



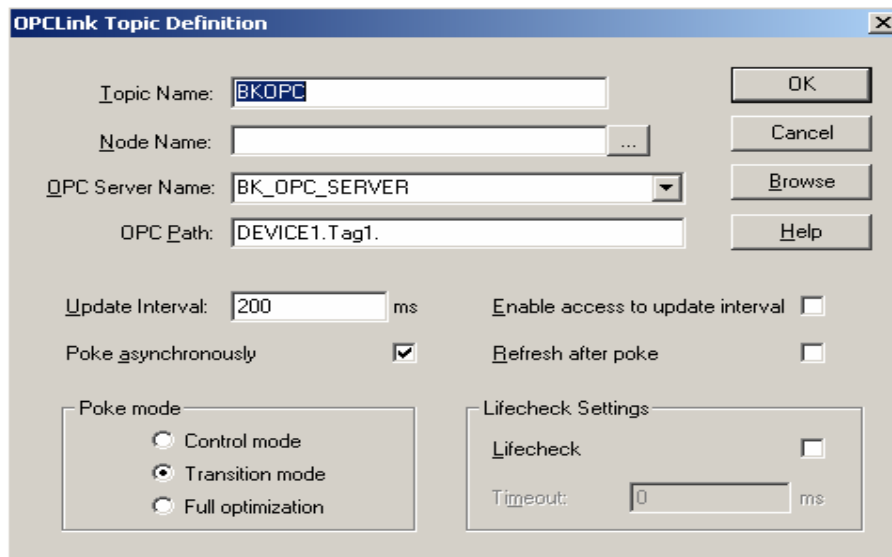
Cửa sổ cho phép chọn các thành phần cài đặt của I/O Server

- Tiếp tục các bước cài đặt cho đến khi hoàn tất quá trình cài đặt. Sau khi cài xong OPCLink ta vào cấu hình cho OPCLink.



Cửa sổ chính cho phép cấu hình OPCLink kết nối đến OPC Server

- Chọn Configure/Topic Definition/New.

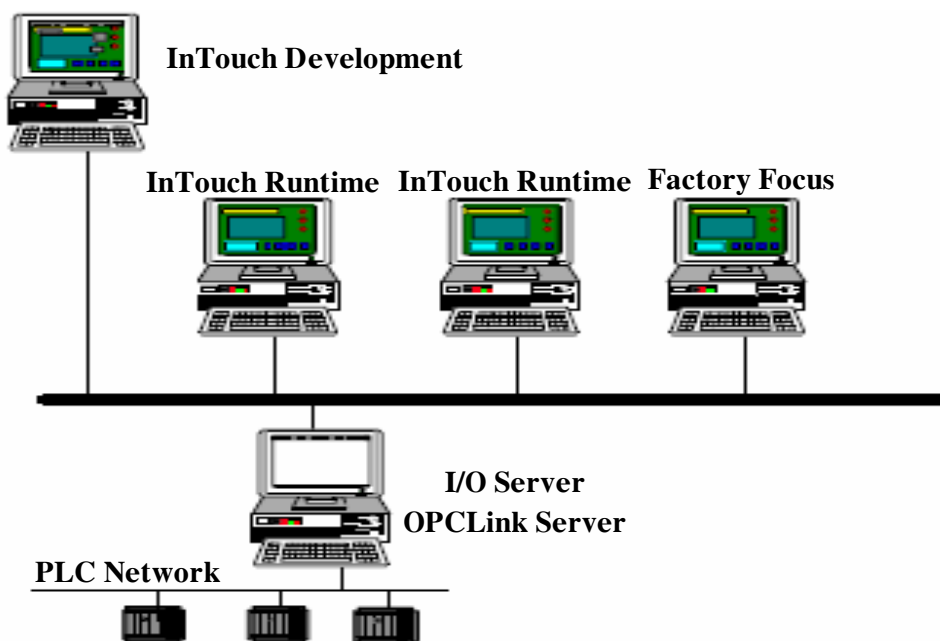


Cửa sổ nhập các thông số cần thiết kết nối đến OPC Server.

- Topic Name: là tên cấu hình cho OPCLink.
- Node Name: là tên của máy tính chạy OPC Server.
- OPC Server Name : là tên của OPC Server mà OPCLink sẽ kết nối đến.
- OPC Path : là đường dẫn đến những tag của thiết bị nằm trong OPC Server.

Để giám sát một dây chuyền sản xuất thì việc tạo một giao diện HMI mô tả hoạt động của một dây chuyền trên máy tính sẽ tạo điều kiện thuận lợi cho người vận hành hệ thống có thể quan sát quá trình thực thi một các trực quan và linh động hơn. Wonderware cung cấp một công cụ rất mạnh để xây dựng giao diện HMI đó chính là InTouch.

- Sau đây là mô hình kết nối của Intouch trong hệ thống: [1]



- Ta thấy InTouch sẽ liên kết với thiết bị phần cứng qua I/O Server. Cụ thể là OPCLink Server. Trong đó InTouch sẽ phân ra từng phần riêng biệt, mỗi phần sẽ thực hiện một công việc nhất định.

Các thành phần trong InTouch: Công dụng của InTouch là cho phép tạo ra những giao diện HMI. Với InTouch người sử dụng có thể tạo ra những ứng dụng mạnh khai thác được những đặc tính của hệ điều hành Windows bao gồm ActiveX Control, OLE, hình ảnh, hệ thống mạng và còn nhiều hơn nữa. InTouch bao gồm 3 phần chính là Application Manager, WindowMaker và WindowViewer.

Application Manager: cho phép quản lý tất cả những ứng dụng đã tạo ra.

WindowMaker: là môi trường phát triển ứng dụng. Cho phép tạo ra những object với hình ảnh của các thiết bị cũng như dây chuyền. Những object này có thể liên với các thiết bị phần cứng thông qua I/O Server và liên kết với các ứng dụng Window khác.

WindowViewer : là môi trường thực thi, dùng để hiển thị hoạt động thiết bị hay dây chuyền sản xuất mà ta đã tạo trong WindowMaker.

Xây dựng giao diện HMI bằng InTouch