IoT Communication

Karel Slavicek
Vaclav Oujezsky

2024

Outline

- Overview of communication busses
- I2C
- SPI
- 1-Wire communication
- MODBUS
- CANbus

Structure of an IoT System

Internal Communication

- **SPI**
  - SD
- **I2C**
  - TWI
  - SMBus
- **1-Wire**
  - UPDI
- CSI, I2S, ...

External Communication

- Structure
- **MODBUS**
- **CANBUS**
- M-BUS
- FLEXRAY
- ARIC-429

Communication Busses Structure

- ISO-OSI reference model
- Physical Layer
- Coding and Modulation
- Data Units
- Conversation Protocols
- Number of nodes
- Transmission speed

ISO-OSI Reference Model

7. Application layer

6. Prezentation layer

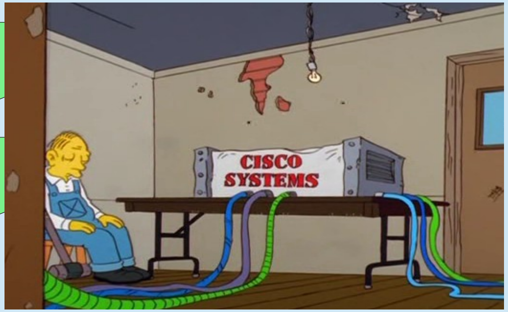
5. Session layer

4. Transport layer

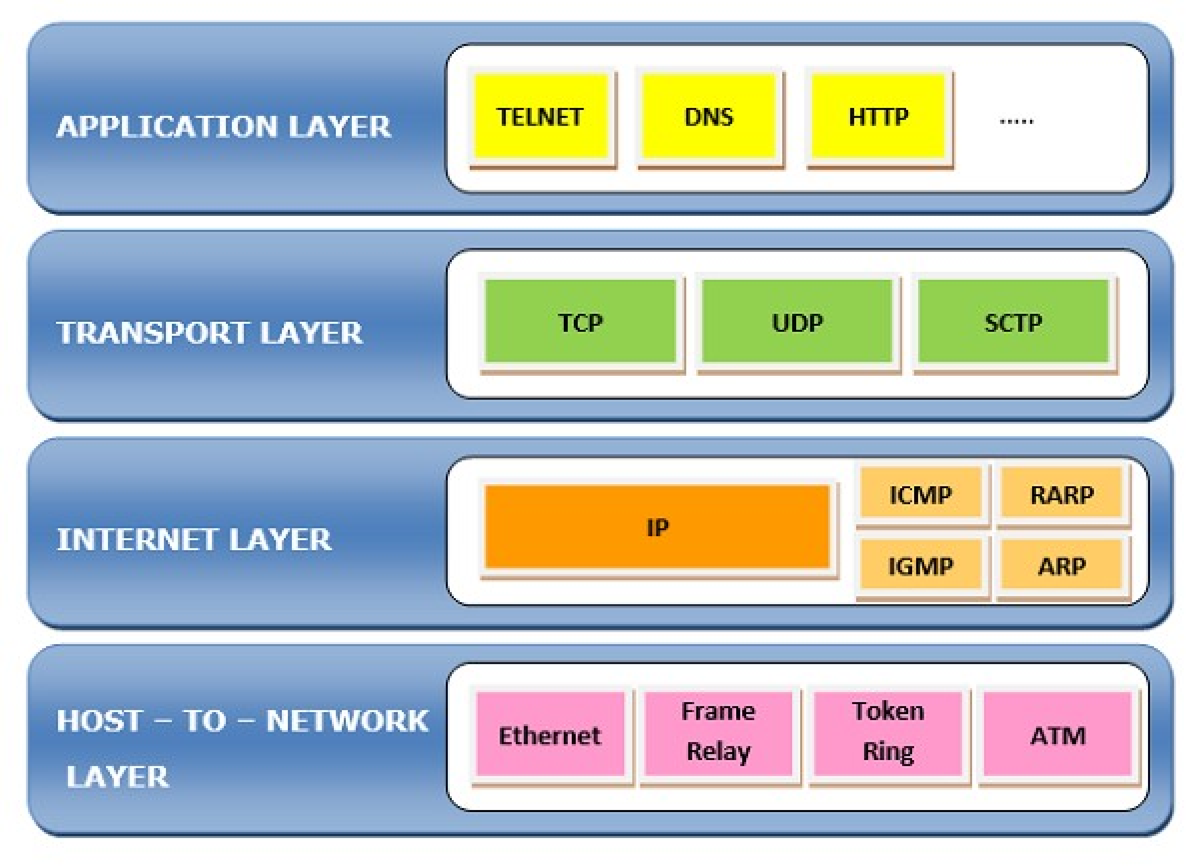
3. Network layer

2. Link layer

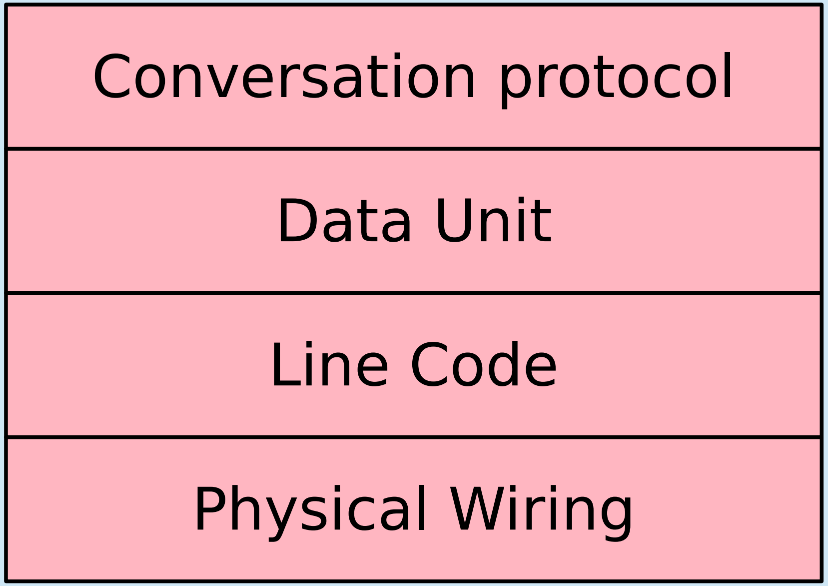
1. Physical layer



TCP/IP Reference Model



Industrial busses

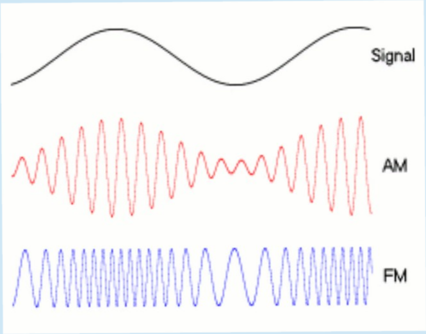


Physical Layer

- Number of conductors (wires)
- Unipolar / Bipolar / Differential signal
- Twisted pair
- Clock recovery usually homodyne = from the received signal or on separate conductors

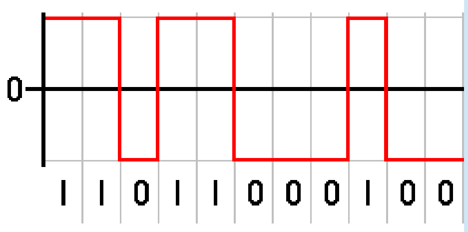
Coding and Modulation

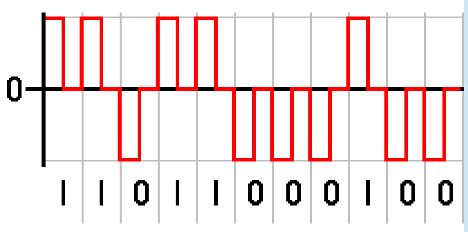
- A tour to domain of Faculty of Electrical Engineering
- Coding = translation of data (usually 8-bit) word to codeword transported via communication media
- Modulation = process of varying properties of a periodic waveform (carrier signal) with a separate signal that contains information to be transmitted

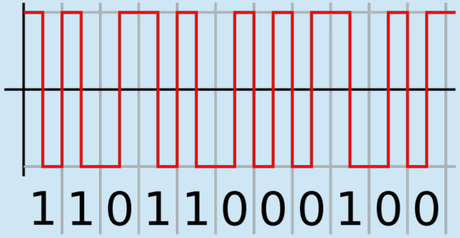


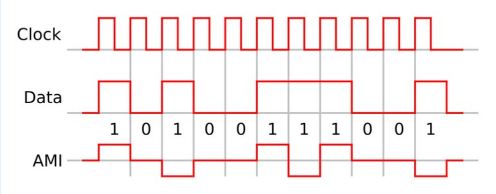
Line codes

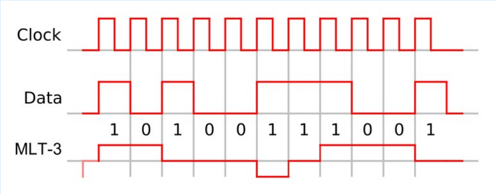
- In case of transmission in base band
- Pattern of voltage, current, or photons used to represent digital data transmitted
- Most common: RZ, **NRZ**, Manchester, HDB3, AMI











Coding and Modulation

- Coding commonly not used
- Line code NRZ
- Symbols for logical „1“, logical „0“
- Special symbol for start/end of the data unit
- Optional Parity bit

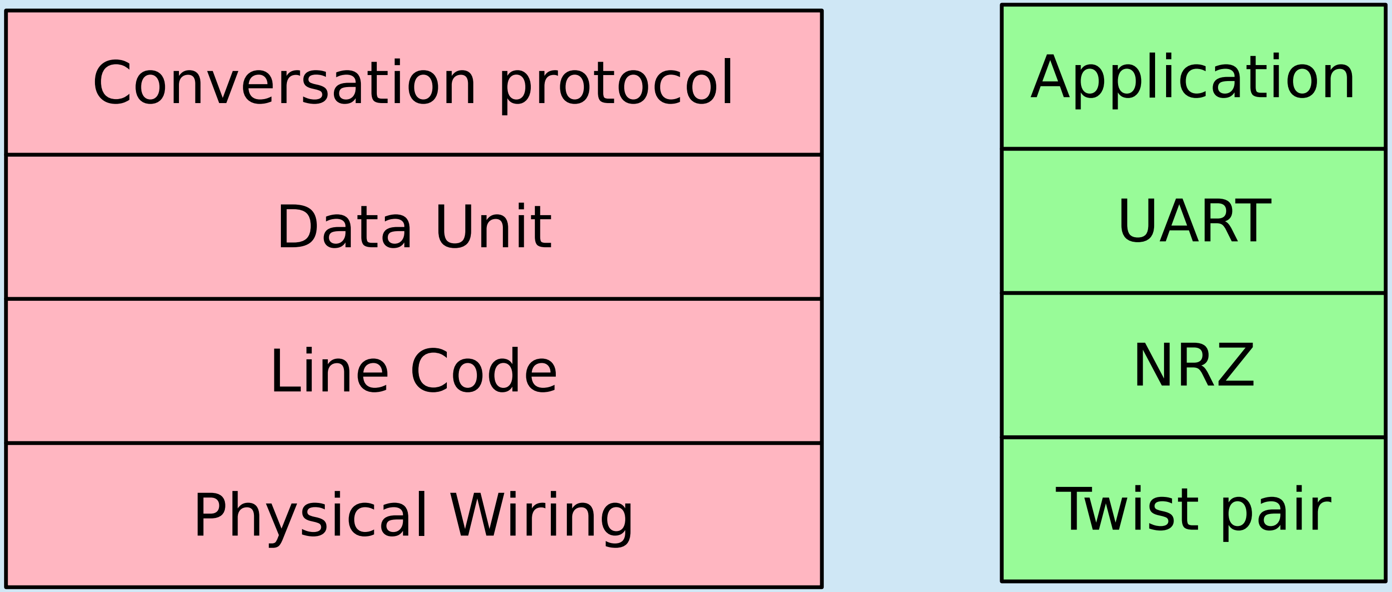
Internal Communication

The main branches

- I2C – two wire bus
- SPI – performance
- 1-Wire – minimizing

Historical remarks

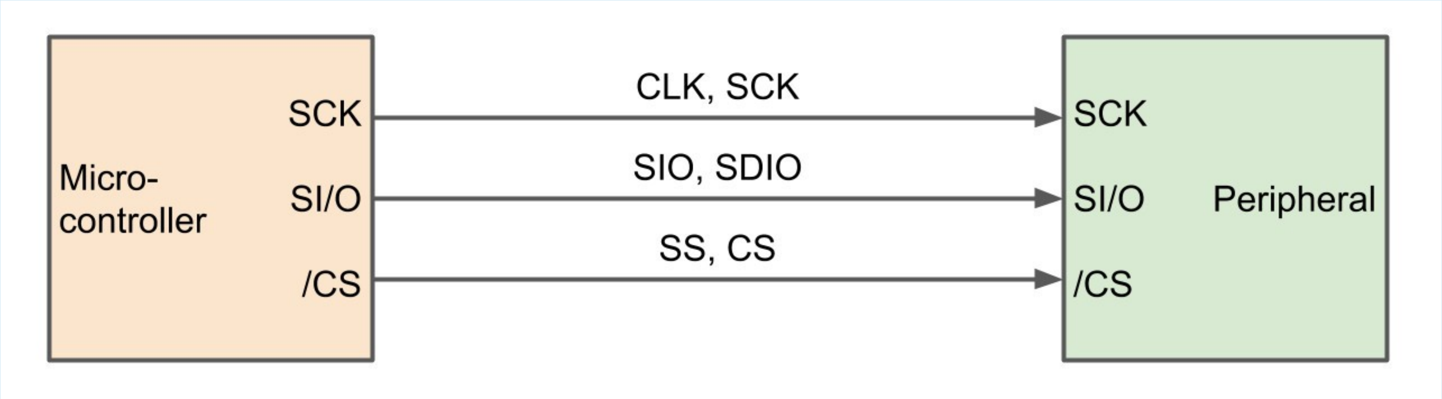
- These protocols founded later then RS-232 (UART),...
- Designed for small distance
- Original idea to simplify and unify internal structure of devices
- Evolution driven by new applications


Conversation protocol
Data Unit
Line Code
Physical Wiring
Application
UART
NRZ
Twist pair

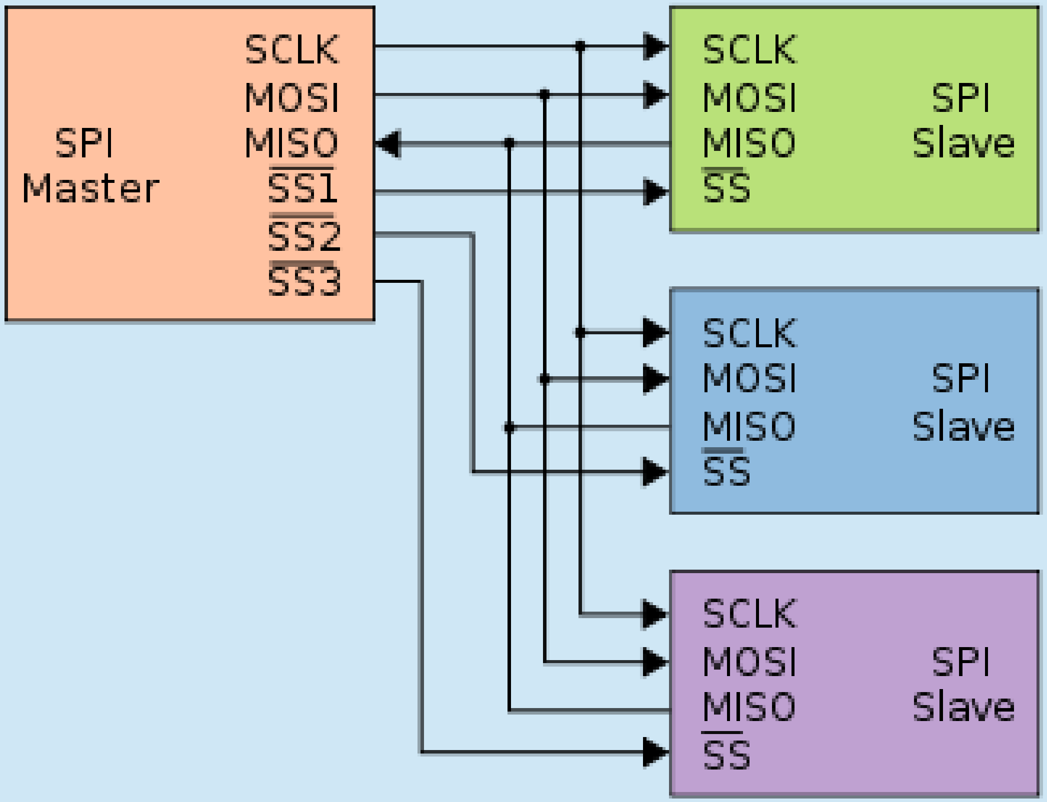
SPI

- Serial Peripheral Interface
- Motorola 1980
- Can be full-duplex
- For higher throughput
- Small number of devices
- Addressing realized by separate wires

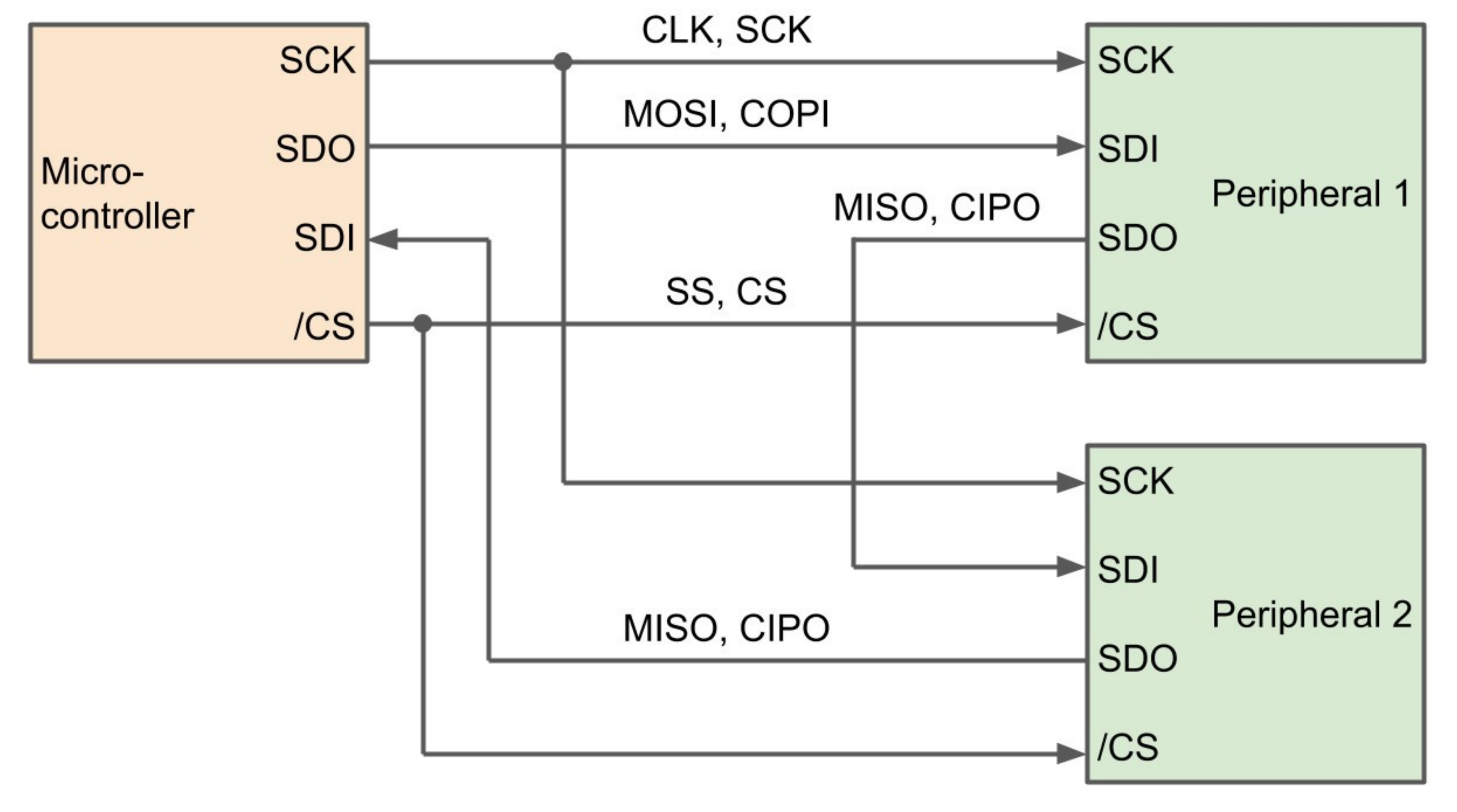
SPI – physical layer – topology



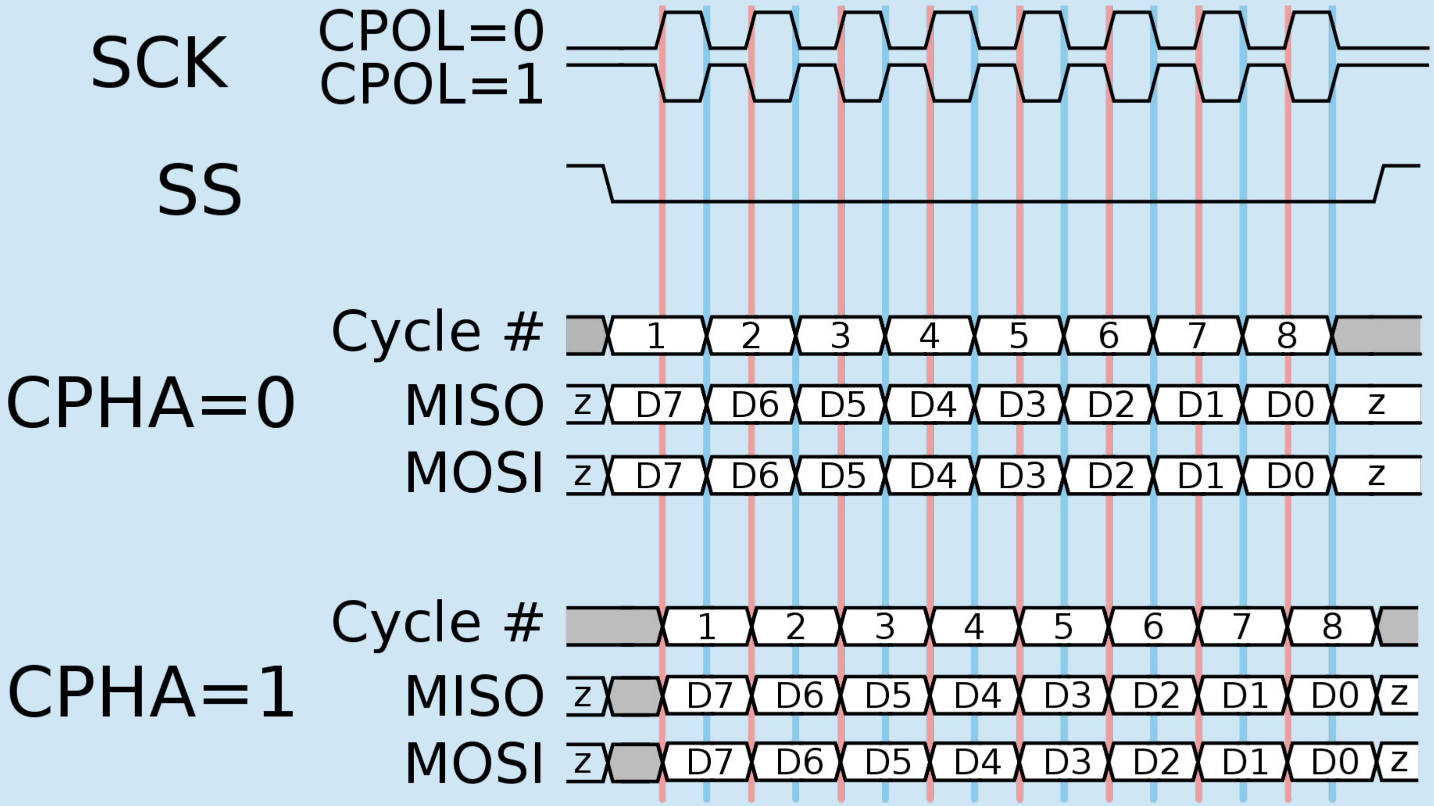
SPI – physical layer – topology



SPI – physical layer – topology



SPI – physical layer



SPI – physical layer

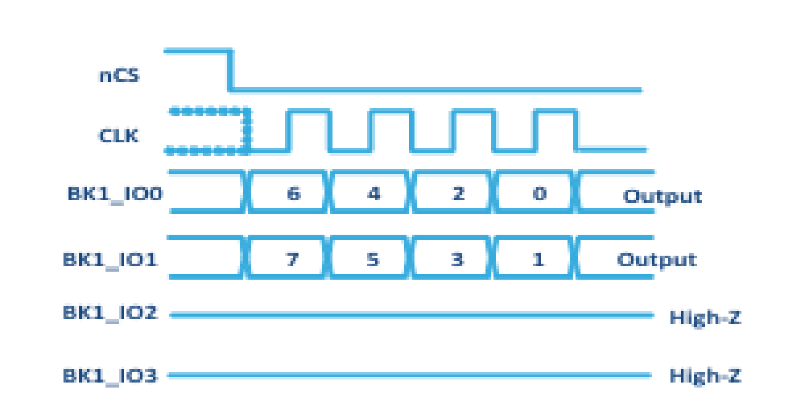
- 1 – 10 Mhz
- 8 / 16 bit
- 3.3 V / 5 V

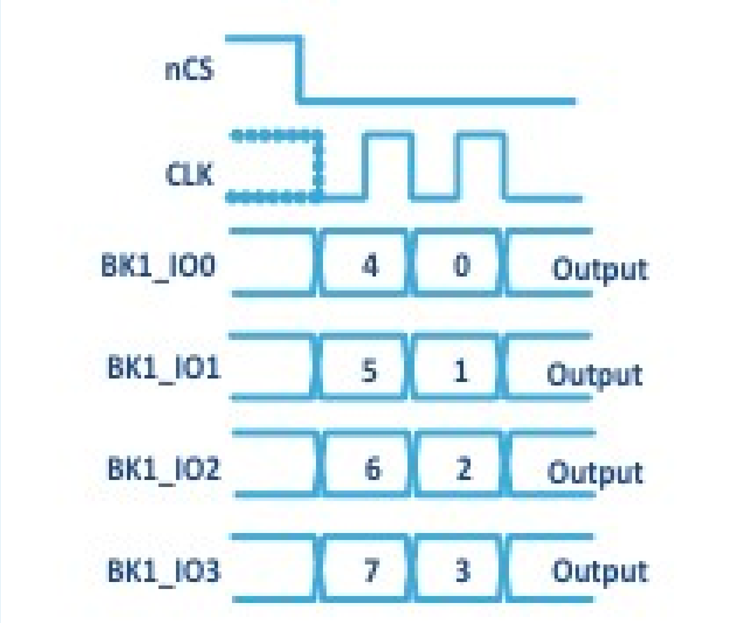
SPI Modes

Mode	CPOL	CPHA
0	0	0
1	0	1
2	1	0
3	1	1

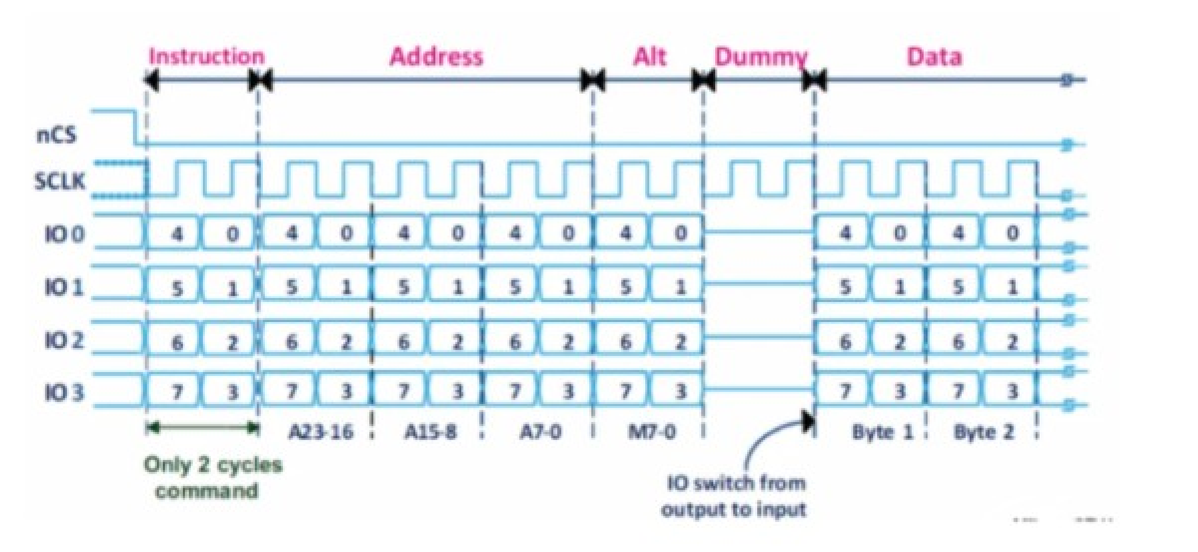
SPI – related protocols

- Dual SPI
- Quad SPI (QSPI)
-

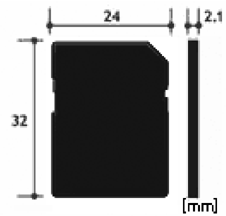
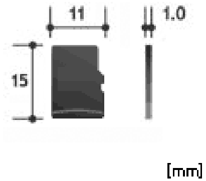




SPI for memory access



SD cards

Form Factor		SD	microSD
Dimension			
Card Capacity Type		SD, SDHC, SDXC and SDUC	
Physical	Number of pins	High Speed and UHS-I : 9 pins UHS-II and UHS-III: 17 pins SD Express 1-lane: 17-19 pins SD Express 2-lane: 25-27 pins	High Speed and UHS-I : 8 pins UHS-II and UHS-III: 16 pins SD Express 1-lane: 17 pins
	Operating Voltage	3.3V VDD range in the first-row: 2.7V – 3.6V 1.8V VDD range in the second-row: 1.70V-1.95V	
	Write-protect Switch	YES	NO

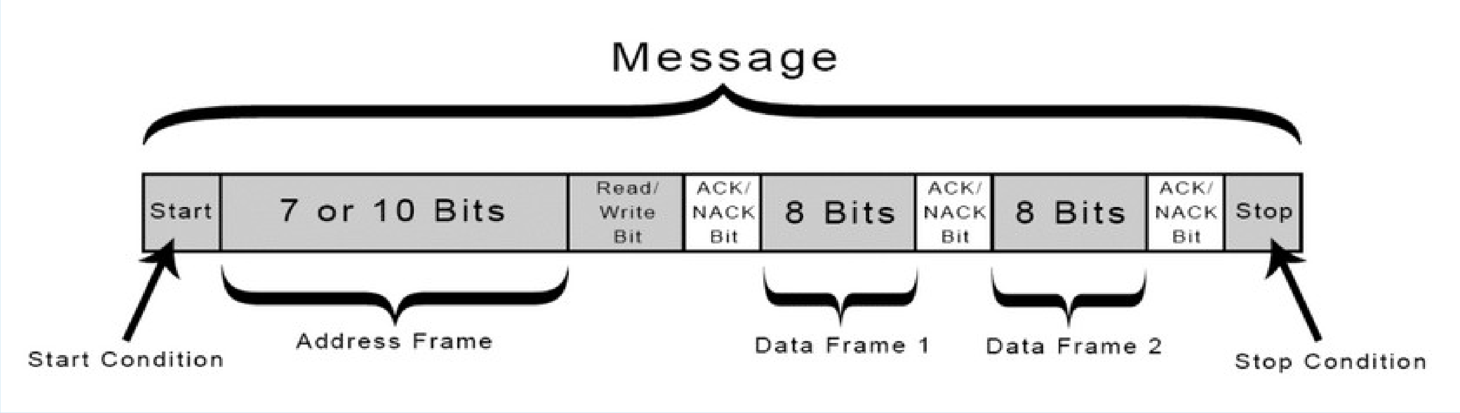
I2C

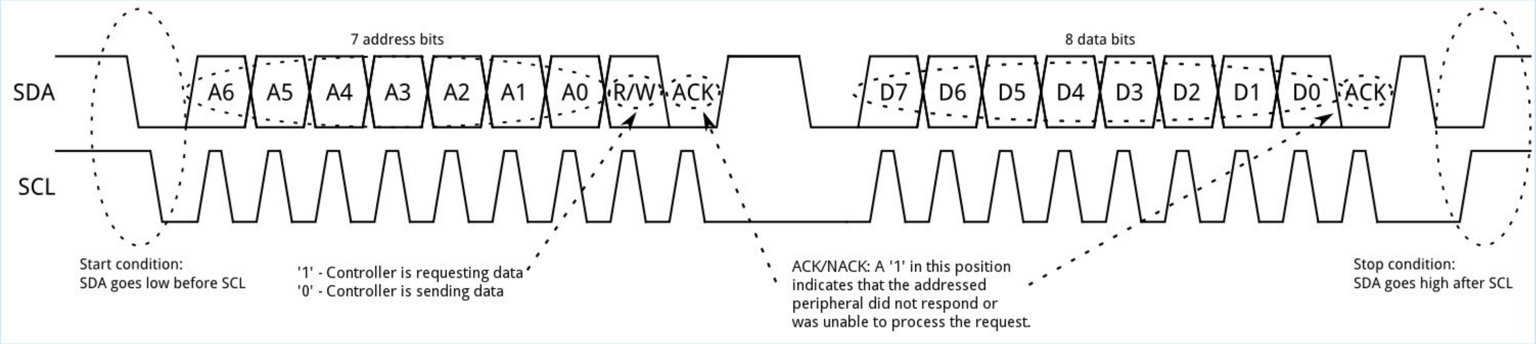
- Two-wire bus
- Planned for slow sensors
- Used, e.g., inside notebooks
- Multiple sensors on a bus
- Master-Slave approach
- Addressing

I2C – physical layer

- Two-wire serial communication
- Master/Slave
- Philips 1982
- Originally 100 kHz, now 400 kHz
- Fast / Ultrafast mode 1 MHz / 5 MHz

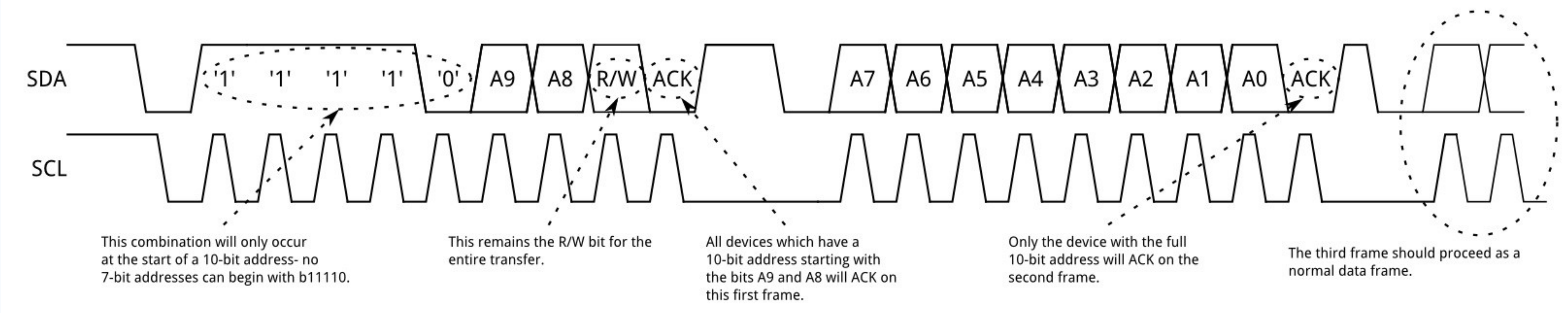
I2C – physical layer





I2C – addressing and conversation

- 7-bit addresses
- 10-bit alternative
-



I2C

- Data transmitted/received byte by byte
- Slave can increment internal counter to send next register
- Clock is generated by master

I2C – related protocols

- TWI
- SMBUS
- SWD

TWI

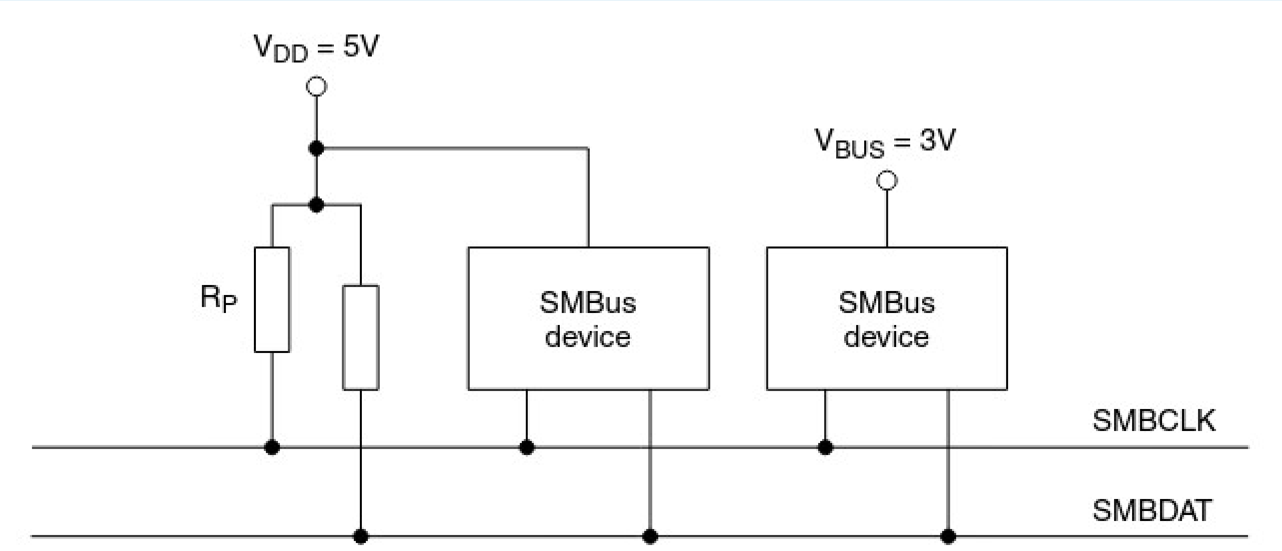
- Two-Wire Interface
- Almost the same as I2C
- Solves licence fees
- Both 100 kHz and 400 kHz
- Same addressing
- Many TWI and I2C devices compatible

SMBUS

- I2C and TWI are strictly Master / Slave
- What if slave device has some urgent info?
- System Management Bus

SMBUS

- Standardization in message format
- Timeouts
-



SWD

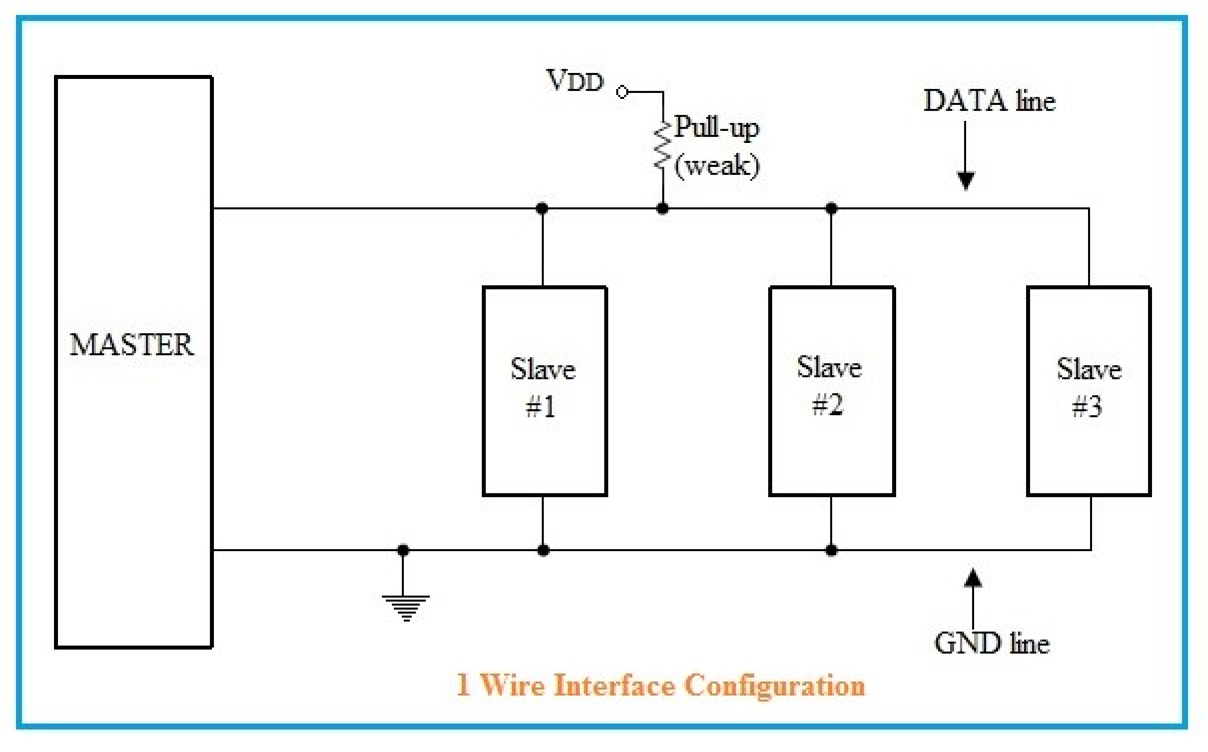
- How the code is uploade into STM32?

1-Wire

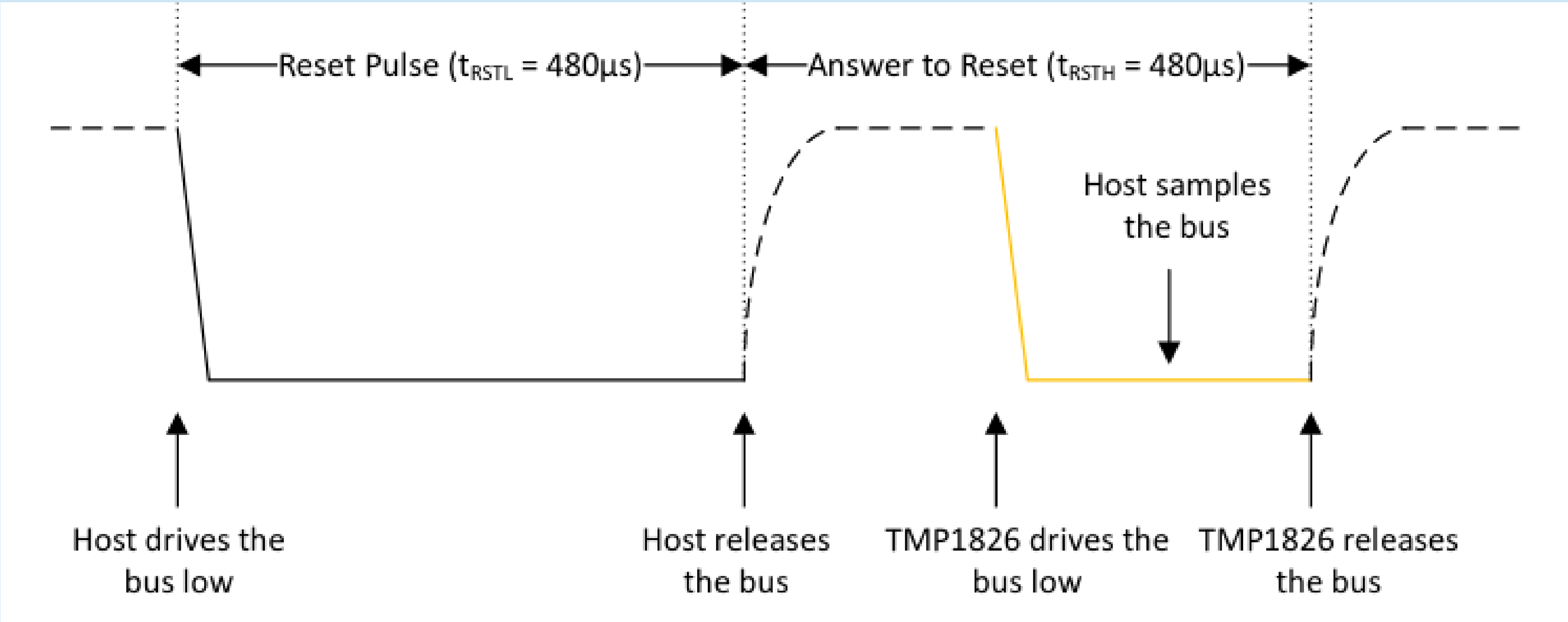
- Proprietary development Dallas Semiconductor
- Lowperformance sensors (16 kb/s)
- Minimizing number of wires
- Master / Slave
- Slave has a 64-bit system ID

1-Wire – physical layer

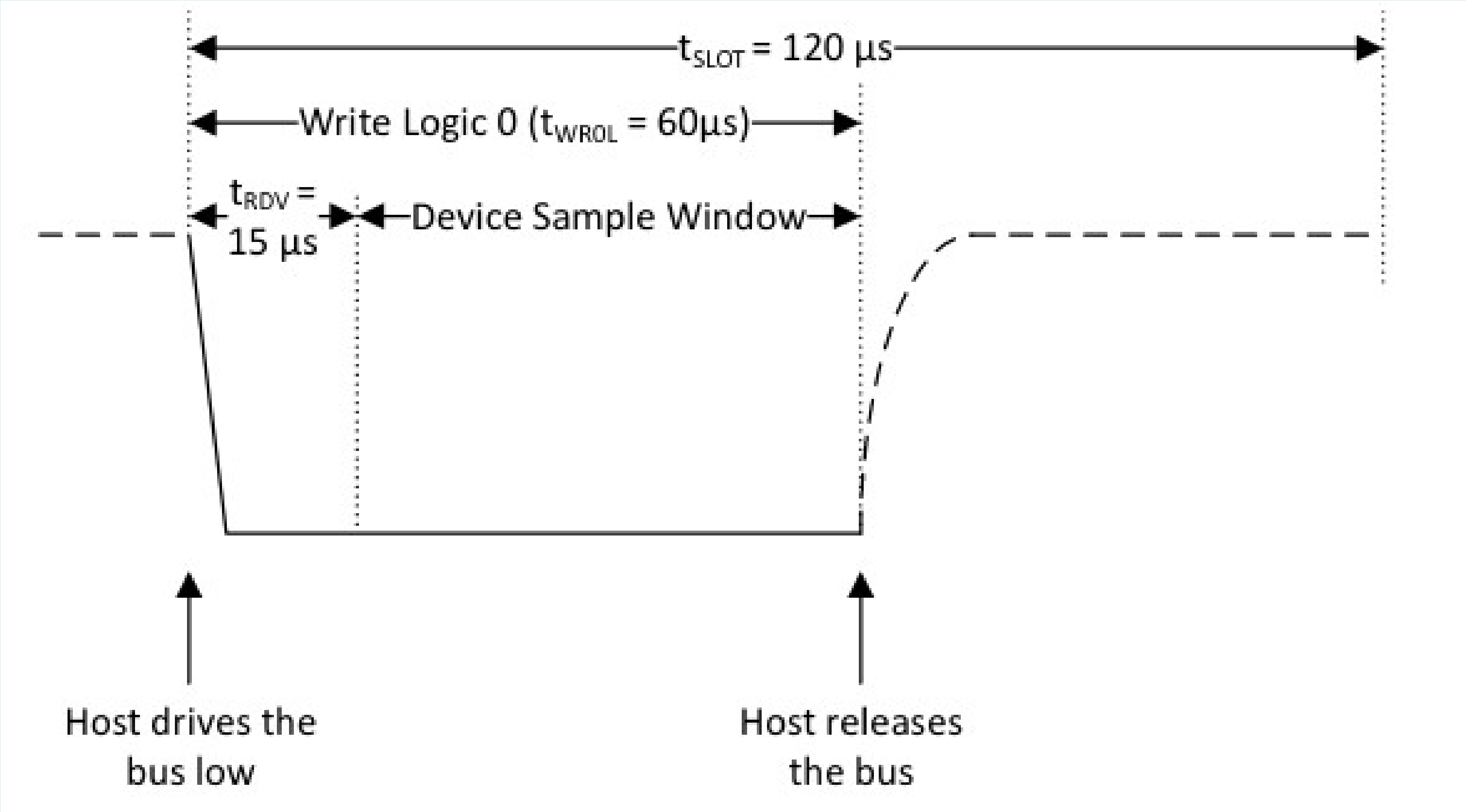
- 3-wire (PWR, GND, DATA)
- Parasitic power
  - 2-wire
  - GND & Data



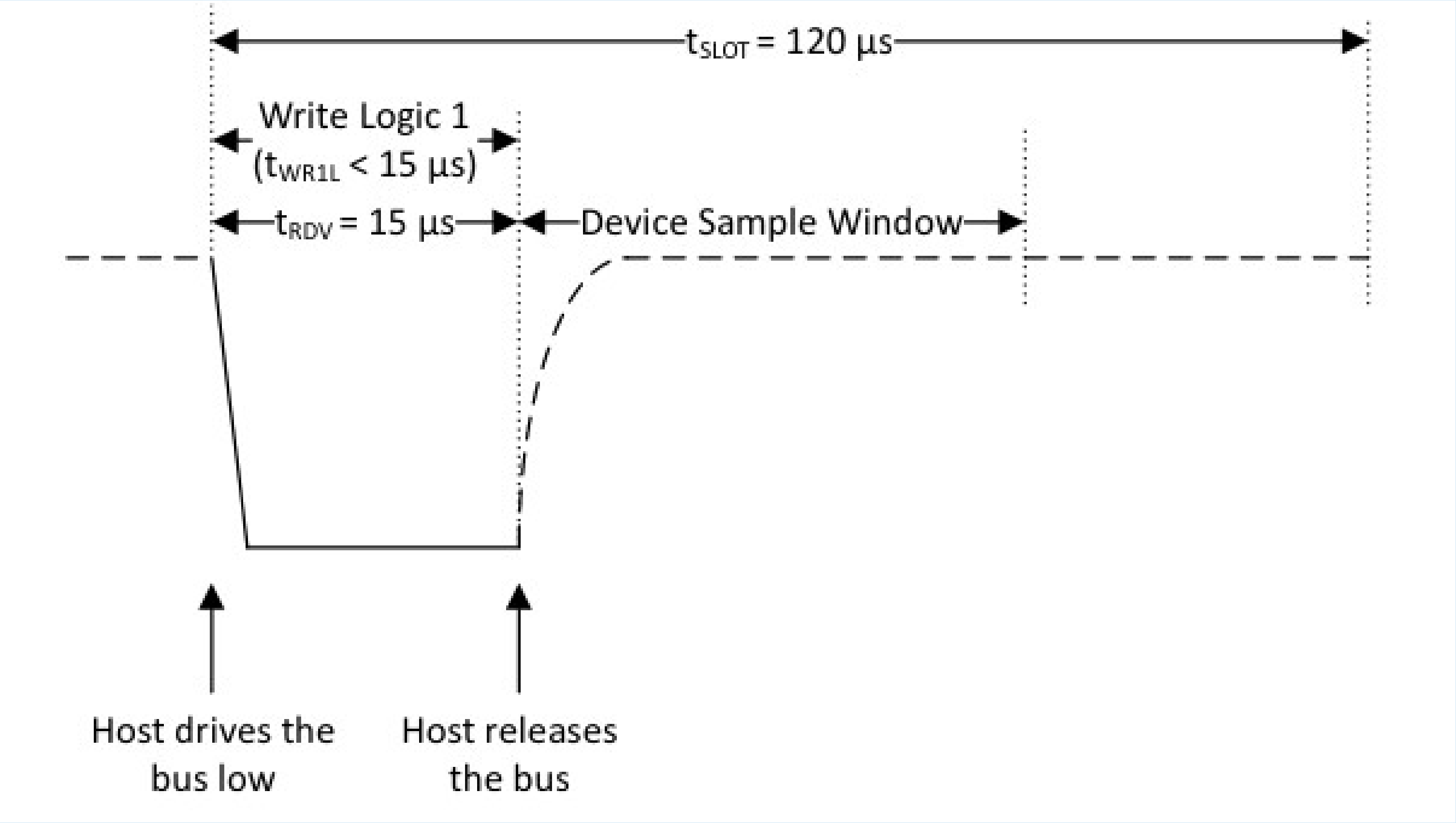
1-Wire – physical layer – reset



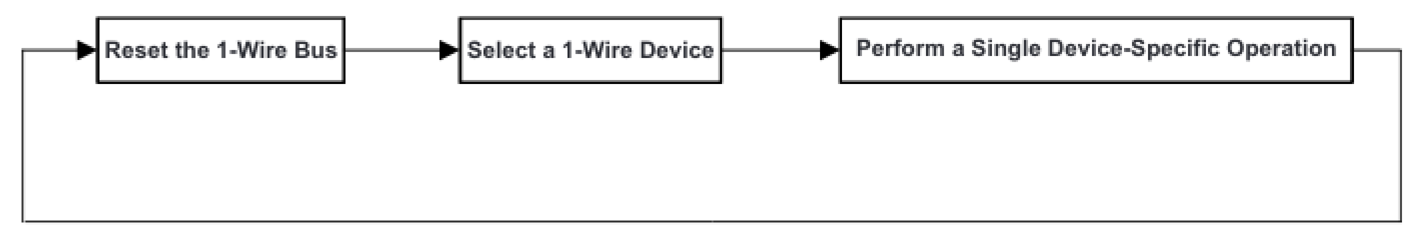
1-Wire – physical layer – log. 0

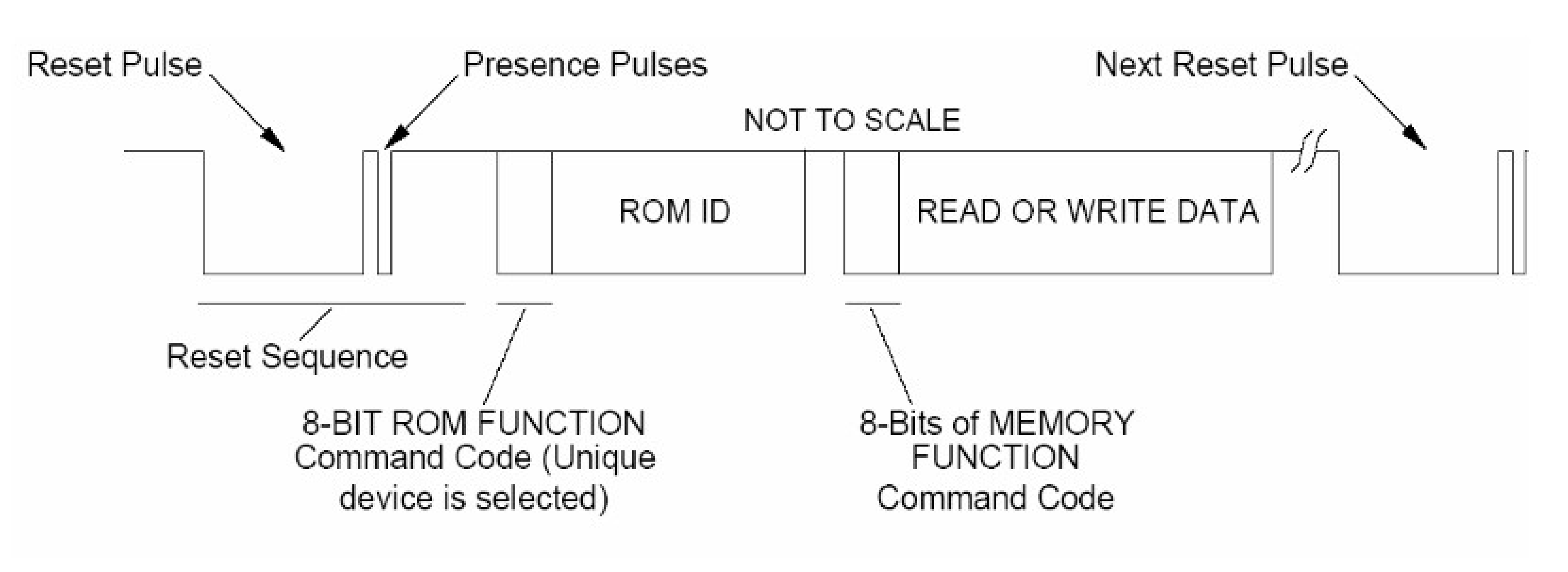


1-Wire – physical layer – log. 1



1-Wire – procedure





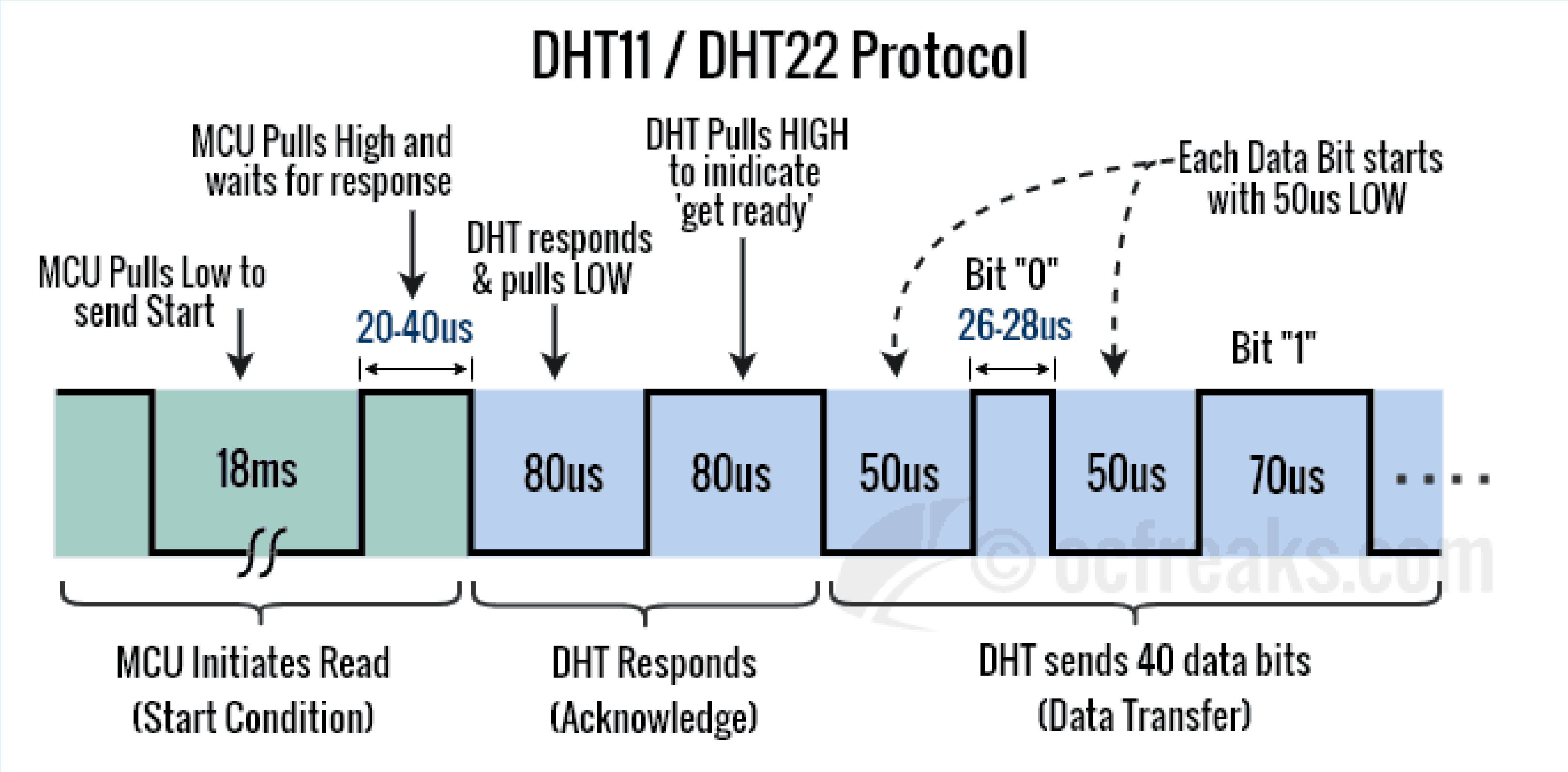
1-Wire – protocol

ROM Command	Code
Search Rom	F0h
Read ROM	33h
Match ROM	55h
Skip ROM	CCh
Alarm Search	ECh

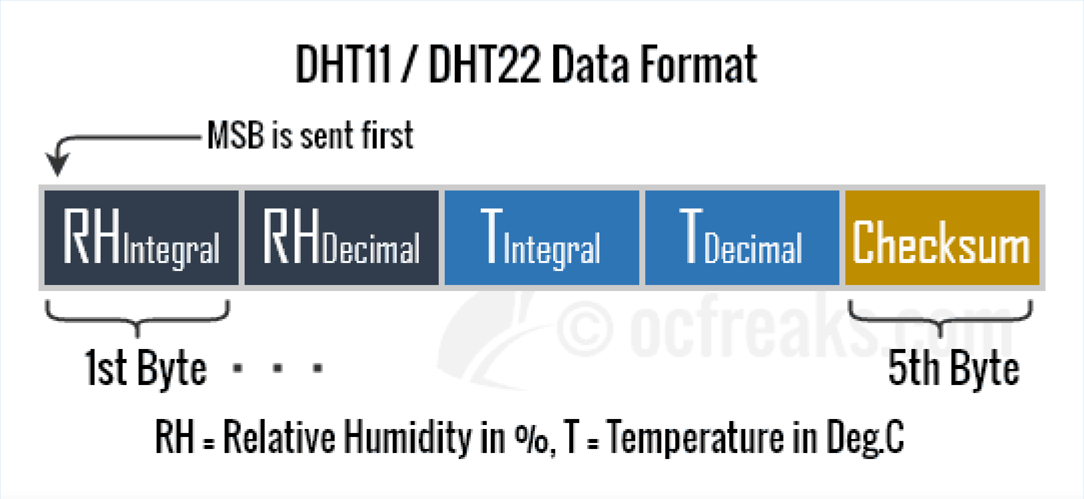
1-Wire – related protocols

- Proprietary (DHT-11 / DHT-22)
- UPDI

DHT-xx


DHT11 / DHT22 Protocol
MCU Pulls High and waits for response
DHT Pulls HIGH to inidicate 'get ready'
Each Data Bit starts with 50us LOW
DHT responds & pulls LOW
MCU Pulls Low to send Start
Bit "0"
26-28us
20-40us
Bit "1"
18ms
80us
80us
50us
50us
70us
© ocfreaks.com
MCU Initiates Read (Start Condition)
DHT Responds (Acknowledge)
DHT sends 40 data bits (Data Transfer)

DHT – 11/22



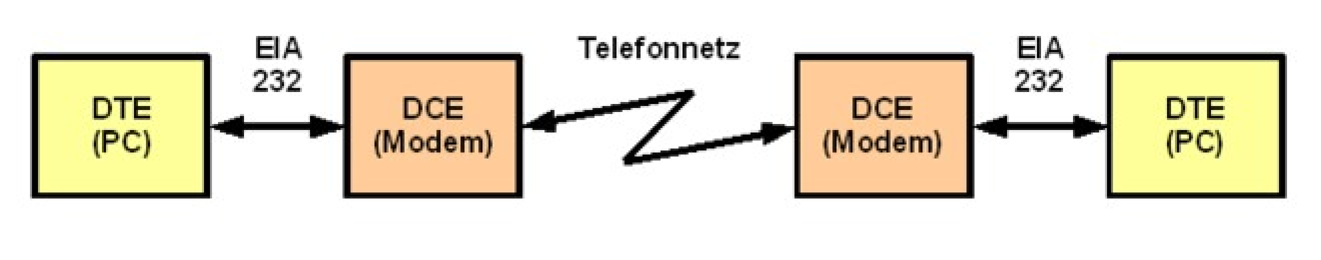
DHT11 / DHT22 Data Format
MSB is sent first
RHIntegral
RHDecimal
TIntegral
TDecimal
Checksum
© ocfreaks.com
1st Byte . . .
5th Byte
RH = Relative Humidity in %, T = Temperature in Deg.C

UPDI

- Unified Program and Debug Interface
- Attiny

External Communication

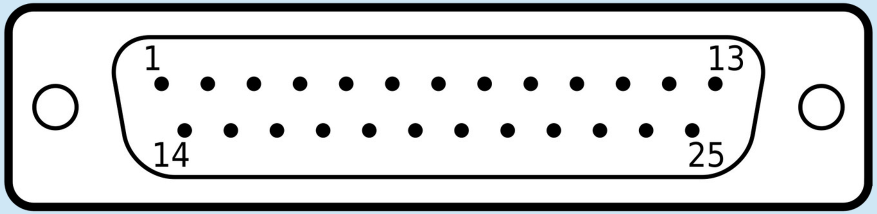
Interface RS-232

- EIA-232 (Electronic Industries Association)
- ITU-T/CCITT V.24
- 

- First introduced in 1960
- The character format and transmission bit rate are set by the serial port hardware - UART

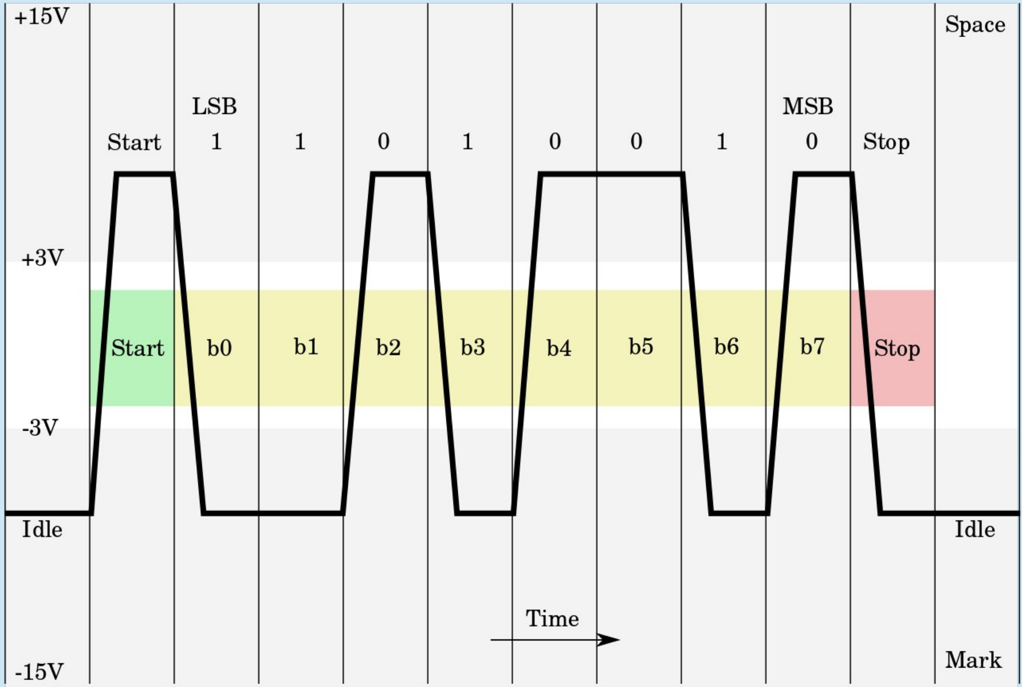
RS-232

- Both synchronous and asynchronous transmission
- Signal levels: ±5 V, ±10 V, **±12 V**, ±15 V
- Modem signals
- Cable length up to 20 m

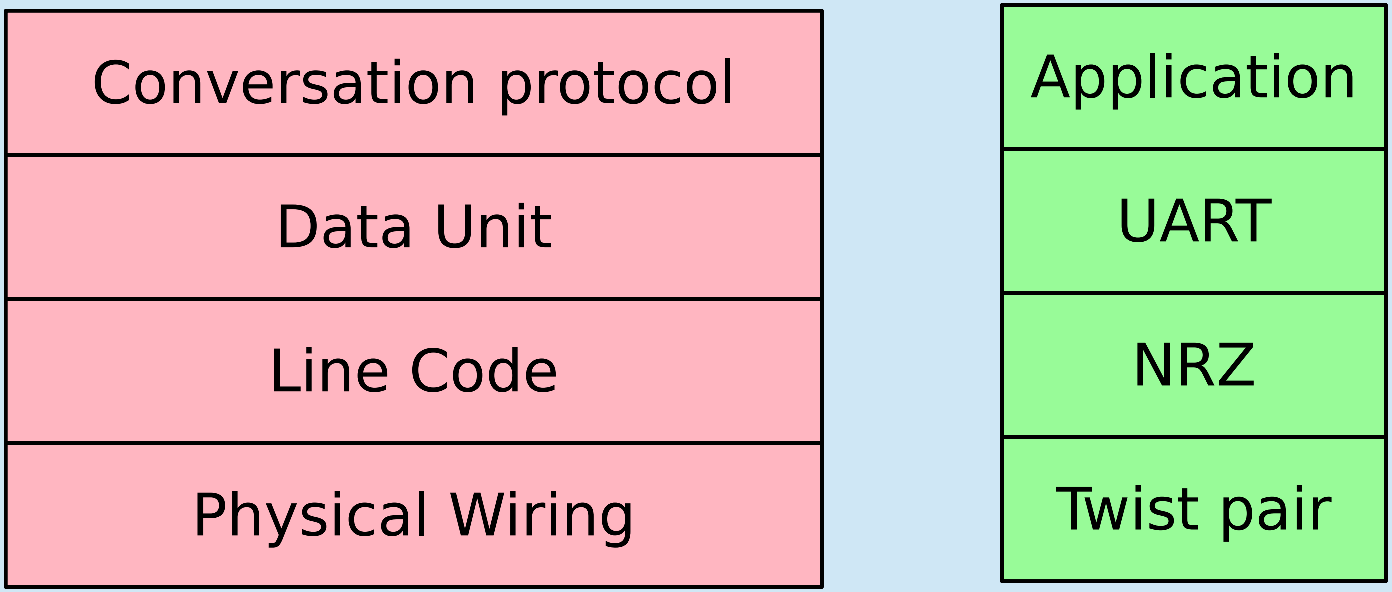


RS-232

- Last release RS-232C defined in 1969
- Many baudrates up to 115200 Bd
- 9600 Bd common console



RS-232


Conversation protocol
Data Unit
Line Code
Physical Wiring
Application
UART
NRZ
Twist pair

UART

- Baud Rate
- Parity bit
- Data bits size
- Stop bits size
- Flow Control
- Default: 8N1

1	5-9	0-1	1-2
Start bit	Data bits	Parity	Stopbits

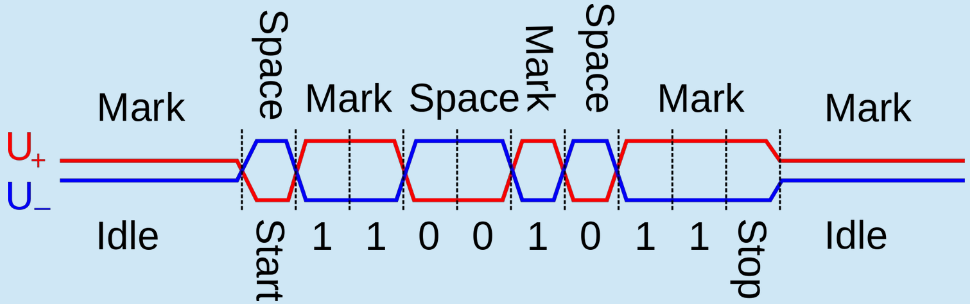
XOFF	CTRL+S	0x13
XON	CTRL+Q	0x11

RS-485

- Defined in 1983 (EIA)
- Two-wire multipont network
- Similar basis as RS-232
- No modem signals
- Up to 32 nodes
- Up to 1200 m

RS-485

- Conductors A and B
- Binary 1 – A negative with respect to B
- Binary 0 – A positive with respect to B
- Termination on long lines
- Hardware conversion RS-232 / RS-485
- Transmission +- 2V
- Receving min +- 200 mV



MODBUS

- Communication protocol published by Scheider Electric 1979
- Frequently used by SCADA (Supervisory Control and Data Acquisition) systems to connect remote terminal units (RTU)
- Popular in building automation – cooling, heating, ...
- 3 main versions:
  - MODBUS RTU
  - MODBUS ASCII
  - MODBUS TCP
  - Many extensions, not so important

MODBUS

- Client/Server (Master/Slave) architecture
- 8-bit address field, max 247 nodes on data link
- Client node must routinely poll each field device (server)
- No security mechanisms

MODBUS RTU Frame Format

Start of Frame	Node Address	Function Code	Data	Checksum	End of Frame
Min 28 bits	8 bits	8 bits	N x 8 bits	16 bits	Min 28 bits

MODBUS TCP Frame Format

Transcation ID	Protocol	Length	Unit Identifier	Function Code	Data
2B	2B	2B	1B	1B	N Bytes
For synchronization	0 for MODBUS TCP	Number of remaining bytes	For MODBUS TCP to/from MODBUS RTU gateways. 255 if unused		

MODBUS Object Types

Object Type	Access	Size	Addresses space
Coil	read-write	1 bit	00001 – 09999
Discrete Input	read	1 bit	10001 – 19999
Input Register	read	16 bits	30001 – 39999
Holding Register	read-write	16 bits	40001 – 49999

Function Codes

- Read Discrete Inputs (2)
- Read Coils (1)
- Write Single Coil (5)
- Write Multiple Coils (15)
- Read Input Registers (4)
- Read Multiple Holding Registers (3)
- Write Single Holding Register (6)
- Write Multiple Holding Registers (16)

M-Bus

- Meter Bus
- Developed for water, electricity and gas meters
- 1 Master, up to 250 Slaves
- Span up to 1000 m / 300 Bd or 350 m / 9600 Bd
- Special physical layer usable also for powering the meter
- https://m-bus.com/

M-Bus

- Developed by Uni Paderborn + TI Deutschland + Techem around 2000
- Respects ISO-OSI model
- Two-wire telephone cable
- EN 13757-2 physical and link layer
- EN 13757-3 application layer
- EN 13757-4 wireless M-Bus

M-Bus physical layer

- Master communicates on Volage level
  - Steady state = logical 1 Master sets the line to 36 V
  - Logical 0 = 24 V
- Master can power slaves
- Slaves communicate by change in current consumption
  - Steady state 1.5 mA sharp = logical 1
  - Logical 0 = current consumption increase by 11-20 mA

M-Bus Data Frame and Addressing

- 8-bit serial communication
- 8 bit addresses, optional network layer with secondary addresses, in this case address 253
- Link layer based on IEC 870-5
- Several data types:
  - Single byte: 0xE5 – used as acknowledgement
  - Short frame: 0x10 + Control + Address + Checksum + 0x16
  - Long frame: 0x68 + Length + Length(twice) + 0x68 + Controll + Address + Control_Information + Data(0-252B) + Checksum + 0x16

CAN

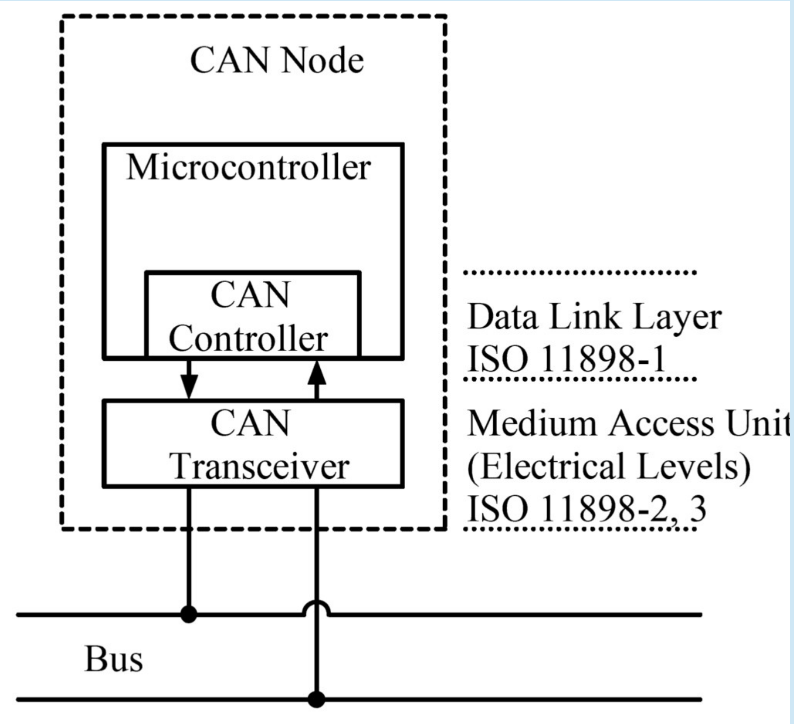
- Controler Area Network
- Started in 1983 at Robert Bosch GmbH
- Started in automotive
- Latest specification CAN 2.0 (1991)
- CAN 2.0A – 11-bit identifier
- CAN 2.0B – 29-bit identifier
- In 1993 standardized as ISO 11898
- In 2012 CAN FD – Flexible Data (Bosch)
- One of OBD-II protocols used for onboard car diagnostic

CAN bus

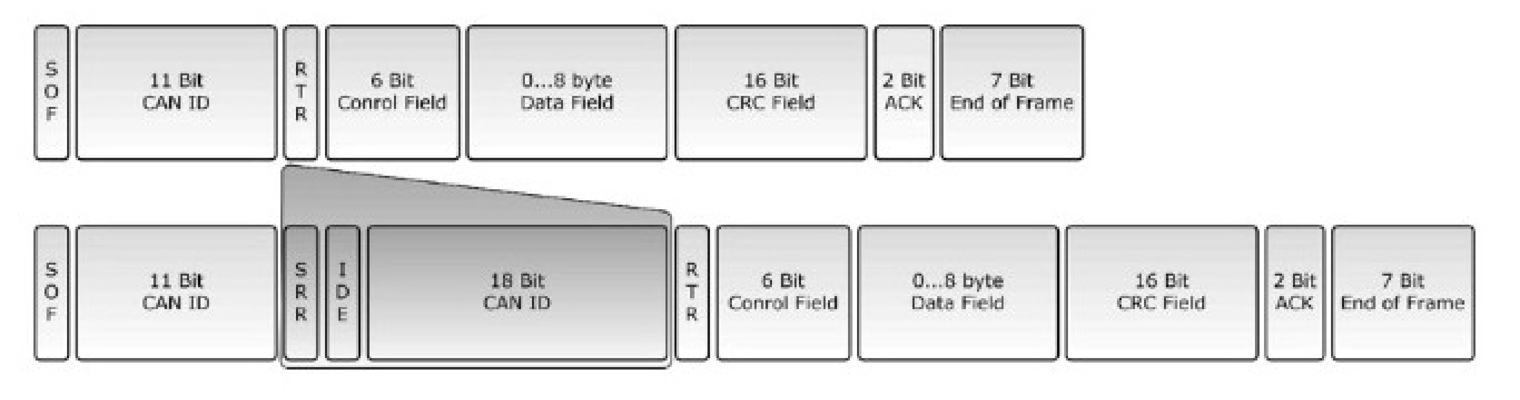
- Multimaster serial bus
- Two-wire line – twistpair with nominal impedance 120Ω
- Terminated with resistors
- CAN high (CANH) and CAN low (CANL)
- Dominant state (CANH > CANL) = 0 – CANH → 3.5 V, CANL → 1.5V
- Recesive state (CANL > CANH) = 1
- Baudrate 1 Mbps (5 Mbps on CAN-FD)
- 125 kbps low speed fault tolerant CAN
- Span 40 m

CAN node

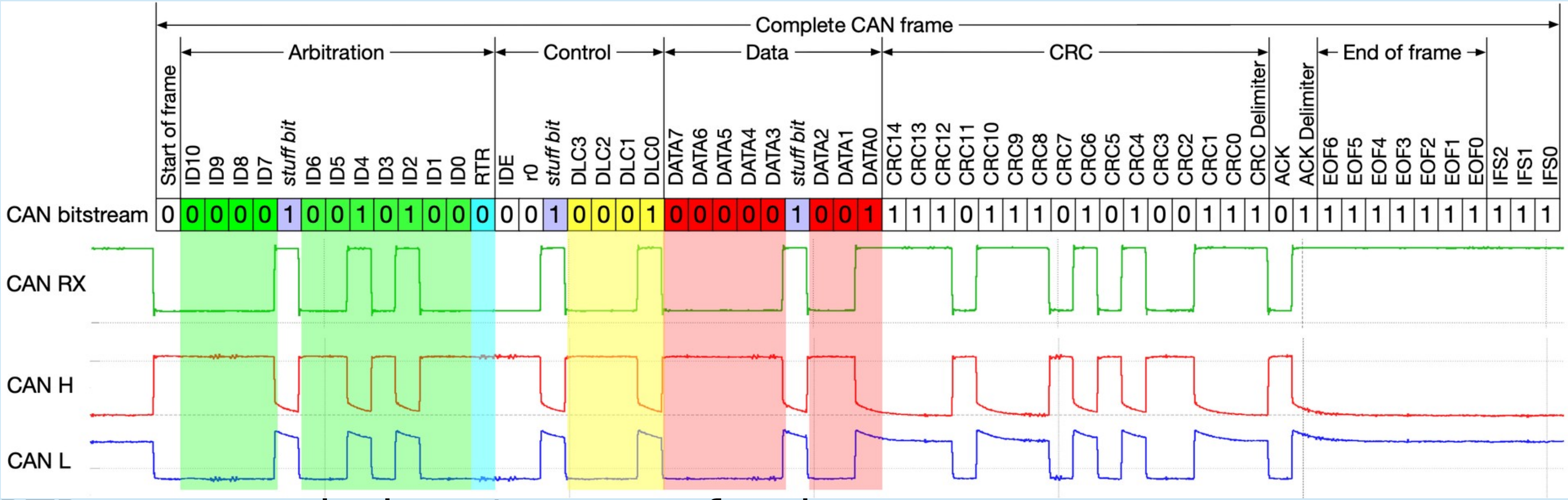
- MCU
- CAN Controller
- CAN Transceiver



CAN bus Data Frame



CAN bus Data Frame



- RTR = transmit data / request for data
- IFS = Inter Frame Space
- Bit stuffing

CAN bus

- Line Code: NRZ
- Bit stuffing: after 5 consecutive bites of the same polarity a bit with opposite polarity is inserted
- Exceptions: CRC delimiter, ACK, End-of-Frame
- Frame size not necessarily multiple of 8 bits

CAN bus

- Identifier is not and address but a priority
- Receivers should be bit synchronous
- Lower identifier = higher priority
- If I send „1“ and receive „0“ (bit), someone else with hihger priority transmits → loss of arbitration
- Identifier must be unique
- Different philosophy: I’m transmitting data if I have some, who is interested can receive it

FlexRay

- Automotive
- Consortium of developers led by Bosch
- Since 2009 disbundled
- ISO 17458
- A bus and set of ECUs (Electronic Control Units)
- Stronger time determinism than CAN
- First used in BMW X5
- First fully powered by FlexRay BMW X7

FlexRay

- Speed up to 10 Mbps, 2047 nodes, 24 m
- Both copper and fiber lines (POF)
- ECUs have independent clock
- Drift from the reference no more tna 0.15%
- Only 1 ECU transmits at a time
- Each bit is transmitted in 8 cycles
- Each ECU has buffer for at least 5 cycles
- Majority of at least 5 samples

FlexRay

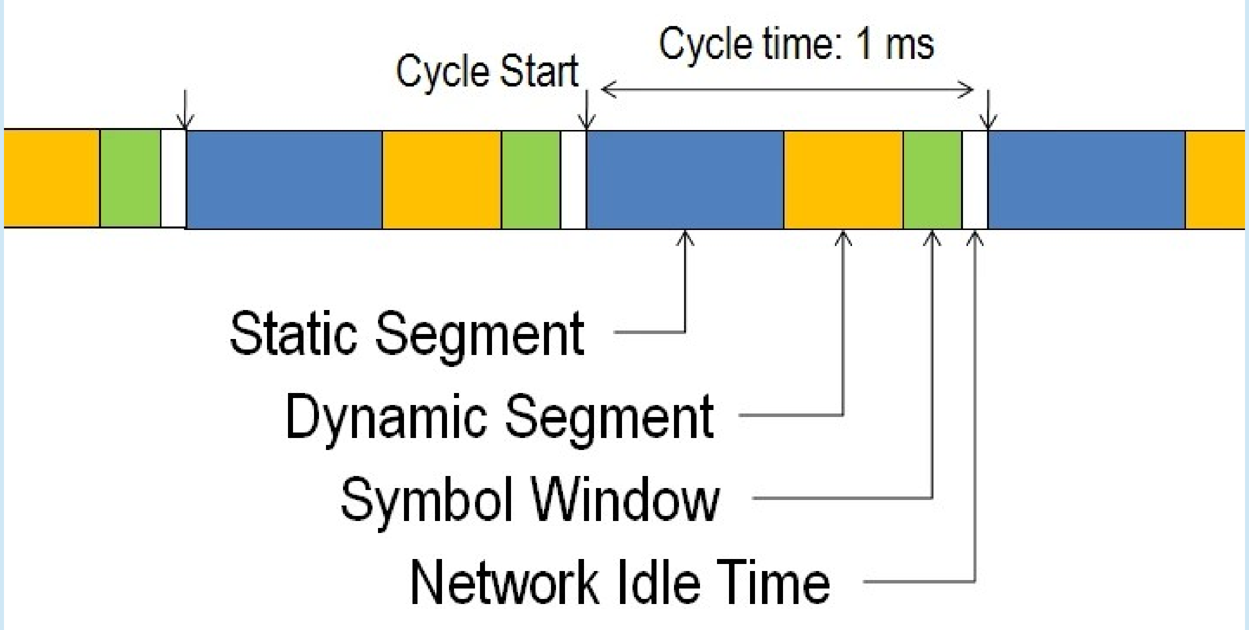
To be sent	1	0	1
Sent	11111111	00000000	11111111
Received	11111111	00010010	11111101
Voter	1	0	1

FlexRay

- Single or dual channel
- Multidrop bus
- Star topology
- Hybrid topology
- TDMA principle

FlexRay

- Communication cycle 1 – 5 ms
- Determined at network design

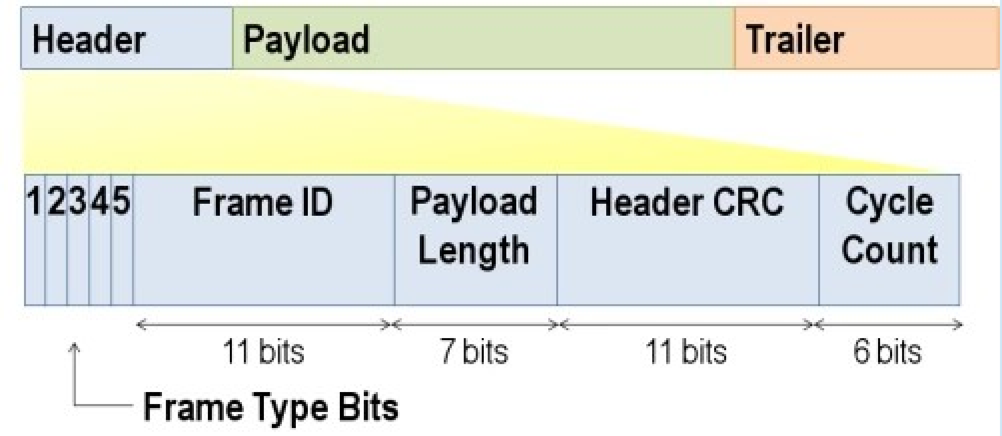


FlexRay

- Static segments broken up to slots
- Each slot reserved for given ECU
- Dynamic segment has fixed size
- Is broken to minislots (1us typicaly)
- Higher priority data pushes out lower priority ones
- Each slot contains a FlexRay Frame

FlexRay Frame

- 40 bit header
- 0-254 Bytes payload
- 24 bits trailer



ARINC-429 / ARINC-629 / AFDX

Serial Busses Security remarks

- Designed for closed networks
- Security not addressed
- Now become more opened and vulnerable

Thank you for your attention!

Questions and comments?