Industrial Automation
MKT4152

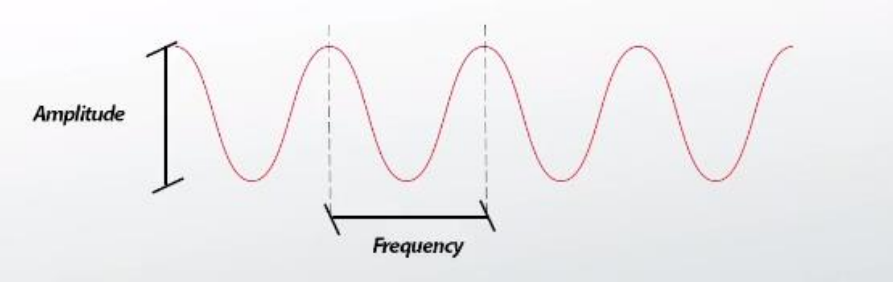
Analog vs. Digital Signal Communication

Analog Signal:

Continuous wave denoted by a sine wave with varying signal strength (Amplitude) and varying frequency

Theoretically infinite resolution

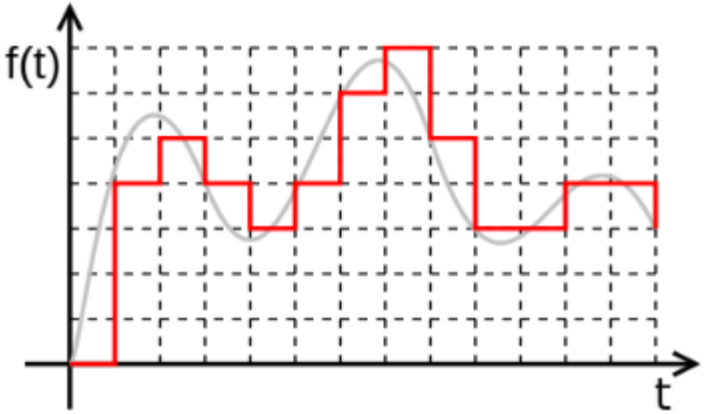
Subjected to electronic noise and distortion (information degradation)

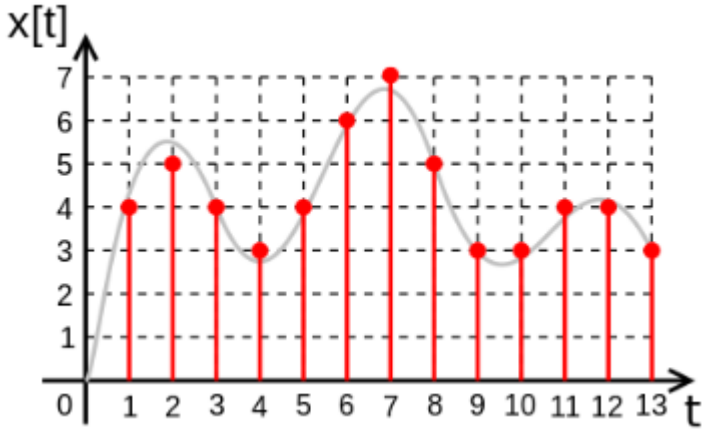


Digital Signal:

Discrete, finite values of amplitude (quantization)

Not continuous: discrete sample time





Comparison of Analog and Digital Communication

Analog Signal Transmission

Advantages

- Infinite resolution
- continuous

Disadvantages

- Noise
- Distortion
- Processing is difficult
- Transmission is limited

Digital Signal Transmission

Advantages

- Easy to transmit over greater distances, with no losses
- No noise from signal transmission
- Error checking possible
- Easy to process (example: multiplication)

Disadvantages

- Quantization of amplitude
- Discrete sampling/discrete time signal
- More complex

Digital Communication & Distributed Systems in Automation

FIELDBUS SYSTEMS

Automation

PLC's:

- digital I/O and analog I/O
- Digital signal processing
- Large input/output systems
- Process automation
  - Measurement
  - Open-loop control
  - Closed loop control

Process Automation Example: BASF Ludwigshafen / Germany

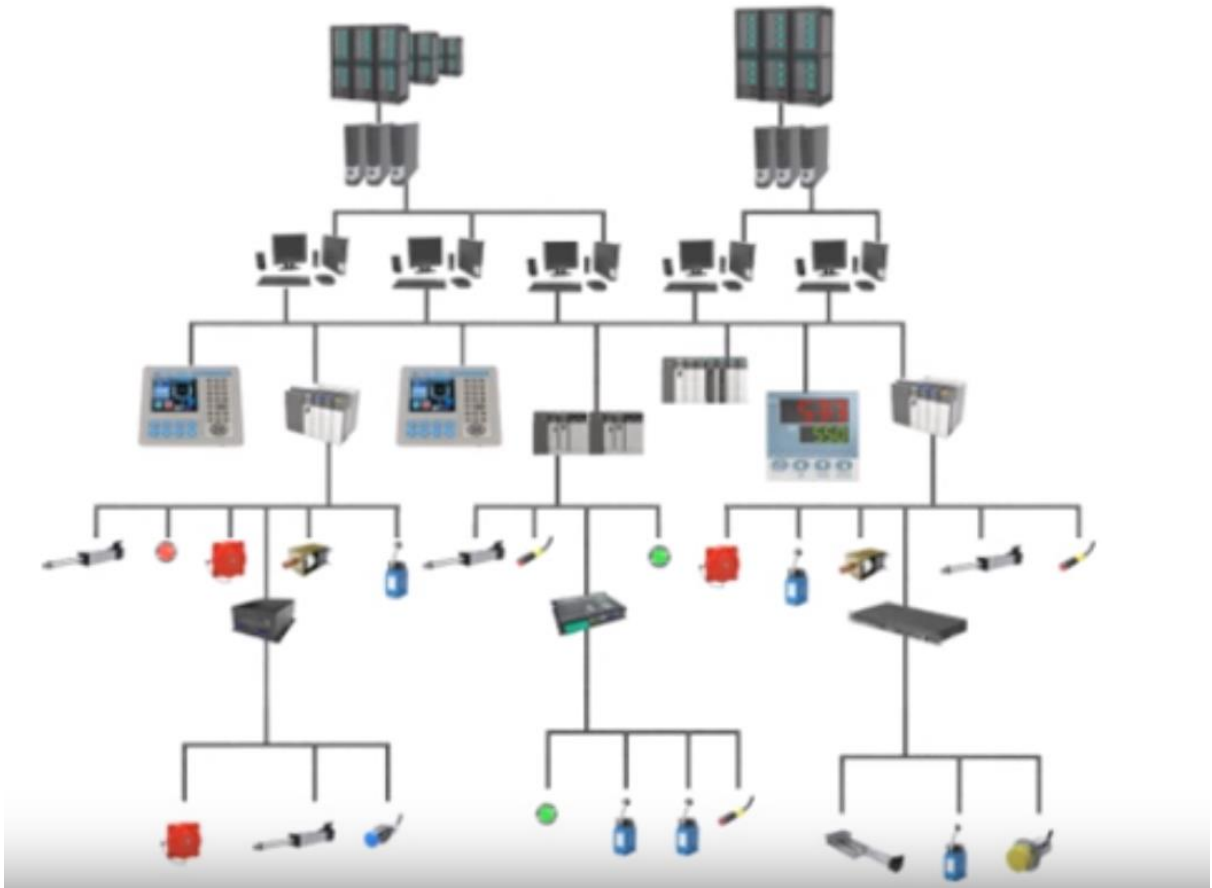
Fieldbus Systems

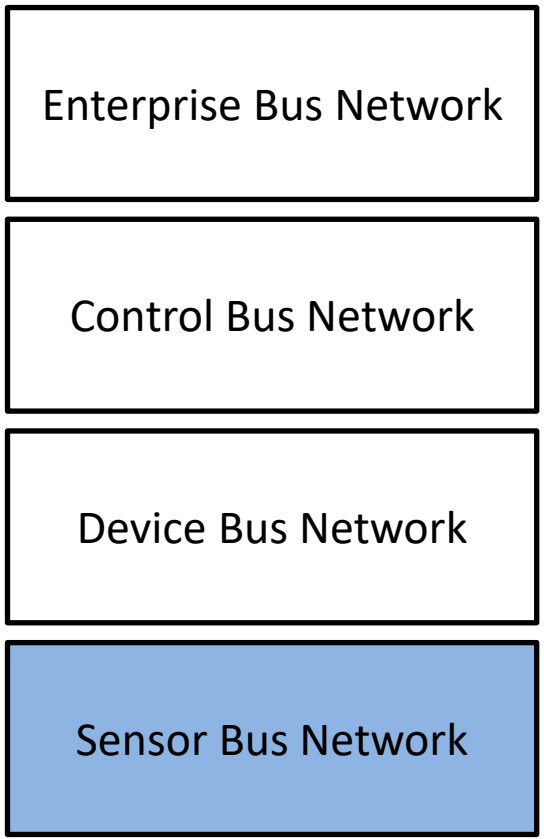
Connection of computers, PLC's, …, instruments, sensors to a single network in a manufacturing plant.

Fieldbus systems allow real-time control, monitoring, design of distributed systems.

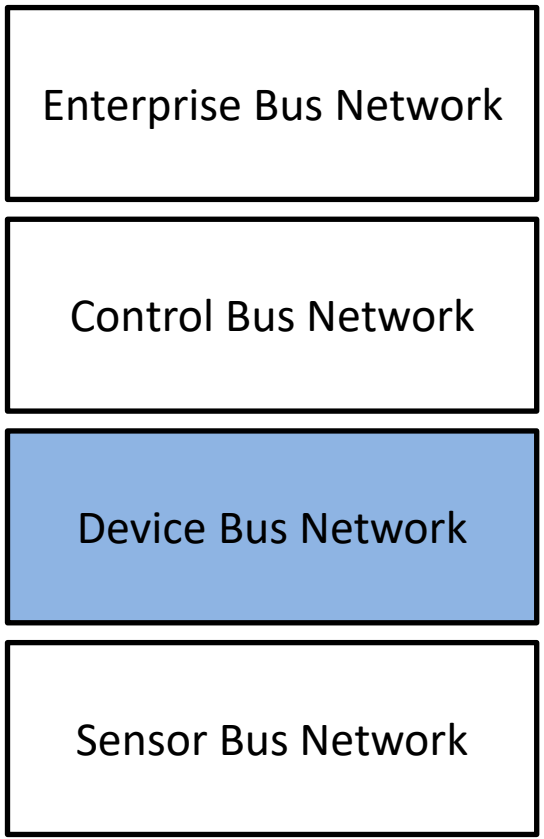
Fieldbus Systems can be broken down to four different levels of complexity:

- Enterprise Bus Network
- Control Bus Network
- Device Bus Network
- Sensor Bus Network





- Least complex networks
- Developed for industrial application
- Multiple basic field devices, such as limit switches, or level optical sensors are connected through one cable to a **controller**. Field devices can also be output devices, such as indicator lamps, alarms, or actuator devices
- Range: short
- Examples:
  - I2C
  - RS232
  - SPI
  - 1-Wire



- Similar to device bus networks
- Works on larger scale, connecting many sensors and actuators together
- Examples:
  - Modbus
  - Profibus
  - CAN
  - CANopen
  - Ethercat

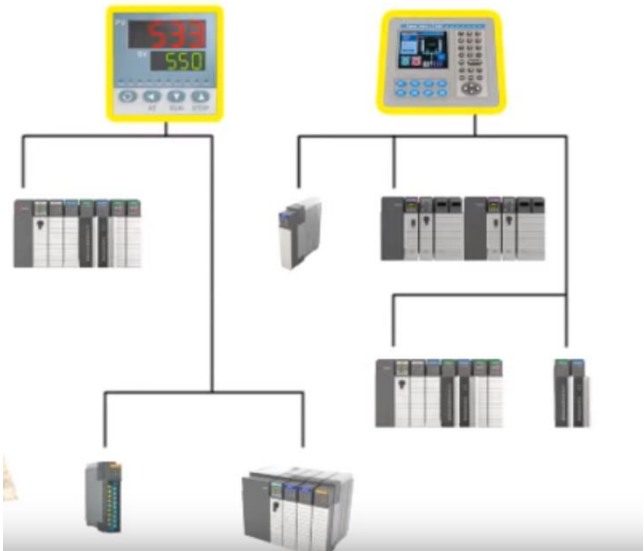
- High level data communication
- PLC are connected to each other
- Connection of PLC’s and HMI devices
  HMI: Human Machine Interface
- Configuration and control of instruments and devices on a network

Enterprise Bus Network

Control Bus Network

Device Bus Network

Sensor Bus Network



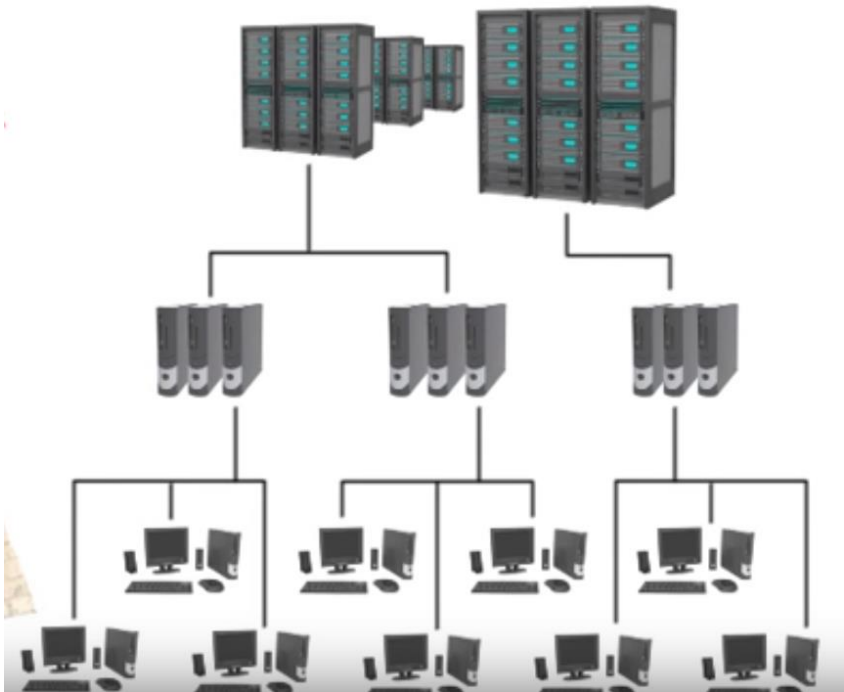
- Information Level , connects computers and departments
- Highest level of complexity
- Mostly computer driven
- **Data collection + data monitoring**
- **Higher level process control on large scale** (example: BASF)
- Based on LAN (Local Aera Networks) using TCP/IP Protocols.
- Supervisory Control and Data Acquisition (SCADA)

Enterprise Bus Network

Control Bus Network

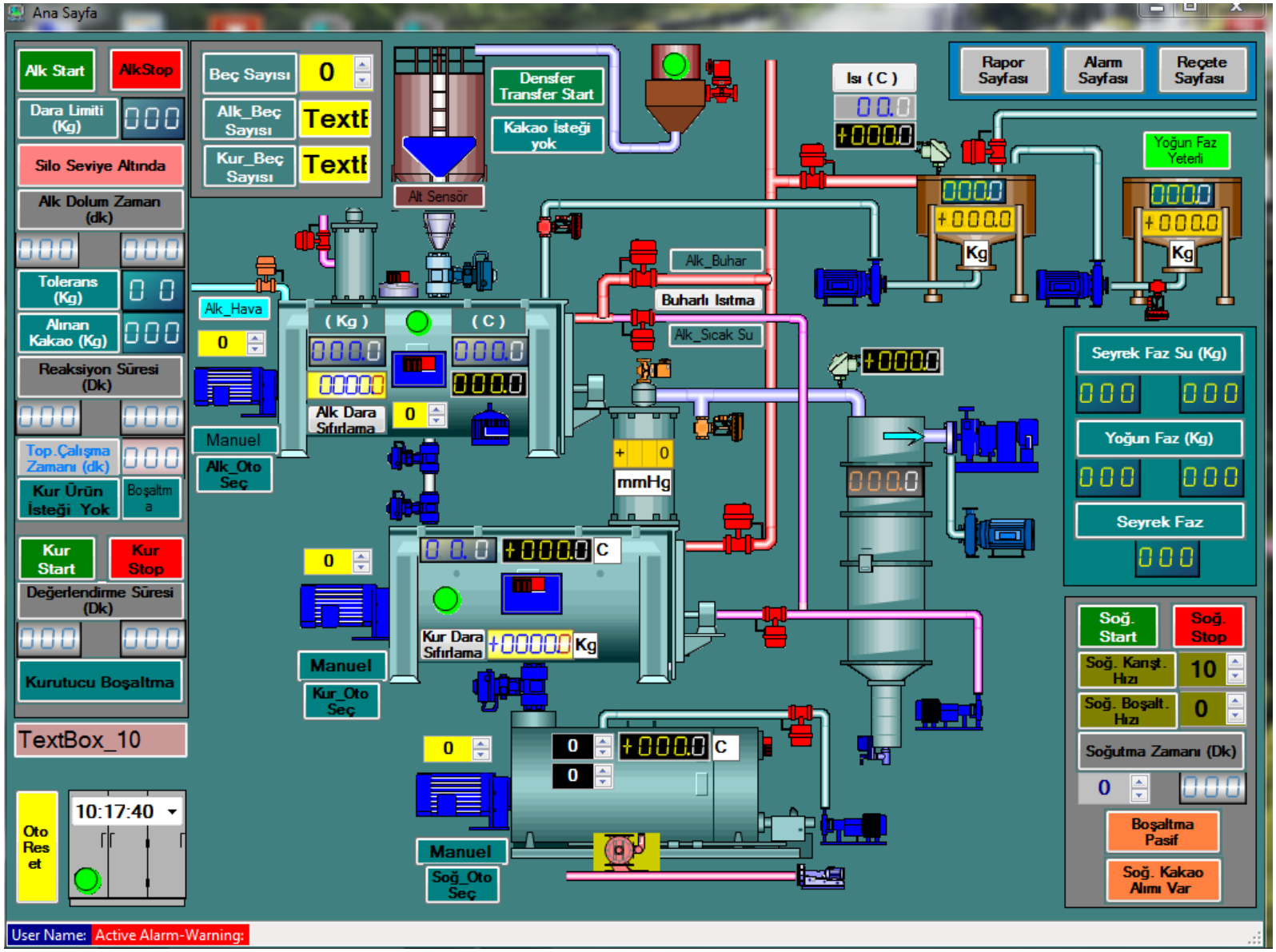
Device Bus Network

Sensor Bus Network

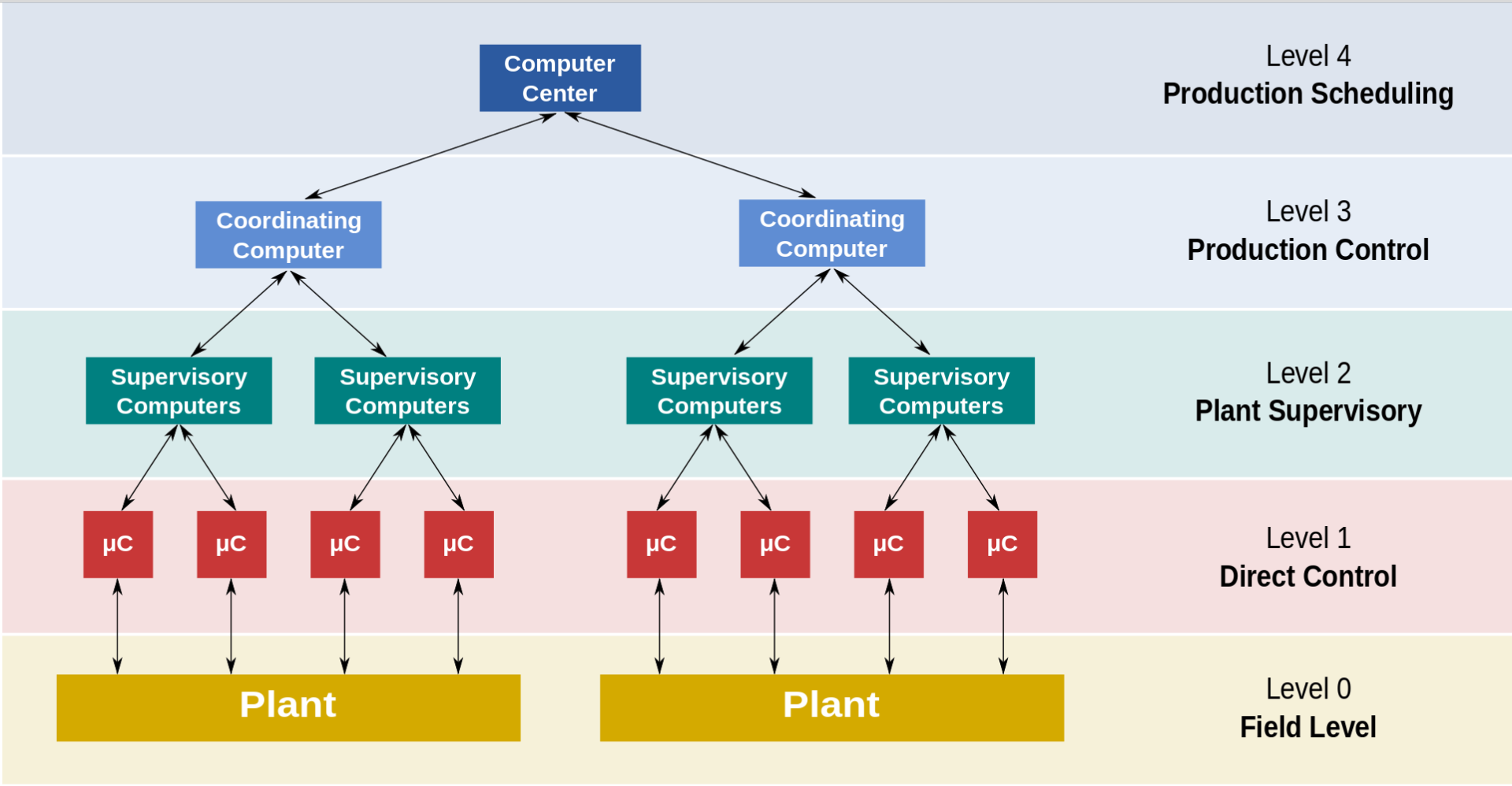


SCADA – Supervisory Control And Data Acquisition

SCADA – Supervisory Control And Data Acquisition



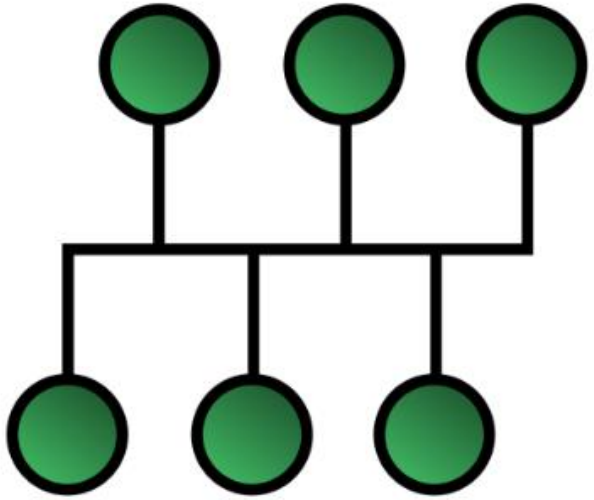
SCADA in production systems



BUS – what is a BUS?

- BUS-Network:
  - Multiple nodes are connected through a common digital communication medium
  - Every node will receive all network traffic

- Communication is controlled centrally by a Bus-Master or defined within the communication protocol.
  Examples:
  EtherCat: single Master + multiple slaves
  CAN: decentralized, multi-master-multi-slave network. Each node is master and slave simultaneously.



Bus network with linear topology

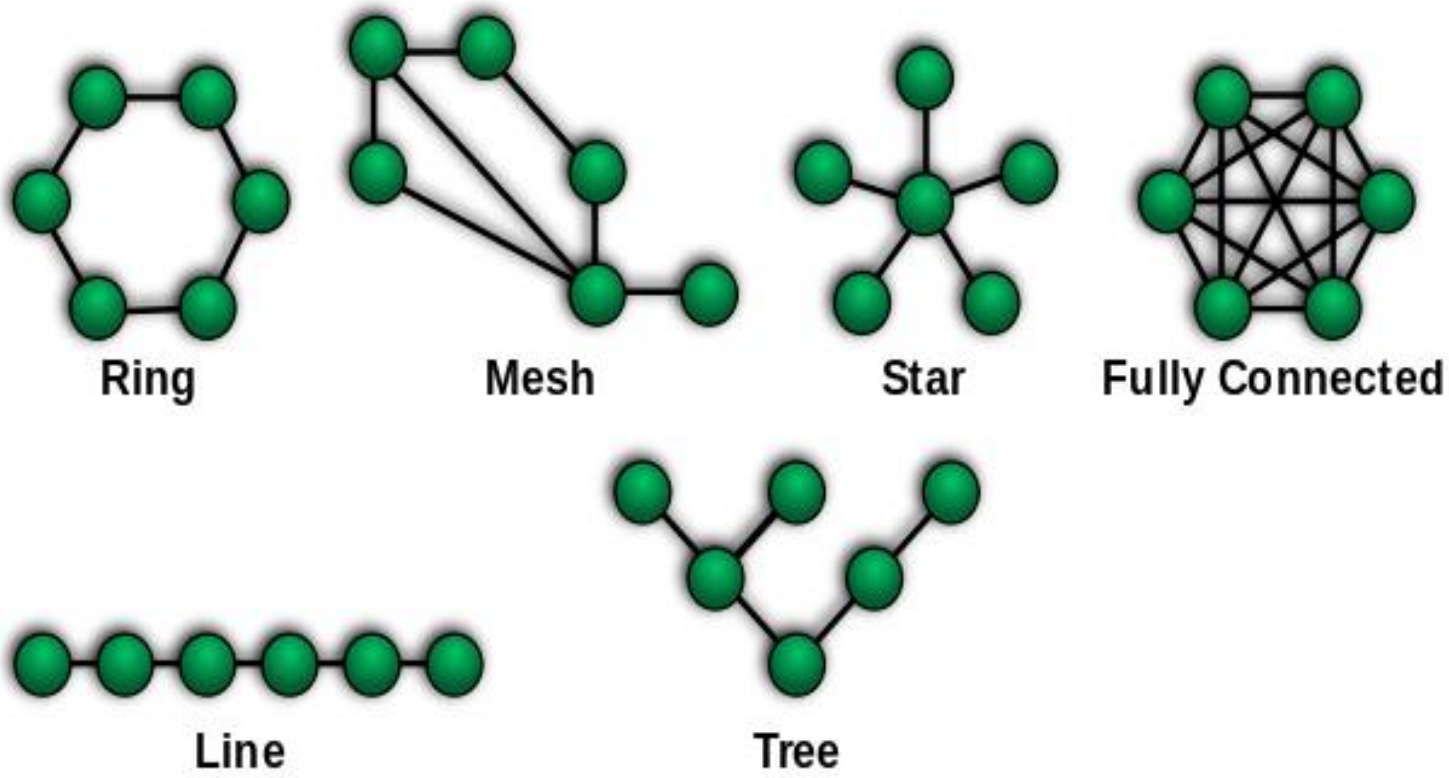
Advantages

- Very easy to connect a computer or peripheral to a linear bus
- Requires less cable length than a star topology resulting in lower costs
- It works well for small networks
- It is easy to extend by joining cable with connector or repeater
- If one node fails, it will not have effect on the whole network

Disadvantages

- Entire network shuts down if there is a break in the main cable or one of the T connectors break
- Large amount of packet collisions on the network, which results in high amounts of packet loss
- Slow with many nodes in the network

Network Topologies



7 Layer OSI Reference Model

- reference model for how applications communicate over a network
- Describes a concept framework
- Purpose: guide vendors and developers to create products and software with better interoperability
- OSI model defines 7 abstraction layers. Each layer serves the layer above it, and depends on the layer below.

7 Layer OSI Reference Model

Layer		*Protocol data unit (PDU)*	*Function*
Host layers	7. Application	Data	High-level APIs, including resource sharing, remote file access. *Examples: FTP, SMTP, Telnet*
	6. Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5. Session		Managing communication sessions, i.e. continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4. Transport	Segment (TCP) Datagram (UDP)	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media layers	3. Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2. Data link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1. Physical	Bit	Transmission and reception of raw bit streams over a physical medium

Common Fieldbus Examples

Bus Type	Bandwidth	Hardware Layer
CAN-Bus	1Mbit/s max	1-wire or 2-wire
CANopen		2-wire
J1939		2-wire
EtherCAT	100Mbit/s	Ethernet
Ethernet/IP	100Mbit/s	Ethernet
Modbus RTU	Typical: 9600Bit/s or 19200Bit/s	Serial communication RS485 RS232
Modbus TCP/IP		Ethernet
Profibus		
Profinet IO		

Fieldbus Protocols

CAN-BUS

CAN – Controller Area Network

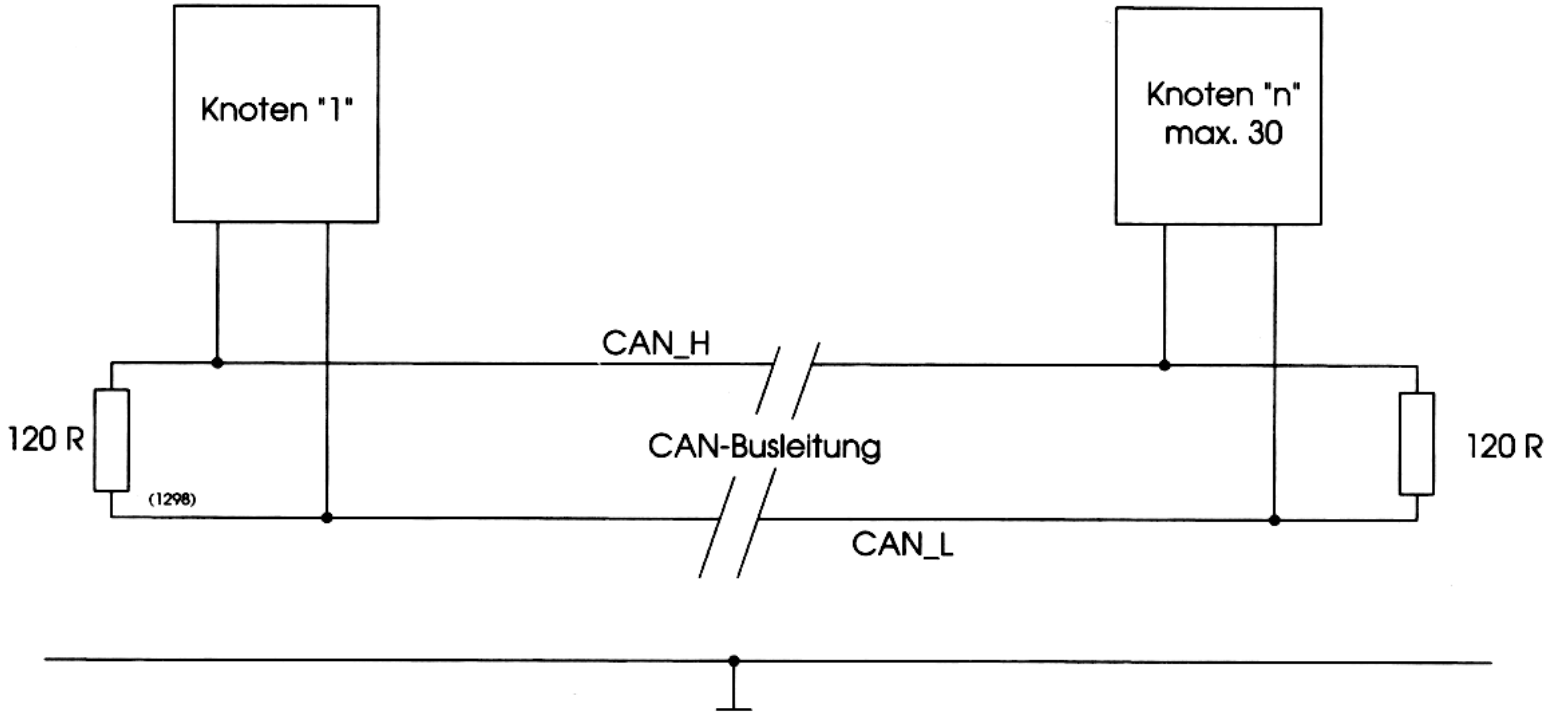
- Developed by Bosch. Latest specification is CAN 2.0 published in 1991
- Purpose:
  Allow microcontrollers and devices in vehicles to communicate with each other, without a host computer
- Properties:
  - Robust
  - Flexible
  - Cost-efficient
- Application in vehicles: Communication between different electronic control units, such as:
  - Engine Control Unit
  - Transmission Control
  - Airbags
  - Power Windows
  - ….

CAN - Bus

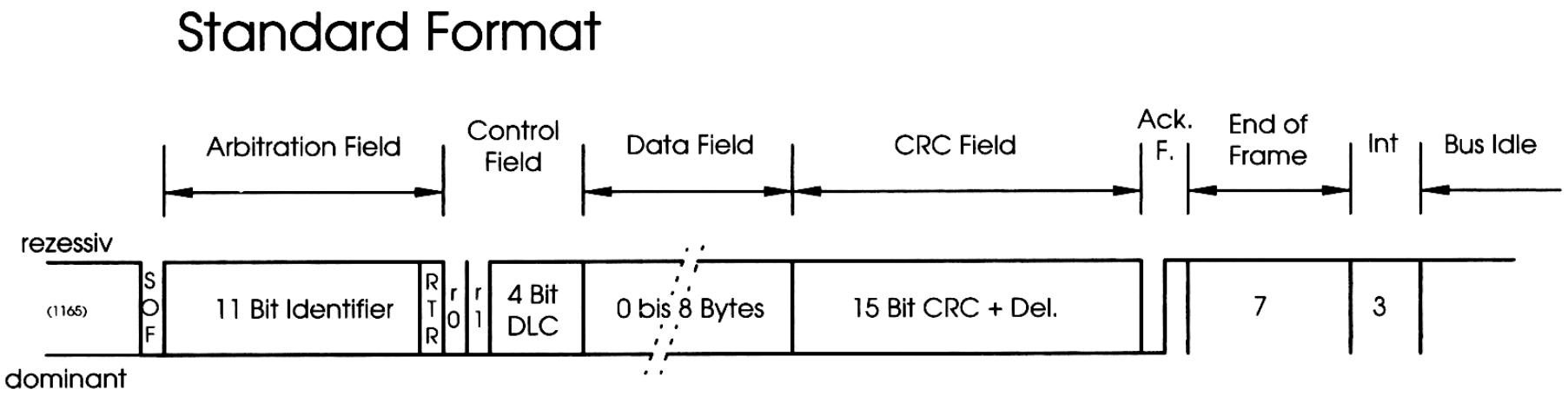
- Multi-Master: No master node/controller required. A minimum of two CAN controllers form a Bus. Collisions of data frames are solved without damaging frames.
- Physical Implementations:
  - **High-Speed CAN:**
    2-wires, CAN_HIGH and CAN_LOW. Voltage between both defines logical states “True” or “False”.
    Bandwidth: user-defined in the range of up to 1MBit/s. Higher bandwidth results in shorter possible cable length.
    Examples:
    - 1MBit/s -> 40m
    - 500kBit/s -> 100m
    - 125kBit/s -> 500m
  - **Low-Speed CAN:**
    1 wire or 2 wires. 125kBit/s max. bandwidth.

Bus Topology

- Linear Bus Topology
- 120Ohm resistors on both ends of bus network

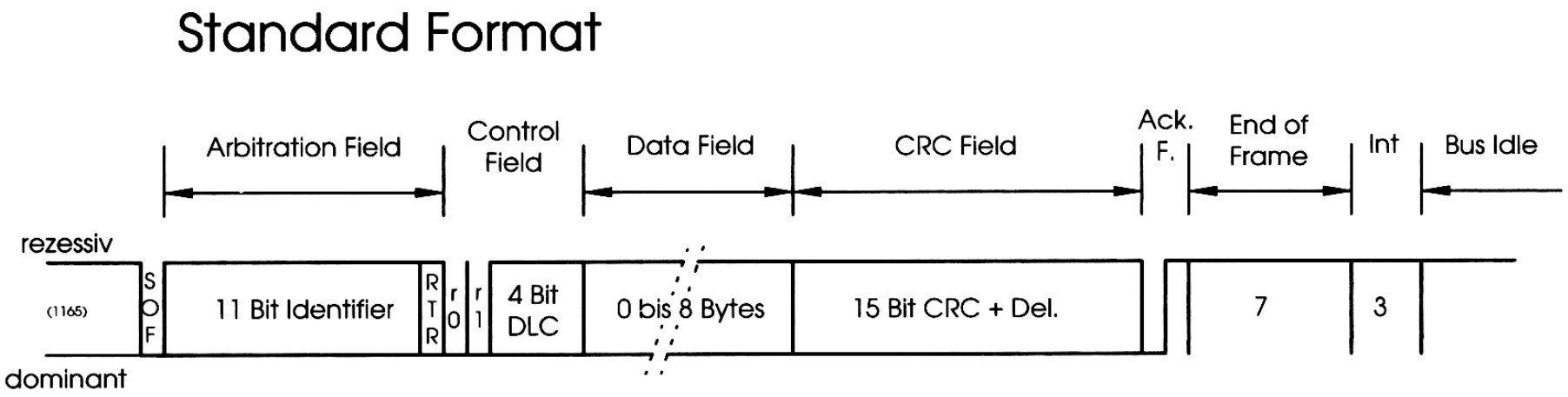


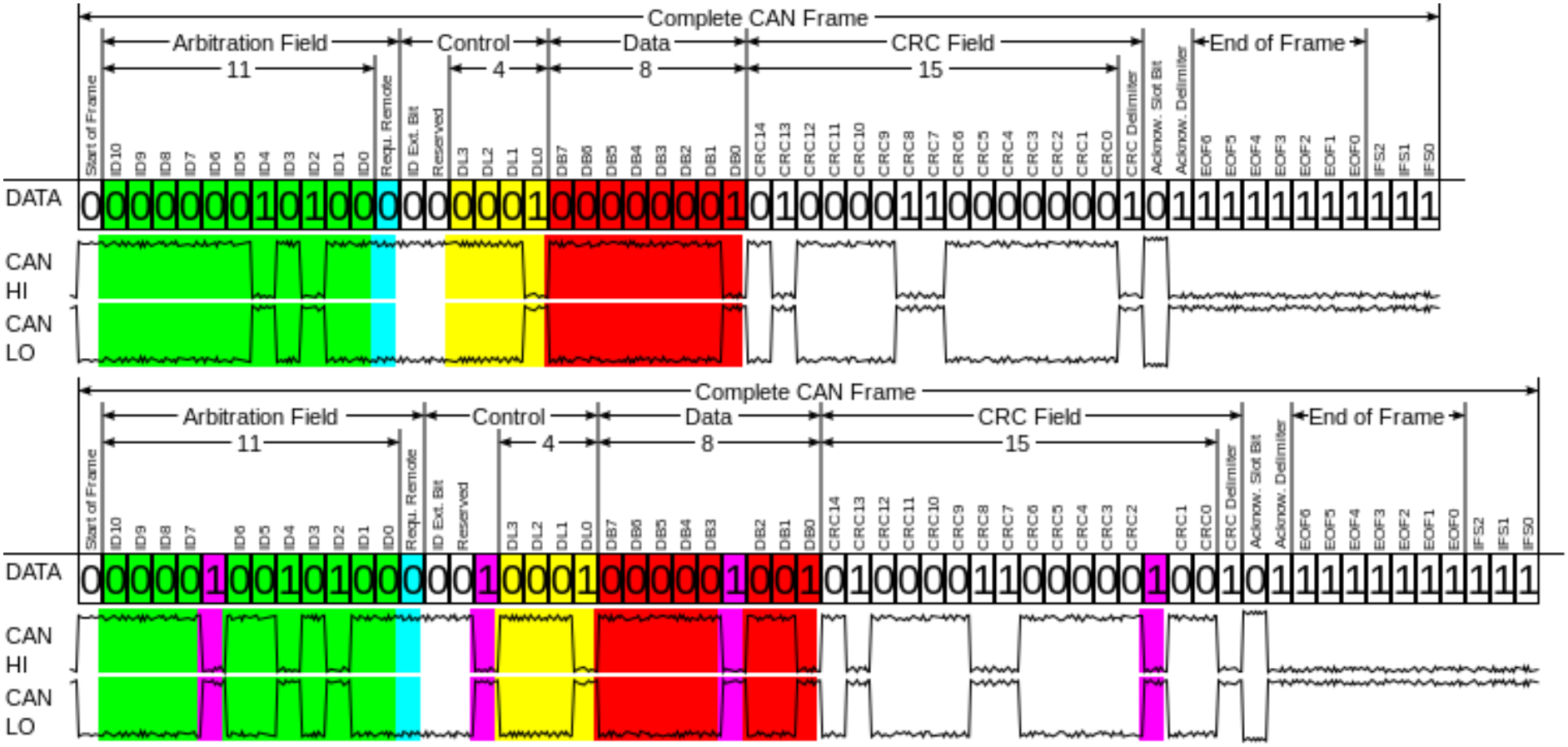
- Packages of defined specification transmit information. These “packages” are called **frames**.
- Each frame has two sets of information:
  - Identifier (11Bit standard or 29Bit extended identifier)
  - Data field (8 Bytes, or 64Bit)
- Total length of frame: 100Bit
- Time required for transmission with standard identifier, with 1Mbit/s:
  1Bit = 1µs, transmission time: 100µs



Detailed transmission of a single frame

- Bus levels are called dominant and recessive. Recessive is of lower priority. Any node can therefore set the bus state to dominant at any time.
- SOF- Start of frame. A node sets bus on “dominant”. All nodes synchronize on the level and now wait for communication
- Sending node starts to send identifier, followed by the control field. All nodes now know, which frame with which content is going to be sent
- Data field (64 Bits) is transmitted
- CRC field is transmitted.
  CRC is a checksum, generated from the data field. Each receiving node generates the equivalent crc value from received data.
- When transmitted CRC and node-calculated CRC are equal, the bus state will fall back to “recessive”. Otherwise, this frame will be marked with an error, and the sender will know that transmission failed, and will resend the frame.
- End of frame: when bus state is “recessive” for at least 7 bit, the end of the message has been reached.




Complete CAN Frame
Arbitration Field
11
Control
4
Data
8
CRC Field
15
End of Frame
Start of Frame
ID10 ID9 ID8 ID7 ID6 ID5 ID4 ID3 ID2 ID1 ID0
Requ. Remote
ID Ext. Bit
Reserved
DL3 DL2 DL1 DL0
DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
CRC14 CRC13 CRC12 CRC11 CRC10 CRC9 CRC8 CRC7 CRC6 CRC5 CRC4 CRC3 CRC2 CRC1 CRC0
CRC Delimiter
Acknow. Slot Bit
Acknow. Delimiter
EOF6 EOF5 EOF4 EOF3 EOF2 EOF1 EOF0
IFS2 IFS1 IFS0
DATA
CAN HI
CAN LO
Complete CAN Frame
Arbitration Field
11
Control
4
Data
8
CRC Field
15
End of Frame
Start of Frame
ID10 ID9 ID8 ID7 ID6 ID5 ID4 ID3 ID2 ID1 ID0
Requ. Remote
ID Ext. Bit
Reserved
DL3 DL2 DL1 DL0
DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0
CRC14 CRC13 CRC12 CRC11 CRC10 CRC9 CRC8 CRC7 CRC6 CRC5 CRC4 CRC3 CRC2 CRC1 CRC0
CRC Delimiter
Acknow. Slot Bit
Acknow. Delimiter
EOF6 EOF5 EOF4 EOF3 EOF2 EOF1 EOF0
IFS2 IFS1 IFS0
DATA
CAN HI
CAN LO

CAN- frame types

- **DATA FRAME**
  transmits data (maximum 64bit)

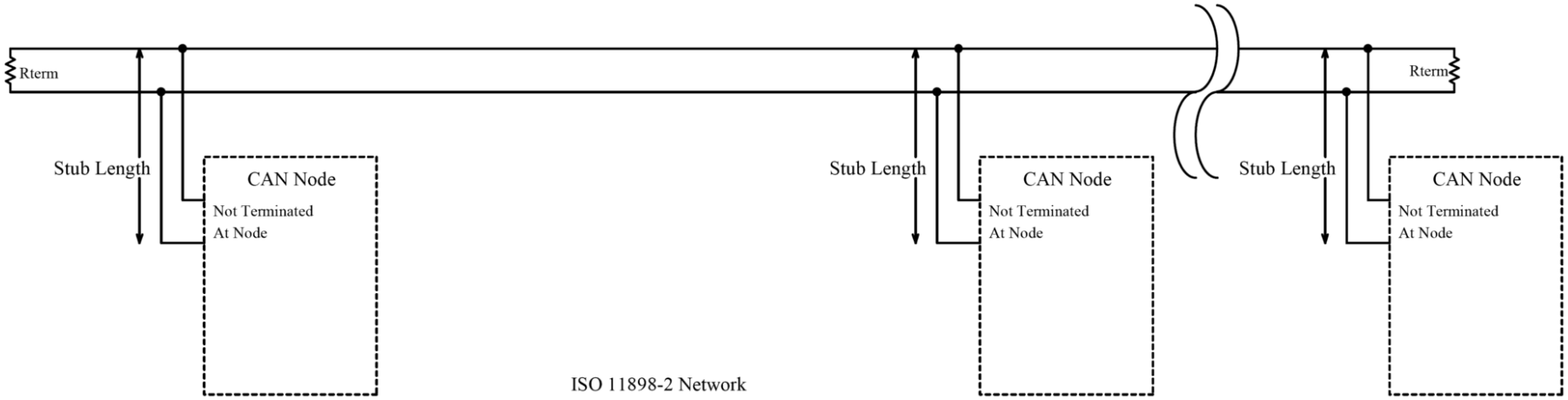
- **REMOTE FRAME**
  sends a request to all bus nodes, requesting to send a frame with specific identifier

- ERROR FRAME
  any node can send an error frame, when a frame was not transmitted correctly. This will cause the respective node to resend the frame.

- OVERLOAD FRAME
  is send when bus load is too high. Causes bus nodes to pause longer between frames

OK.
All nice and well.
Now, how does that work now?

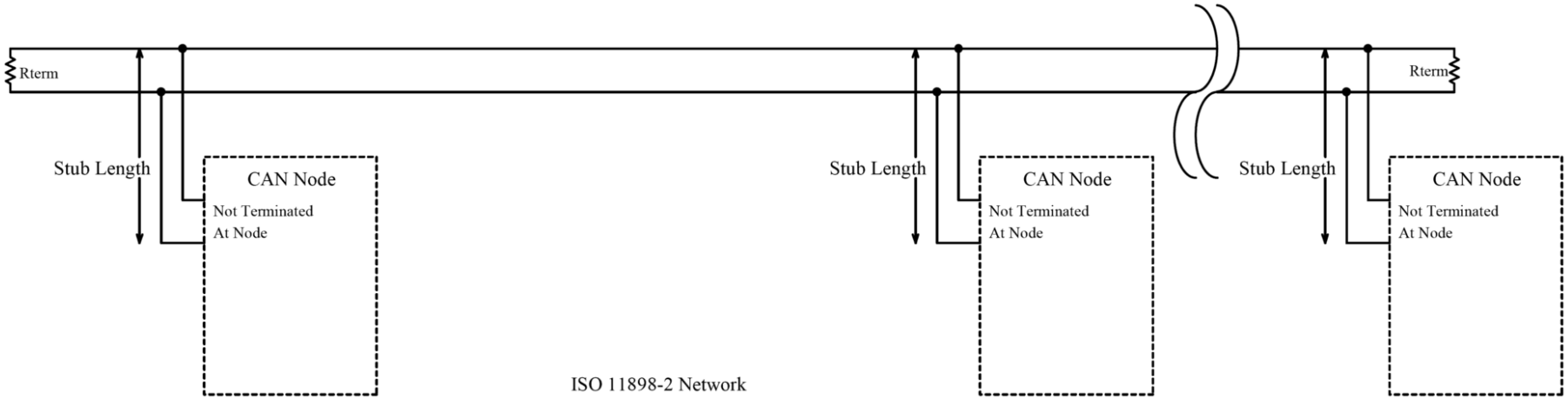
CAN Network



- Example network with three CAN nodes. There is no limit to the number of CAN nodes.
- A limit only exists in the number of different identifiers (11Bit standard, 29Bit extended), aka different messages
- Each node can start sending any time, all others will listen

But: how do we really communicate? And What?

CAN Network



- Each node has a list of identifiers, that can be sent, or received
- For each identifier, there is a list defining how to interpret (encode/decode) the 64Bit contained for that particular identifier

CAN – Example data frame with id=100

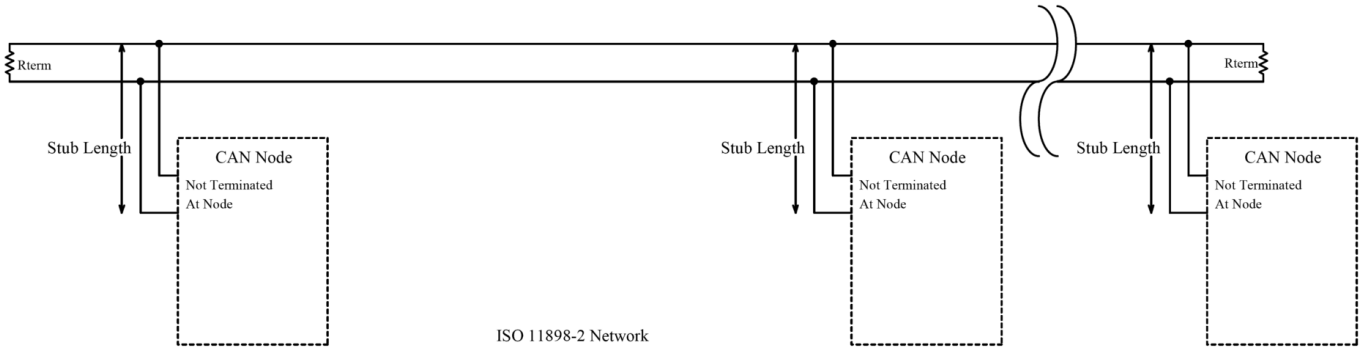
		Bit Position							
		0	1	2	3	4	5	6	7
Byte index in CAN Frame	0	0	1	2	3	4	5	6	7
	1	8	9	10	11	12	13	14	15
	2	16	17	18	19	20	21	22	23
	3	24	25	26	27	28	29	30	31
	4	32	33	34	35	36	37	38	39
	5	40	41	42	43	44	45	46	47
	6	48	49	50	51	52	53	54	55
	7	56	57	58	59	60	61	62	63

CAN – Example data frame with id=100

Byte index in CAN Frame	Bit Position 0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

- Byte 0, Bit 0: Light on/off
- Byte 0, Bit 1: Ignition on/off
- Byte 1, Bit 8-15:
  8Bit UINT (values of 0..256), “name”=“temperature T1”, scaling: gain=5, offset=0

CAN Network



- With each node given a list, i.e. a database of information regarding how to interpret information in the data field of different identifiers, different controllers can communicate with each other

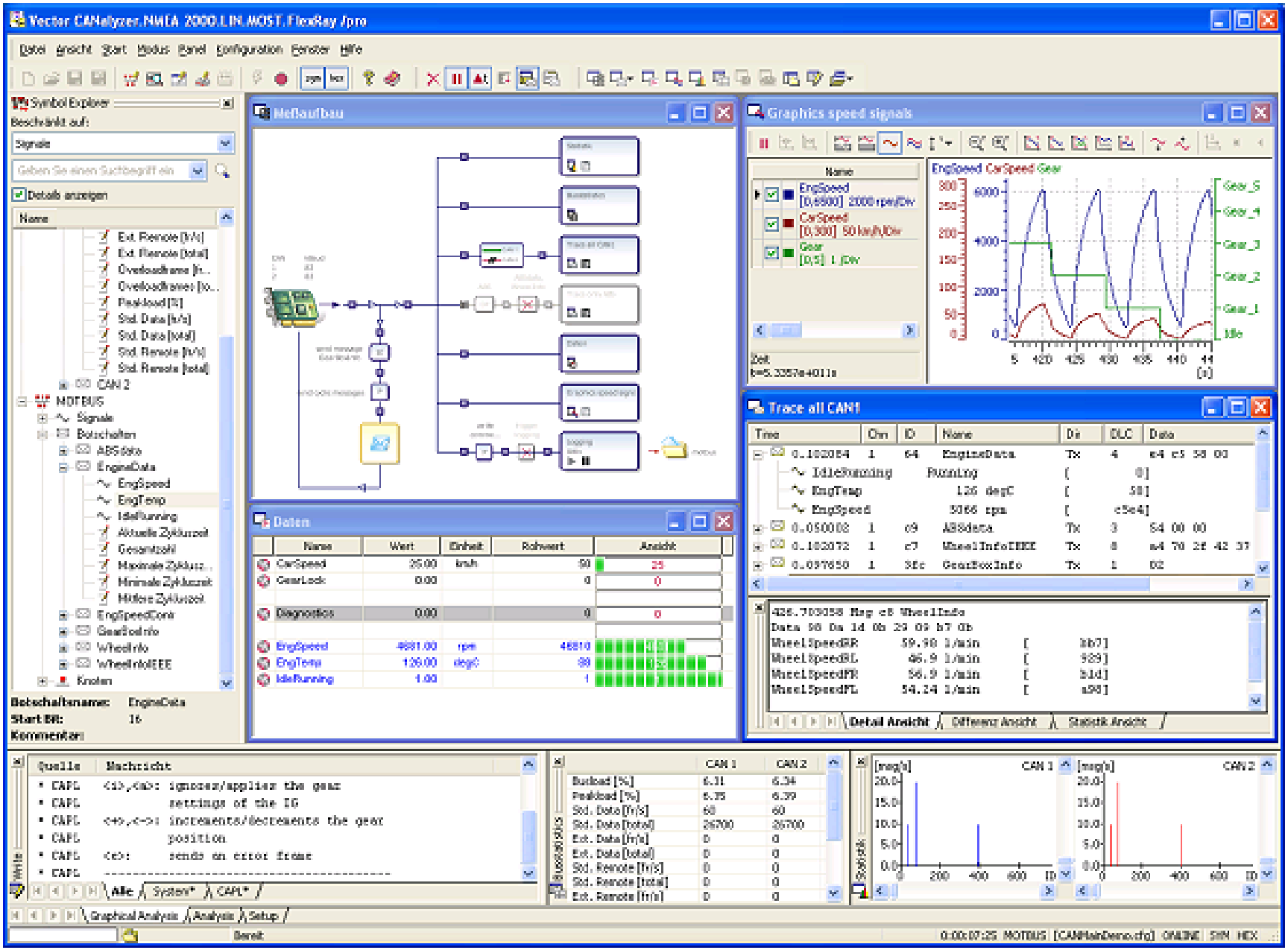
CAN Network Monitoring

- With a CAN adapter for PC's, you can connect to a given CAN network, and start reading BUS communications
- Without a CAN database description, you will see a list of raw indentifiers and contents of the data package
- **Only with a given database description, the information becomes usefull**

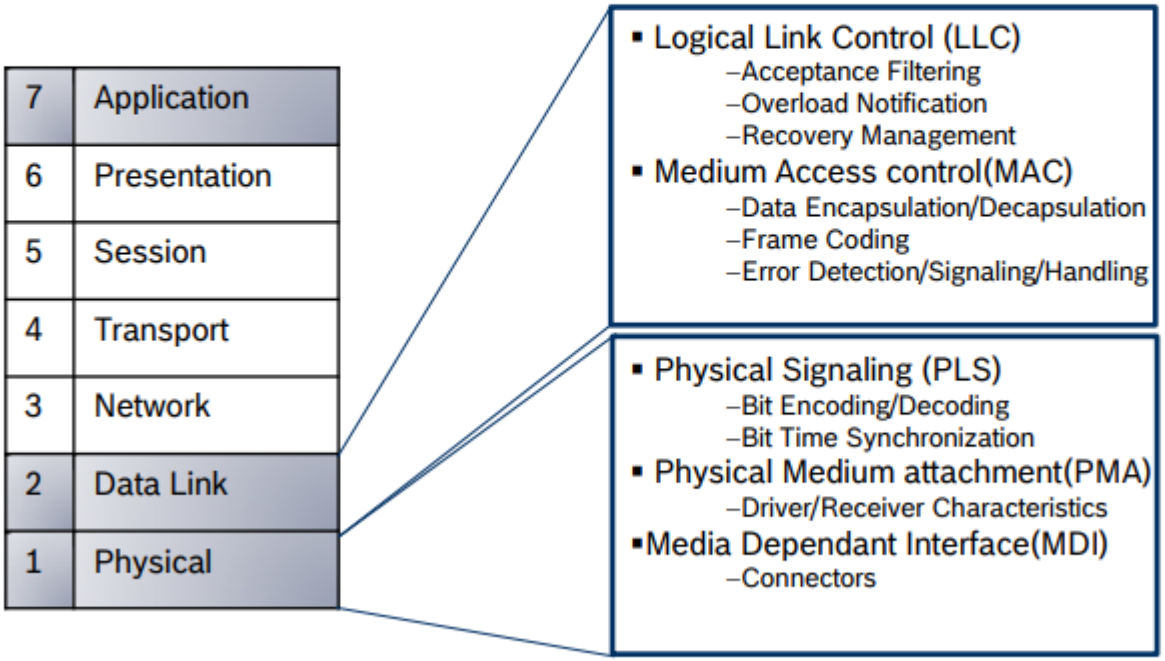
timestamp	Identifier	Byte			
		0	1	2	3
1	100	0F	2A	0	1
2	50	1A	1B	4E	E1
3	13	0F	2A	0	1
4	17	1A	1B	4E	E1
5	20	0F	2A	0	1
6	100	1A	1B	4E	E1
7	50	0F	2A	0	1
8	13	1A	1B	4E	E1
9	17	0F	2A	0	1
10	20	1A	1B	4E	E1
11	100	0F	2A	0	1
12	50	1A	1B	4E	E1
13	13	0F	2A	0	1
14	17	1A	1B	4E	E1

Hex.	Dualsystem	Dez.
0	0 0 0 0	00
1	0 0 0 1	01
2	0 0 1 0	02
3	0 0 1 1	03
4	0 1 0 0	04
5	0 1 0 1	05
6	0 1 1 0	06
7	0 1 1 1	07
8	1 0 0 0	08
9	1 0 0 1	09
A	1 0 1 0	10
B	1 0 1 1	11
C	1 1 0 0	12
D	1 1 0 1	13
E	1 1 1 0	14
F	1 1 1 1	15

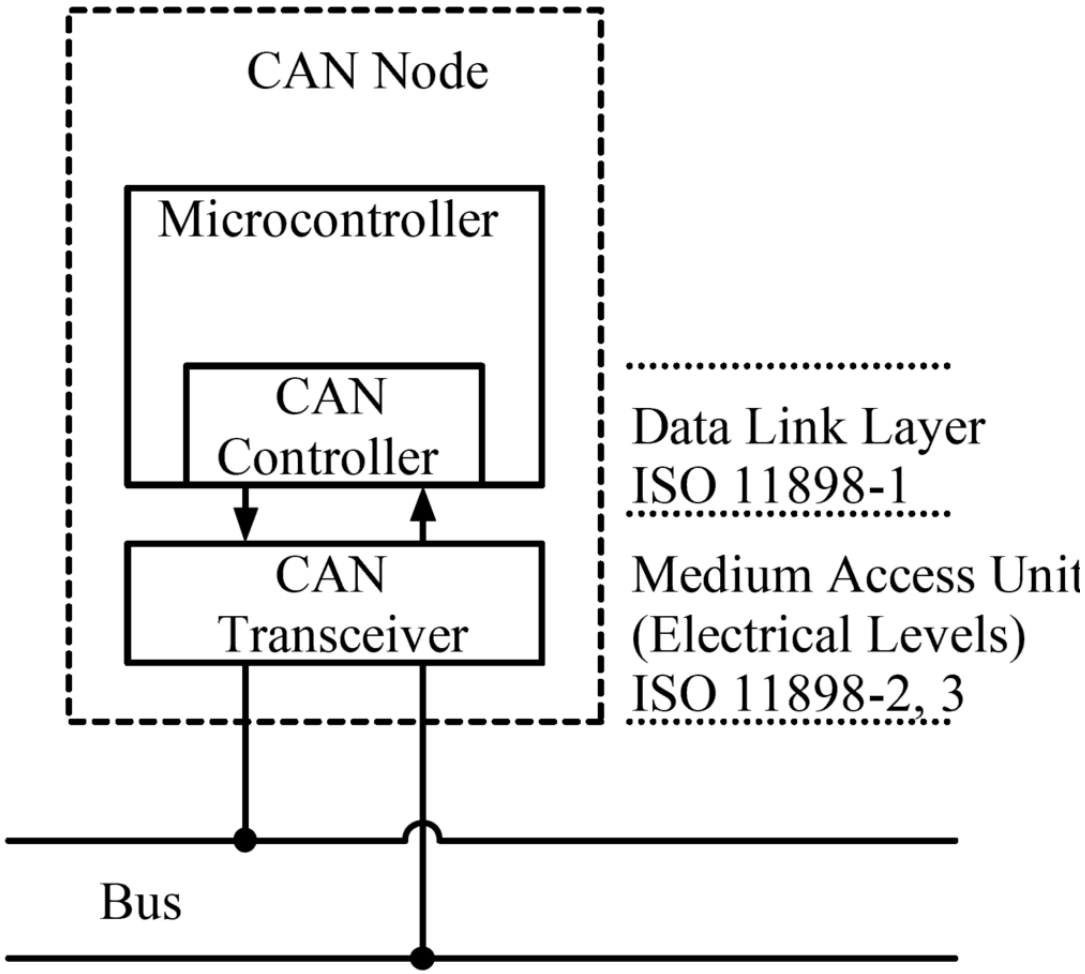
Example CAN Bus Analysis Software



Automotive Bus Systems in the OSI Model: Example of the CAN Bus



Overview of a CAN node



Protocoll Efficiency

$$efficiency = \frac{sum\ of\ information\ bits\ in\ telegram}{sum\ of\ information\ bits + overhead - bits\ in\ telegram}$$

-> How much information can be transmitted with a given bandwidth???

CAN Bus:
Telegram length with standard identifier: 100Bit

Information bits: data-field (8Byte = 64Bit)

CAN Bus efficiency: 64%

Advantages of CAN-Bus Communication

- Different controllers can communicate “lossless” with each other
- In vehicles, **cabling was reduced considerably**, despite an increase in functionality
- Communication can be verified
- CAN controllers are cheap!!!
- Data, and Communication can be logged (stored)
- Resistant to EMC (Electromagnetic Compatibility) -> noise in signals
- Fault tolerant
- Flexible
- Without CAN database file, information is useless (almost…) ->security!!
- Transmission causes delay in signals

Disadvantages of CAN-Bus Communication

- Communication is not strictly real-time
- Discrete transmission of information
- Low Bandwidth (max 1 Mbit/s) -> modern cars have many, many different CAN Bus networks on a single car to overcome low bandwidth
- Every car manufacturer, actually every car, has different communication structures- gets complicated!!!!!

Network Architecture
A6, Q7 and A8 2006

Diagnose-CAN
500 kBaud

Kombi-CAN
500 kBaud

Gateway

Motor Elektronik
Getriebe Elektronik
ESP 5.7
4CL +CDC
Airbag

ACC
E-CAN
500 kBaud
LWR
AFS
EPB
Lenkwinkel Sensor
Lenksäulen Modul
MFL

TSG Fahrer
TSG hinten links
Stand-heizung
TSG Beifahrer
TSG hinten rechts
Klima
FZHli
FZHre
Gebläse
FSH
Einpark-hilfe
ILM hinten
Neigungs Sensor

WWS
ILM Fahrer
FBS / EZS
Fingerprint

ILM Beifahrer
UGDO
Energie Manager
Klimabed. Fond
RDK
Antennen
Dachmodul
SAD

HeadUnit
Radio
Standard Soundsystem
CD-Single
CD-Wechsler
Bose Soundsystem
Navigation
Telefon
Telematik
TV-Tuner
Sprach-bedienung

MMI-Display
MMI Bedienteil
Lautsprecher
Mikrofone
Schnurlos Hörer

Powertrain-CAN
500 kBaud

Comfort-CAN
100 kBaud

LIN-Bus
20 kBaud

MOST-Bus
21 MBaud

Basis
M-Ausstattung

Fieldbus Protocols

MODBUS

Modbus

- Master/Slave Architecture
- open protocol
- Originally developed for connecting PLC’s
- 3 variations of Modbus
  - Modbus RTU
    - RS232 (max 15m, only 2 devices)
    - RS485: multiple devices (32 nodes) , and 1200m max length, very commonly used
    - RS422
    - Typical Baudrates: 9600-19200 Bit/sec.
    - Maximum of 247 slaves + master
  - Modbus Ascii
  - Modbus TCP/IP
    - Higher Bandwidth than with Modbus RTU (LAN network speed: 100MBit/s-1Gbit/s)
    -

Modbus RTU Message

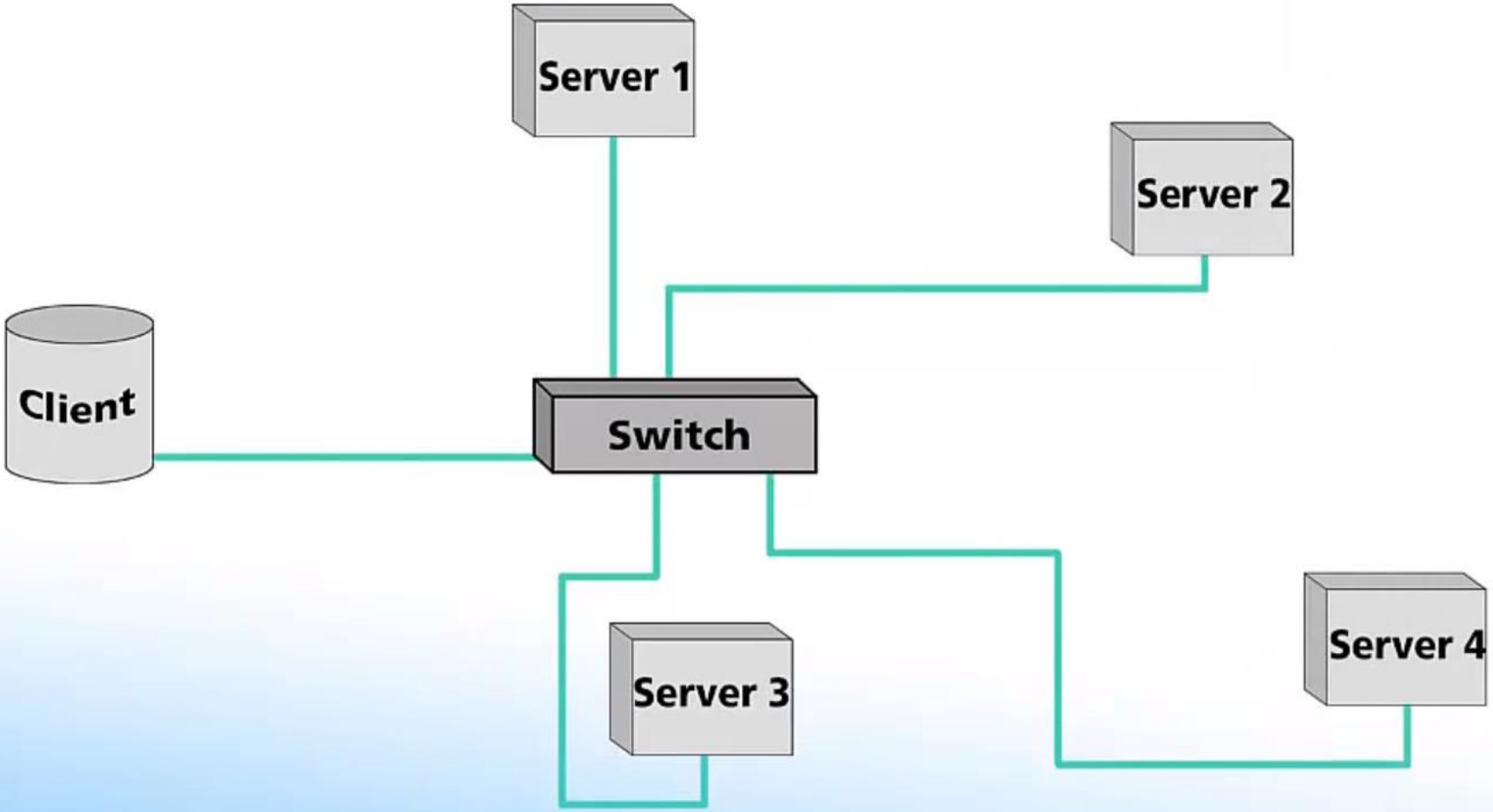
- Slave id: 1 to 247
- Function code:
- Data:
- CRC: error checking

Modbus TCP/IP Message

- MBAP: Modbus Application Header
  contains all information needed to route the message to the correct device
- Function code:
- Data:
- CRC: included in MBAP

- Modbus TCP/IP uses Port 502
- Can be integrated in existing local area network (LAN)

TCP/IP



Modbus Addressing System

4 different datatables

2 read only, 2 read/write

Coils: discrete, 1Bit inputs

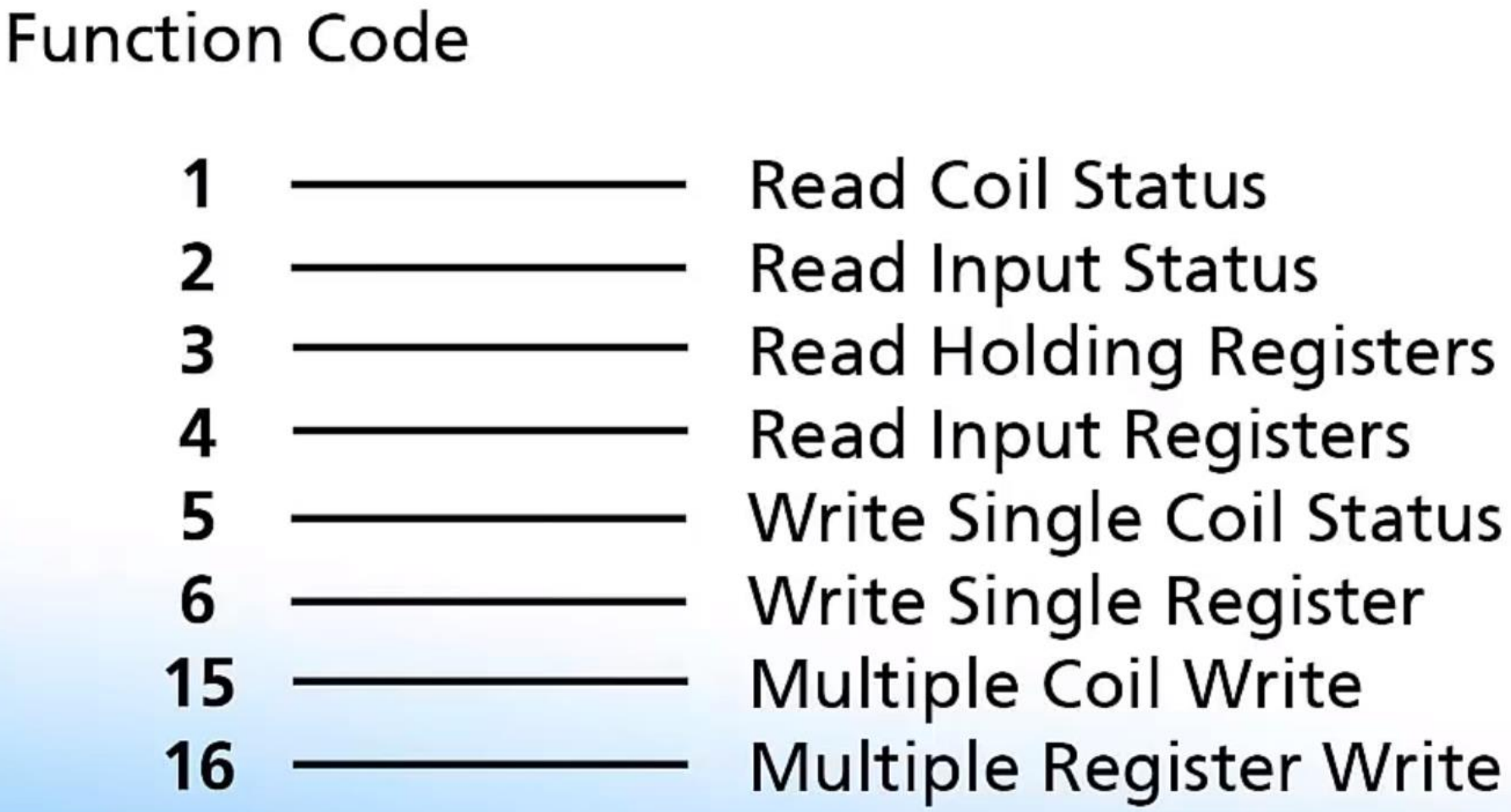
Registers: “words”, 2Byte, 16Bit

No floating point values in Modbus definition.

coils - read/write	**00001 - 09999**
discrete inputs - read only	**10001 - 19999**
input registers - read only	**30001 - 39999**
holding registers - read/write	**40001 - 49999**

Function Code

Define purpose of a message, and tell slave what to do. Read, write operationsn etc.

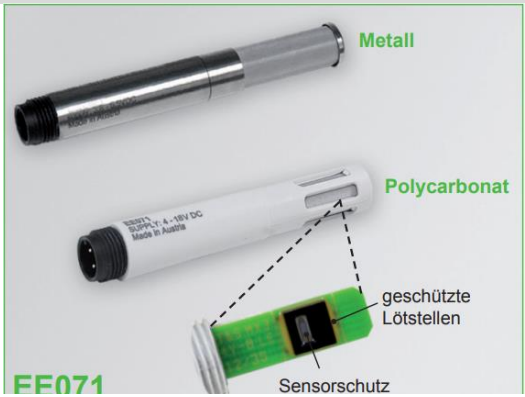


Modbus Example: EE071 Temperature Sensor

Uses Modbus RTU on RS485

Supplier supplies information on how information is shared on the bus

http://downloads.epluse.com/fileadmin/data/product/ee071/Datenblatt_EE071.pdf



Modbus Map

Die Messwerte werden als 32Bit *float* Wert von 0x19 bis 0x25 und als 16Bit *signed integer* zwischen 0x27 und 0x2D gespeichert.
Die Slave- ID beträgt werkseitig 247 als *integer* 16Bit Wert. Diese ID kann im Register 0x00 kundenseitig überschrieben werden (Wertebereich 1 - 247 zulässig).
Die Seriennummer befindet sich als ASCII-Zeichen an Registeradresse 30001-30008.

FLOAT (Leseregister):

Register-adresse	Kommunikations-adresse	Parameter Name	
30026	0x19	Temperatur	[°C]
30028	0x1B	Temperatur	[°F]
30030	0x1D	Feuchte rel	[%]
30032	0x1F	Feuchte abs	[g/m³]
30034	0x21	Taupunkt	[°C]
30036	0x23	Taupunkt	[°F]
30038	0x25	Mixing ratio	[g/kg]

INTEGER (Leseregister):[1)]

Register-adresse	Kommunikations-adresse	Parameter Name	
30040	0x27	Temperatur	[°C]
30041	0x28	Temperatur	[°F]
30042	0x29	Feuchte rel	[%]
30043	0x2A	Feuchte abs	[g/m³]
30044	0x2B	Taupunkt	[°C]
30045	0x2C	Taupunkt	[°F]
30046	0x2D	Mixing ratio	[g/kg]

INTEGER (Schreibregister):

Register-adresse	Kommunikations-adresse	Parameter Name
60001	0x00	Slave-ID

FLOAT (Lese-/Schreibregister):

Register-adresse	Kommunikations-adresse	Parameter Name
5001[2)]	0x1388	Luftdruck[3)]

1) Werte sind mit einer Skalierung von 1:100 abgelegt (z.B.: 2550 entspricht 25,5°C)
2) Lesen Funktion Code: 0x03 Schreiben Funktion Code: 0x10
3) Umgebungsdruck in mbar, mit 2 Nachkommastellen (z.B. 1008,25)

Modbus-Protokoll-Einstellungen siehe Application Note Modbus (www.epluse.com/EE071).