Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

12.24196

Introduction to Embedded Systems

Prof. Dr.-Ing. Stefan Kowalewski | Julius Kahle, M. Sc.

Summer Semester 2025

Part 2

Data Buses

Introduction

- Most embedded systems are interconnected
  - Microcontroller and external devices (board)
  - Engine control unit and tachometer (car)
  - Process control center and magnetic valve (plant)
- All communication share common principles
- Implementations vary in
  - Costs
  - Safety and reliability
  - Real-time capability
  - Data rate
  - Flexibility

Part A

Data Communication Basics

- Fourier analysis and sampling theorem
- Topology
- ISO/OSI
  - Mechanical and electrical properties
  - Bit encoding
  - Frames
  - Error detection and correction
  - Medium access

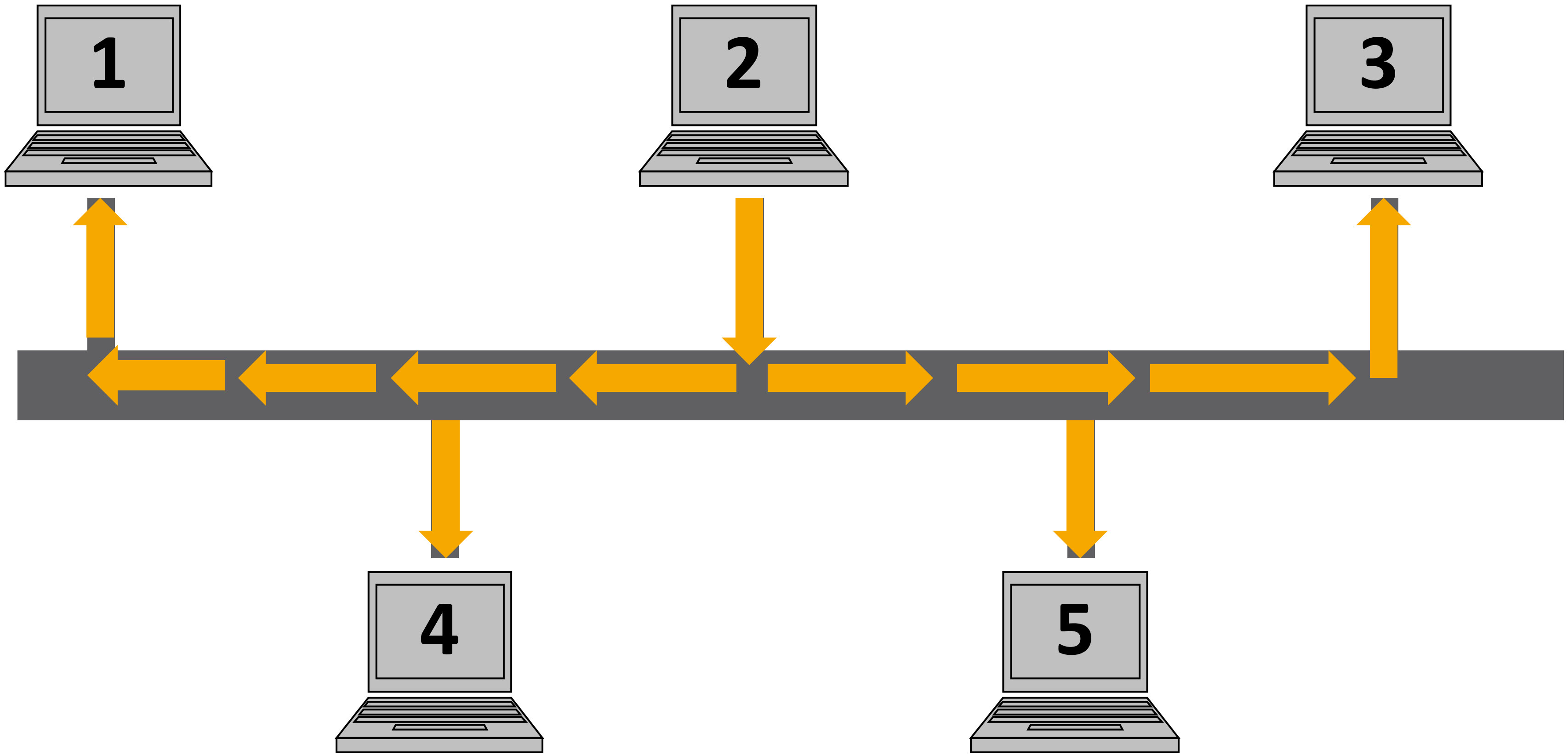
Layer 1

Layer 2

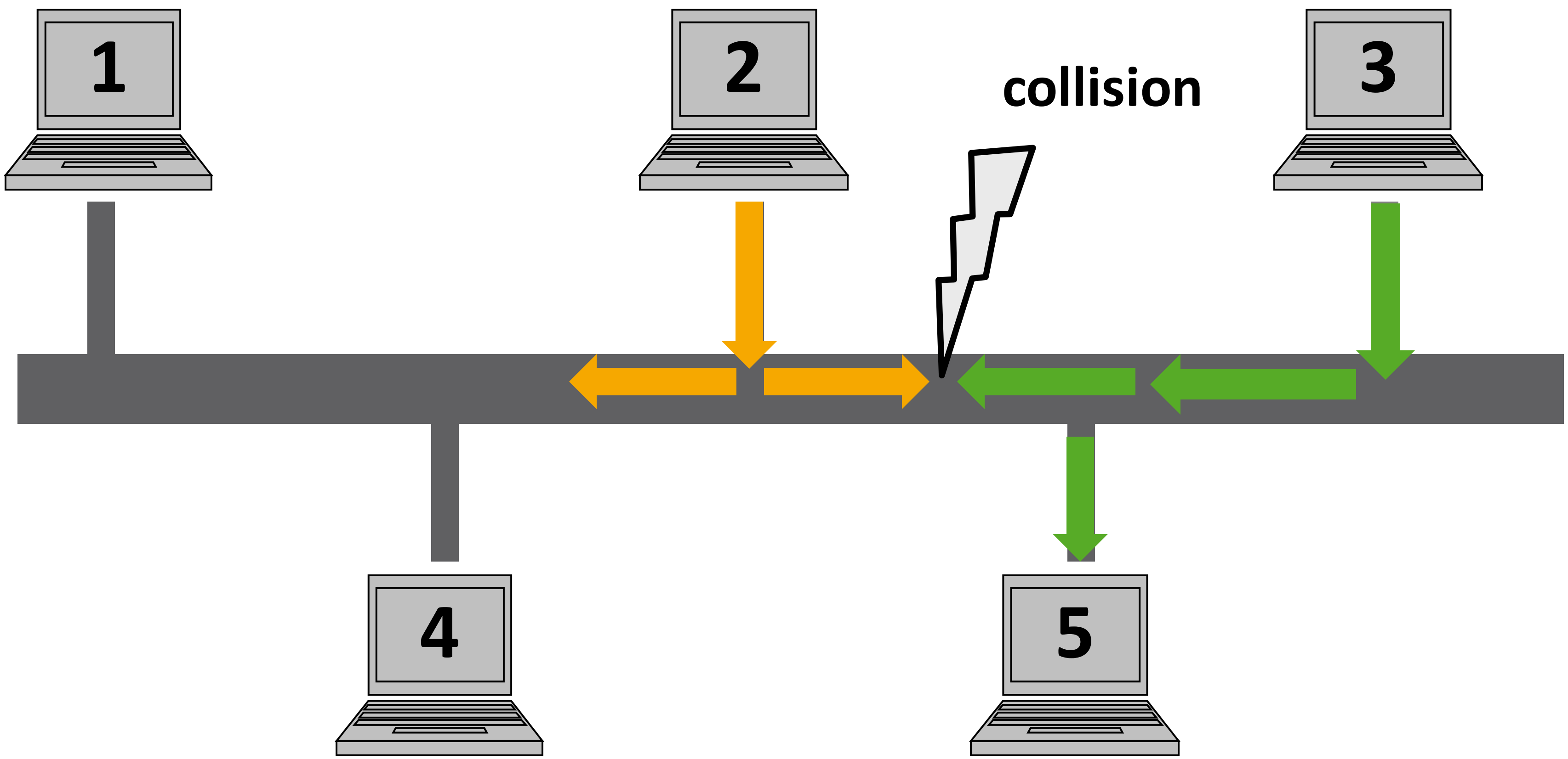
Bus Topology

- Linear line (with terminators)
- Passive connection (no repeating)
- Only one partner can send at a time
- All partners can listen to all communication
- Advantages
  - Cheap
  - Simple
- Disadvantages
  - Multiple access (Babbling idiot, security)
  - Single point of failure

Bus Topology



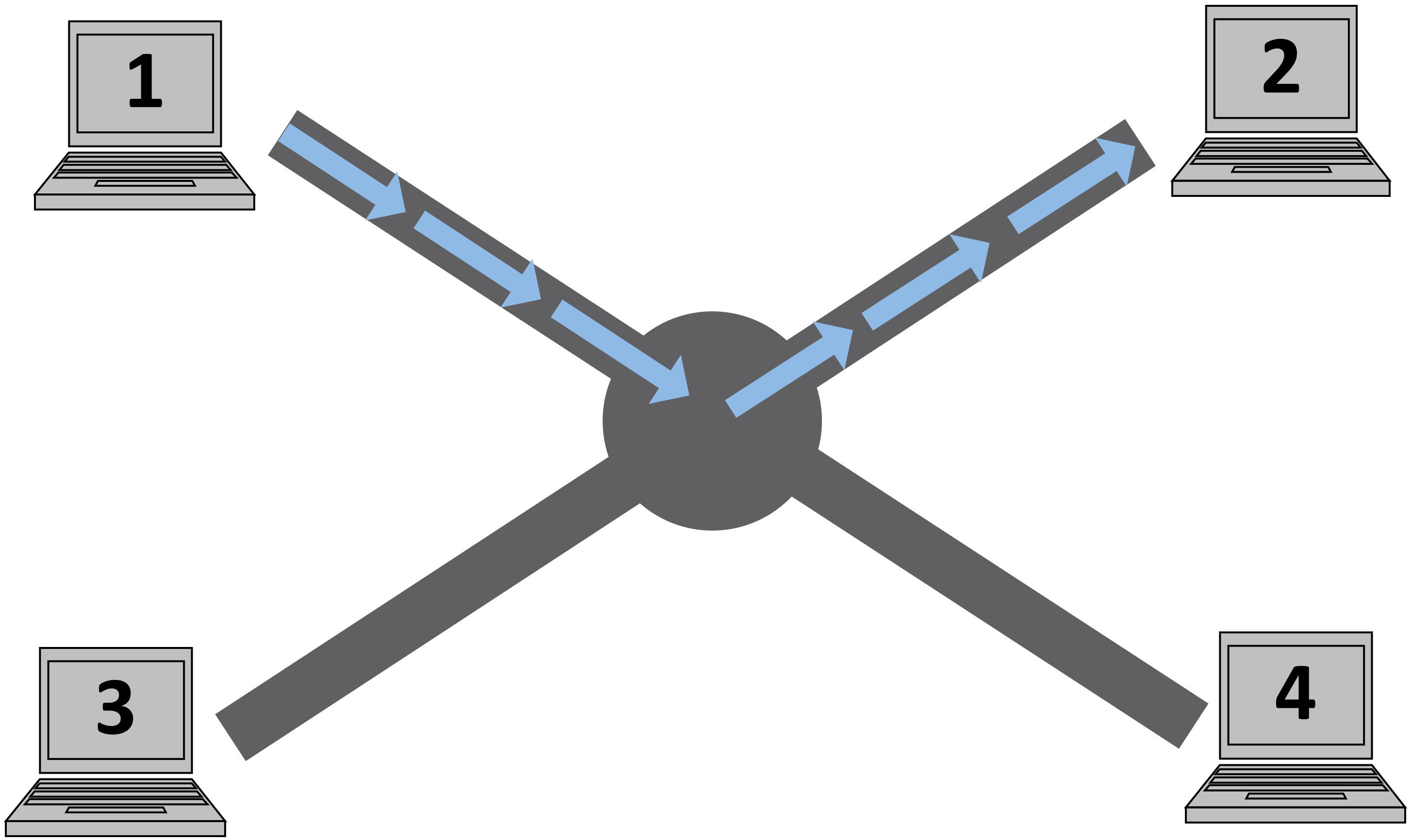
Bus Topology



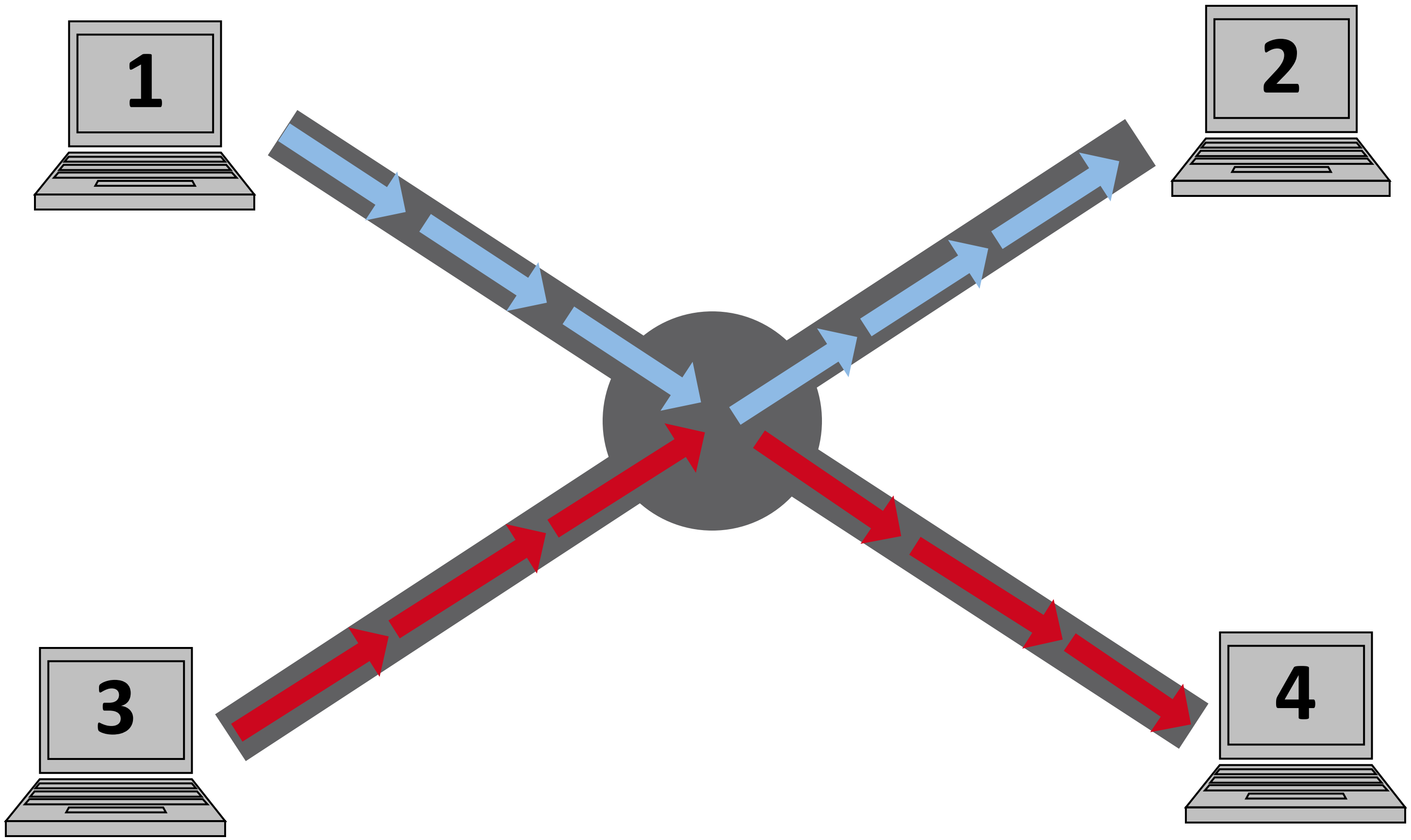
Star Topology

- Dedicated connection to central station
- Buffering and repeating
- Multiple partners can send at the same time
- Only sender and receiver can listen to communication
- Advantages
  - Multiple access, no collisions
  - Only central station is single point of failure
- Disadvantages
  - Expensive central station
  - More wiring

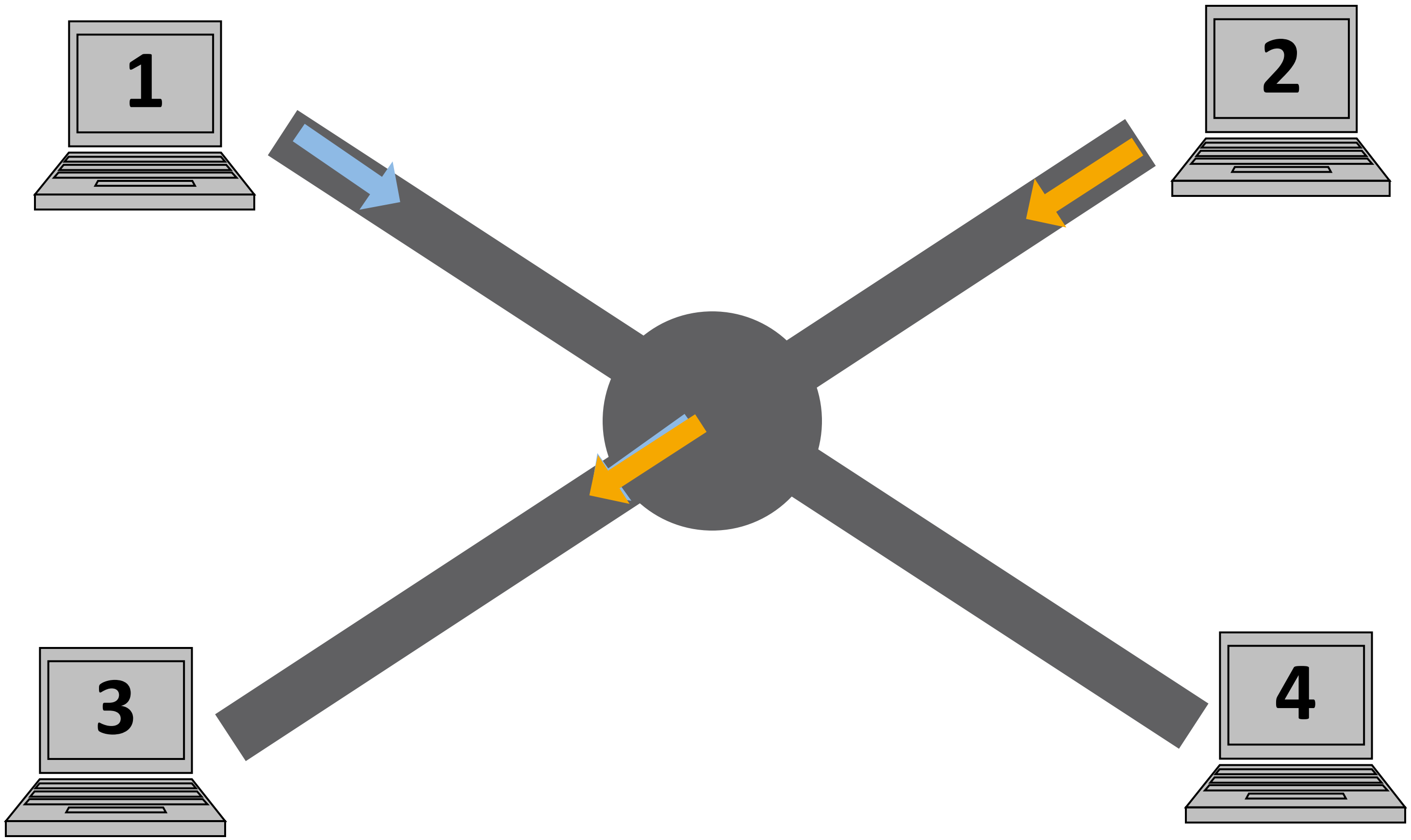
Star Topology



Star Topology



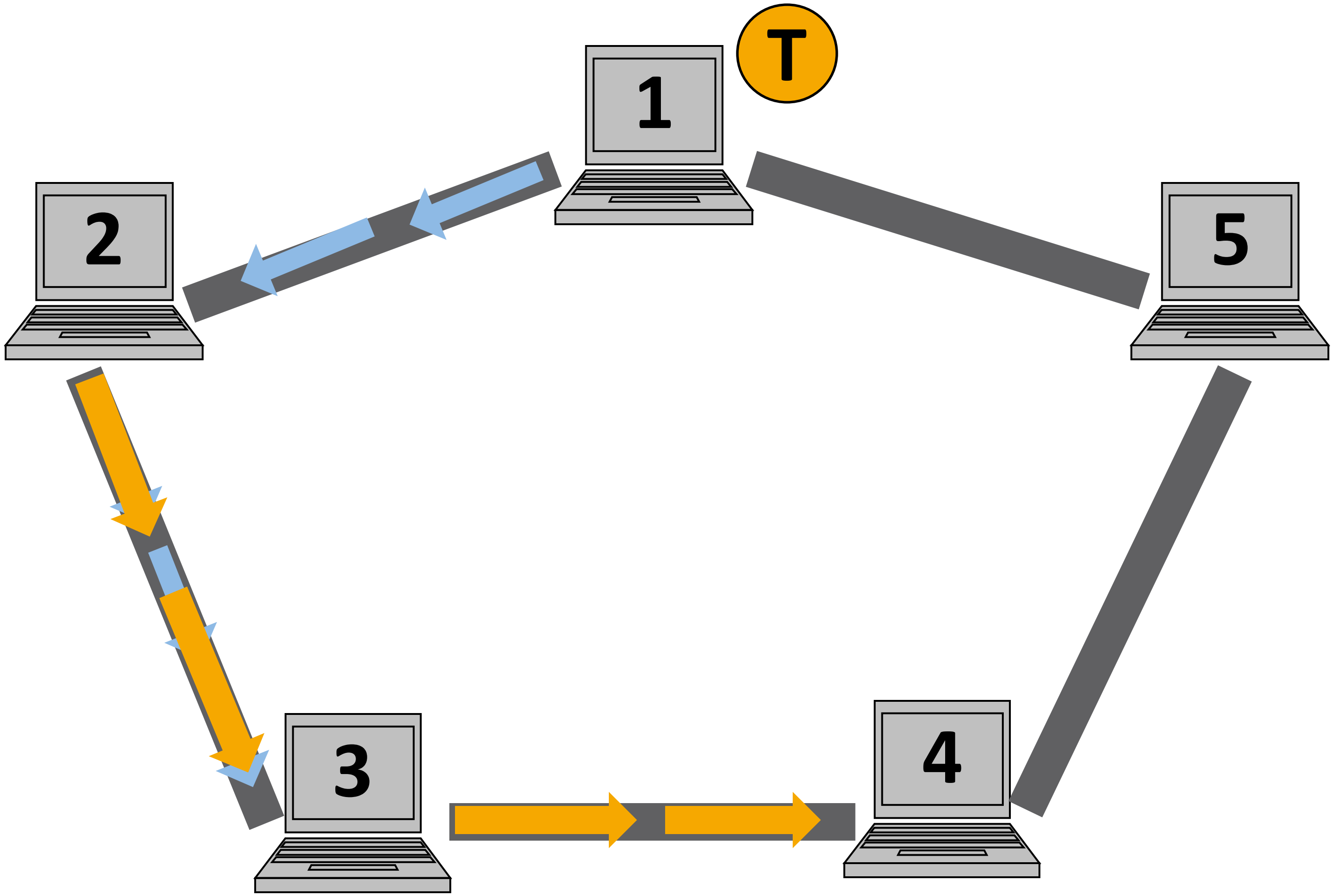
Star Topology



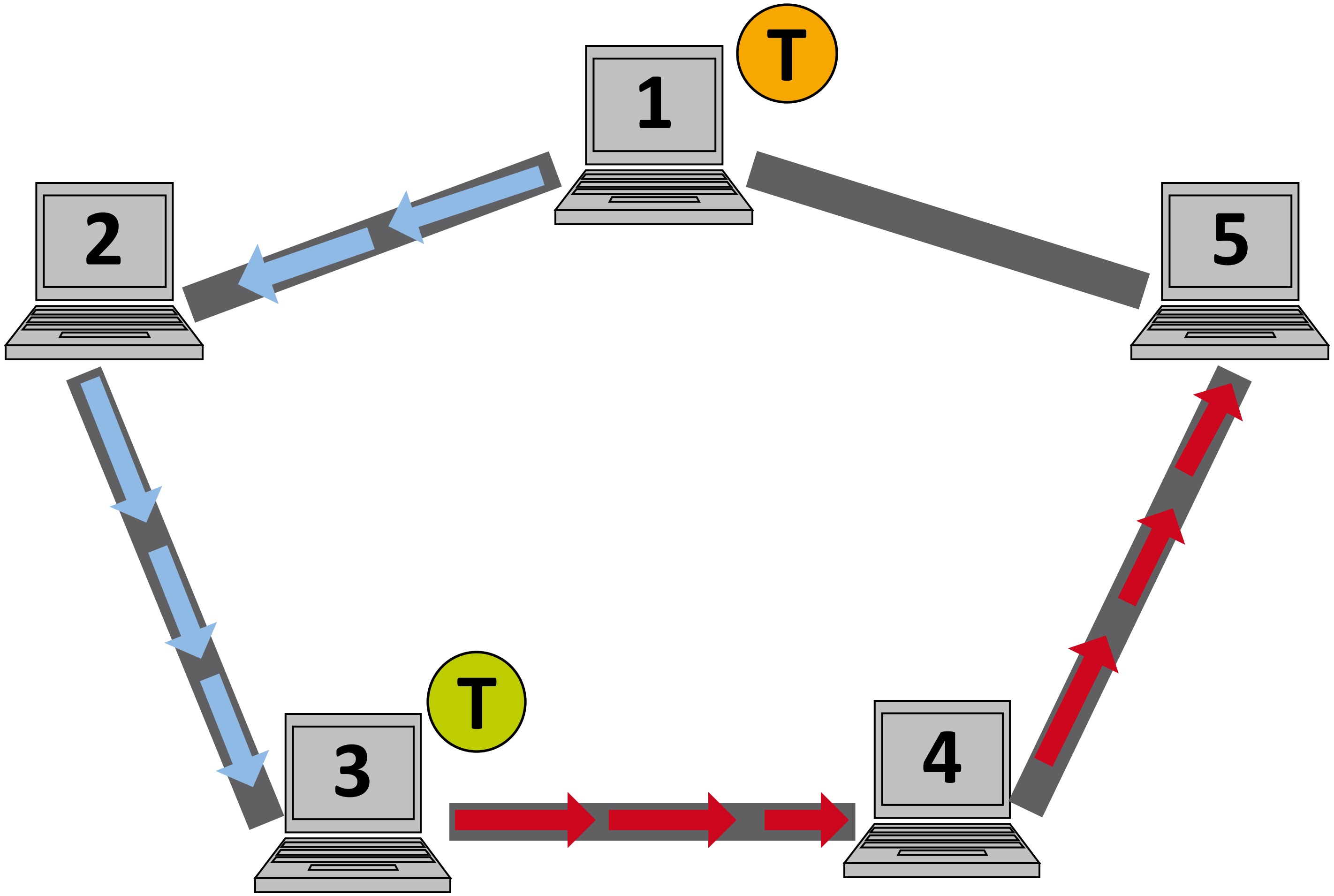
Ring Topology

- Circular line (unidirectional)
- Active connection (repeating / changing)
- Multiple partners can send at the same time
- Some partners can listen to communication
- Advantages
  - High quality of service
  - Multiple access (to some extend)
- Disadvantages
  - Complex (expensive)
  - Single point of failure

Ring Topology



Ring Topology



ISO/OSI

- International Organization for Standardization
- Open Systems Interconnection
- 7 Layer architecture
- One task per layer
- Very complex
- Reference model
- Here: layers 1 & 2

Layer 7: Application Layer
Layer 6: Presentation Layer
Layer 5: Session Layer
Layer 4: Transport Layer
Layer 3: Network Layer
Layer 2: Data Link Layer
Layer 1: Physical Layer

Physical Layer

- Defines mechanical properties
  - Medium: copper, optical fiber, air, EM waves
  - Connectors: form and pin assignment
- Defines electrical / optical properties
  - Voltage
  - Frequencies
  - Baud rate
  - Bit encoding
- Hardware
  - Cable, connector, terminator, antenna, amplifier
  - Transceiver, repeater, hub

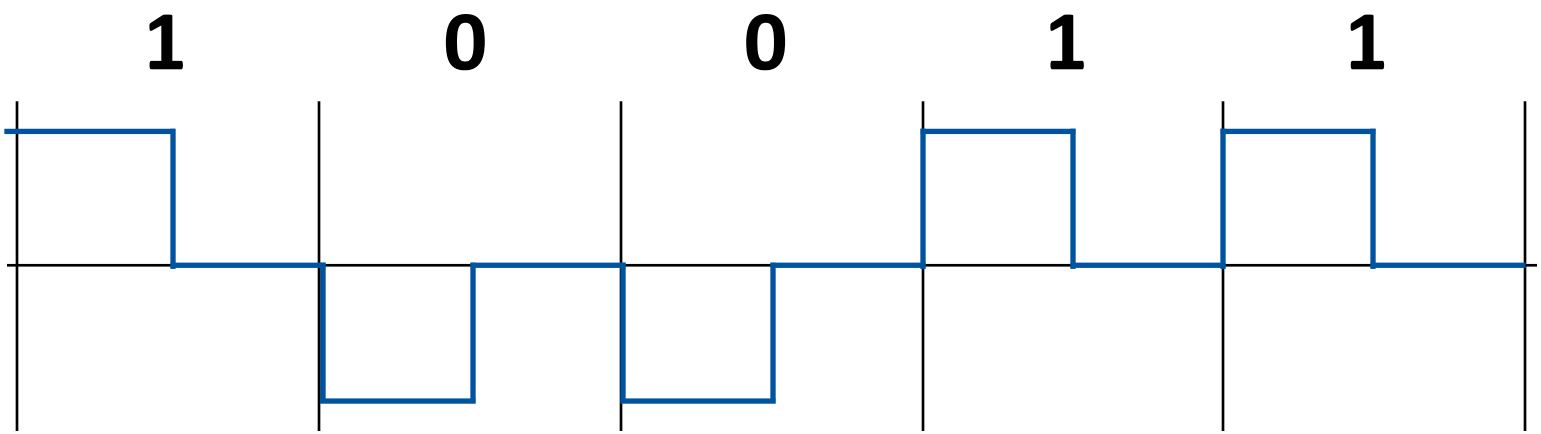
Bit Encoding

- Return to Zero (RZ)
- Non Return to Zero (NRZ)
- Differential NRZ
- Bit stuffing
- Manchester Code
- 4B/5B Code

Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY

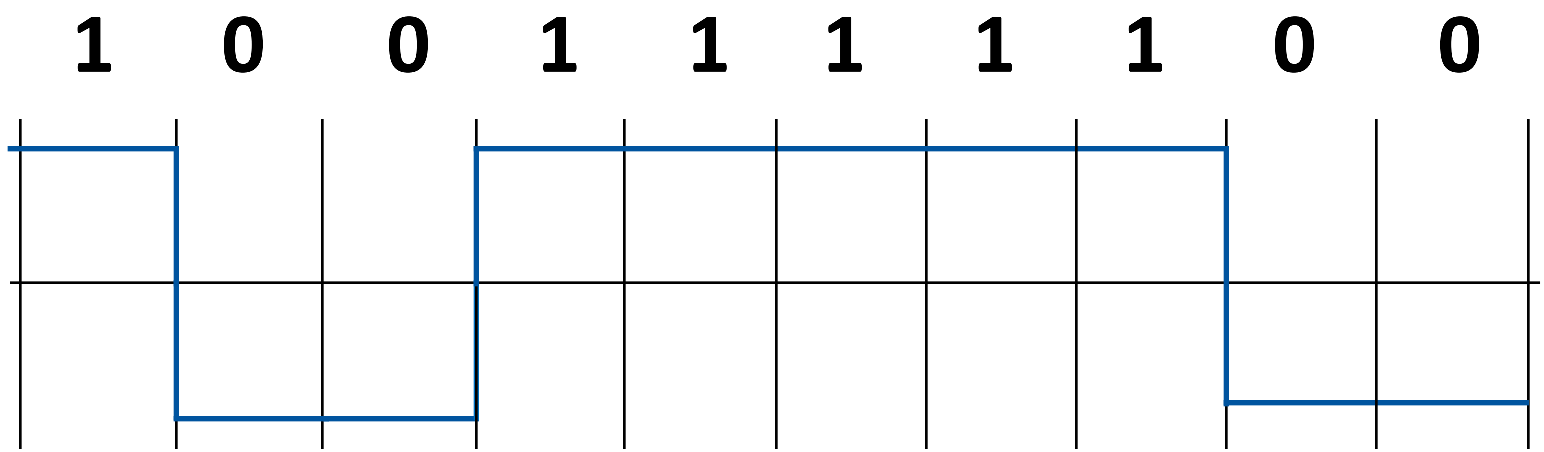
Return to Zero

- Return to neutral state between all pulses
- Needs three states
- Self-synchronizing
- Half data rate



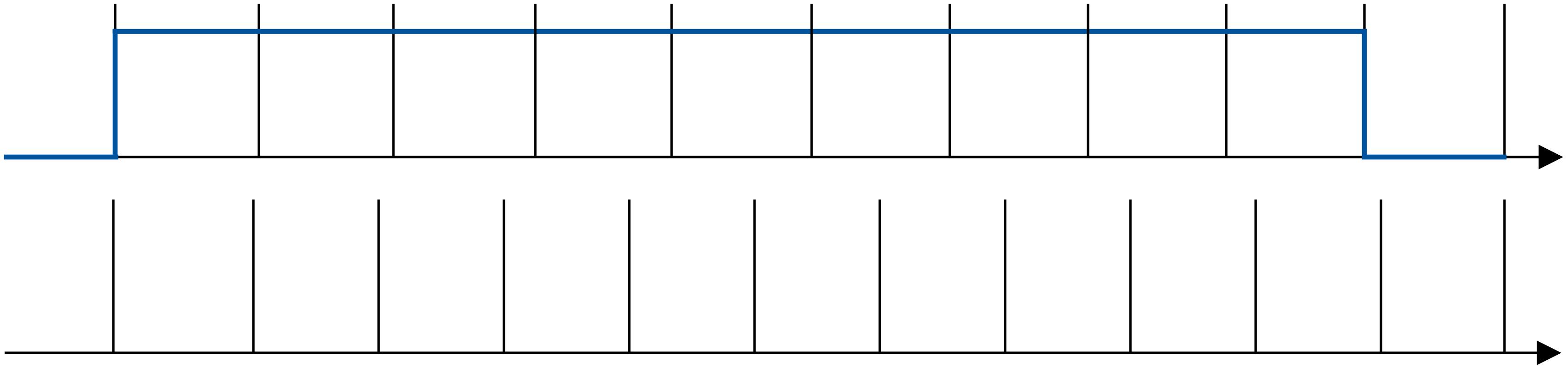
Non Return to Zero

- No neutral state
- Needs synchronization
- Capacitive problems
- Full data rate



Synchronization

- Clocks are never perfectly synchronous
- Clock A ticks every 100µs
- Clock B ticks every 90µs
- Clock A sends nine ones → high level for 900µs
- Clock B interprets this as ten ones

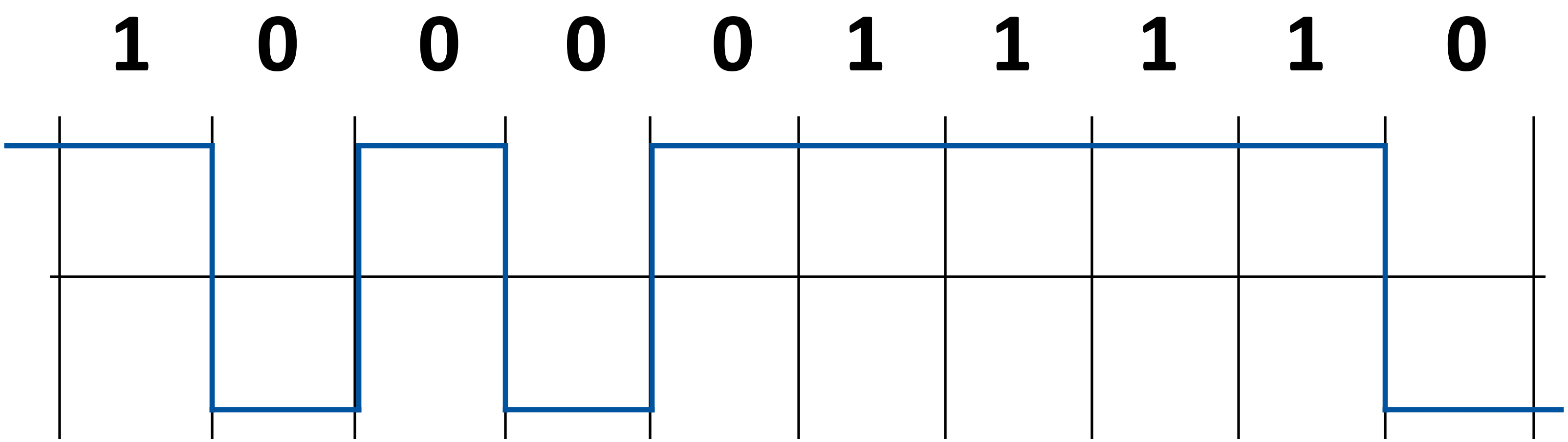


Informatik 11
Embedded Software

RWTH AACHEN UNIVERSITY

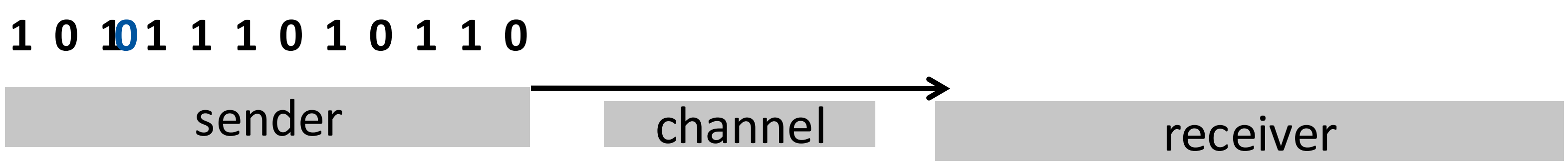
Differential NRZ

- 0 is represented by level change
- 1 is represented by no level change
- No problems for long sequences of 0s



Bit Stuffing

- Prevents long sequences of 1s
- Sender inserts a 0 after a sequence of n 1s (n = 6 for USB)
- Receiver (checks and) removes inserted 0s
- Code violations (frame delimiters)

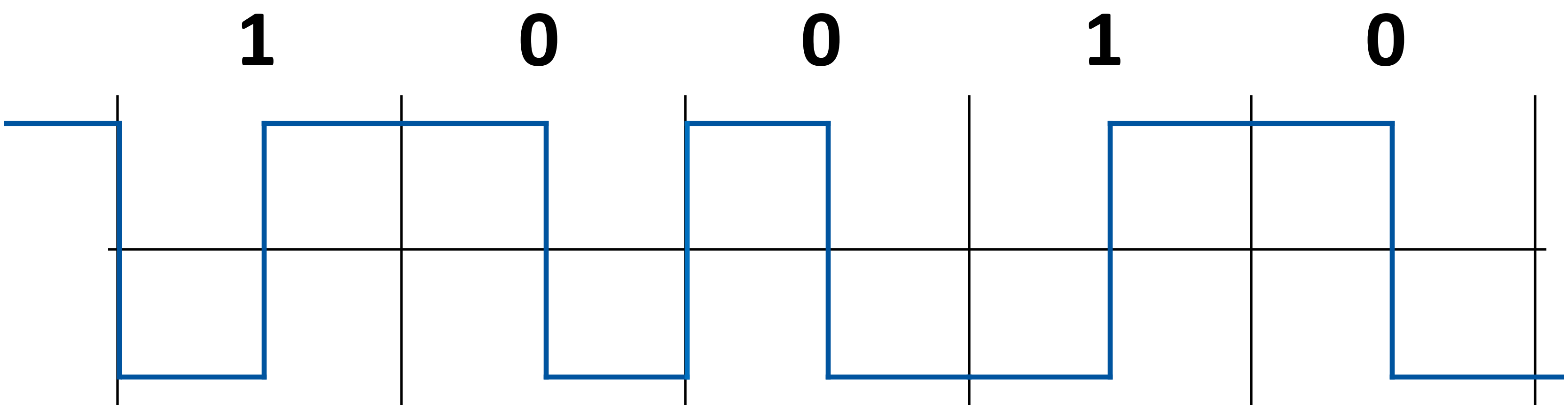


Manchester Code (IEEE 802.3)

- No neutral state
- Self-synchronizing
- No capacitive problems
- Half data rate

Rising Edge = 1
Falling Edge = 0

(inverted in G.E. Thomas version)



4B/5B

- Uses some form of NRZ
- 80% data rate
- Encoding table prevents long sequences (FDDI with NRZI)

Name	4B	5B	Name	4B	5B	Name	5B	Desc
0	0000	11110	8	1000	10010	Q	00000	Quiet
1	0001	01001	9	1001	10011	I	11111	Idle
2	0010	10100	A	1010	10110	J	11000	Start #1
3	0011	10101	B	1011	10111	K	10001	Start #2
4	0100	01010	C	1100	11010	T	01101	End
5	0101	01011	D	1101	11011	R	00111	Reset
6	0110	01110	E	1110	11100	S	11001	Set
7	0111	01111	F	1111	11101	H	00100	Halt

ISO/OSI

Layer 7: Application Layer
Layer 6: Presentation Layer
Layer 5: Session Layer
Layer 4: Transport Layer
Layer 3: Network Layer
Layer 2: Data Link Layer
Layer 1: Physical Layer

Data Link Layer

- Encapsulates data (bits) into frames
- Frame synchronization
- Logical link control
  - Automatic repeat request (ARQ)
  - Forward error correction (FEC)
  - Flow control
- Media access control
- Hardware: Switch, Bridge
- Two sublayers
  - Logical Link Control (LLC)
  - Media Access Control (MAC)

Frames

- On layer 2 data is transferred in frames
- Typical parts of a frame
  - Start delimiter
  - Sender address
  - Receiver address
  - Identifier
  - Acknowledgement
  - Length of frame / data
  - User data
  - Checksum
  - End delimiter

SD	SA	RA	ID	ACK	L	Data	CS	ED

Error Detection

- Parity bit
  - Append parity bit such that the sum off all bits is even / odd
  - One bit error detection
- Cyclic Redundancy Check (CRC)
  - Hash function based on polynomial division
  - Detection of burst errors
- Hamming Code
  - Set of parity bits
  - Single error correction
  - (Double error detection)

Hamming Code

- Published by Richard Hamming in 1950

- For (up to) 2^n-1 Bits in a hamming encoded message:
  - Positions that are powers of 2 are Parity Bits (n Parity Bits)
  - Remaining Bits are the data Bits (2^n-n-1 Data Bits)
  - A Data Bit at position x is protected by the Parity Bits that comprise x

- Example: 14 Bits in the encoded message
  - Positions 1, 2, 4 and 8 are Parity Bits
  - Positions 3, 5-7, 9-14 are Data Bits
  - Data Bit 11 is protected by the Parity Bits 1, 2 and 8 (1 + 2 + 8 = 11)

Hamming Code – Example (Encoding)

- Encode the 8 Bit message $(01010010)_2$ with <u>even</u> parity:
- Use 4 Parity Bits: ($2^3 - 3 - 1 < 8 <= 2^4 - 4 - 1$)
- Structure of encoded message:
  - $p_1 p_2 d_3 p_4 d_5 d_6 d_7 p_8 d_9 d_{10} d_{11} d_{12}$
- Fill in the message:
  - $p_1 p_2 0 p_4 101 p_8 0010$
- Calculate Parity Bits:
  - $p_1 + d_3 + d_5 + d_7 + d_9 + d_{11}$ has to be <u>even</u> -> $p_1 = 1$
  - $p_2 + d_3 + d_6 + d_7 + d_{10} + d_{11}$ has to be <u>even</u> -> $p_2 = 0$
  - $p_4 + d_5 + d_6 + d_7 + d_{12}$ has to be <u>even</u> -> $p_4 = 0$
  - $p_8 + d_9 + d_{10} + d_{11} + d_{12}$ has to be <u>even</u> -> $p_8 = 1$
- Encoded message is: $(100010110010)_2$

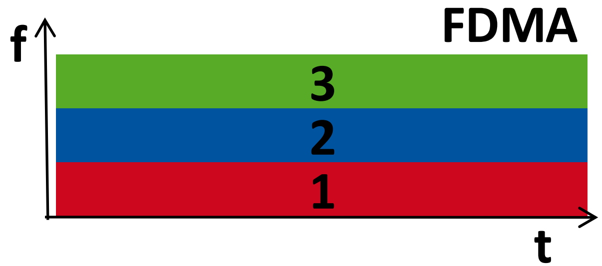
Informatik 11 Embedded Software
RWTH AACHEN UNIVERSITY

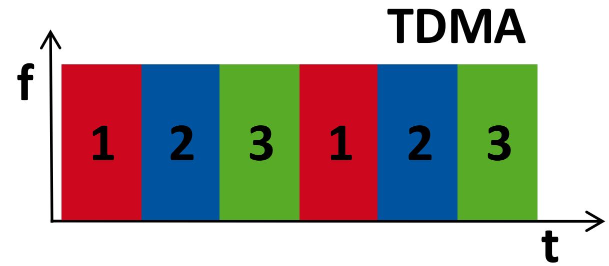
Automatic Repeat Request

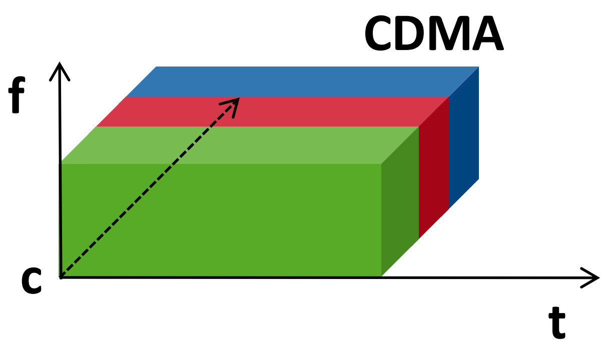
- Successful transmission
  - A sends frame to B
  - B acknowledges frame
- Unsuccessful transmission (transmission error)
  - A send frame to B
  - B detects error
  - B sends negative acknowledgement (“NACK”)
  - A resends frame to B
- Unsuccessful transmission (transmission lost)
  - A sends frame to B
  - Timeout occurs
  - A resends frame to B

Media Access Control

- Regulates access to a shared medium
- Static MAC
  - Frequency (FDMA)
  - Time (TDMA)
  - Code (CDMA)
  - Hybrid



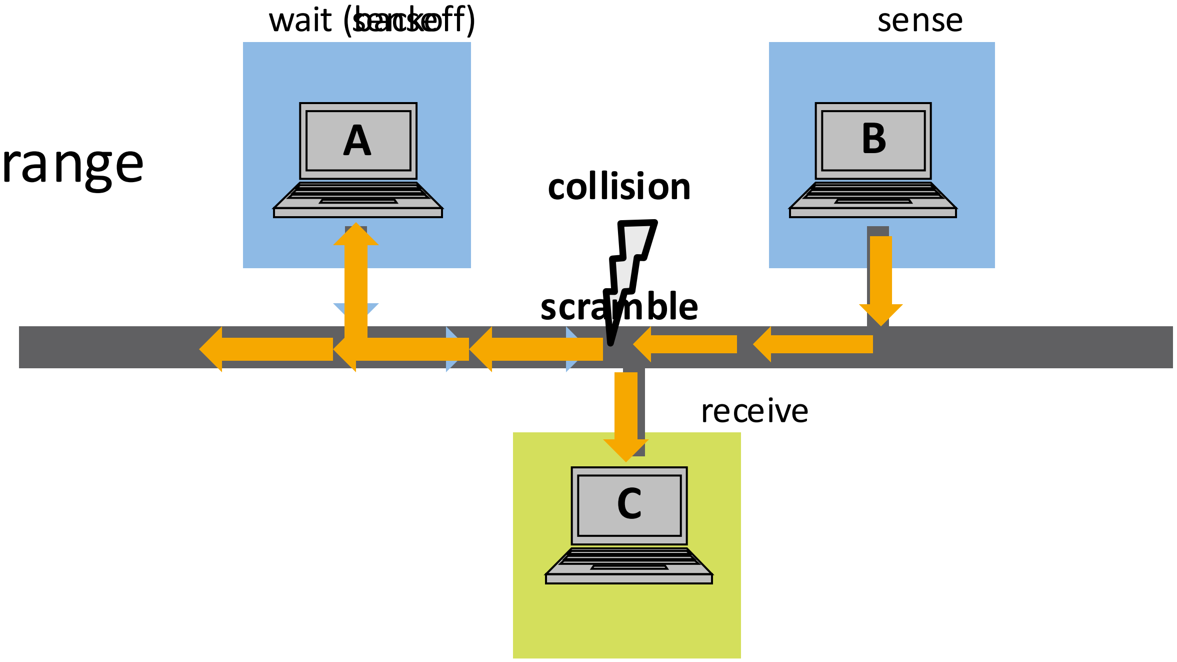




- Dynamic MAC
  - With collisions: CSMA/CD
  - Without collisions: CSMA/CR

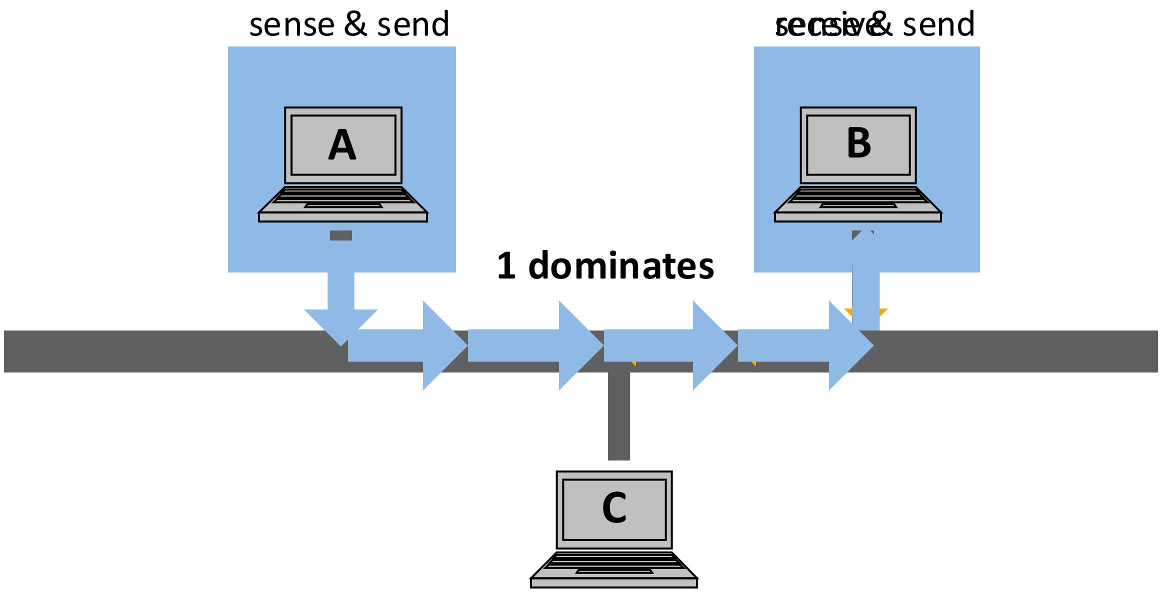
CSMA/CD

- Carrier sense multiple access / collision detection
- Wait until medium is free
- Start sending
- If collision is detected
  - Scramble
  - Back off
- High data rate / long range



CSMA/CR

- Carrier sense multiple access / collision resolution
- Wait until medium is free
- Start sending
- If collision is dominated
  - Stop sending
  - Start receiving
- No Collisions
- Either dominant
  - 1 (“wired or”) or
  - 0 (“wired and”)



Part B

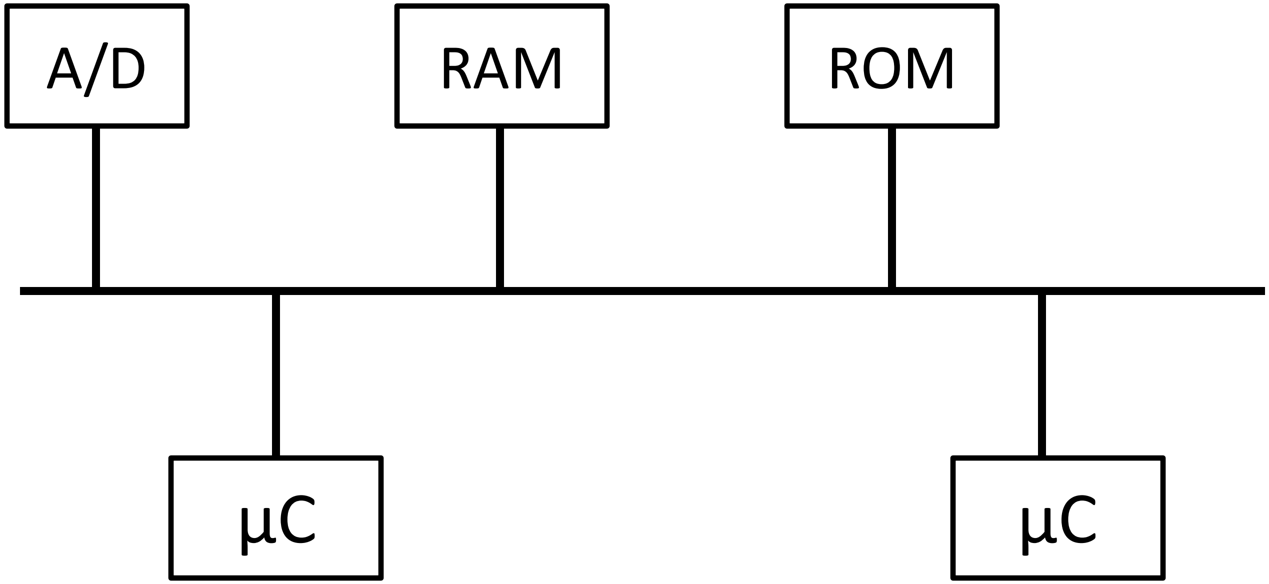
Data Bus Standards

- I^2C bus
- CAN bus
- FlexRay
- PROFIBUS

Inter-Integrated Circuit (I^2C) Bus

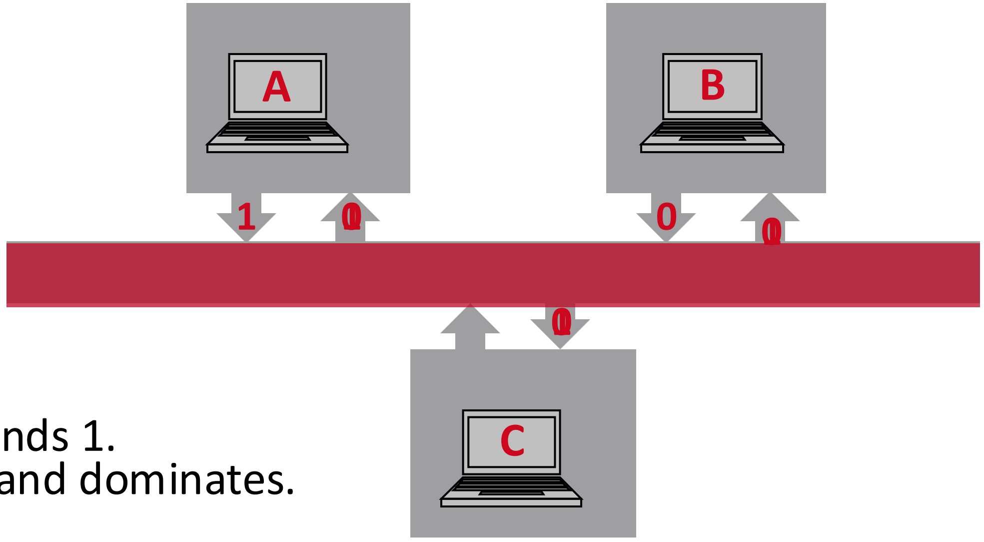
- Connects multiple devices on the same board
- Developed by Philips in 1980s
- Also known as Two Wire Interface (TWI)
- Five modes
  - Standard mode: 100 kbit/s
  - Fast mode: 400 kbit/s
  - Fast mode plus: 1 Mbit/s
  - High speed mode: 3.4 Mbit/s
  - Ultra-high speed mode: 5 Mbit/s
- Noise-prone (used inside shielded casings)
- Simple and cheap
- Very popular

I²C – Exemplary Setup



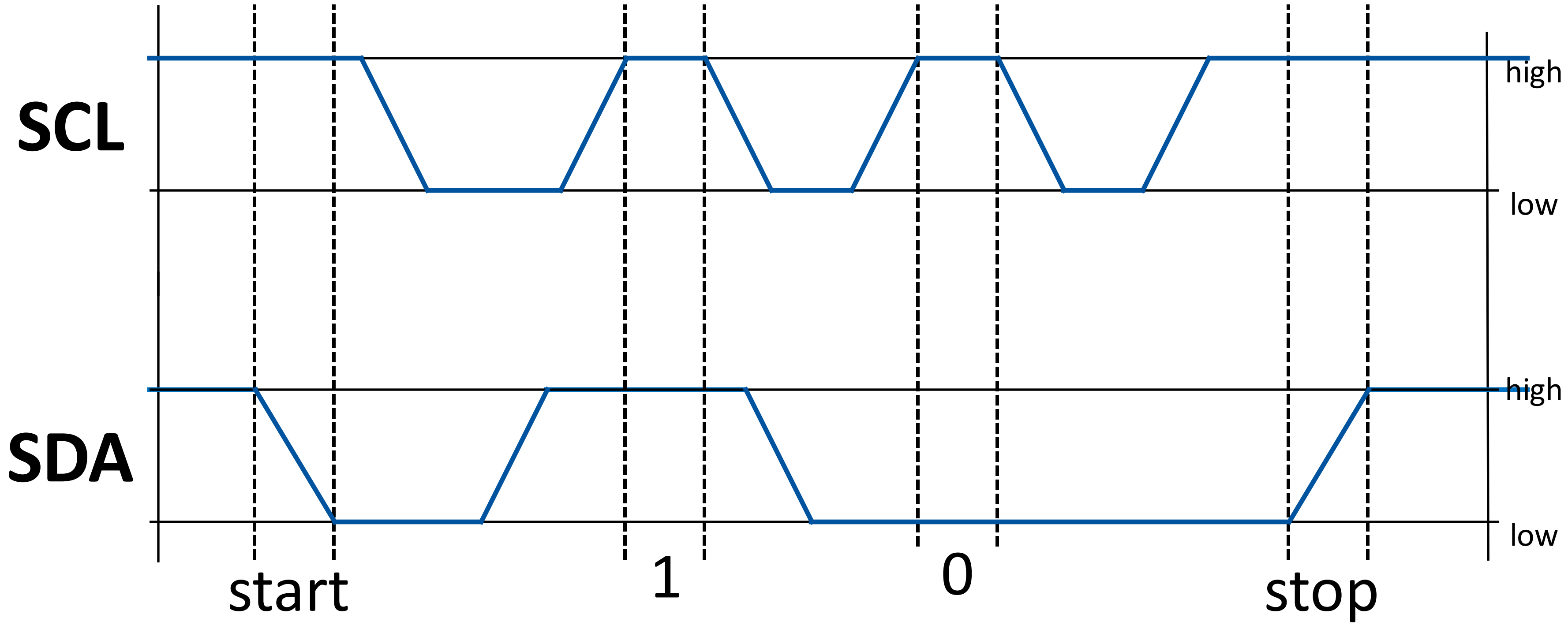
I²C – Physical Layer

- Two lines connected to pull-up resistors
  - SCL: serial clock line - SDA: serial data line
- Devices are connected via open connectors
- High level (logical 1): >0.7V (usually 3.3V – 5V)
- Low level (logical 0): -0.5V – 0.3V
- Maximum Capacity 400pF (few meters)
- Wired-AND (dominant 0)



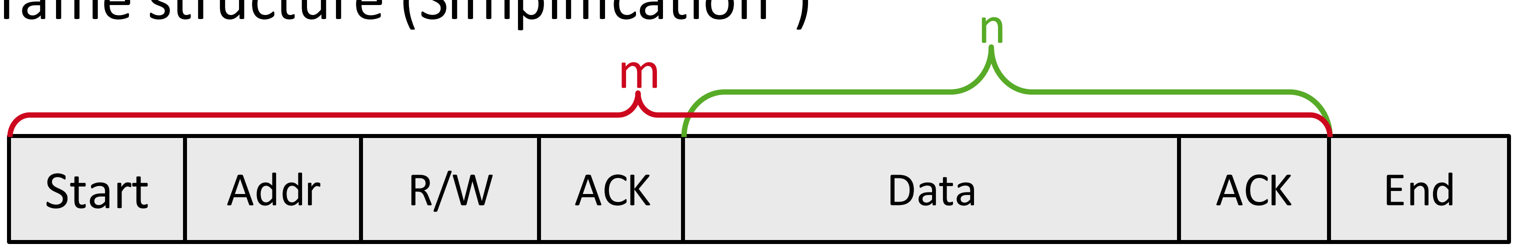
A sends 1.
B sends 0 and dominates.

I²C – Bit Encoding



I²C – Data Link Layer

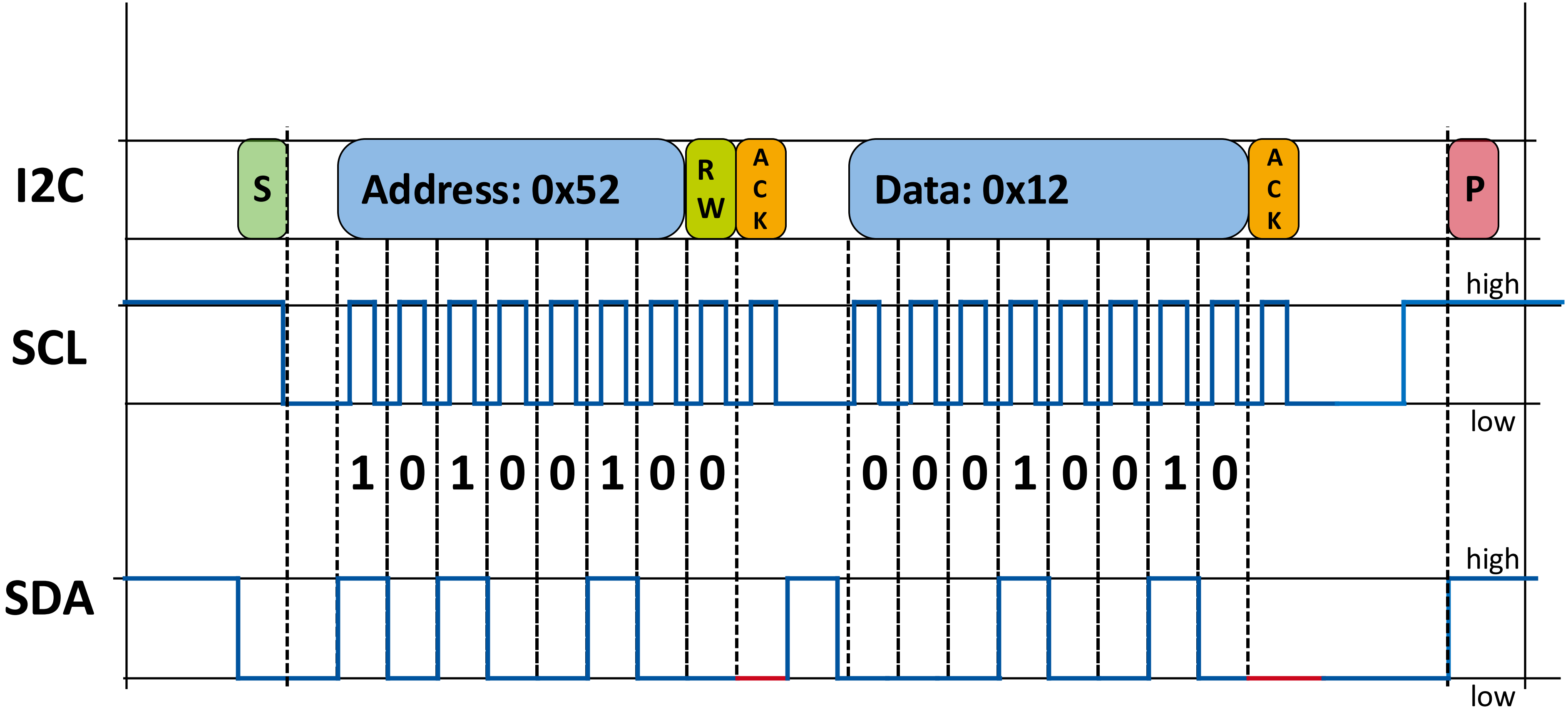
- Each device has a unique **7 bit** address (priority)
- Frame structure (Simplification*)



- Master-Slave principle
  - Master polls / pushes data
- Multi-master
  - arbitration by CSMA/CR: first 0 wins
- No error detection / correction
- Flow control by
  - Acknowledgement
  - Clock stretching

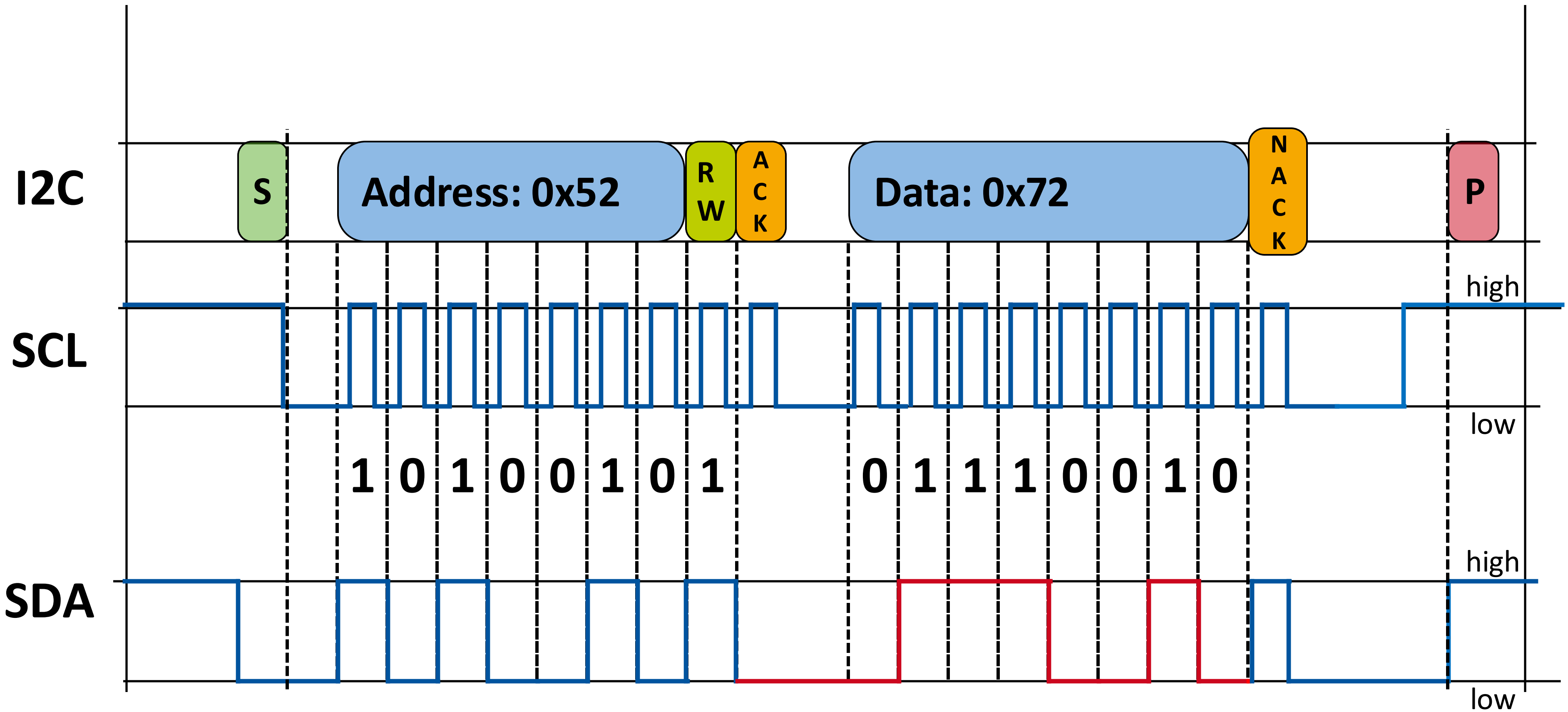
*I²C additionaly uses register addresses, these are excluded in the lecture

I²C – Example: Simple Write



Informatik 11 Embedded Software

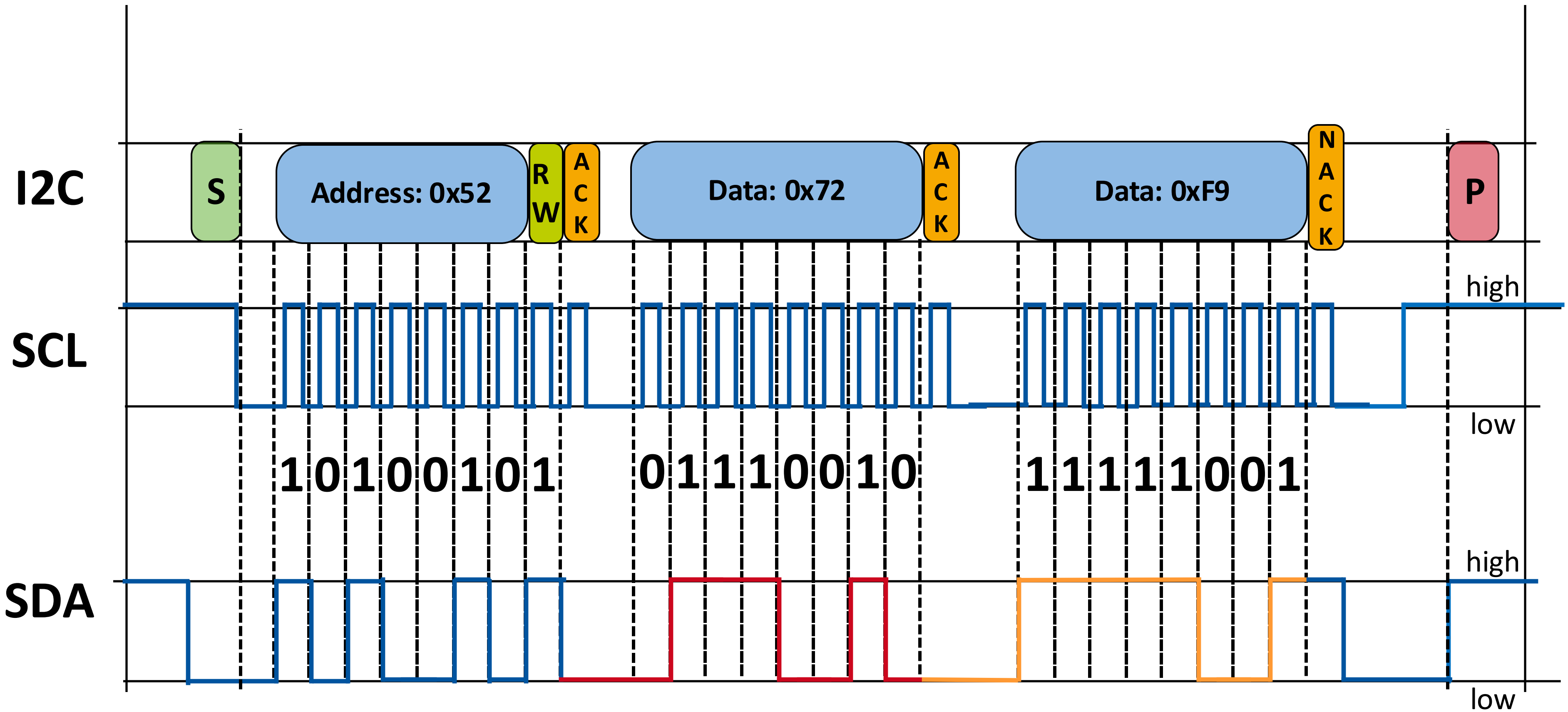
I²C – Example: Simple Read



- Master doesn't want to receive more bytes
  - → Sends NACK

Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY

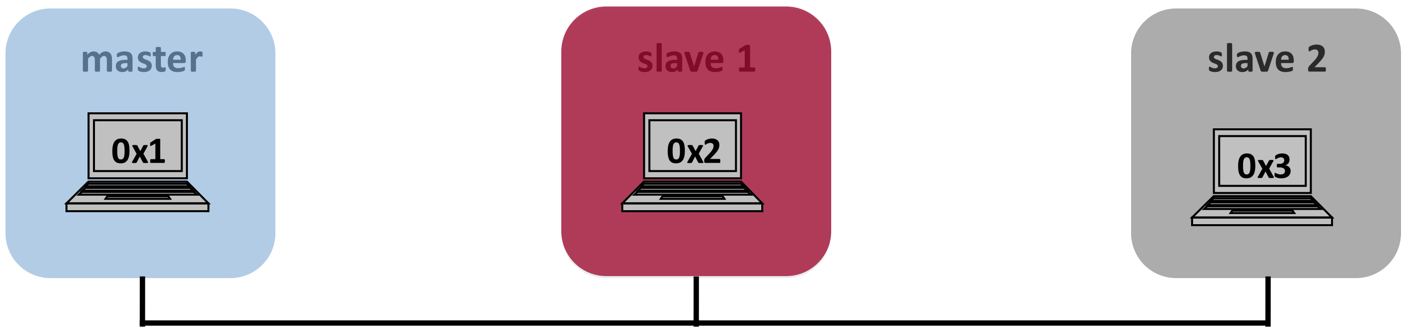
I²C – Example: Efficient Double Read

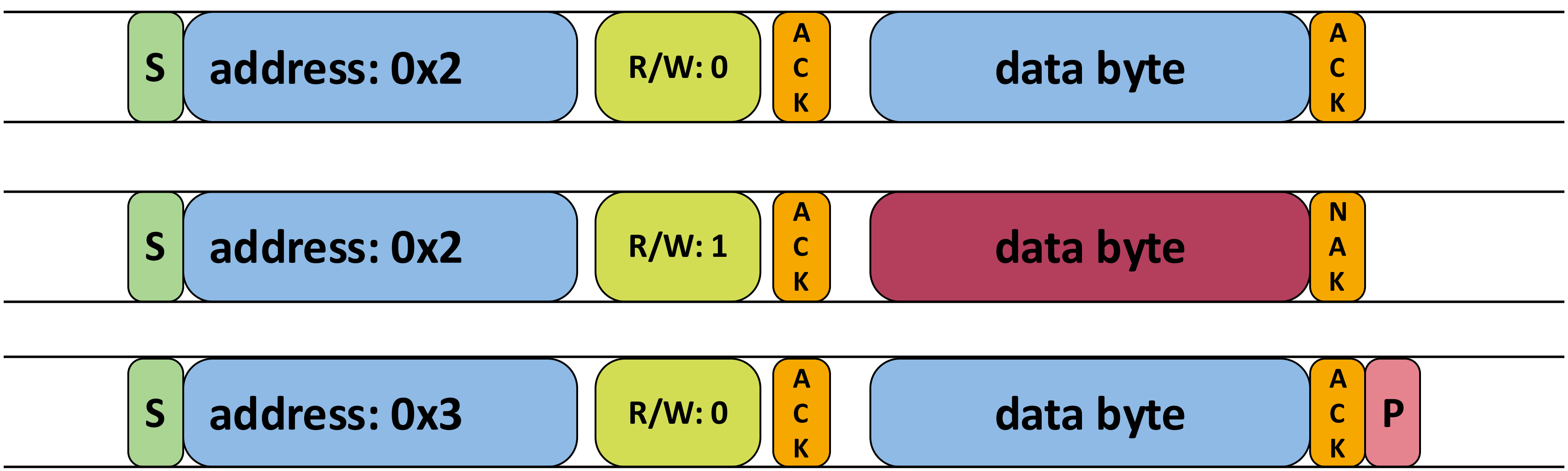


- Master doesn't want to receive more bytes
  - → Sends NACK

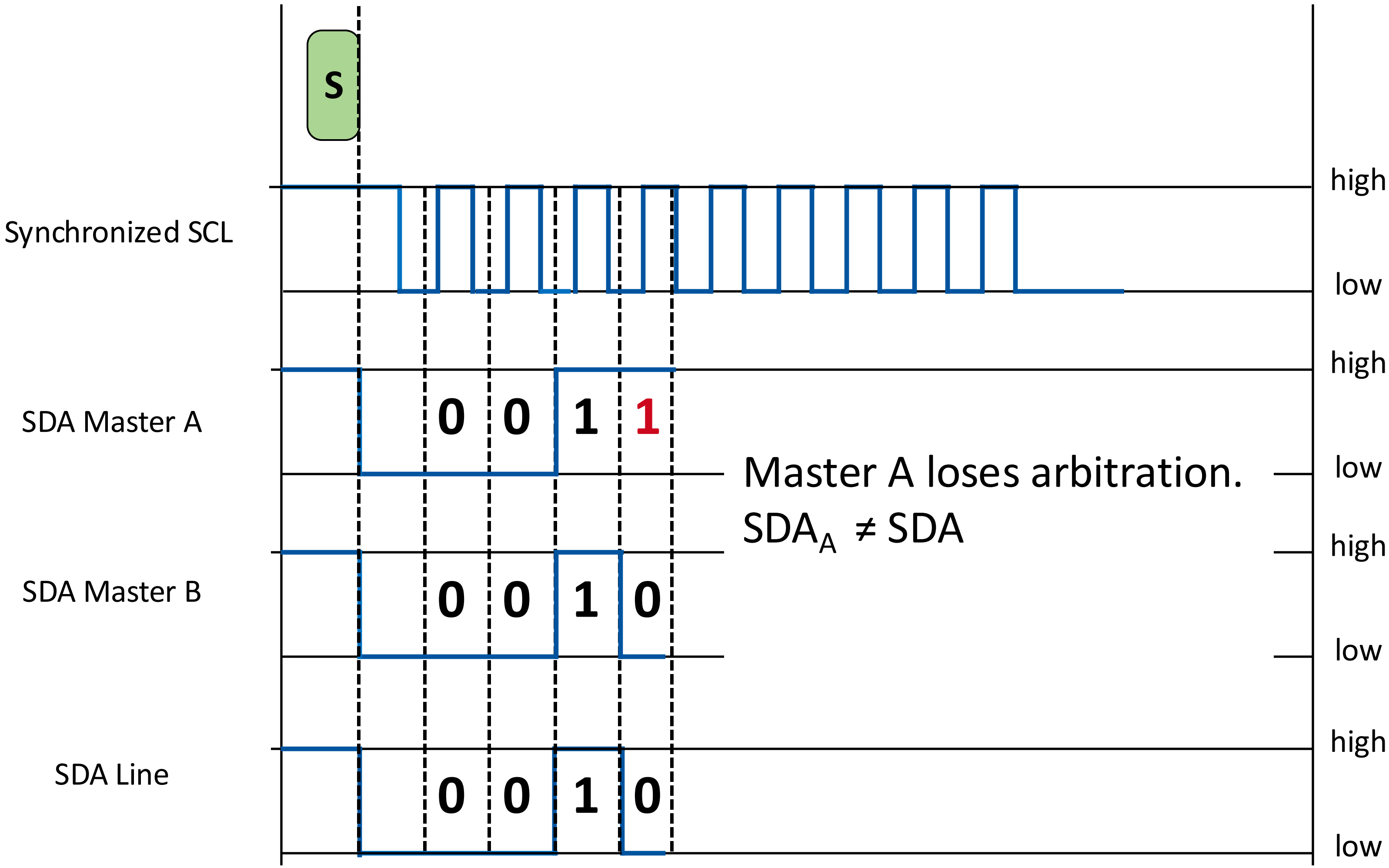
Informatik 11
Embedded Software

I²C – Example: Restart





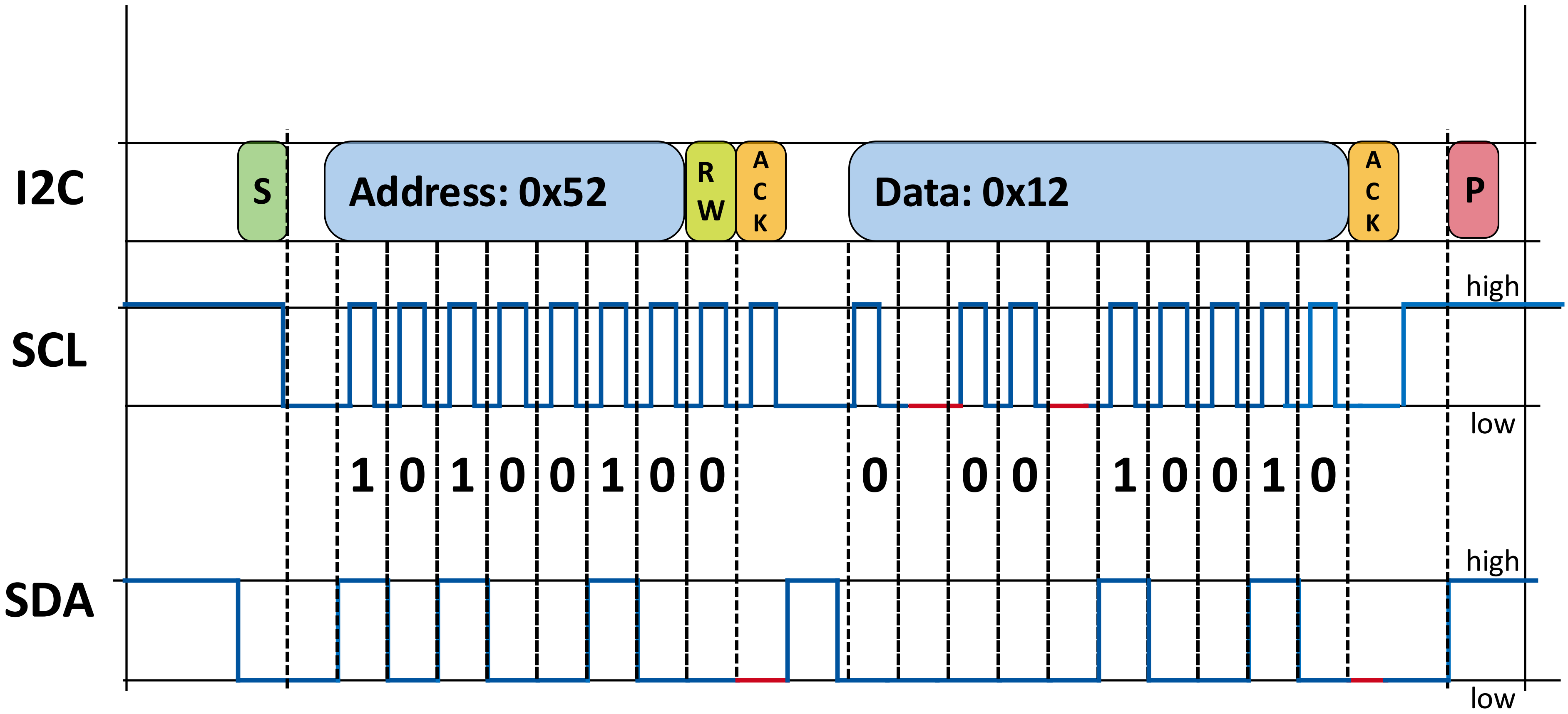
I²C – Example: Master Arbitration



Informatik 11 Embedded Software

RWTH AACHEN UNIVERSITY

I²C – Example: Clock stretching

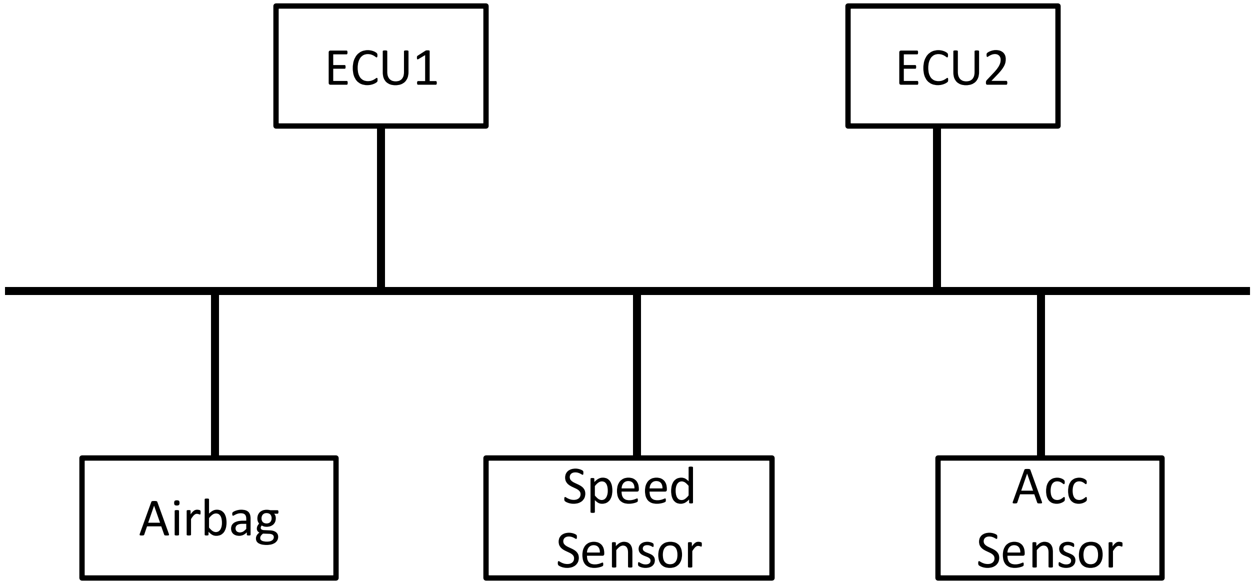


Informatik 11
Embedded Software
RWTH AACHEN UNIVERSITY

Controller Area Network (CAN) Bus

- Connects multiple controller units in harsh environments
  - Up to 5km at 10kbit/s
  - Up to 25m at 1Mbit/s
- Developed by Bosch in 1983
  - Reduce number and length of cables (weight / cost)
  - Reduce number and types of connectors (wiring errors)
- Standardized in ISO 11898
- Main application as automotive bus
- Also used as industrial field bus (CANopen)

CAN – Exemplary Setup



CAN – Physical Layer

- Shielded twisted pair (also: optical fiber)
- Comfort bus (low speed) also possible with single line
- Higher data rates use difference signals
  - Prevents common-mode interference (Gleichtaktstörung)
  - Can use optional third line CAN_GND
- Simple NRZ with bit stuffing after 5 equal bits
- Wired-AND (dominant 0; see I^2C)
- Usually up to 32 participants
- 64, 110, and 128 (with limitations) possible
- More participants with repeaters and bridges
- Bus termination with 120Ω resistor

CAN – Data Link Layer

- Each message type has unique identifier (priority)
- Devices have no address
- 4 frame types
- Arbitration by CSMA/CR: first 0 wins
- CRC (15 bit) error detection

CAN – Object Identifier

- Unique bit mask
  - 11 bits: base frame format (CAN 2.0A)
  - 29 bits: extended frame format (CAN 2.0B)
- Each object ID should only be sent by one device
- Each device can have multiple object IDs
- Object ID is used for arbitration
- Assignment of object IDs is fixed in specification
  - CAN matrix document
  - Reserve object IDs for future extensions

CAN – Matrix

- Message: Motor 1
- Identifier: 0x0280
- Length: 8 byte
- Receiver: all
- Data rate: 500kBaud
- Periodicity: 10ms

Signal	Byte	Bit	Init	Range	Interpretation
…	…	…	…	…	…
Pedal check	1	1			0: pedal ok 1: use default value
Kick-down switch	1	2			0: no kick-down 1: kick-down
…	…	…	…	…	…

CAN – Frames

- **Data frame**

1 Bit	11 Bits	1 Bit	2 Bit	4 Bit	0…64 Bit	16 Bit	2 Bit	7 Bit
SOF	ID	RTR=0	R Bits	DLC	Data	CRC	ACK	EOF

- **Remote frame**

1 Bit	11 Bit	1 Bit	2 Bit	4 Bit	16 Bit	2 Bit	7 Bit
SOF	ID	RTR=1	R Bits	DLC	CRC	ACK	EOF

SOF= Start of Frame

RTR= Remote transmission request

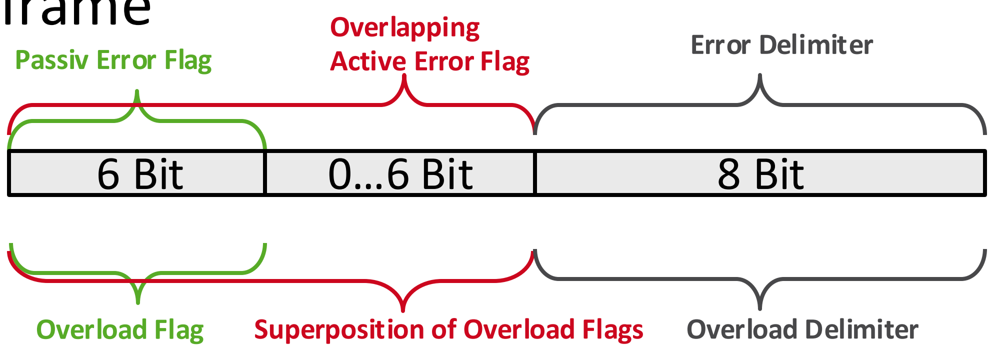
R Bits = two reserved Bits

DLC = Data Length Control

CRC = Cyclic Redundancy Check

EOF = End of Frame

- **Error frame**



- Overload frame (analog to active error frame)
- Interframe space: three recessive bits (pause)

CANopen

- Application layer protocol
- Based on CAN bus
- Developed ESPRIT project (lead developer: Bosch)
- Since 1995 by CAN in Automation (CiA)
- EN 50325-4
- Four basic services
  - Request: application requests service
  - Indication: system notifies application of event
  - Response: application replies to an indication
  - Confirmation: system confirms service execution

FlexRay

- Automotive network communication protocol
- Developed FlexRay consortium (2000)
- Core partners
  - BMW
  - Daimler
  - Motorola (Freescale)
  - Philips (NXP Semiconductors)
  - Later: Bosch, General Motors, Volkswagen
- Deterministic timing → real time capable (X-by-wire)
- Hybrid MAC: TDMA + dynamic part (reservation)
- Integrates parts of ByteFlight protocol (BMW)

FlexRay – Exemplary Setup

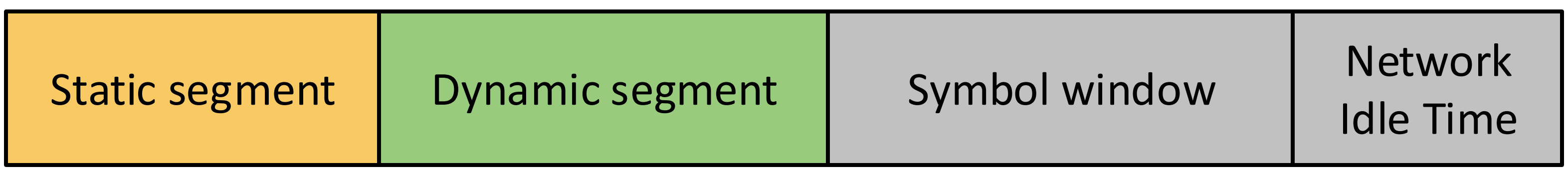
- 2 channels with up to 10Mbit/s each
  - Use both channels for redundancy
  - Use single channel for higher data rate
- Distributed clock synchronization (no master)
- Bus, star, and star with buses topologies

FlexRay – Physical Layer

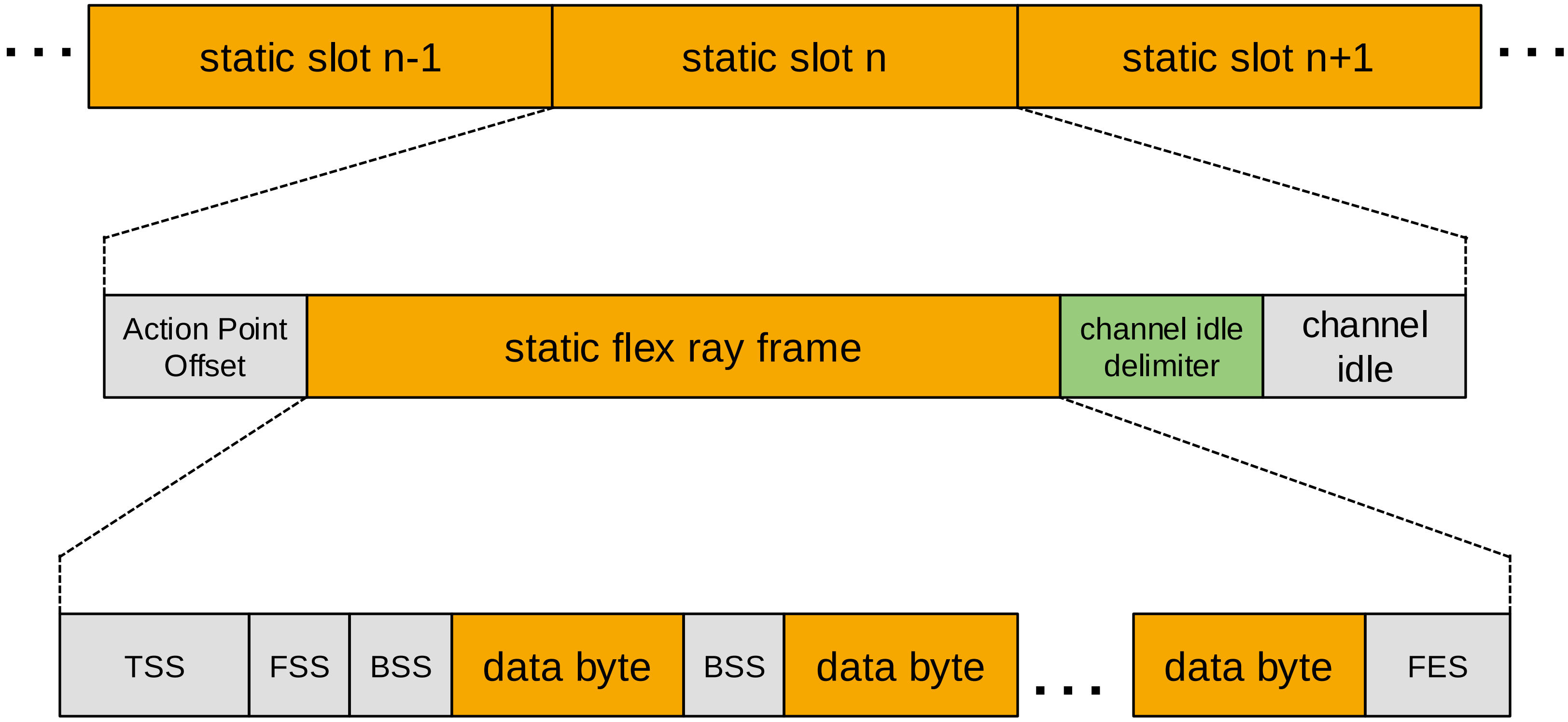
- Shielded twisted pair
- NRZ
  - High 3.5 V
  - Low 1.5 V
  - Idle 2.5 V
  - Suspension 0 V
- Clock drift < 0.15% compared to reference clock
- Maximum signal delay < 2.5µs
- 8 samples per bit (majority vote of 5 samples)

FlexRay – Data Link Layer

- Repeating communication cycle
- Static part for real-time communication (TDMA)
- Dynamic part for other communication (FTDMA)
- CRC (11 bit) error detection for header
- CRC (24 bit) error detection for payload

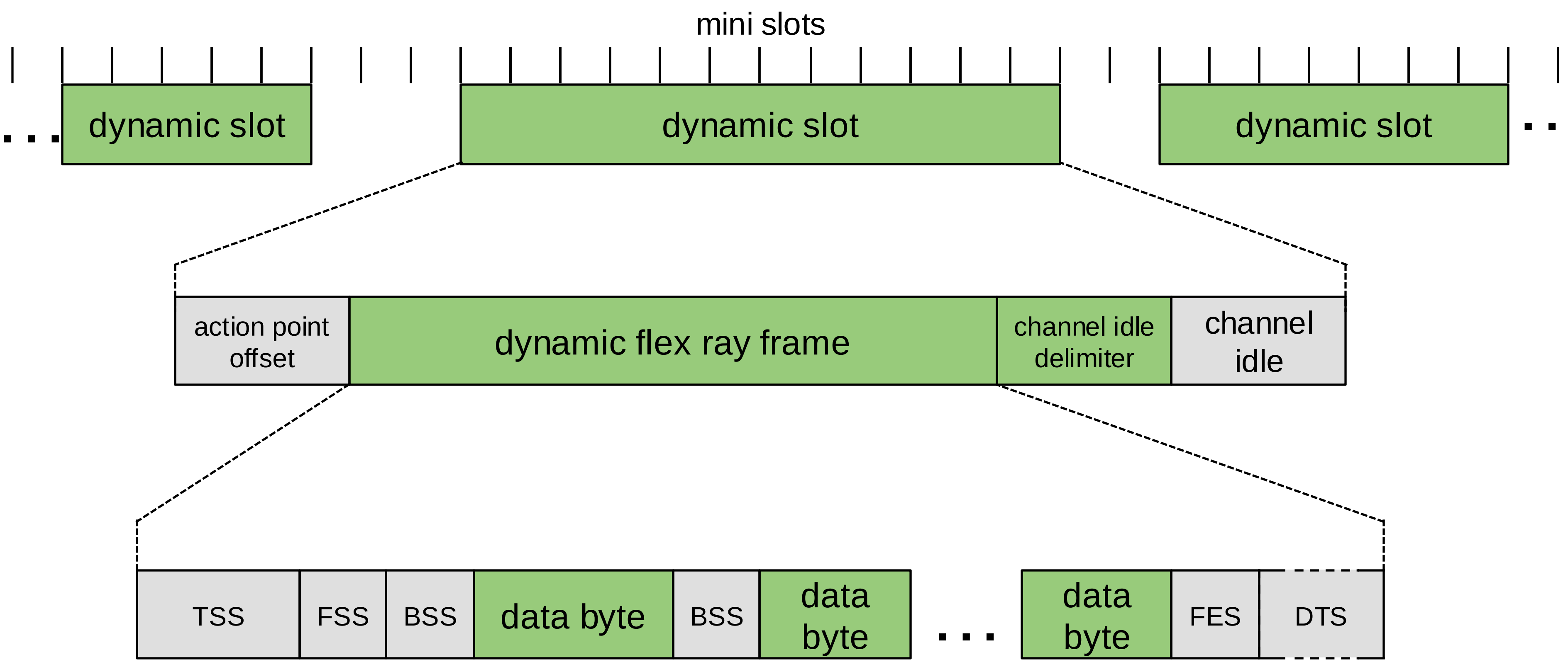


FlexRay – Static Slot



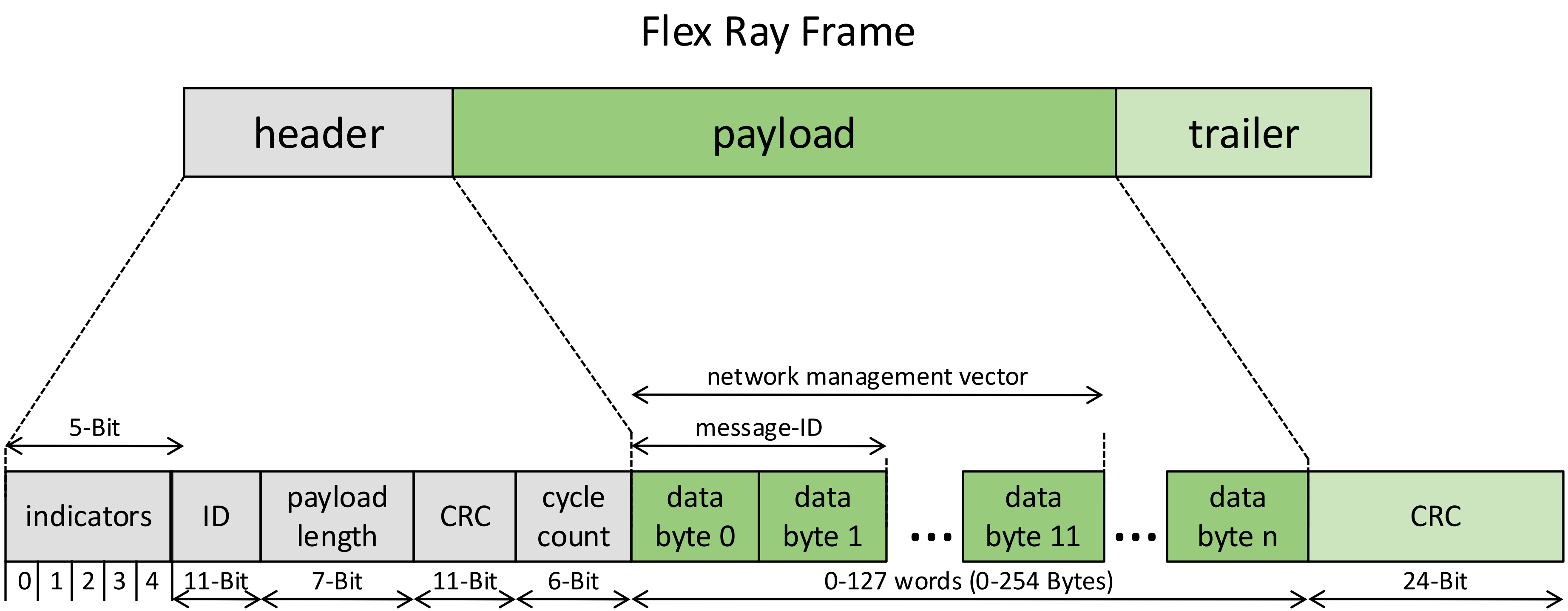
TSS: Transmission Start Sequence
FSS: Frame Start Sequence
BSS: Byte Start Sequence
FES: Frame End Sequence

FlexRay – Dynamic Slot



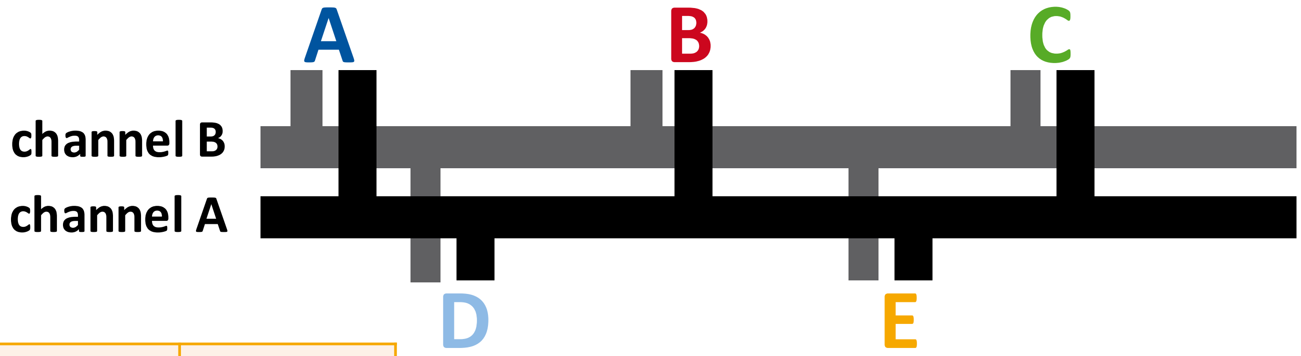
TSS: Transmission Start Sequence
FSS: Frame Start Sequence
BSS: Byte Start Sequence
FES: Frame End Sequence
DTS: Dynamic Trailing Sequence

FlexRay – Payload

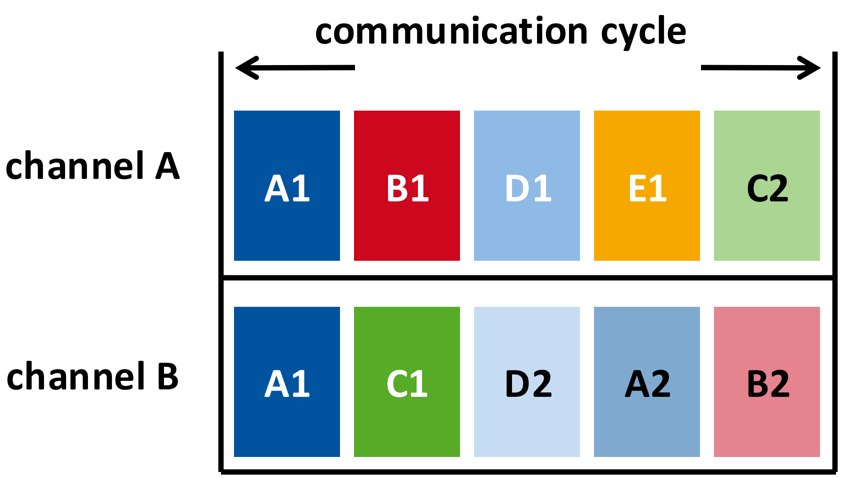


0: reserved
1: payload preamble indicator
2: sync frame indicator
3: null frame indicator
4: startup frame indicator

FlexRay – Static Segment

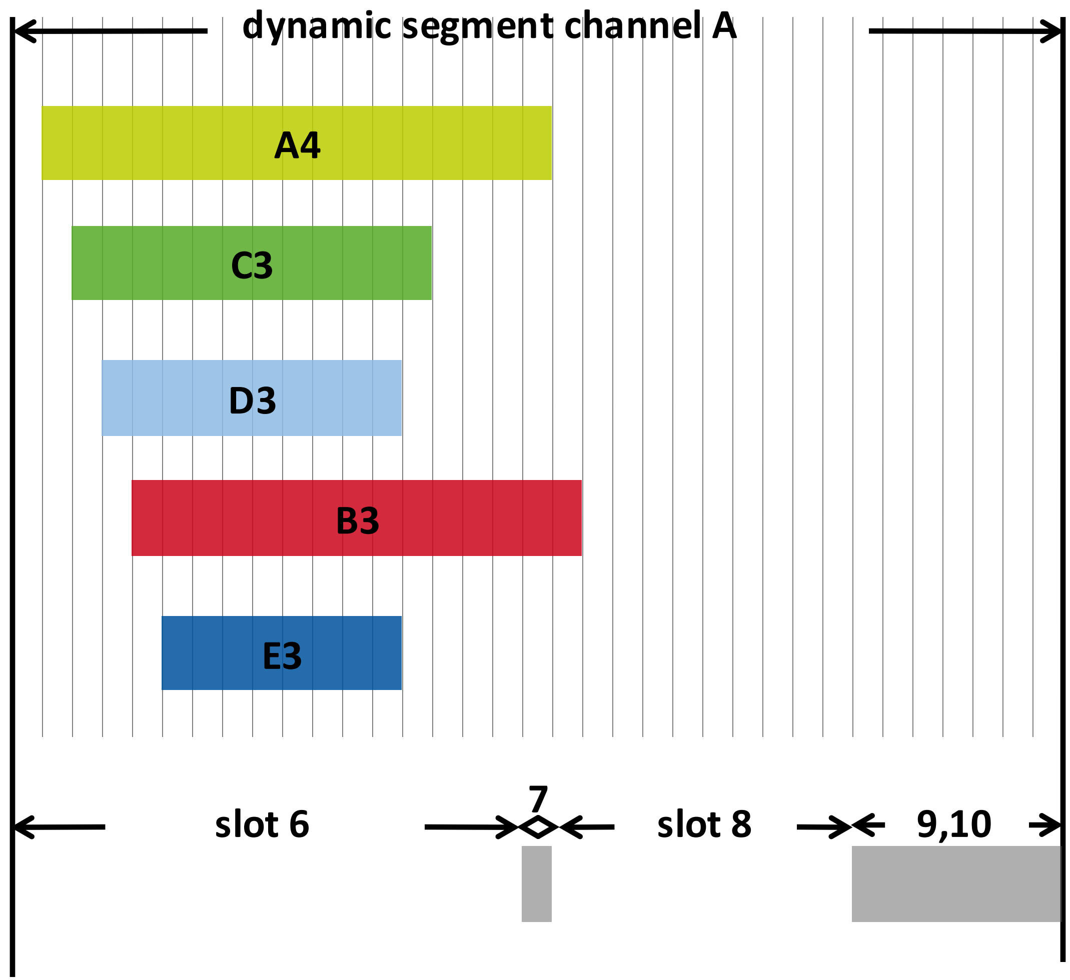


slot	node	message	channel
1	A	A1	A
		A1	B
2	B	B1	A
	C	C1	B
3	D	D1	A
		D2	B
4	E	E1	A
	A	A2	B
5	C	C2	A
	B	B2	B



FlexRay – Dynamic Segment

slot	node	message	event
6	A	A4	
7	C	C3	
8	D	D3	
9	B	B3	
10	E	E3	

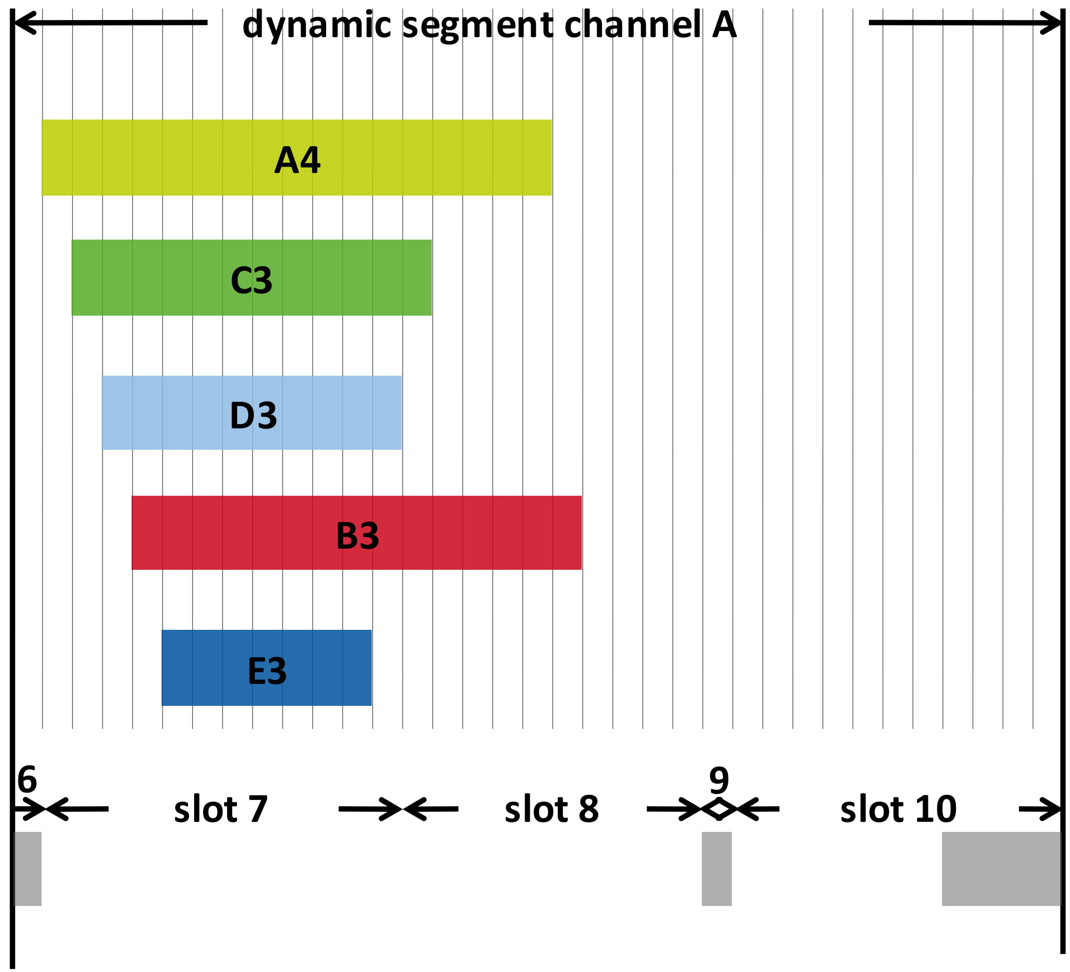


resulting dynamic slots:

- slots 7,9 and 10 are filled with minislots
- message B3 doesn't fit into the dynamic segment

FlexRay – Dynamic Segment

slot	node	message	event
6	A	A4	
7	C	C3	
8	D	D3	
9	B	B3	
10	E	E3	



resulting dynamic slots:

- slots 6 and 9 are filled with minislots
- all triggered messages can be send

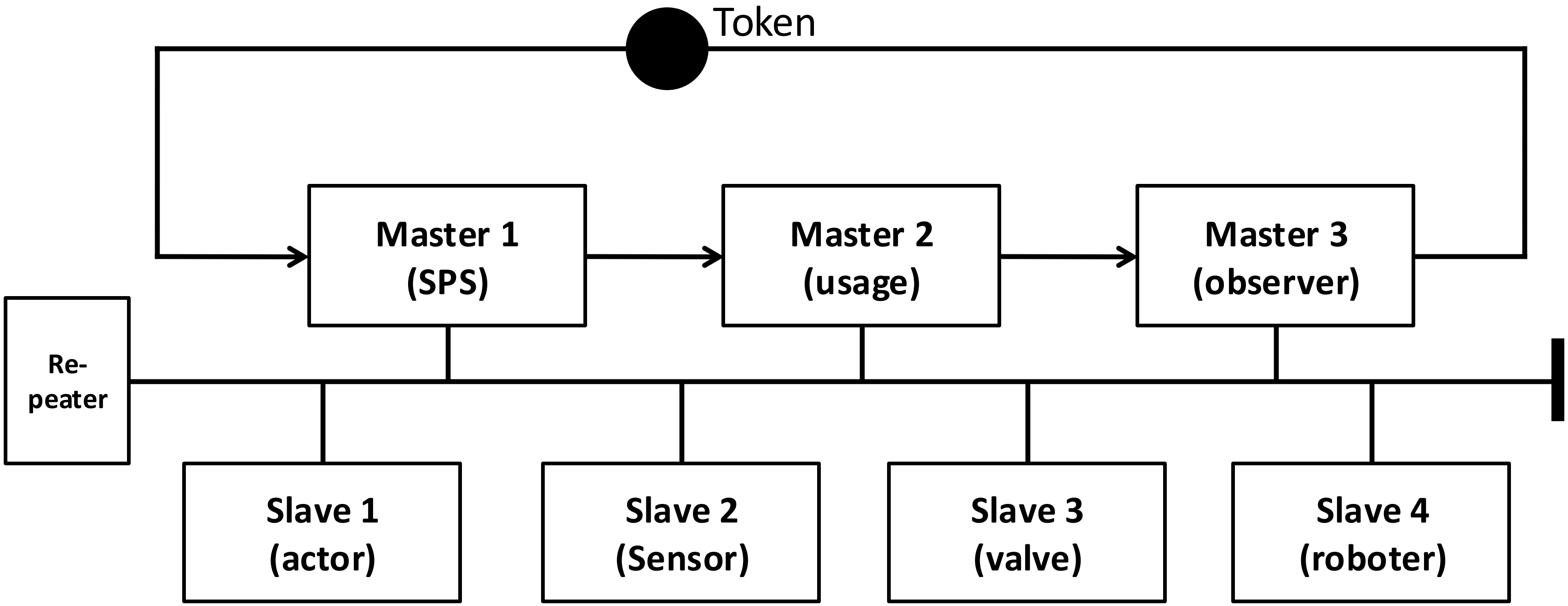
Process Field Bus (PROFIBUS)

- Field bus for automation in industrial environments
- Publicly founded German research project (BMBF)
- “Dezentrale Peripherie” (DP) (1993)
  - Focus on central controller using remote sensors and actors
  - Simple, fast (12Mbit/s)
- “Prozessautomation” (PA)
  - Limited current (explosion protection)
  - Very slow (31.25kbit/s)
- “Fieldbus Message Specification” (FMS) (1991)
  - First version
  - Very complex, replaced by DP
- IEC 61158 / IEC 61784

PROFIBUS-DP – Exemplary Setup

single master setup

multi master setup



PROFIBUS-DP – Physical Layer

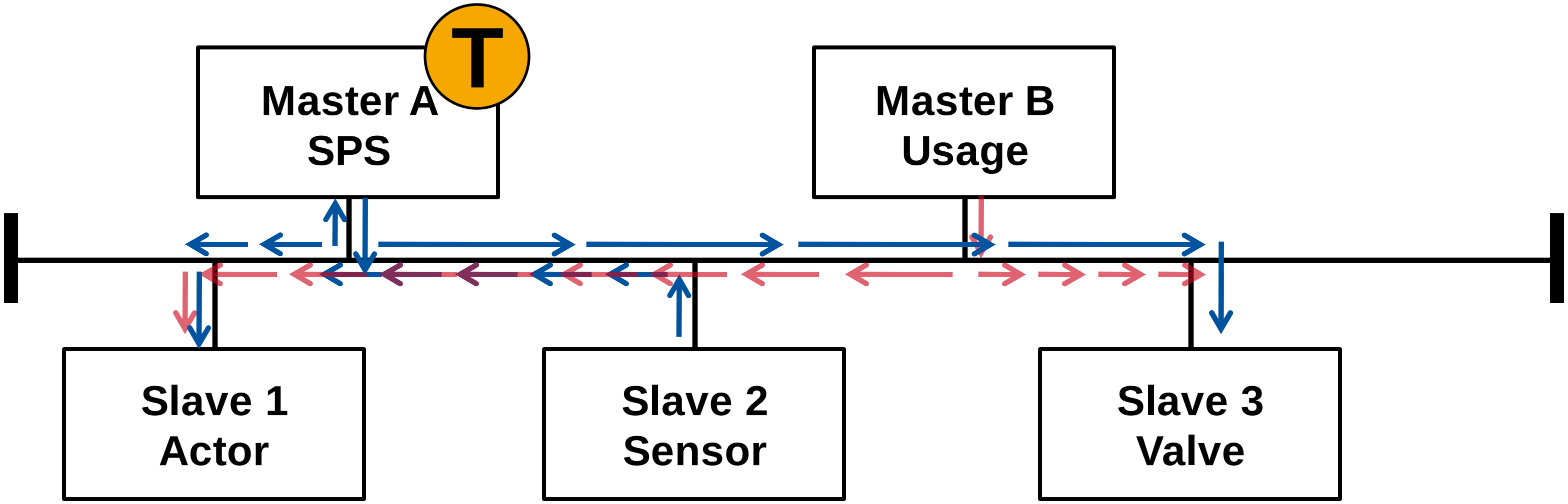
- RS-485
  - Shielded twisted pair
  - 9600 bit/s – 12Mbit/s
  - Bus topology with 150Ω terminators
  - 100m – 1200m between repeaters (depends on data rate)
  - NRZ
- Optical fiber
  - Star, bus, or (redundant) ring topology
  - Up to 15km between repeaters

PROFIBUS-DP – Data Link Layer

- Fieldbus Data Link (FDL)
- Each device has 7 bit address
- 5 frame types
- Master polls / pushes data
- Multi-master by token passing
- CRC (8 bit) error detection
- Hamming distance of 4 for delimiters

PROFIBUS-DP – Token Passing

SPS controls actor using latest sensor data



PROFIBUS-DP – Frames

- No Data

SD 1	DA	SA	FC	FCS	ED

- Variable data length
- Fix data length
- Token
- Short Confirmation

SD 1: *Start delimiter*, signals type of data

DA: *Destination adress*

SA: *Source adress*

FC: *Function code*, extension of data type

FCS: *Frame checking sequence,* error handling

ED: End delimiter

PROFIBUS-DP – Frames

- No Data

SD 1	DA	SA	FC	FCS	ED

- Variable data length

SD 2	LE	LEr	SD 2	DA	SA	FC	DSAP	SSAP	PDU	FCS	ED

- Fix data length
- Token
- Short Confirmation

LE: *Length of PDU*

LEr: *Repetition of LE*

DSAP: *Destination Service Access Point*

SSAP: *Source Service Access Point*

PDU: *Protocol Data Unit*

PROFIBUS-DP – Frames

- No Data

SD 1	DA	SA	FC	FCS	ED

- Variable data length

SD 2	LE	LEr	SD 2	DA	SA	FC	DSAP	SSAP	PDU	FCS	ED

- Fix data length

SD 3	DA	SA	PDU	FC	FCS	ED

- Token

- Short Confirmation

PROFIBUS-DP – Frames

- No Data

SD 1	DA	SA	FC	FCS	ED

- Variable data length

SD 2	LE	LEr	SD 2	DA	SA	FC	DSAP	SSAP	PDU	FCS	ED

- Fix data length

SD 3	DA	SA	PDU	FC	FCS	ED

- Token

SD 4	DA	SA	ED

- Short Confirmation

PROFIBUS-DP – Frames

- No Data

SD 1	DA	SA	FC	FCS	ED

- Variable data length

SD 2	LE	LEr	SD 2	DA	SA	FC	DSAP	SSAP	PDU	FCS	ED

- Fix data length

SD 3	DA	SA	PDU	FC	FCS	ED

- Token

SD 4	DA	SA	ED

- Short Confirmation

SC

PROFIBUS-DP – Application Layer

- DPV0
  - Original specification
  - Cyclic exchange of data
  - Automation technology
- DPV1
  - Acyclic data communication
  - Alarm management
  - Chemical engineering
- DPV2
  - Isochronous data communication
  - Slave to slave communication
  - Robot control

Literature

- Vector E-Learning: http://www.vector-elearning.com
- PROFIBUS Handbuch: http://www.profibus.felser.ch/
- Heinz Wörn, Uwe Brinkschulte: Echtzeitsysteme, Springer-Verlag, 2005. ISBN 3-540-20588-8.

Summary

- All data communication follows basic principles
  - Limitation by bandwidth
  - Degradation due to damping
- Embedded data communication has special demands
  - Cheap and simple
  - Real-time capable
  - Robust
- Physical Layer defines how bit streams are transported
  - Mechanical and electrical characteristics
  - Bit encoding
  - Synchronization

Summary

- Data Link Layer defines how messages are transported
  - Frame formats
  - Medium access
  - Error correction and flow control
- Different areas of application use fitting protocols
  - I^2C for intra board communication
  - CAN for intra car communication
  - FlexRay for intra car communication with real time and higher data rates
  - PROFIBUS for industrial controllers