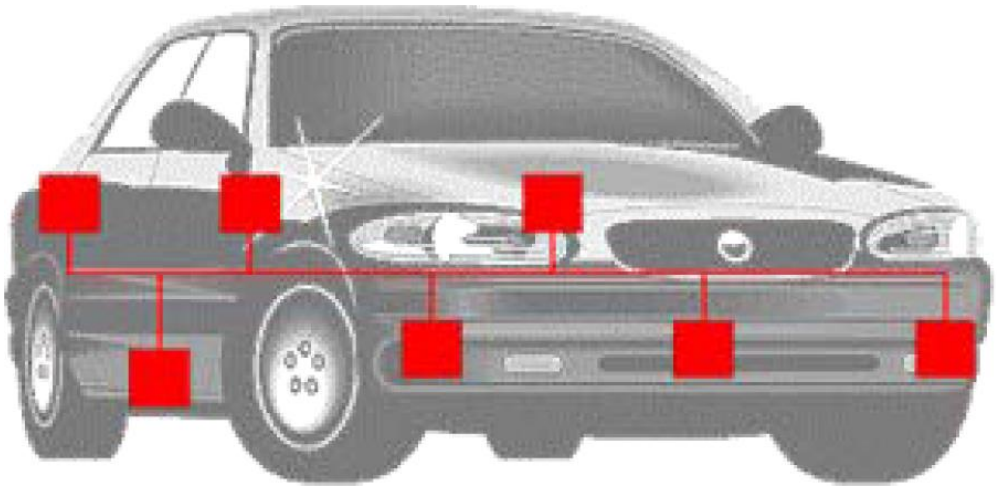
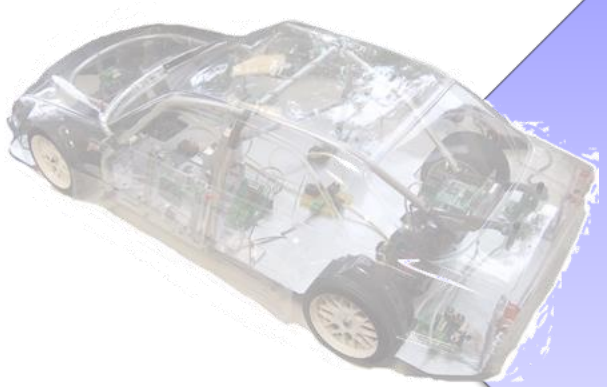
CPS Networks

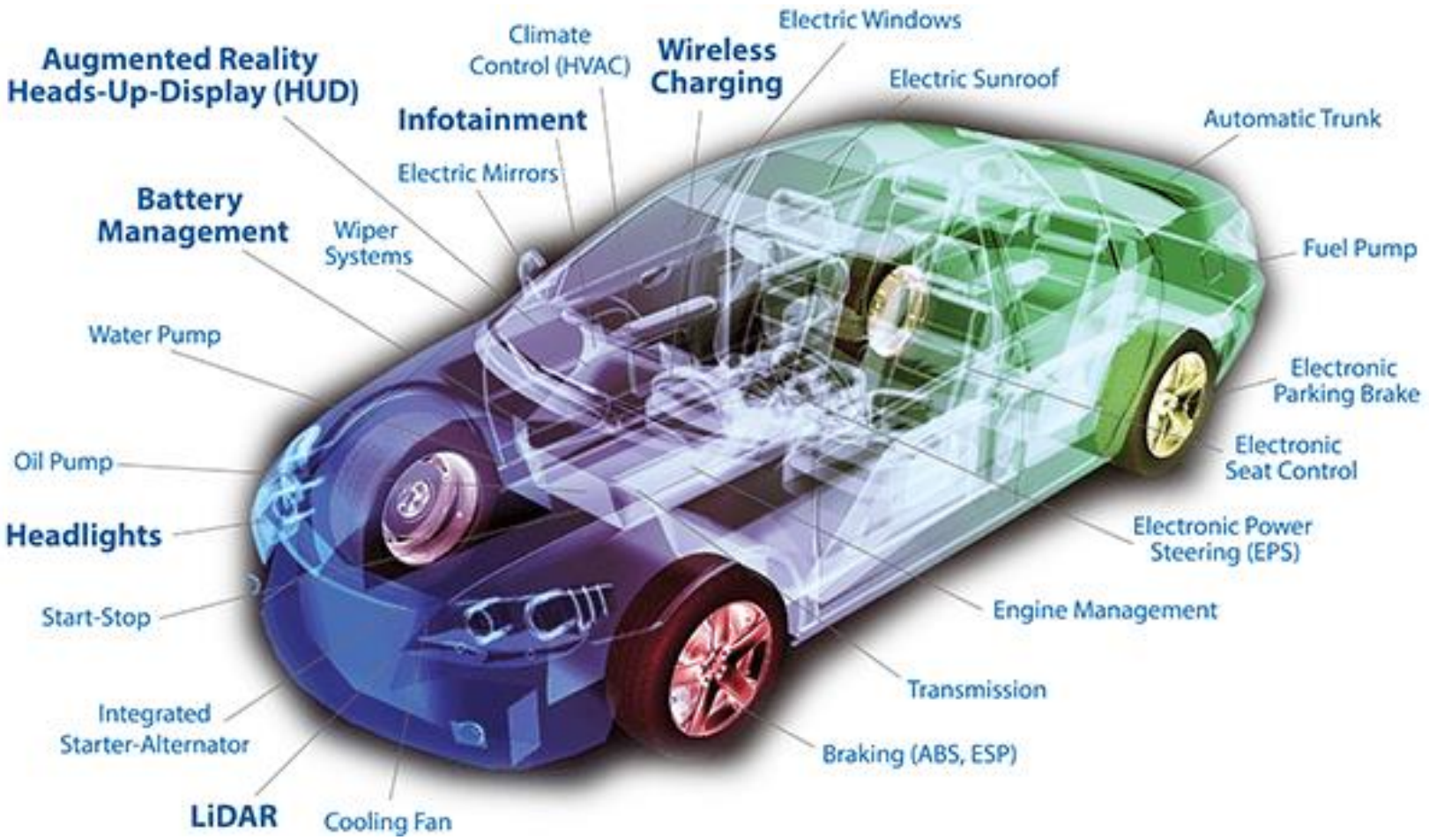
Institute of Technology, Nirma University



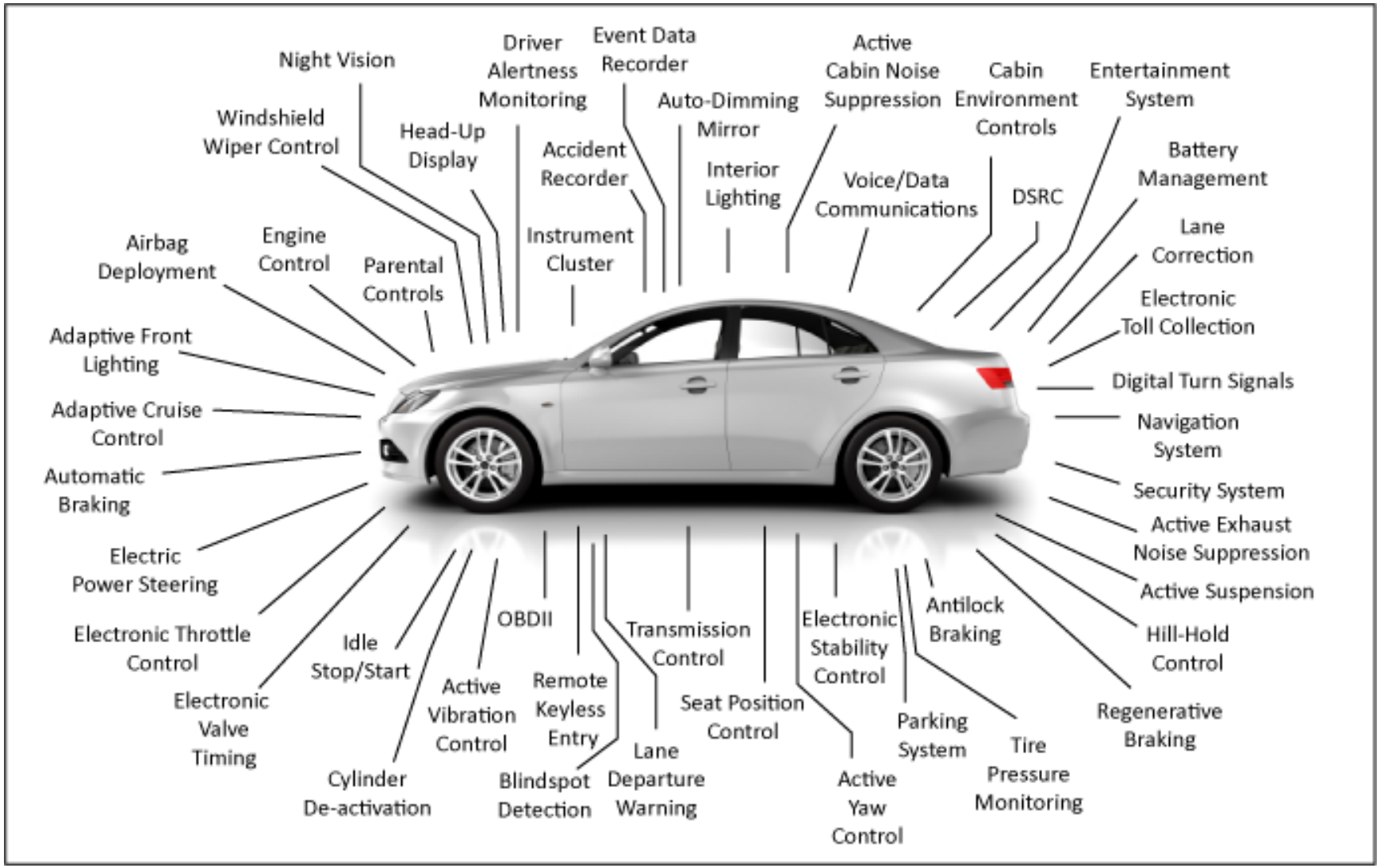


CAN
Controller Area Network

Microcontrollers in Automotive



Microcontrollers in Automotive



Outline

- CANBUS Introduction
  - What is CANBUS?
  - WHY?
  - Who uses CANBUS?
  - CANBUS history
  - CANBUS timeline
- CANBUS Characteristics
  - OSI Model
  - Physical Layer
  - Transmission Characteristics
- Message Oriented Communication
- Message Format
- Bus Arbitration

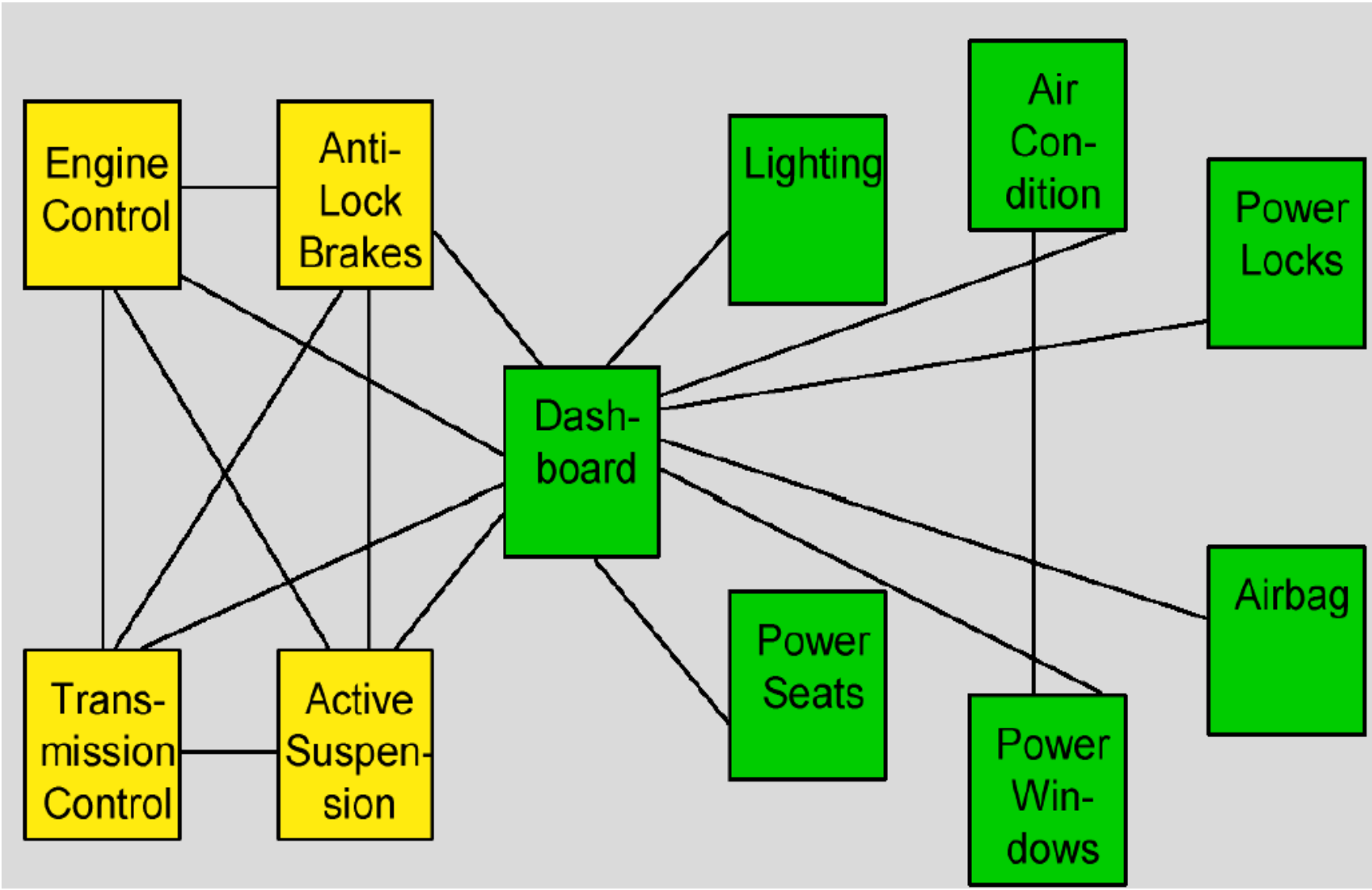
What is CANBUS?

CANBUS or CAN bus – **C**ontroller **A**rea **N**etwork **bus**

An automotive serial bus system developed to satisfy the following requirements:

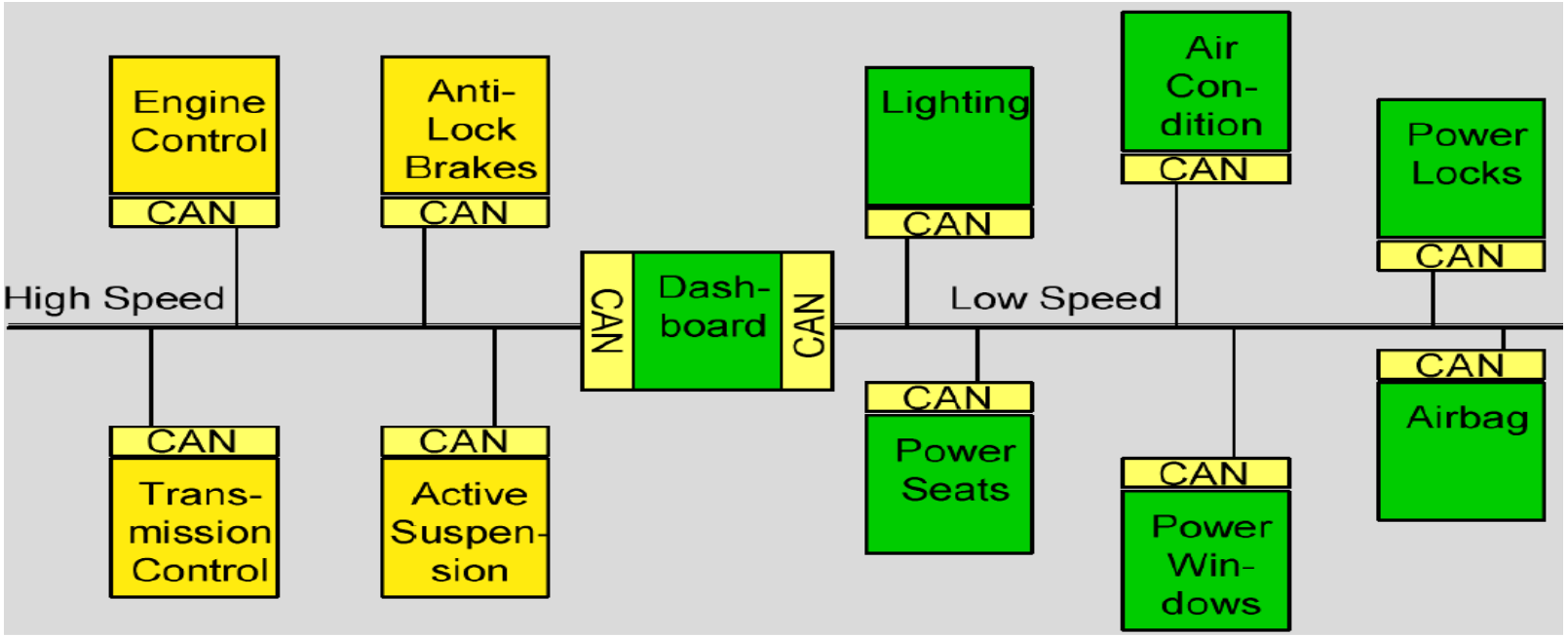
- Network multiple microcontrollers with 1 pair of wires.
- Allow microcontrollers communicate with each other.
- High speed, real-time communication.
- Provide noise immunity in an electrically noisy environment.
- Low cost
- Asynchronous Serial Bus
- Simple 2-wire differential bus
- Absence of node slave addressing

Before CAN


Engine Control
Anti-Lock Brakes
Lighting
Air Con-dition
Power Locks
Dash-board
Airbag
Power Seats
Trans-mission Control
Active Suspen-sion
Power Win-dows

With CAN

The solution to this problem was the connection of the control systems via a serial bus system. This bus had to fulfill some special requirements due to its usage in a vehicle. With the use of CAN, point-to-point wiring is replaced by one serial bus connecting all control systems. This is accomplished by adding some CAN-specific hardware to each control unit that provides the "rules" or the protocol for transmitting and receiving information via the bus.



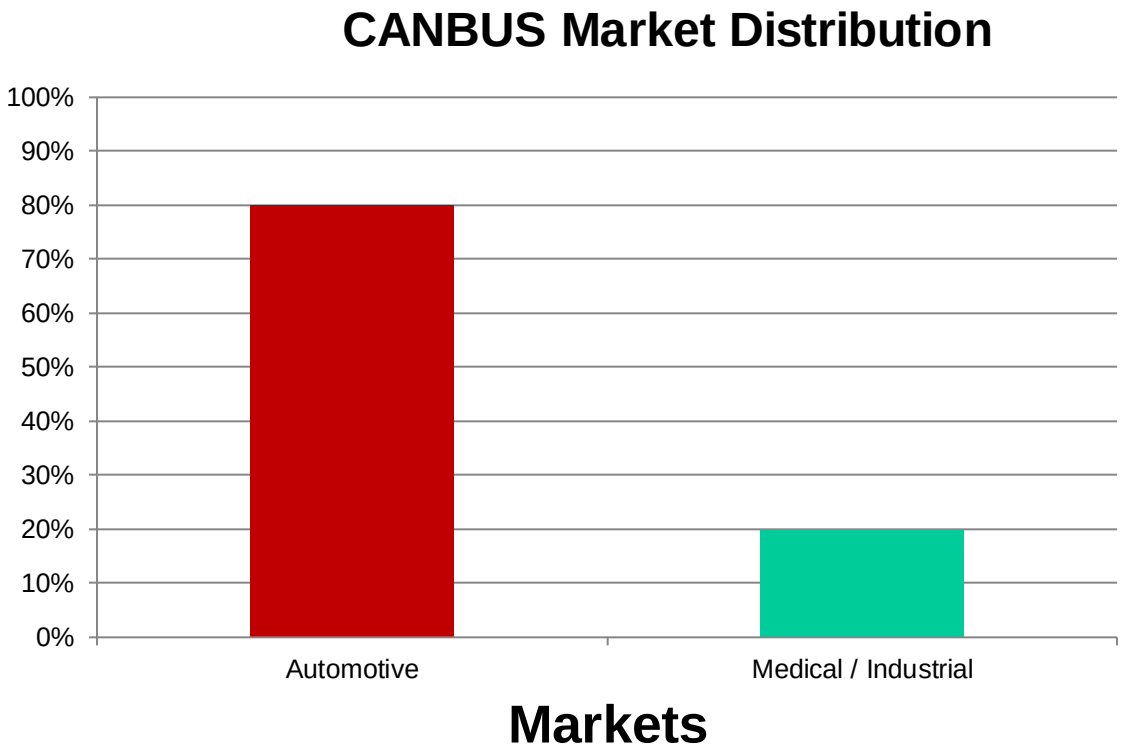
Why CAN ?

- **Mature Standard**
  - ✓ CAN protocol more than 16 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 (up to 1Mb/s)
- **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**
  - ✓ CRC error detection mechanism
- **Fault Confinement**
  - ✓ Built-in feature to prevent faulty node to block system

Most used protocol in industrial and automotive world

Who uses CANBUS?

- Designed specifically for automotive applications
- Today - industrial automation / medical equipment



Manufacturers

Over 20 different chip manufacturers produce microcontrollers with on-chip CAN interfaces.

- Some more notable ones are:
  - Cygnal
  - Intel
  - Motorola
  - NEC
  - Phillips
  - Toshiba

CANBUS History

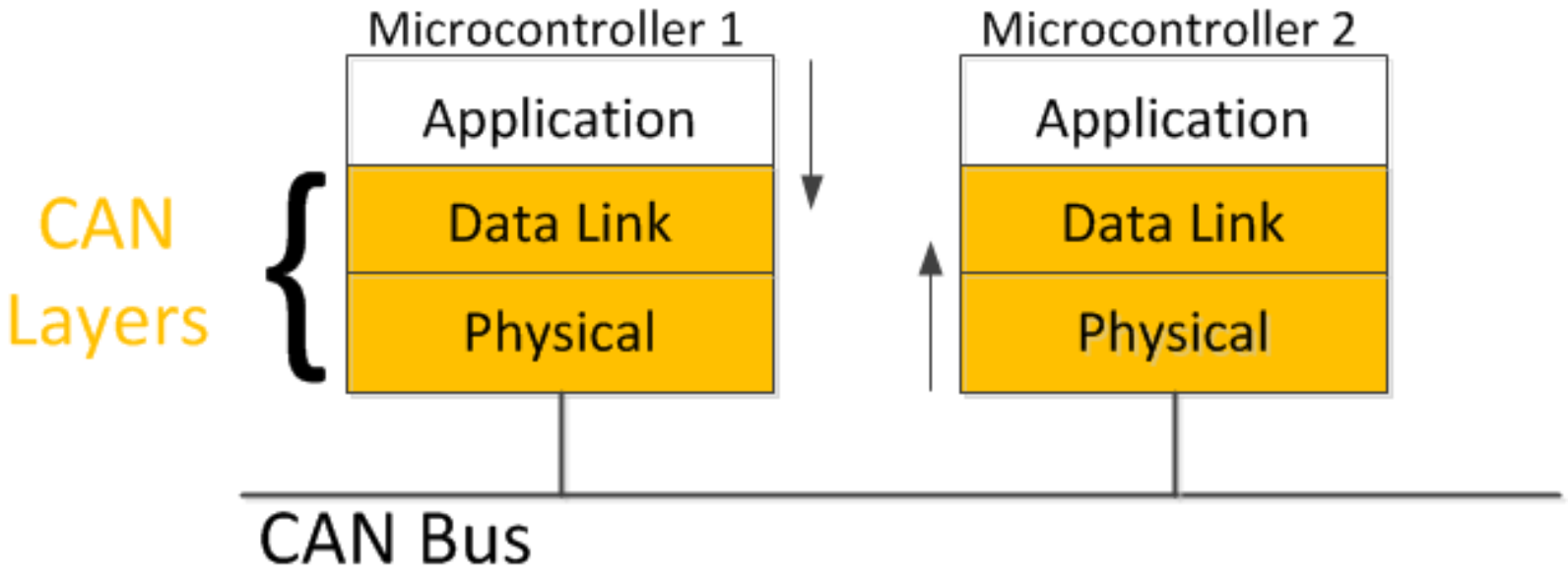
- First idea - The idea of CAN was first conceived by engineers at Robert Bosch in Germany in the early 1980s.
- Early focus - develop a communication system between a number of ECUs (electronic control units).
- New standard - none of the communication protocols at that time met the specific requirements for speed and reliability so the engineers developed their own standard.

CANBUS Timeline

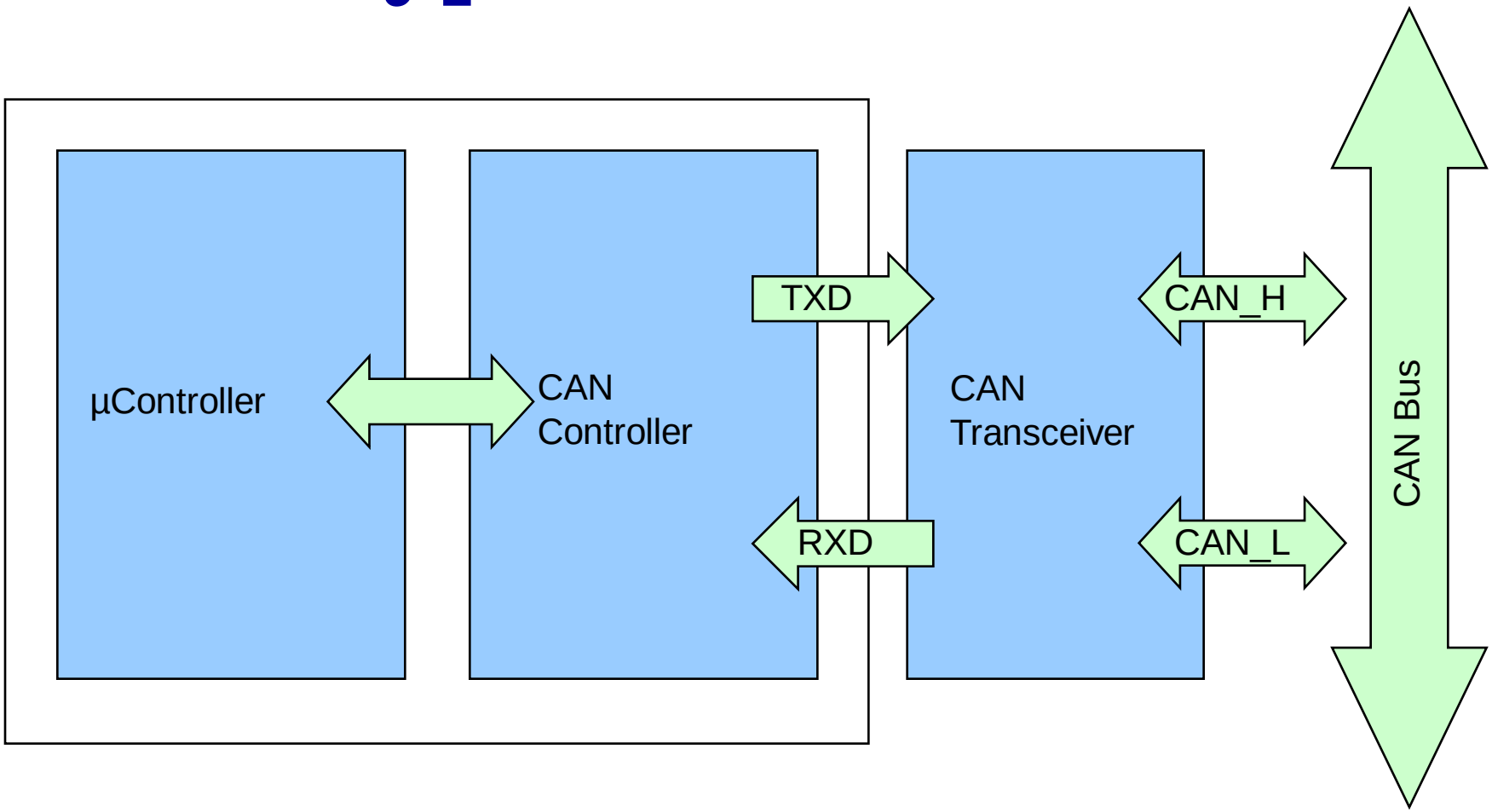
- 1983 : First CANBUS project at Bosch
- 1986 : CAN protocol introduced
- 1987 : First CAN controller chips sold
- 1991 : CAN 2.0A specification published
- 1992 : Mercedes-Benz used CAN network
- 1993 : ISO 11898 standard
- 1995 : ISO 11898 amendment
- Present : The majority of vehicles use CAN bus.

CANBUS and the OSI Model

- CAN is a closed network
  - – no need for security, sessions or logins.
  - - no user interface requirements.
- Physical and Data Link layers in silicon.



Typical CAN Node



CAN Bus is a simple 2-wire differential serial bus

CAN Bus is terminated on each side by a 120 Ohm resistor

CANBUS Physical Layer

- Physical medium – two wires terminated at both ends by resistors.
- Differential signal - better noise immunity.
- Benefits:
  - Reduced weight, Reduced cost
  - Fewer wires = Increased reliability

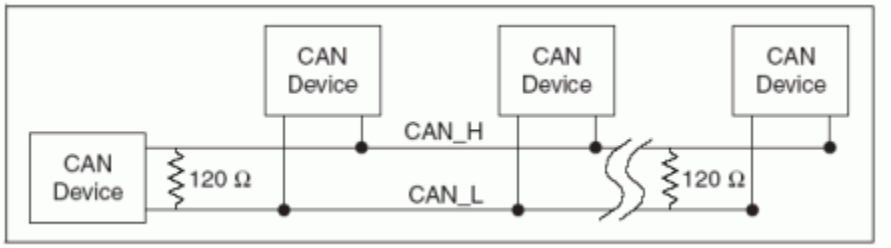


Fig. 1 High Speed CAN wiring

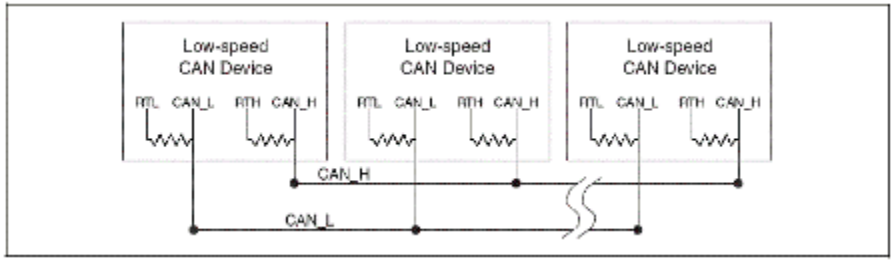


Fig. 2 Low Speed CAN wiring

Differentilal Signaling

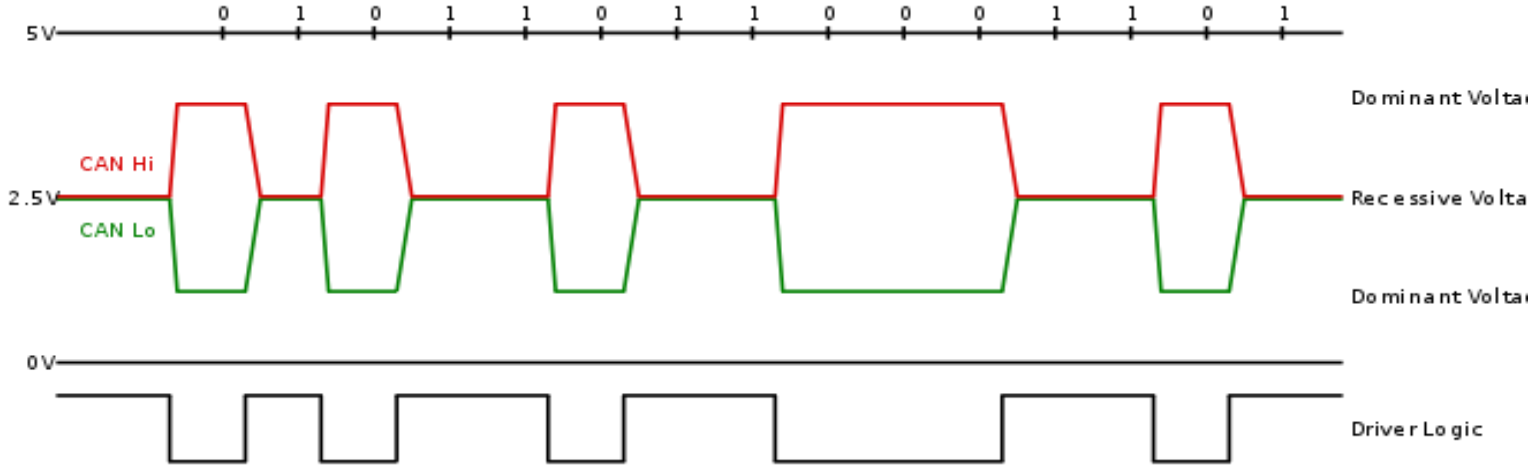
There are two types of CAN network based on speed and they both uses different voltage levels of signal.

- **High Speed CAN** offers baud rates from **40 Kbit/s to 1 Mbit/sec**, depending on cable length. This is the most **popular** standard for the physical layer, since it allows for simple cable connection between devices. This is the physical standard used in the **DeviceNet** and **CANopen** specifications. High speed CAN networks are terminated with 120 ohm resistors on each end of the network.

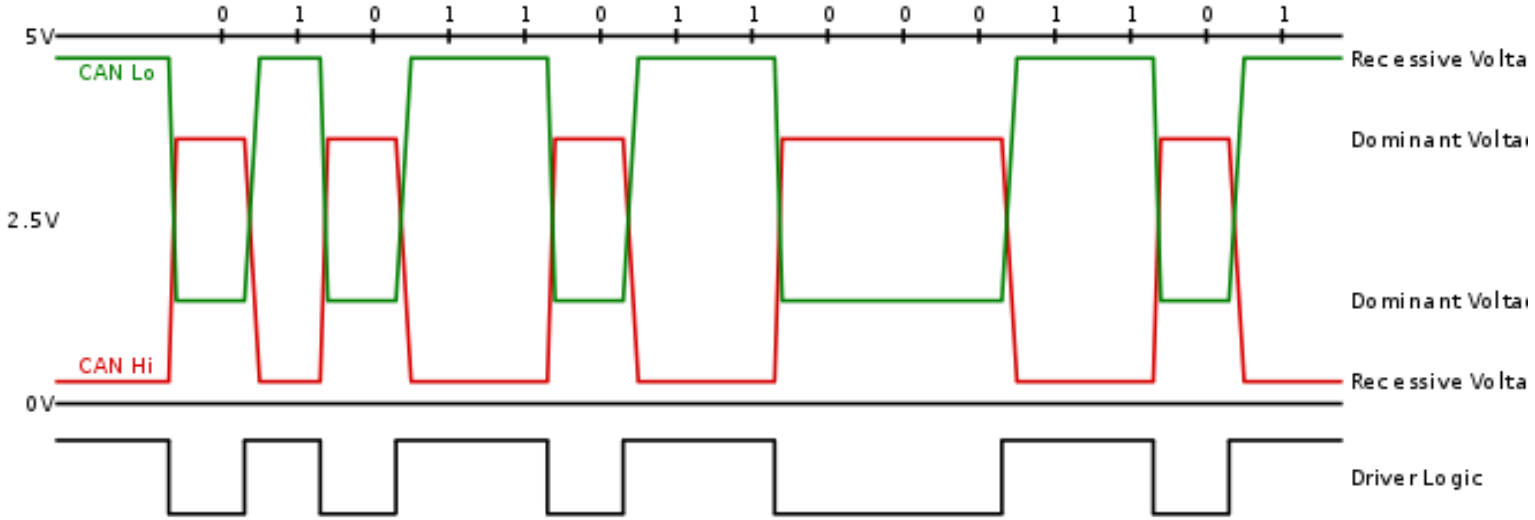
- **Low Speed/Fault Tolerant CAN** offers baud rates from **40 Kbit/s to 125 Kbits/sec**. This standard allows CAN bus communication to continue in case of a wiring failure on the CAN bus lines. In low speed/fault tolerant CAN networks, **each device has its own termination.**

Differential signaling

High Speed CAN Signaling. ISO 11898-2

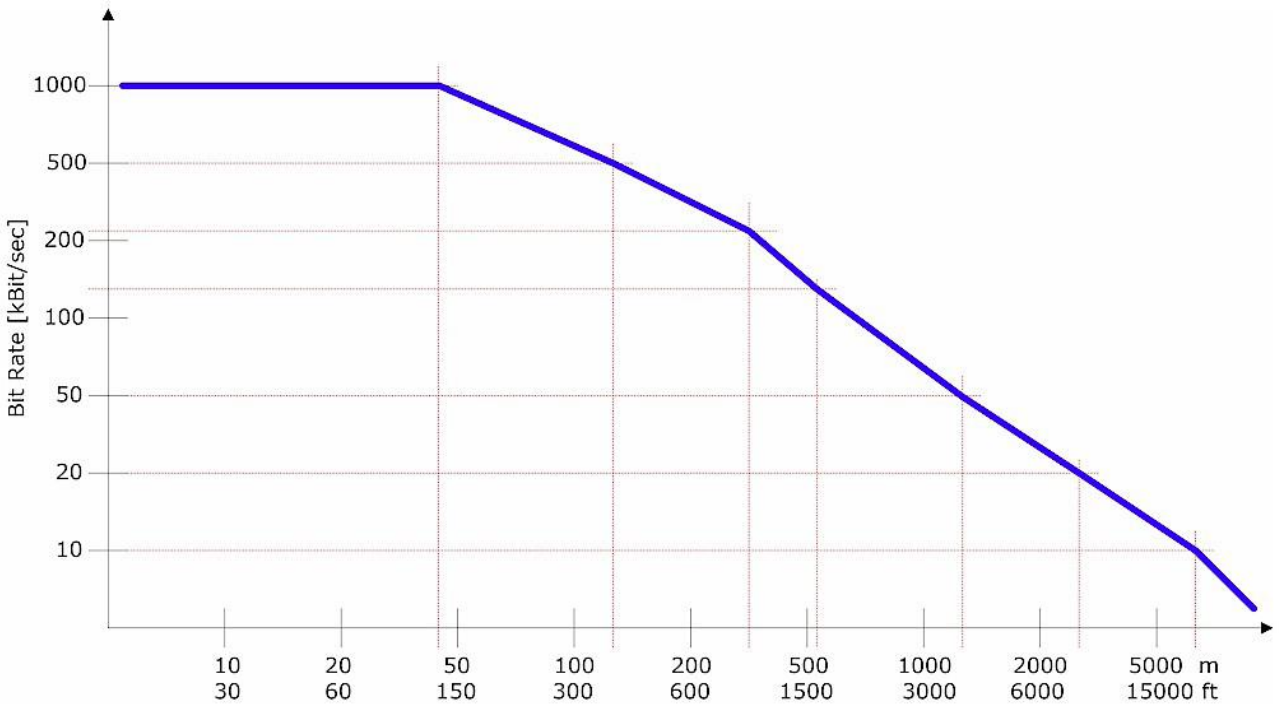


Low Speed Fault Tolerant CAN Network. ISO 11898-3



Transmission Characteristics

- Up to 1 Mbit/sec.
- Common baud rates: 1 MHz, 500 KHz and 125 KHz
- All nodes – same baud rate
- Max length:120’ to 15000’ (rate dependent)

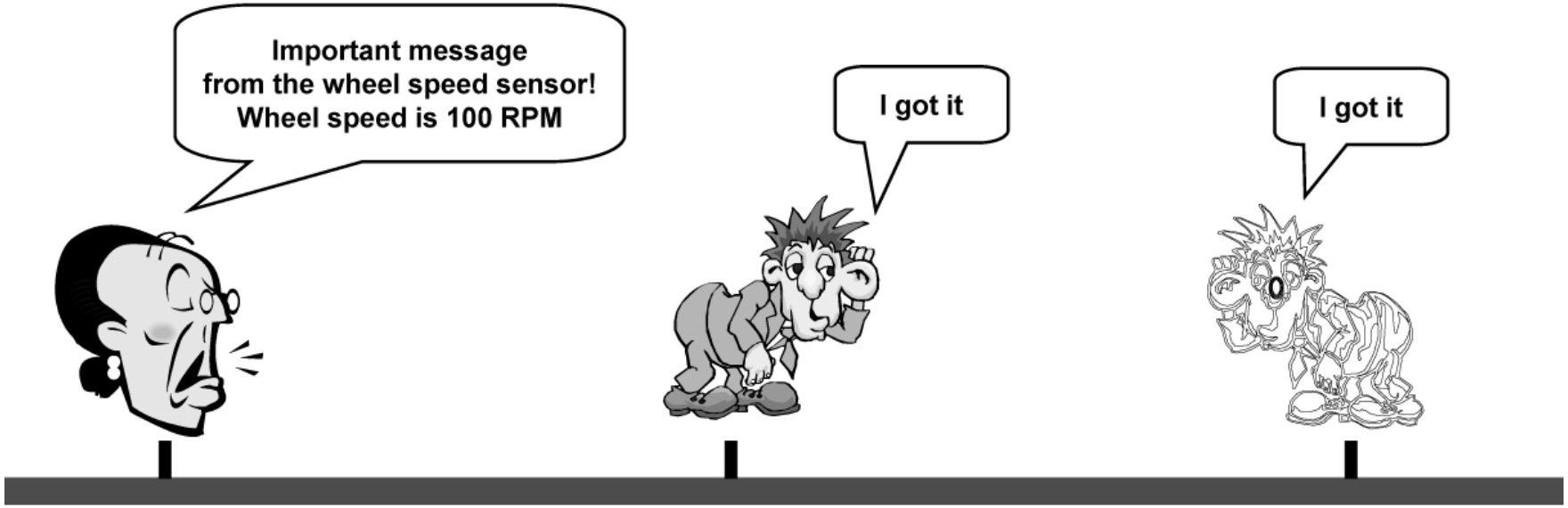


Bus Length/Bit rate trade-off

- 1M bit/sec 40meters (131 feet)
- 500k bit/sec 100 meters (328 feet)
- 250k bit/sec 200 meters (656 feet)
- 125k bit/sec 500 meters(1640 feet)

Message Oriented Transmission Protocol

- Each node – receiver & transmitter
- A sender of information transmits to all devices on the bus
- All nodes read message, then decide if it is relevant to them
- All nodes verify reception was error-free
- All nodes acknowledge reception

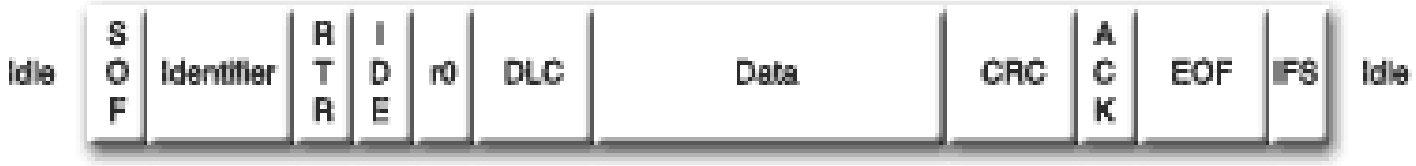


Types of Frame

There are four types

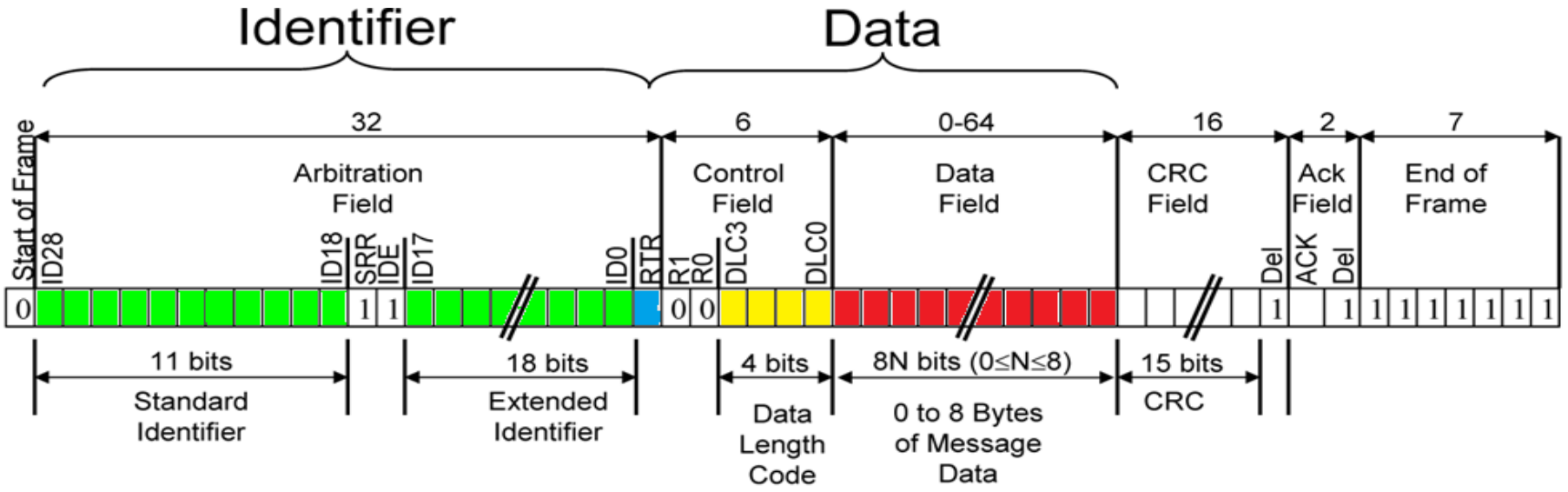
- Data frame – when a node sends data
- Remote Frame – when a node ask data from other node
- Error Frame – When error occurs the node that detects the error sends immediately this frame and all the other nodes on reception of this frame sends the error frame and the transmitter of original data frame stops on reception of error frame. Error frame is made up of six 0's or dominant bit followed by eight 1's or recessive bits. Transmitter resends the message.
- Overload Frame – this is sent by a node when it encounters congestion in order to notify the other nodes to wait a bit. Its format is same as the error frame but the difference is that error frame is sent at instance where error is detected, while overload frame is send after the End of Frame and before Inter Frame Space of the ongoing transmission frame. Transmitter does not resend the data frame.

Frame Format



- SOF – Start of Frame
- Identifier – Tells the content of message and priority
- RTR – Remote Transmission Request
- IDE – Identifier extension (distinguishes between CAN standard,11 bit identifier, and CAN extended, 29 bit identifier.)
- DLC – Data Length Code
- Data – holds up to 8 bytes of data
- CRC – “Cyclic Redundant Check” sum
- ACK – Acknowledge
- EOF – End of Frame
- IFS – Intermission Frame Space. Minimum number of bits separating consecutive messages.

Message Format



Example of Message Transaction

- Instrument panel ECU says "can anyone tell me what the block temperature is?

ID	Data
400	

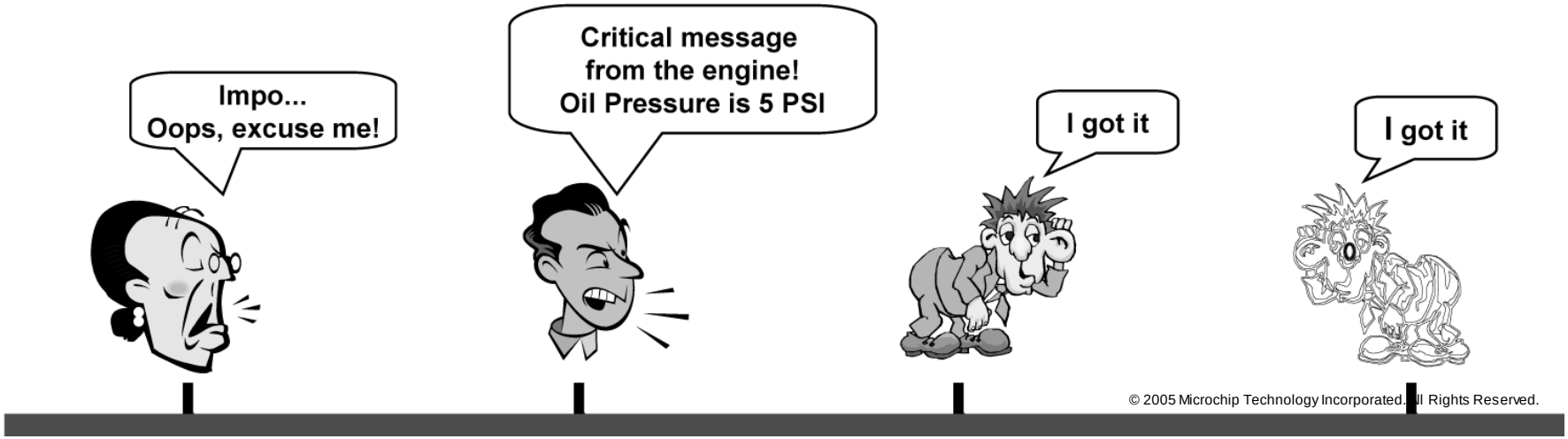
- Block ECU sees this message and issues a message "block temperature is 76 Celsius"

ID	Data
400	076

- Instrument panel ECU sees block temperature message and displays it on console

Bus Arbitration

- Arbitration – needed when multiple nodes try to transmit at the same time
- Only one transmitter is allowed to transmit at a time.
- A node waits for bus to become idle
- Nodes with more important messages continue transmitting



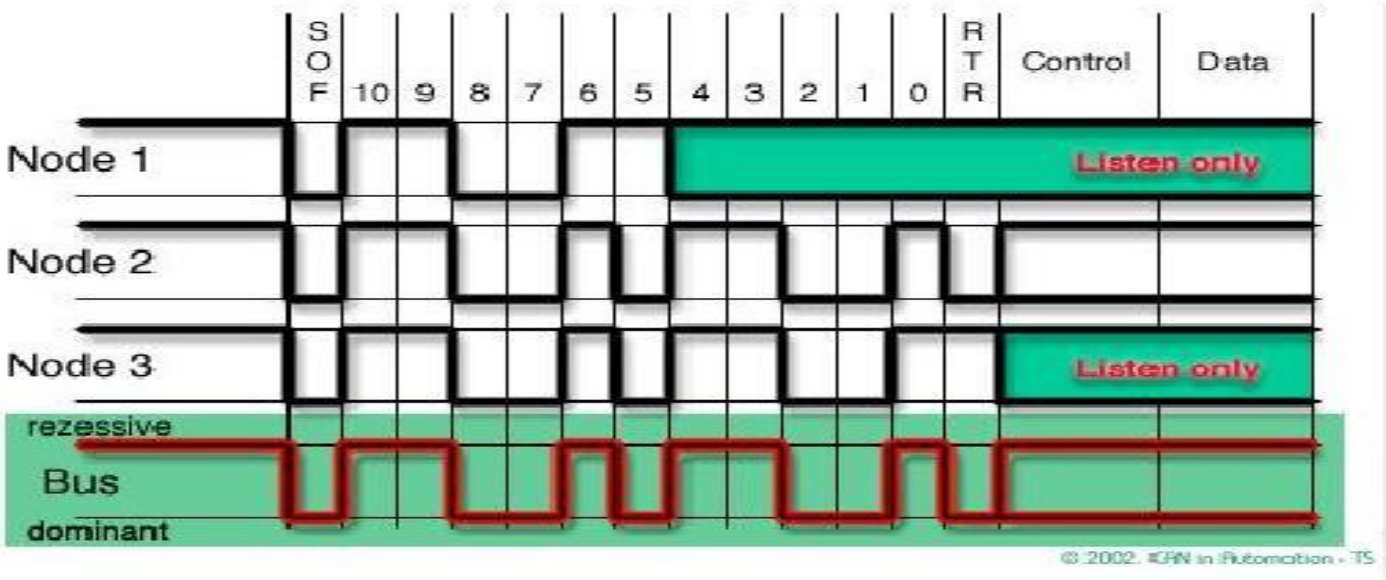
CAN bus

Bus Arbitration

- Message importance is encoded in message ID.

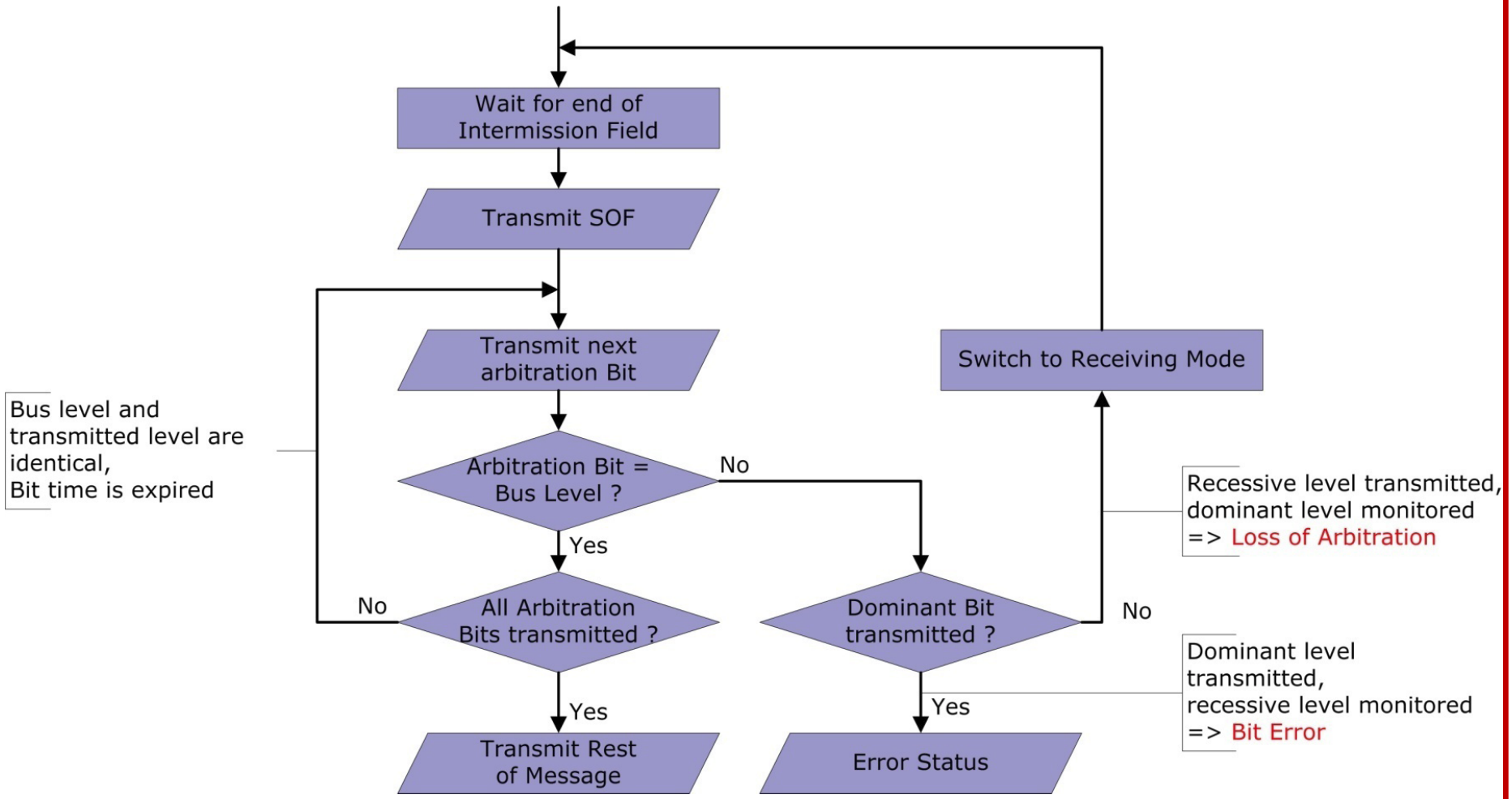
  Lower value = More important

- As a node transmits each bit, it verifies that it sees the same bit value on the bus that it transmitted.
- A “0” on the bus wins over a “1” on the bus.
- Losing node stops transmitting, winner continues.



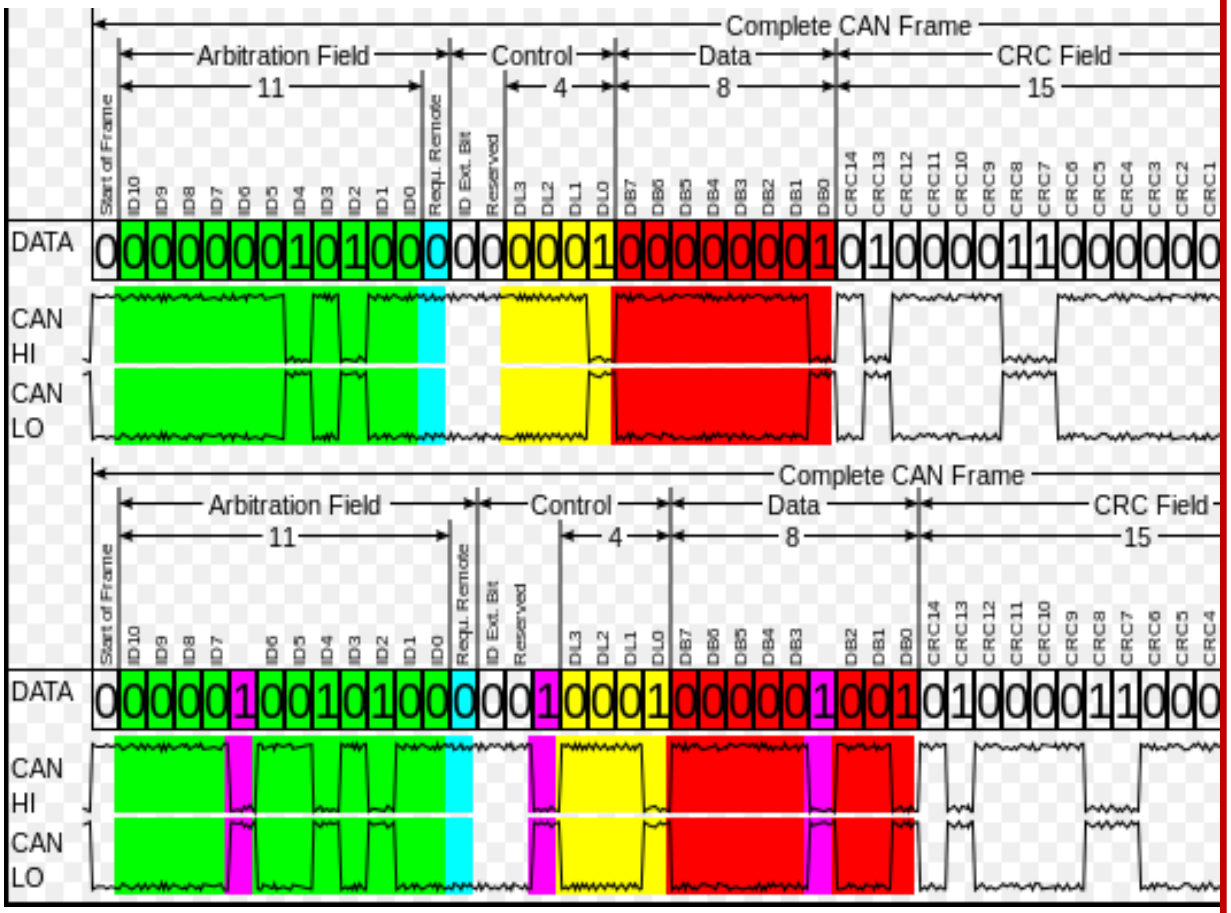
- In a CAN message frame, the RTR bit is located within the control field of the frame.
- A dominant (0) RTR bit indicates a data frame.
- A recessive (1) RTR bit indicates a remote frame.
- Suppose Node A wants to request data from Node B. Node A will send a message with the RTR bit set to recessive (1) to indicate that it is a remote frame requesting specific data.
- Node B, upon receiving the remote frame, will recognize the request and respond accordingly by sending the requested data in a separate data frame.

Bus Arbitration Flowchart



Bit Stuffing

- 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: a Large Field of Applications

- Building Automation:
  - ✓ Heating control, Air Conditioning, Security, Access & Light Control…
- Domestic & Food distribution appliances
  - ✓ Washing machines, dishes cleaner, self-service bottle distributors…
- Automotive & Transportation
  - ✓ Dash Board, Power train & Car Body
  - ✓ Train, bus and truck equipment…
- Robotic
- Production Automation:
  - ✓ Control & link of production machines…
- Medical
- Agriculture
  - ✓ Harvester, seeding, sowing machines, tractor control…

Useful Links

- Manufacturer and Product List
  - http://www.can-cia.org/products/can/chips/
- CAN Information
  - http://www.canbus.us/
  - http://www.can-cia.org/can/

Thank You