Atmel Wireless & Microcontrollers
CAN Tutorial

Gerhard Goller
gerhard.goller@atmel-wm.com

Agenda

- Introduction or: What is CAN?
- Why CAN?
- CAN Protocol
- CAN higher Layer Protocols
- CAN Applications
- CANary: Atmel CAN Microcontrollers & Roadmap
- Application Support
- Conclusion

Introduction

- The CAN is an ISO standard (ISO 11898) for serial communication
- The protocol was developed 1980 by BOSCH for automotive applications
- Today CAN has gained widespread use:
  - Industrial Automation
  - Automotive, …etc.
- The CAN standard includes:
  - Physical layer
  - Data-link layer
    - Some message types
    - Arbitration rules for bus access
    - Methods for fault detection and fault confinement

Why CAN?

- **Mature Standard**
  - CAN protocol more than 14 years
  - Numerous CAN products and tools on the market
- **Hardware implementation of the protocol**
  - Combination of error handling and fault confinement with high transmission speed
- **Simple Transmission Medium**
  - Twisted pair of wires is the standard, but also just one wire will work
  - Other links works, too: Opto - or radio links
- **Excellent Error Handling**
  - Strong point of the protocol: § Extensive error detection mechanism
- **Fine Fault Confinement**
  - Built-in feature to prevent faulty node to block system
- **Most used protocol in industrial and automotive world**
- **Best Performance / Price ratio**

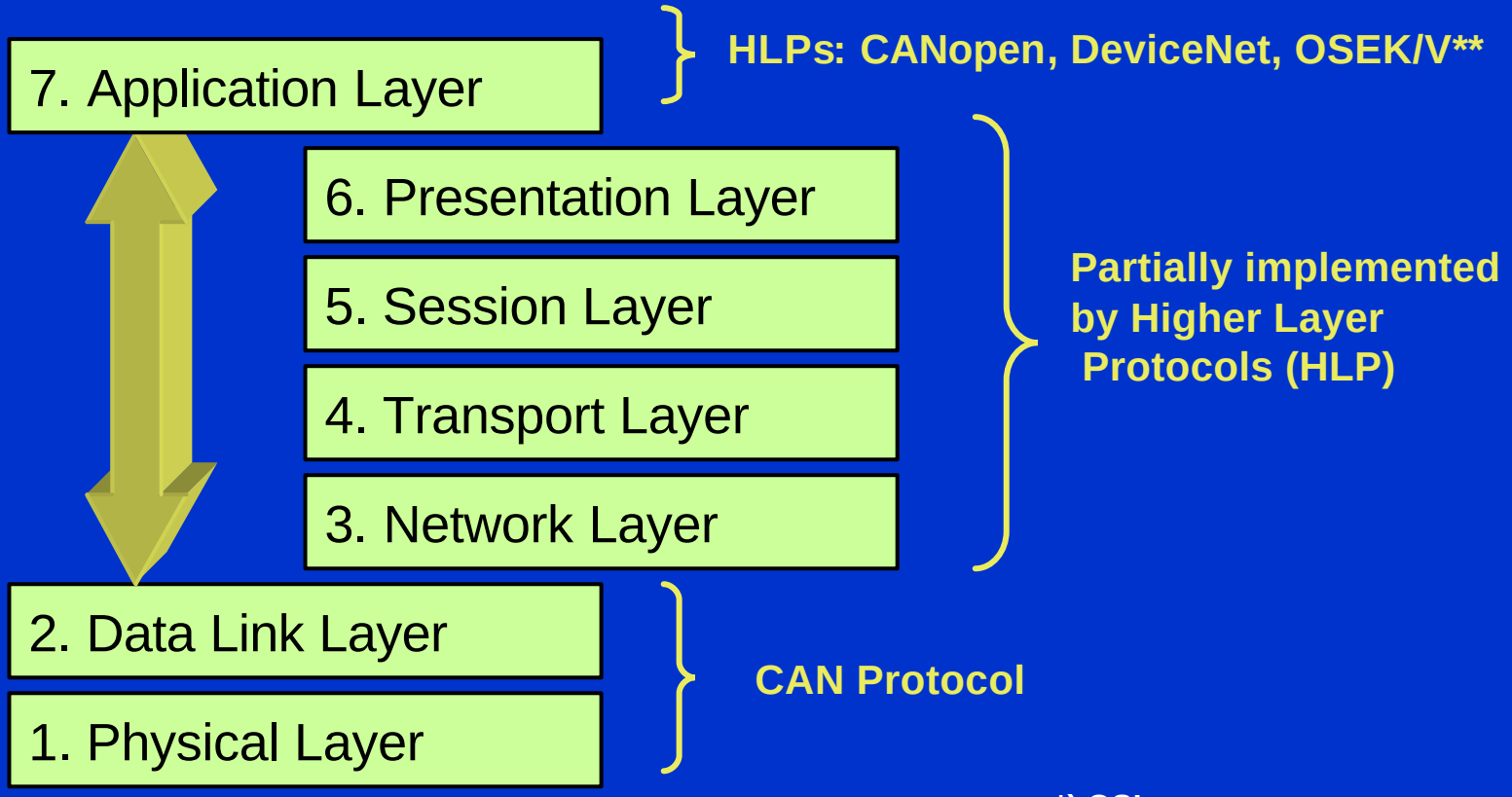
CAN Protocol

- What is CAN?
- ISO-OSI Reference Model
- CAN Bus Logic
- Typical CAN Node
- CAN Bus Access and Arbitration
- CAN Bit Coding & Bit Staffing
- CAN Bus Synchronization
- CAN Bit Construction
- Relation between Baud Rate and Bus Length
- Frame Formats (1)
- Frame Formats (2)
- Frame Formats (3)
- Frame Formats (4)
- Fault Confinement (1)
- Fault Confinement (2)
- Undetected Errors

What is CAN?

- Controller Area Network
  - Invented by Robert Bosch GmbH
  - Asynchronous Serial Bus
  - Absence of node addressing
    - Message identifier specifies contents and priority
    - Lowest message identifier has highest priority
  - Non-destructive arbitration system by CSMA with collision detection
  - Multi-master / Broadcasting concept
  - Sophisticated error detection & handling system
  - Industrial and Automotive Applications

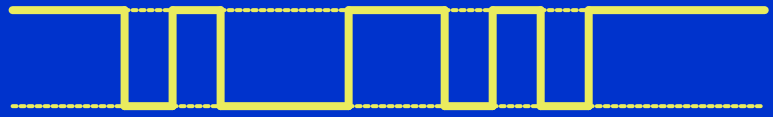
ISO-OSI* Reference Model



*) **OSI -** Open System Interconnection

CAN Bus Logic

Two logic states on the CAN bus



"1" = recessive

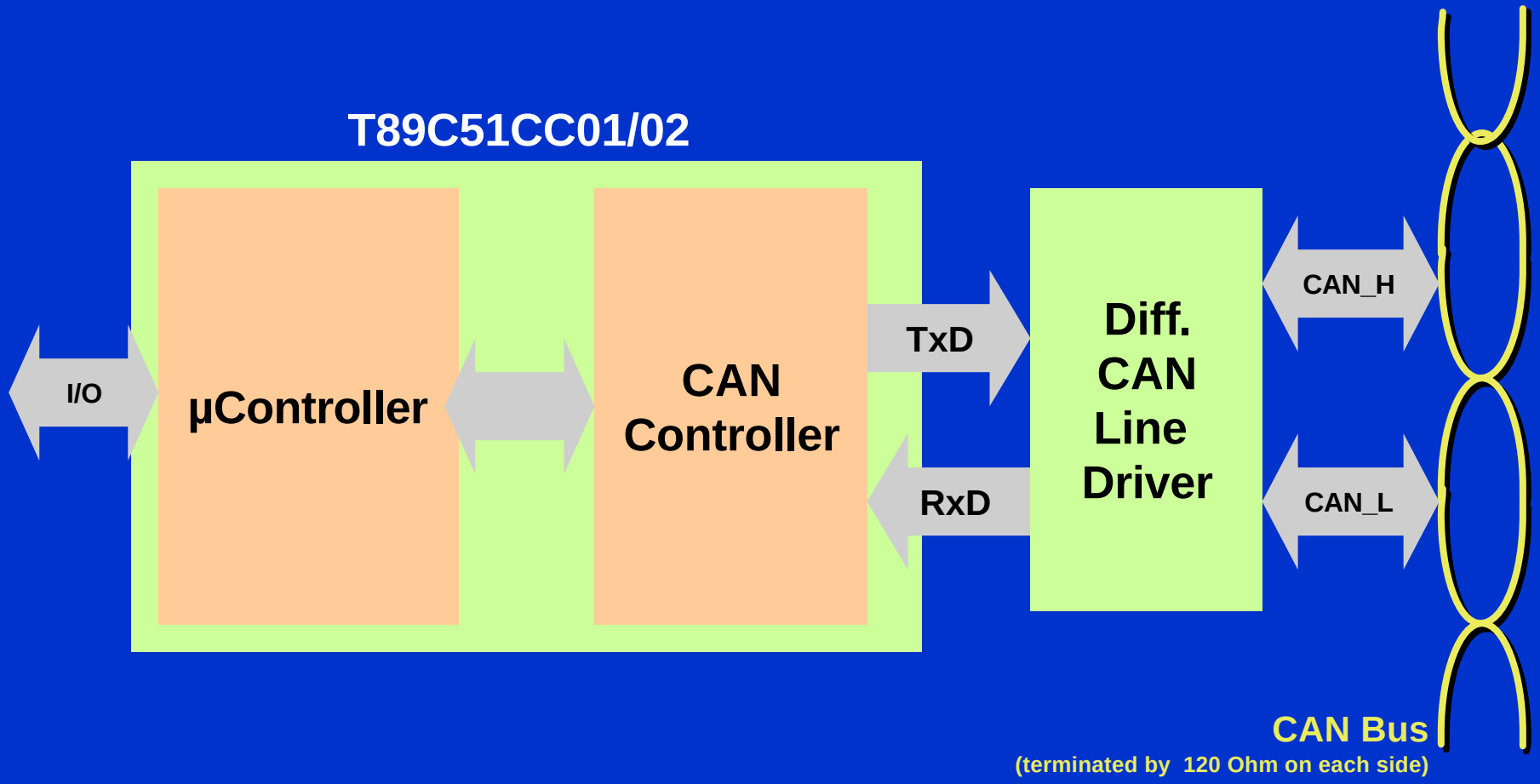
"0" = dominant

Node A	Node B	Node C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

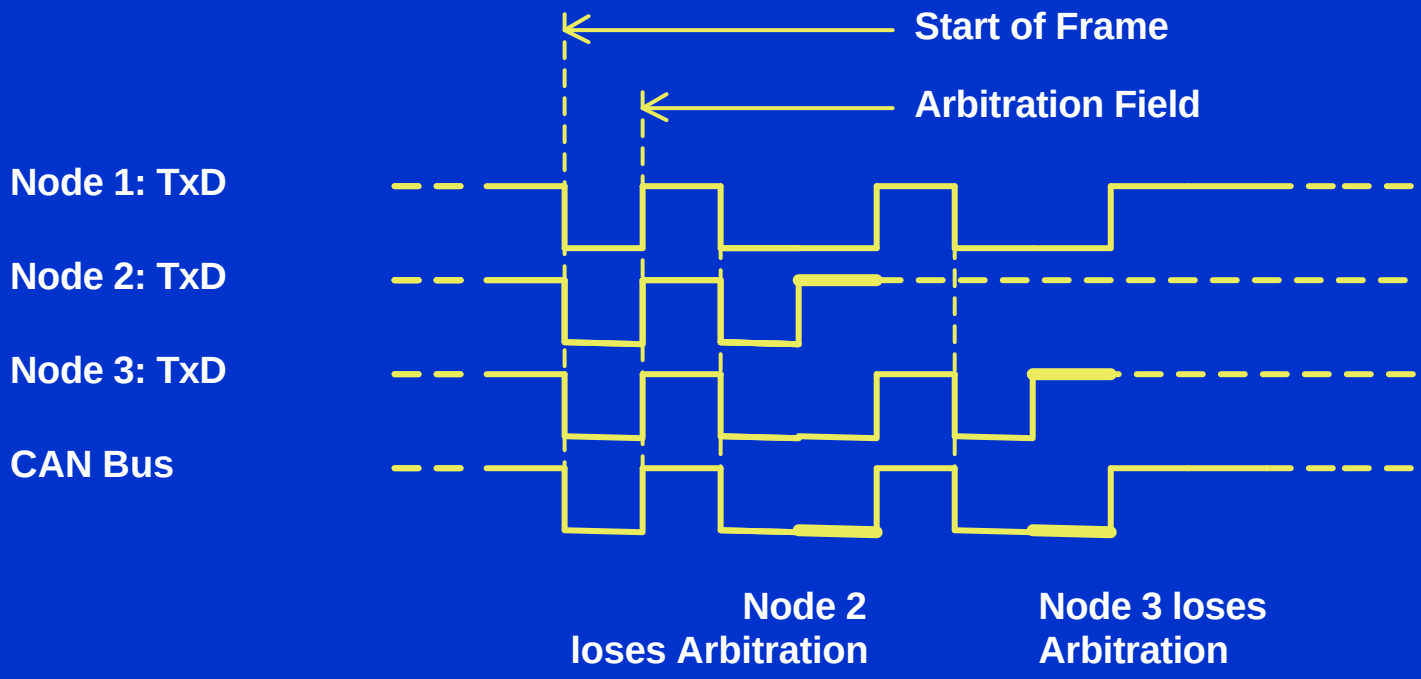
"Wired-AND" function:
as soon as one node transmit a dominant bit (zero)
the bus is in the dominant state

Only if all nodes transmit recessive bits (ones)
the Bus is in the recessive state

Typical CAN Node



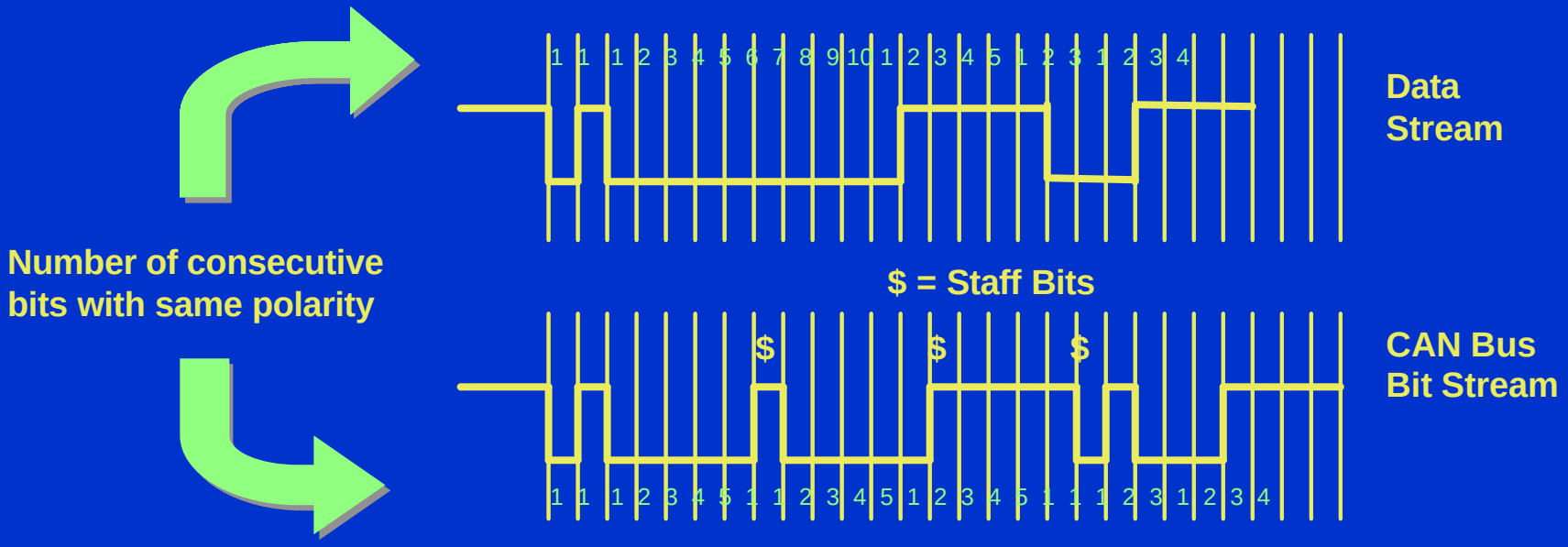
CAN Bus Access and Arbitration: CSMA/CD and AMP *)



*)Carrier Sense Multiple Access/Collision Detection and Arbitration by Message Priority

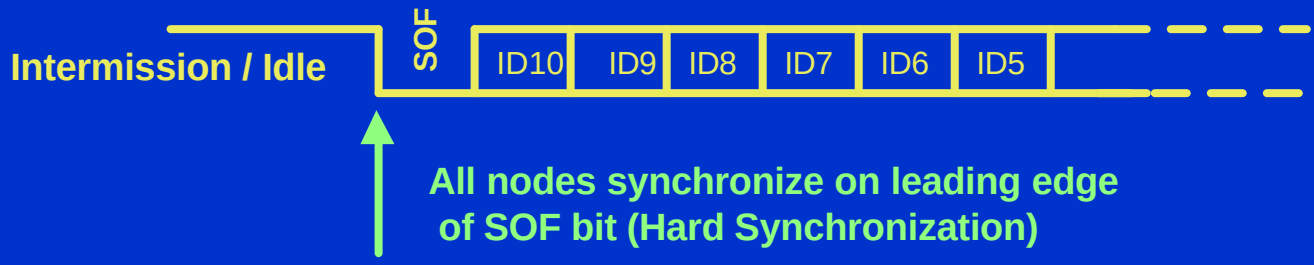
CAN Bit Coding & Bit Staffing

- Bit Coding : NRZ (Non-Return-To-Zero code) does not ensure enough edges for synchronization
- Stuff Bits are inserted after 5 consecutive bits of the same level
- Stuff Bits have the inverse level of the previous bit.
- No deterministic encoding, frame length depends on transmitted data

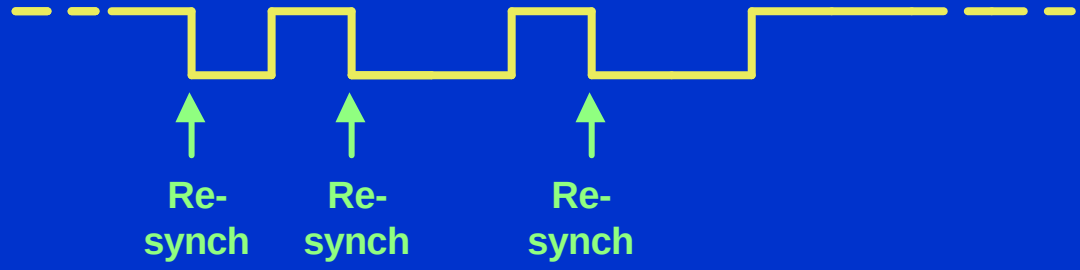


CAN Bus Synchronization

- Hard synchronization at Start Of Frame bit

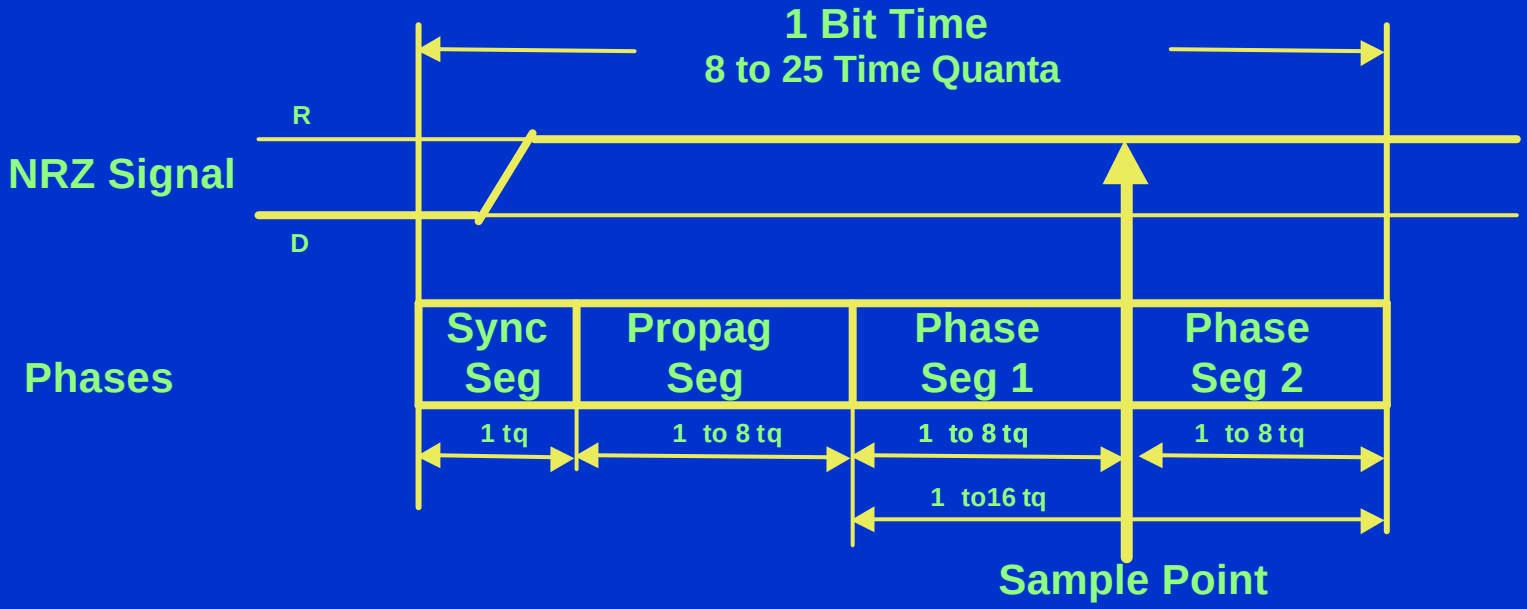


- Re-Synchronization on each Recessive to Dominant bit

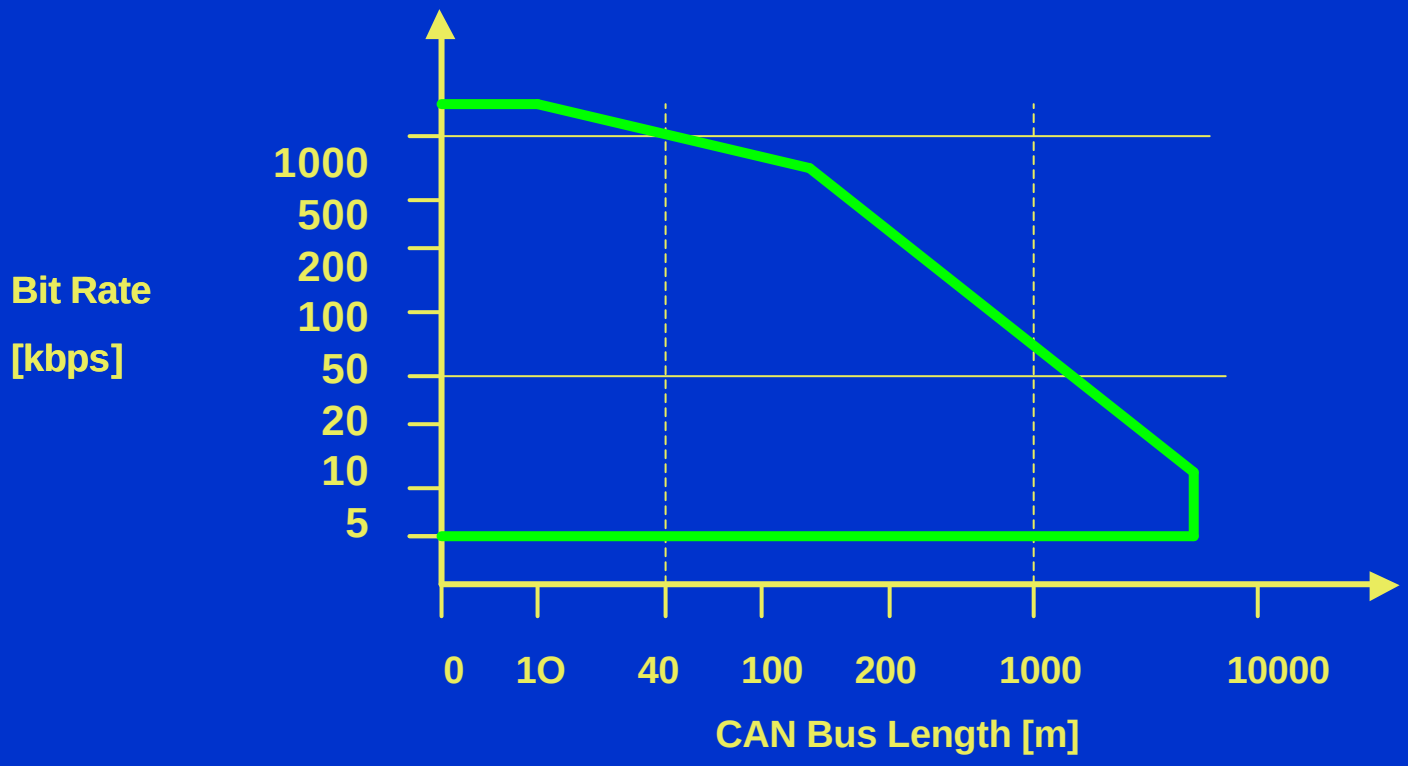


CAN Bit Construction

- Length of one time quanta can be set to multiple of µController clock
- 1 Time quantum = 1 period of CAN Controller base clock
- Number of time quanta in Propag and Phase segments is programmable

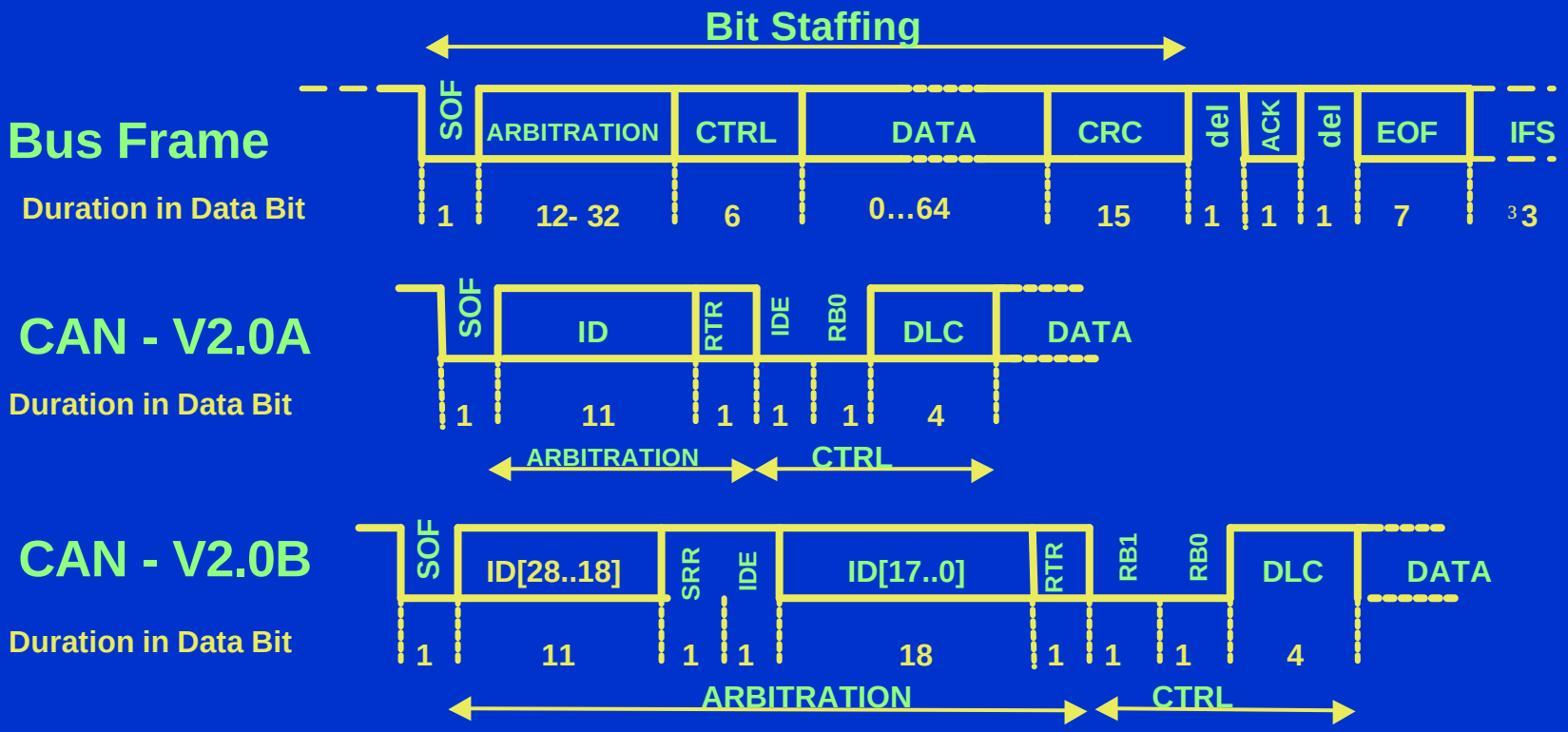


Relation between Baud Rate and Bus Length



Example based on CAN Bus Lines by twisted pair

Frame Formats (1)



SOF	Start of Frame	EOF	End of Frame	RTR	Remote Transmission Request
CRC	Cyclic Redundancy Code	IFS	Inter Frame Spacing	SRR	Substitute Remote Request
del	Delimiter	ID	Identifier	RB0/1	Reserved bits
ACK	Acknowledge	IDE	Identifier Extension	DLC	Data Length Code

Frame Formats (2)

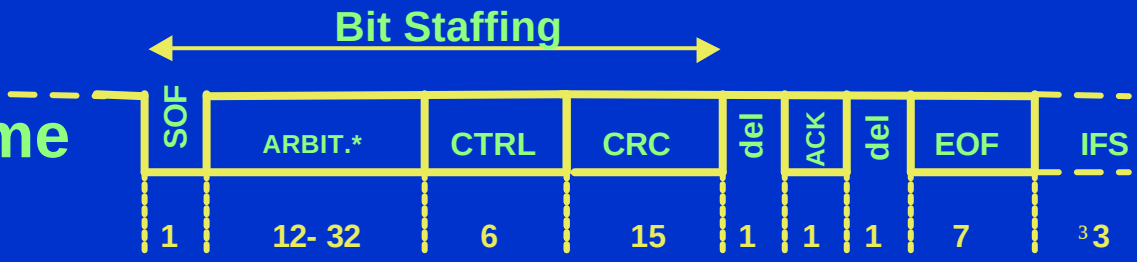
Data Frame

Duration in Data Bit

(*) RTR = dominant

Remote Frame

Duration in Data Bit



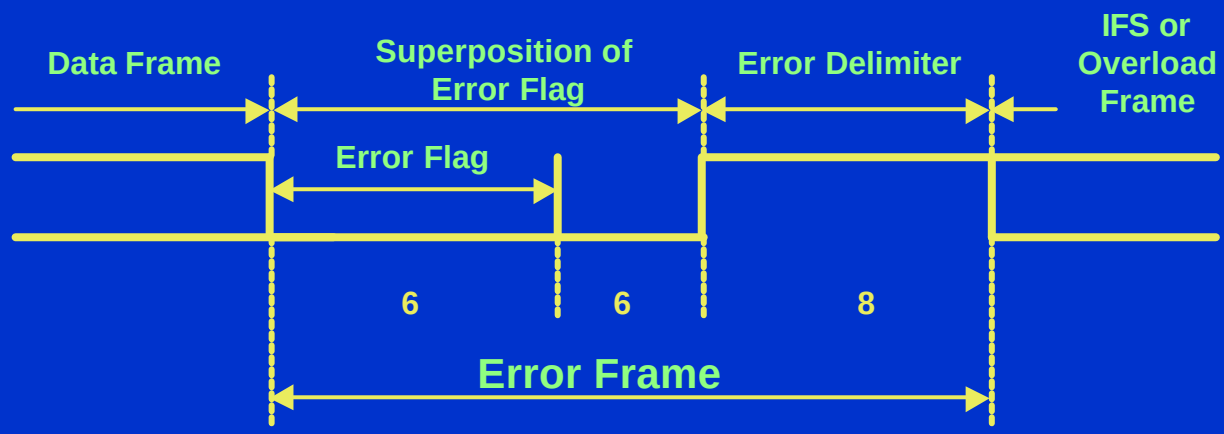
(*) RTR = recessive

Frame Formats (3)

- If any of the CAN nodes detects a violation of the frame format
- or a stuff error, it immediately sends an Error Frame

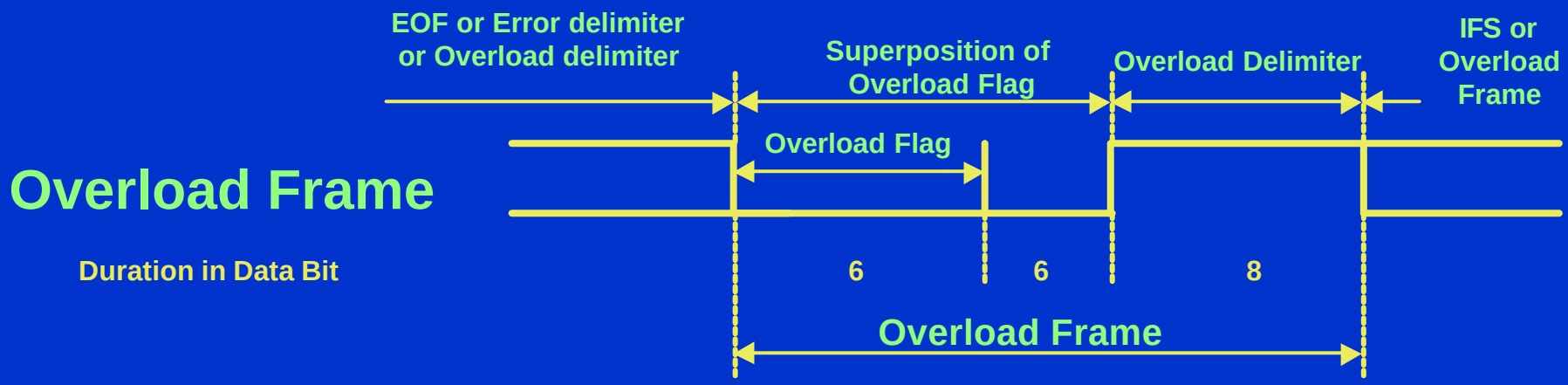
Error Frame

Duration in Data Bit



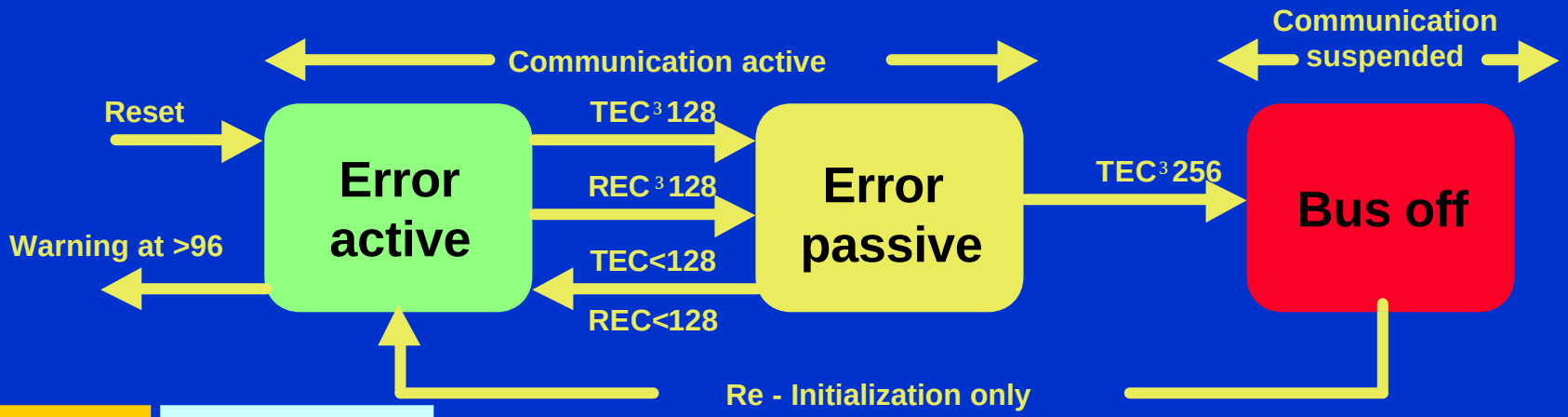
Frame Formats (4)

- If any of the CAN nodes suffers from a “data over flow”, it might send
- up two consecutive Overload Frames to delay the network



Fault Confinement (1)

- Three fundamental states define each node's error signaling
  - Error active: Normal state, node can send all frames incl.error frames
  - Error passive: Node can send all frames excluding error frames
  - Bus off: Node is isolated from bus
- Internal error counts determine the state
  - Transmit error counter (TEC) An error increases the counter by 8
  - Receive error counter (REC) A successful operation decreases by 1
- Aims to prevent from bus dead-locks by faulty nodes



Fault Confinement (2)

- Cyclic Redundancy Check (CRC)
- The CRC is calculated over the non-stuffed bit stream starting with the SOF and ending with the Data field by the transmitting node
- The CRC is calculated again of the destuffed bit stream by the receiving node
- A comparison of the received CRC and the calculated CRC is made by the receiver
- In case of mismatch the erroneous data frame is discarded . Instead of sending an acknowledge signal an error frame is sent.

Undetected Errors

- Error statistics depend on the entire environment
- Total number of nodes
- Physical layout
- EMI disturbance
- Automotive example
- 2000 h/y
- 500 kbps
- 25% bus load

→ One undetected error every 1000 years

CAN Higher Layer Protocols (HLPs)

- Why HLPs
- CANOpen
- DeviceNet
- CAN Kingdom
- OSEK/VDX
- SDS
- J1939

Why HLPs

- The CAN protocol defines only the ‘physical’ and a low ‘data link layer’!
- The HLP defines:
  - Start-up behavior
  - Definition of message identifiers for the different nodes
  - Flow control
  - transportation of messages > 8bytes
  - Definition of contents of Data Frames
  - Status reporting in the system

CANopen

- **Features**
  - CANopen a subset from CAL (CAN Application Layer) developed by CiA!
  - Auto configuration the network
  - Easy access to all device parameters
  - Device synchronization
  - Cyclic and event-driven data transfer
  - Synchronous reading or setting of inputs, outputs or parameters
- **Applications**
  - Machine automatisation
- **Advantages**
  - Accommodating the integration of very small sensors and actuators
  - Open and vendor independent
  - Support s inter-operability of different devices
  - High speed real-time capability

DeviceNet

- **Features**
  - Created by Allen-Bradley (Rockwell Automatisation nowadays), now presented by the users group ODVA (Open DeviceNet Vendor Association)
  - Power and signal on the same network cable
  - Bus addressing by: Peer-to-Peer with multi-cast & Multi-Master & Master-Slave
  - Supports only standard CAN
- **Applications**
  - Communications link for industrial automatisation: devices like limit switches, photo-electric sensors, valve manifolds, motor starters, process sensors, bar code readers,variable frequency drives, panels...
- **Advantages**
  - Low cost communication link and vendor independent
  - Removal and replacement of devices from the network under power

CAN Kingdom

- CAN Kingdom is more then a HLP: A Meta protocol
  - Introduced by KVASER, Sweden
  - A ‘King’ (system designer) takes the full responsibility of the system
  - The King is represented by the Capital (supervising node)
  - World wide product identification standard EAN/UPC is used for
- Applications
  - Machine control, e.g. industrial robots, weaving machines, mobile hydraulics, power switchgears, wide range of military applications
- Advantages
  - Designed for safety critical applications
  - Real time performance
  - Scalability
  - Integration of DeviceNet & SDC modules in CAN Kingdom possible

OSEK/VDX

Offene Systeme und deren Schnittstellen fuer die Elektonik im Kraftfahezeug/Vehicle Distributed eXecutive)

- **Initialized by:**
  - **BMW, Bosch, DaimlerChrysler, Opel, Siemens,VW & IIIT of the University of Karlsruhe / PSA and Renault**
- **OSEK/VDX includes:**
  - **Communication (Data exchange within and between Control Units)**
  - **Network Management (Configuration determination and monitoring)**
  - **Operating System (Real-time executive for ECU software)**
- **Motivation:**
  - **High, recurring expenses in the development and variant management of non-application related aspects of control unit software**
  - **Compatibility of control units made by different manufactures due to different interfaces**
- **Goal: Portability and re-usability of the application software**
- **Advantages: Clear saving in costs and development time**

SAE J1939

- Features
  - Developed by Society of Automotive Engineers heavy trucks and bus division (SAE)
  - Use of the 29 identifiers
  - Support of real-time close loop control
- Applications
  - Light to heavy trucks
  - Agriculture equipment e.g. tractors, harvester etc…
  - Engines for public work

Smart Distributed System (SDS)

- Features
  - Created by Honeywell
  - Close to DeviveNet, CAL & CANopen

CAN: a Large Field of Applications

- Building Automatisation
- Domestic & Food distribution appliances
- Automotive & Transportation
- Robotic
- Production Automatisation
- Medical
- Agriculture

Building Automatisation

- Heating Control
- Air Conditioning (AC)
- Security (fire, burglar…)
- Access Control
- Light Control

Domestic & Food distribution appliances

- Washing machines
- Dishes cleaner
- Self-service bottle distributors connected to internet

Automotive & Transportation

- Automotive
  - Dash board electronic
  - Comfort electronic
- Ship equipment
  - Train equipment
  - Lifts
  - Busses
  - Trucks
  - Storage transportation systems
  - Equipment for handicapped people
- Service & Analysis systems

Robotic

- Tool machines
- Transport systems
- Assembly machines
- Packaging machines
- Knitting machines
- Plastic injection machines
- etc...

Production Automatisation & Robotic

- Control and link of production machines
- Production control
- Tool machines
- Transport systems
- Assembly machines
- Packaging machines
- Knitting machines
- Plastic injection machines
- etc...

Agriculture

- Harvester machines
- Seeding/Sowing machines
- Tractor control
- Control of live-stock breeding equipment

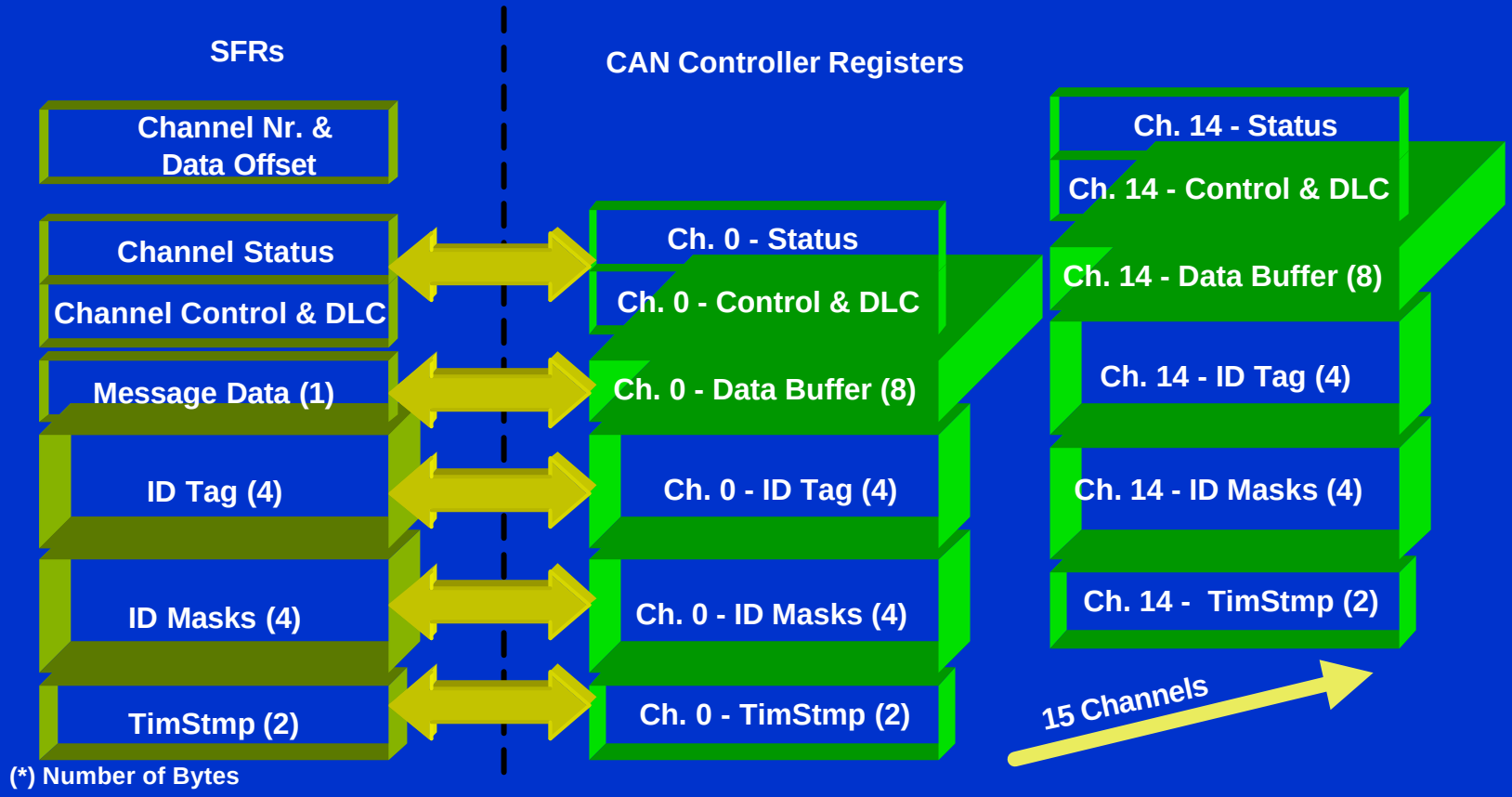
CANary: the ultimate Flash-based CAN microcontrollers

- Atmel CAN Bus Controller
  - Main Features
  - Mailbox concept (1)
  - Mailbox concept (2)
  - Channel Data Buffer (1)
  - Channel Data Buffer (2)
  - Autobaud & Listening Mode
  - Auto Reply Mode
  - Time Triggered Mode
  - Error Analysis Functions
  - CAN Self Test
  - Atmel CANary Controller
  - Conclusion
- CANary Family Benefits
  - In-System-Programming
  - Secured Boot Flash Memory
  - Advanced CAN Controller
  - Advanced C51 Core
- T89C51CC01
  - Block Diagram
  - Features (1) Features (2)
  - Advantages
  - Applications
- T89C51CC02
  - Block Diagram
  - Features (1) Features (2)
  - Advantages
  - Key messages
  - Applications
- Atmel CAN MCU Roadmap

CAN Controller: Main Features

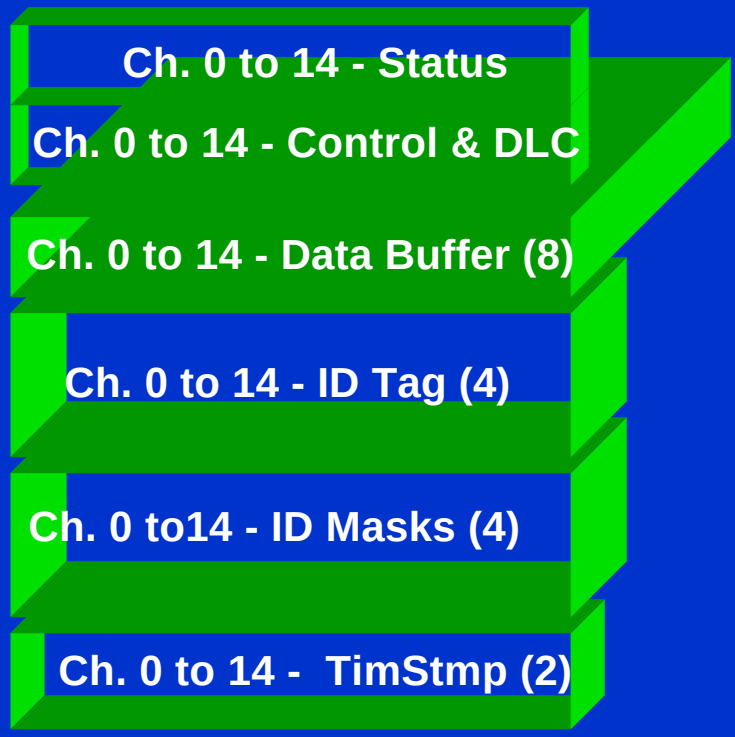
- Full validation by iVS/C&S Wolfenbüttel/Germany
- 1 MHz/sec CAN Bus Data Rate at 8 MHz Crystal
- Frequency in X2 Mode
- CAN 2.0A and 2.0B programmable / Channel
- 15 Channel with 20 Bytes of Control & Data / Channel
- 120 Bytes Reception Buffer
- Support of Time Triggered Communication (TTC)
- Auto Baud, Listening & Automatic Reply mode
- Mail Box addressing via SFRs
- All Channel features programmable on-the-fly

CAN Controller: Mailbox concept (1)



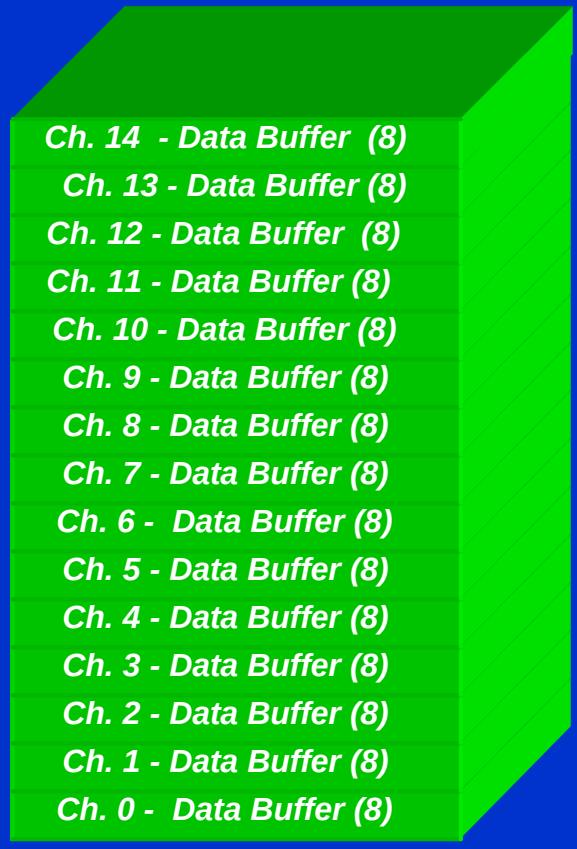
CAN Controller: Mailbox concept (2)

- Channel features
  - 32 bit of ID Mask Register
  - 32 bit of ID Tag Register
  - 64 bit of cyclic Data Buffer Register
  - 16 bit of Status, Control & DLC
  - 16 bit of Time Stamp Register



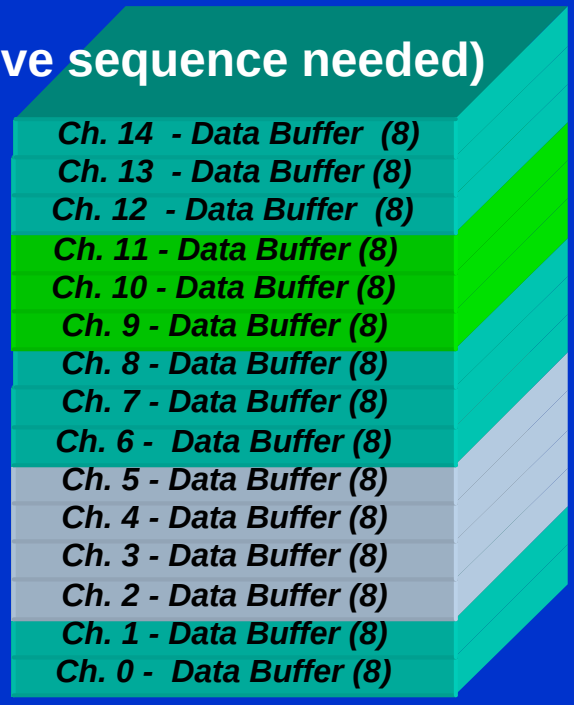
CAN Controller: Channel Data Buffer (1)

- Main Features
  - 15 Channels of 8 Byte (120 Bytes) Data Buffer
  - All Channels programmable as:
    - Receiver
    - Transmitter
    - Receiver Buffer
  - Highest Priority for lowest Channel Nr.
  - Interrupts at:
    - Correct Reception of Message
    - Correct Transmission
    - Reception Buffer full



CAN Controller: Channel Data Buffer (2)

- **Reception Buffer Features:**
  - Several Channels with same ID Mask (no important message will be missed)
  - Lowest Channel Number served first
  - Each Channel can participate (no consecutive sequence needed)



Example:

Reception Buffer

Receiver

Transmitter

CAN Controller: Autoband & listening mode

- CAN monitoring without influence to the bus lines
  - No acknowledge by error frames
  - Error counters are frozen
  - Only reception possible
  - No transmission possible
  - Full error detection possible
- Bit-rate adaption support
  - Hot-plugging of bus nodes to running networks with unknown bit-rate

CAN Controller: Automatic Reply Mode

- Automatic Message Transfer
  - Automatic message transfer after
  - reception of Remote Frame
  - Deferred message transfer after
  - reception of Remote Frame
  - Automatic Retransmission of Data Frames under
  - Software control

CAN Controller: Time Triggered Communication (TTC)

- Support of Real Time Applications
  - Single shot transmission
  - 16 bit CAN timer with IT at overflow
  - 16 bit Time Stamp Register / Channel
  - Trigger for Time Stamp Register at
  - End of Frame (EOF) or Start of Frame (SOF)

CAN Controller: Error Analysis Functions

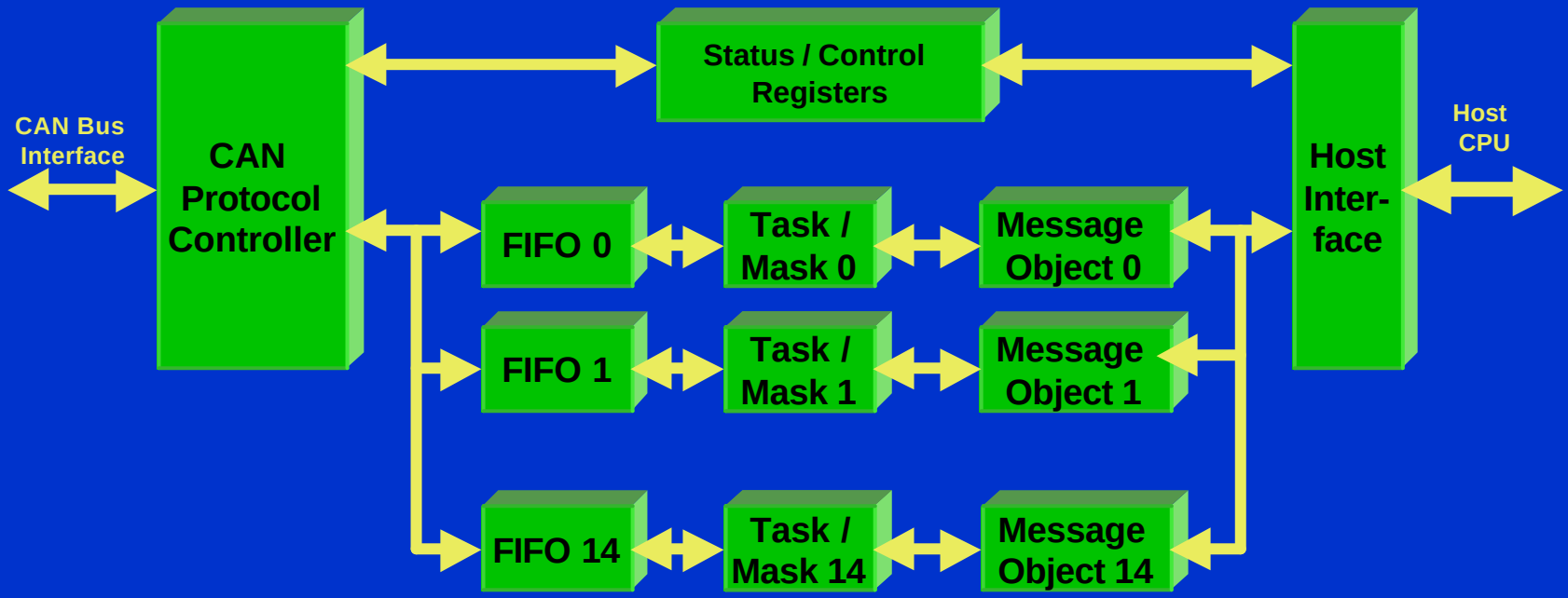
- Channel Status Register (Error Capture Register)
  - Associated to each Channel
  - Type of CAN bus errors: DLC warning, Transmit OK, Receive OK, Bit error (on in transmit), Stuff error, CRC error, Form error, Acknowledgement error
- Error Interrupts
  - Bus errors, Error passive and Error warning
- Readable Error Counters

CAN Controller: CAN Self Test

- Analysis of own transmitted Message
- Support of local self test
- Support of global self test
- Software comparison of Tx & Rx buffer
- Monitoring of CAN bus traffic

CAN Controller: The Atmel CANary controller

- 15 Message objects (Channels), each with filtering, masking and FIFO buffer
- All Channel features programmable on-the-fly



CAN Controller: Conclusion

- 1 MHz/sec CAN Bus Data Rate at 8 MHz Crystal Freq.
- CAN 2.0A and 2.0B programmable / Channel
- 15 Channel with 20 Bytes of Control & Data / Channel
- 120 Bytes Reception Buffer
- Support of Time Triggered Communication (TTC)
- Auto Baud, Listening & Automatic Reply mode
- Mail Box addressing via SFRs
- All Channel features programmable on-the-fly
- All competitors knows how to do a basic CAN, but all CAN differ in acceptance filtering and frame storage capabilities.

CANary Microcontrollers: Advantages (1)

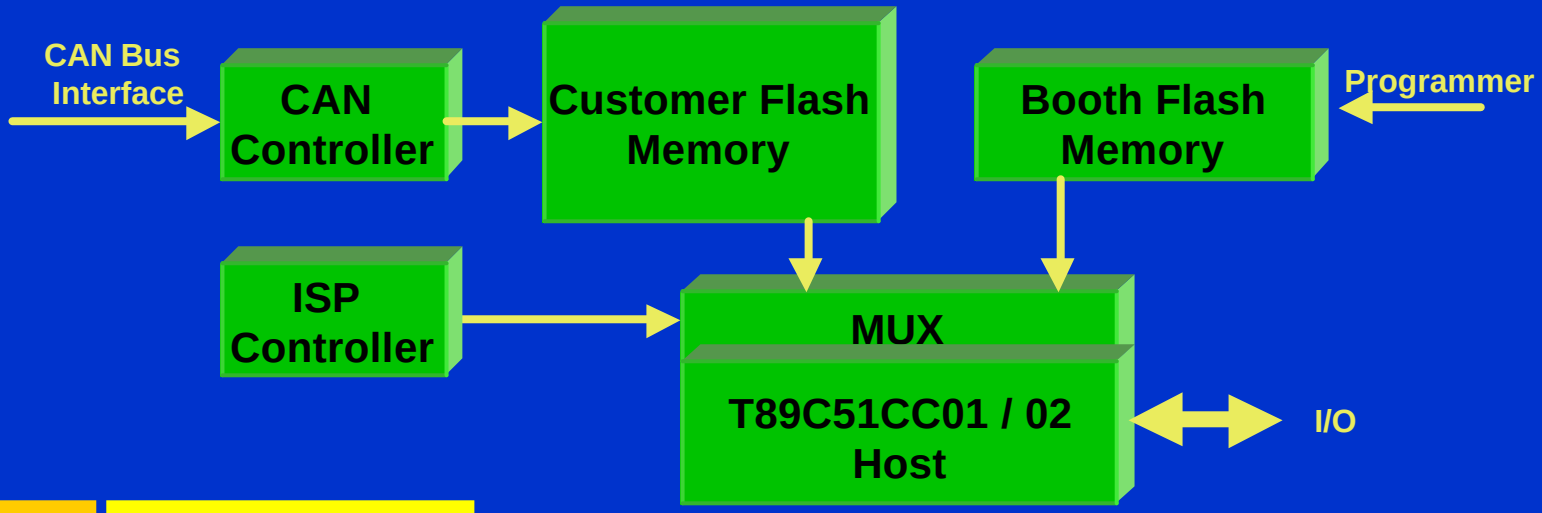
In-System-Programming (ISP)

- Code loading and EEPROM up-date in embedded applications via CAN Bus or UART
- Customer 32kByte Program memory programmable via CAN bus or UART
- Strong Programming Security by separation of 2kByte Flash Boot Loader Memory and 32kByte of Customer Flash programmable Memory
- Application of Atmel or Customer Boot Loader
- Atmel delivers Application Programming Interfaces (API)

CANary Microcontrollers: Advantages (2)

Secure Boot Flash Memory

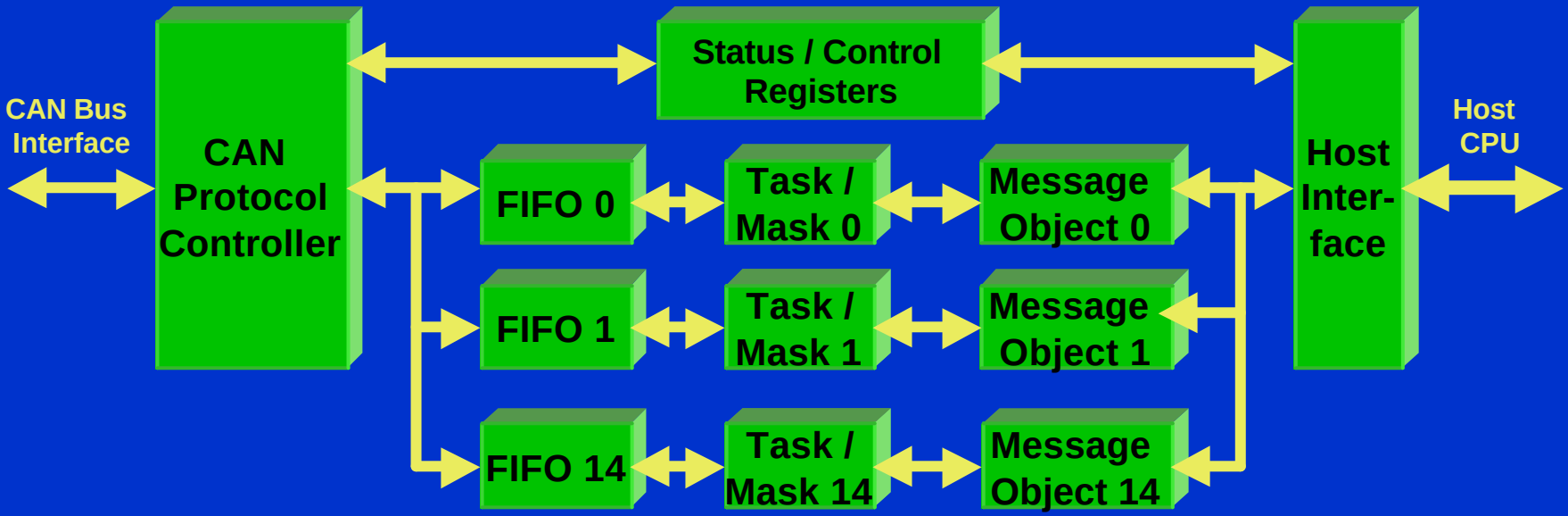
- No involuntary destruction of Booth Flash Memory possible during embedded operations due to its unique parallel access by programmer
- No Idle situation for Micro Controller during ISP of Customer Memory due to separation of Customer and Booth Memory



CANary Microcontrollers: Advantages (3)

Advanced CAN Controller

- 15 Message objects (Channels), each with filtering, masking and FIFO buffer
- All Channel features programmable on-the-fly

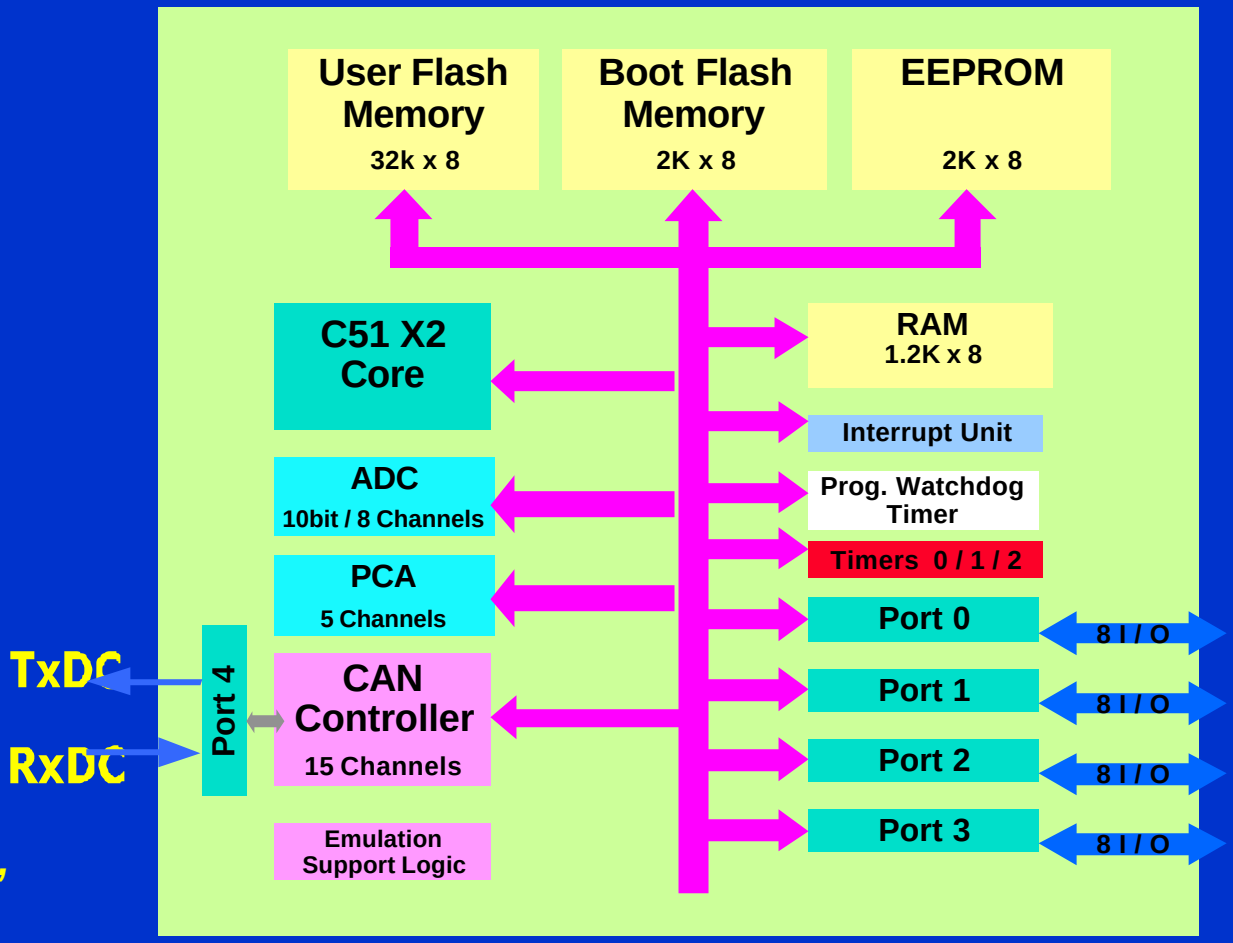


CANary Microcontrollers: Advantages (4)

Advanced C51 Core

- Fully static operation
- Asynchronous port reset
- Second data pointer
- Inhibit ALE
- X2 CORE
- 4 level priority interrupt system
- Enhanced UART
- Programmable Timer 2 clock out
- Power Consumption reduction
- Wake up with external interrupts from Power Down

CANary Microcontrollers: T89C51CC01 Block Diagram



Packages: TQFP44, CA-BGA64

CANary Microcontrollers: T89C51CC01 Features

- C52 Core compatible
- Up to 40 MHz operation (X2 mode)
- X2 Core
- Double Data Pointer
- 32 Kb FLASH ISP, 2 Kb FLASH Boot Loader
- 2Kb EEPROM
- 1.25 k RAM (256b scratchpad RAM + 1kb XRAM)
- 3-16 bit Timers (T0,T1,T2)
- Enhanced UART
- CAN Controller with 15 channels (2.0A and 2.0B)

CANary Microcontrollers: T89C51CC01 Features (Cont 'd)

- 10 bits A/D with 8 Channels
- 5 I/O Ports
- Programmable Counter Array
  - 5 channels, 5 Modes:
    - PWM, Capture, Timer, Counter, Watchdog(Channel 4 only)
- 1Mbit /sec CAN at 8MHz Crystal Frequency (X2mode)
- Temperature: -40 to 85°C
- Voltage: 3 to 5 Volt +/-10%
- Packages: PLCC44, TQFP44, CA-BGA64

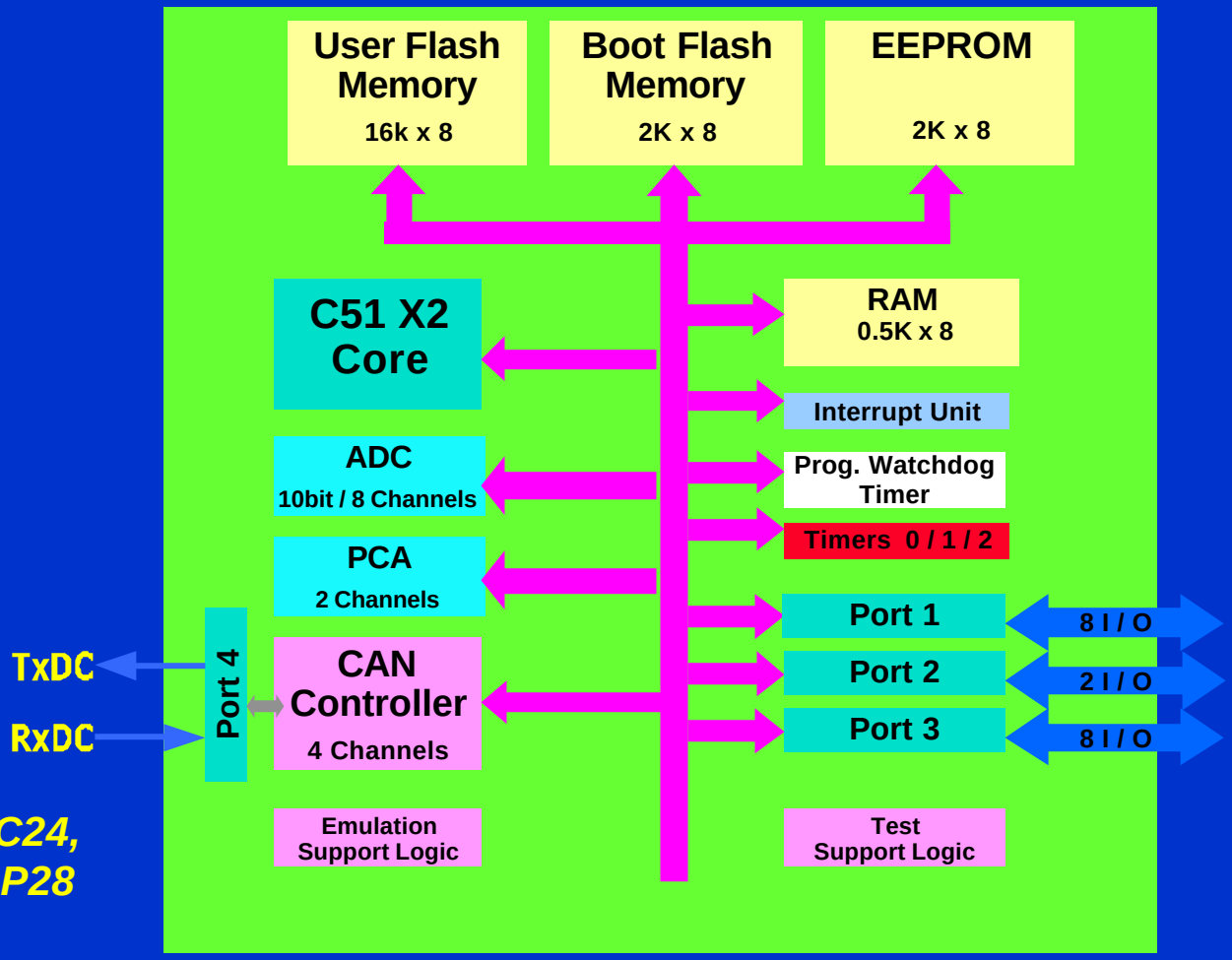
CANary Microcontrollers: T89C51CC01 Advantages

- T89C51CC01 is the first CAN Controller of a new generation for smart embedded applications which offers Flash and ISP Technology for Customer Code & Application Parameter up-date in a 44-pin package .
- For security reasons the 2kB Boot Memory is physically separated from 32kB Customer memory.
- Further for security reasons the Boot memory can be written only in Parallel Mode outside the application.
- 10b ADC & 5 channel PCA allow T89C51CC01 single-chip applications in most cases
- Included in the delivery is a wide range of Application Programming interfaces (API) concerning ISP, EEPROM, Security, Customer & Boot Flash

CANary Microcontrollers: T89C51CC01 Applications

- T89C51CC01 is destinated to embedded CAN bus applications which requests an easy up-date of customer code and application parameters, possible thanks to In-System-Programming (ISP) mode via CAN bus
- T89C51CC01 allows beside the communication with the CAN bus the capture of analog parameters (10b ADC) and the control of eg. a stepper motor (5 channel PCA)
- T89C51CC01 fits in industrial and automotive applications for body and comfort electronic.

CANary Microcontrollers: T89C51CC02 Block Diagram



Packages: SOIC24, SOIC28, TSSOP28

CANary Microcontrollers: T89C51CC02 Features

- C52 Core and T89C51CC01 compatible
- Up to 40 MHz operation (X2 mode)
- X2 Core
- Double Data Pointer
- 16 kb FLASH ISP
- 2 Kb FLASH Boot Loader
- 2 kb EEPROM
- 512b RAM (256b scratchpad RAM + 256b XRAM)
- 3-16 bit Timers (T0,T1,T2)
- Enhanced UART
- CAN Controller with 4 channels (2.0A and 2.0B)

CANary Microcontrollers: T89C51CC02 Features (Cont 'd)

- 10 bit ADC with 8 Channels
- 3 I/O Ports
- Programmable Counter Array (PCA)
  - 2 channels, 5 Modes
    - PWM, Capture, Timer, Counter
- 1MBit/sec CAN at 8MHz Crystal Frequency (X2 mode)
- Temperature: -40 to 85°C
- Voltage: 3 to 5 Volt +/10%
- Package: SOIC28, Plcc28, TQFP32, TSSOP28

CANary Microcontrollers: T89C51CC02 Advantages

- T89C51CC02 is designed for embedded low-end, high volume applications
- Same functions like included in T89C51CC01
- Reduced costs in a 24 pin package.
- Main difference to T89C51CC01:
  - No access possible to external RAM/ROM via Ports 0 & 2
  - Customer Flash Memory 16kBytes
  - On-chip RAM: 512Bytes
  - 4 channel CAN Controller
  - 2 channel PCA
- All other functions will remain identical in the sense that the T89C51CC01 development tools can be used for T89C51CC02.

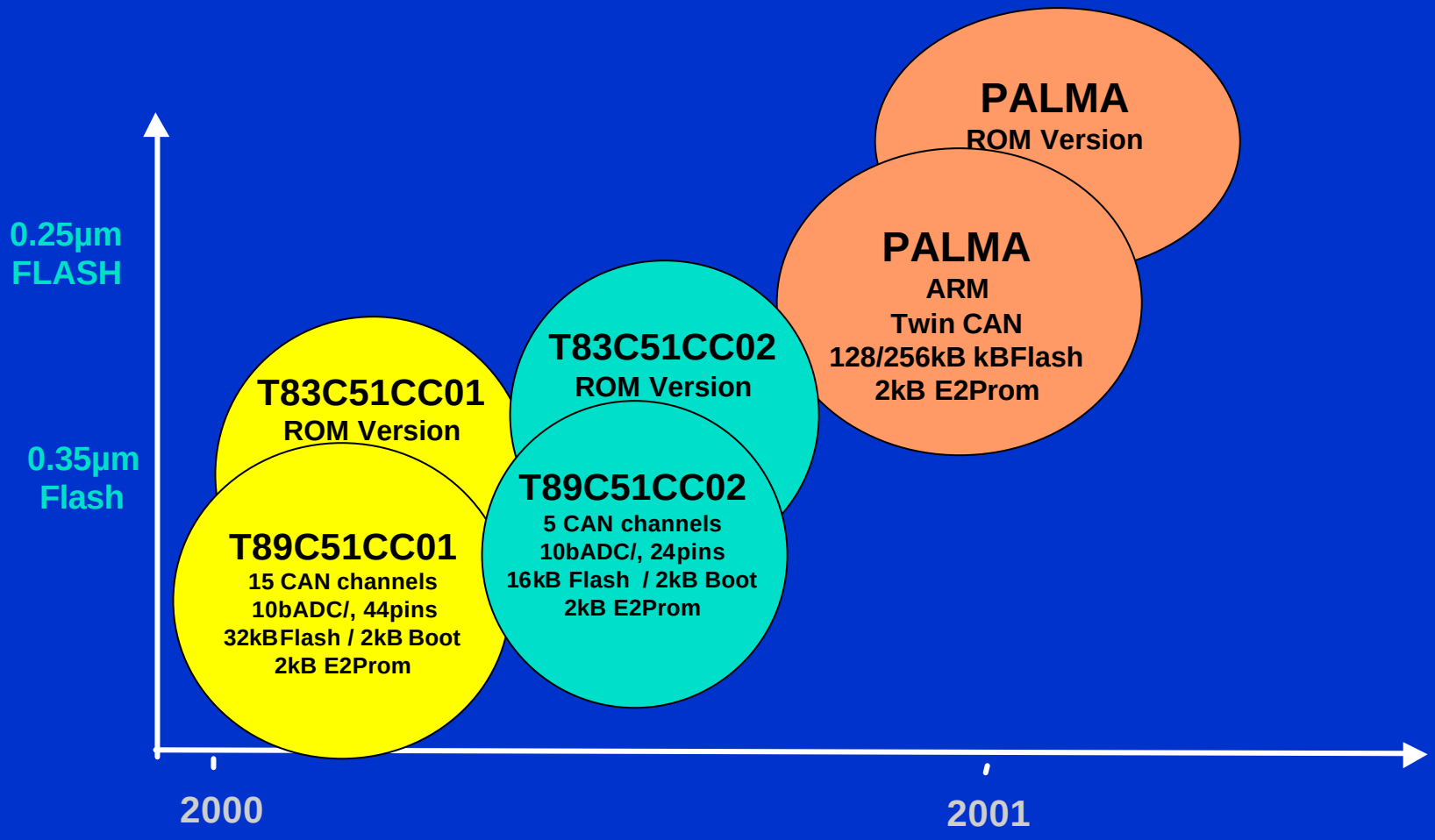
CANary Microcontrollers: T89C51CC02 Key messages

- T89C51CC02 presents a low-pin Count package for embedded high volume applications with the availability of all T89C51CC01 functions like:
  - In-System-programming of Code program & Application Parameters via CAN bus or UART
  - Physical separation of Boot Loader Memory (2kByte) and Customer Flash Memory (16kByte)
  - 2 kB EEPROM
  - 10b ADC
  - 2 channel PCA
- Remark: Low-pin count feature will not allow the access to external RAM/ROM via P0/P2.

CANary Microcontrollers: T89C51CC02 Applications

- Sampling of Parameters (via ADC) and Control Functions (Motors etc. via PWM) of embedded applications in:
  - Building control (eg Air Condition, Security, Light, Access...) ,
  - Production machines (eg. Knitting, packaging machines),
  - Automotive applications (body / comfort electronic)

CANary Microcontrollers: Roadmap



CANary Microcontrollers Tools

A complete tool chain for quick development

C-Compilers / Simulators

KEIL SOFTWARE
RAISONANCE
TASKING
IAR SYSTEMS

Emulators

FPGA bond out emulator:

RAISONANCE

Enhanced Hook emulators:

nohau
hitex
CEIBO Development Tools of Choice
SIGNUM SYSTEMS

Programming tools

DATA I/O
SMS
HI-LO SYSTEMS
BP MICROSYSTEMS
Advin Systems Inc.
System General
EQUINOX
Advantech

Conclusion

- CAN: The most used protocol in industrial & automotive applications
  - strong support by many HLPs & CAN tool vendors
- CANary unique feature in its range:
  - including an advanced powerful CAN Controller
  - In-System-Programming (ISP) of Program Flash via CAN bus
  - separated Flash memories for Program & Boot functions
- CANary: designed for industrial and automotive applications

CANary the ultimate CAN Controller