Controller Area Network

Mathis Schmieder (316110)

TU Berlin
Department of Telecommunication Systems
Telecommunication Networks Group

May 22nd, 2015

Table of Contents

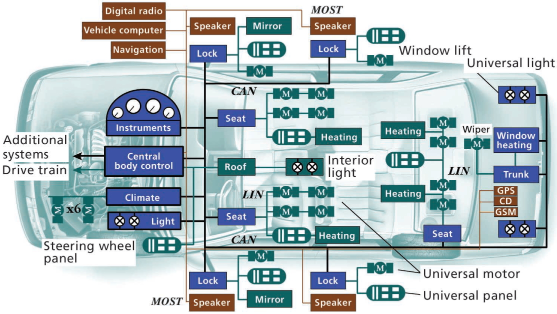
1 Introduction

2 Physical Layer

3 Data Link Layer

4 Application Layer

In Vehicle Networks



CAN	Controller area network
GPS	Global Positioning System
GSM	Global System for Mobile Communications
LIN	Local interconnect network
MOST	Media-oriented systems transport

Figure: Vehicular Network Architecture

Timeline

1983	Development started at Robert Bosch GmbH
1986	Official release at SAE congress
1987	First chips by Intel and Philips
1991	Bosch publishes CAN 2.0 for standardization
1993	Approved and published as ISO 11989
1995	Standard split into 11989-1,2,3
2012	CAN FD 1.0 released by Bosch

CAN protocol basics

- Serial Bus System
- Only physical and data link layer specified
- Physical Layer
  - Manages effective transmission of data
  - Bit timing & Synchronization
  - High & Low Speed standards exist
- Data Link Layer
  - Medium Access Control: CSMA/**CR**
  - Logical Link Control: Interface to higher layers
- Numerous application layers rely on CAN

Physical Layer

- Physical signalling specified in ISO 11898-1
- Medium access units (transceivers) described in two documents
    - ISO 11898-2: High-speed CAN
    - ISO 11898-3: Low-speed fault-tolerant CAN
- Connectors are not standardised by CAN specification

Network Topology

- CAN networks use shared bus topology
- Buses have to be terminated
- Topology should be as linear as possible
- Maximum bitrate and length directly dependent
- No effective way to share line for power and signalling

Transmission Medium

Several kinds of transmission media can be used:

- **Two-wire bus**: Enables differential signal transmission, ensures reliable communication. Requirement for high-speed CAN.
- **Single-wire bus**: Simpler/cheaper alternative, fall back in case of fault.
- **Optical transmission medium**: Ensures immunity to electromagnetic noise, used to interconnect different subnets.

ISO 11898-2: High-speed CAN

- Maximum bit rate 1 Mbps
- Linear bus end-terminated with 120Ω
- Stubs must be shorter than 30 cm

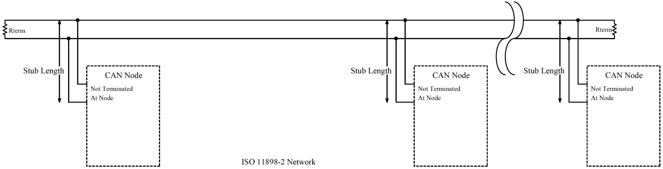


Figure: High Speed CAN Network

ISO 11898-3: Low-speed fault-tolerant CAN

- Maximum bit rate 125 kbps
- Linear or star bus terminated at node with about 100Ω

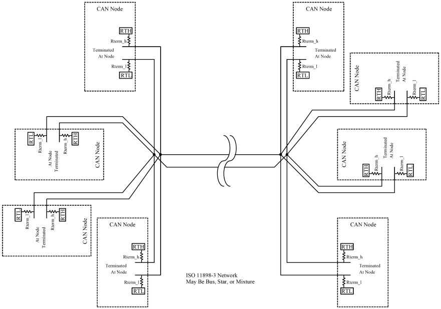


Figure: Low Speed CAN Network

Bit Encoding

- Level on bus can assume two complementary values:
  - *dominant*, usually corresponds to logical value 0
  - *recessive*, usually corresponds to logical value 1
- CAN relies on *non-return to zero* (NRZ) bit encoding

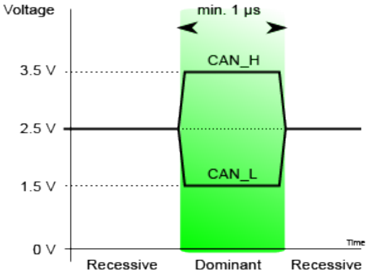


Figure: Levels on the CAN bus

Synchronization

- Timing information extracted from bit stream
- Edges of the signal are used for synchronization
- Bit stuffing to ensure sufficient number of edges

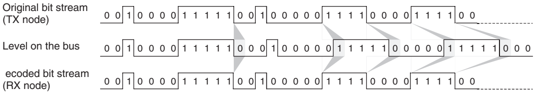


Figure: Bit stuffing technique

Frame Format

- Specifications define standard and extended frame format
  - Standard: 11 bit identifier
  - Extended: 29 bit identifier
- Standard frame format mostly used
- Protocol foresees four kinds of frames: *data, remote, error* and *overload*

Telecommunication Networks Group

Data Frames

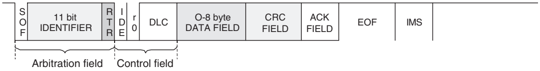


Figure: Format of standard data frames

- Dominant *start of frame* (SOF) bit
- Arbitration field: identifier and *remote transmission request* bit
- *Data length code* (DLC): Length of data field encoded in 4 bits
- *Cyclic redundancy check* (CRC) encoded in 15 bits
- ACK slot: Recessive at transmitter, dominant at receiver
- *End of frame* (EOF) slot: Seven recessive bits

Remote Frames

- Generally, source sends out data autonomously
- Protocol allows to poll for data
- Remote format similar to data format
- RTR field is recessive, data frames have higher priority
- Remote frames carry no data

Error Frames

- Notify nodes that an error has occured
- Consist of two fields:
  - Error flag: Six dominant/recessive bits.
    → Violates bit stuffing rules, error condition is detected
  - Error delimiter: 8 recessive bits.
- Active flag: dominant, transmitted by node in state *error active*
- Passive flag: recessive, transmitted by node in state *error passive*

Fault Confinement

- Supervises correct operation of MAC sublayer
- Disconnect defective node from bus
- Uses two counters: *transmission* and *receive error count*
  - On error detect, counter is increased by a given amount
  - On success, counter is decreased by one
  - Increase amount of detecting node is higher than relying nodes
- When counter exceeds 127, node switches from error active to error passive
- When counter exceeds 255, node switches to bus off

Overload Frames

- Used to slow down operations on the bus by adding delays
- Format similar to the error frames
- Hardly ever used because today's CAN controllers are very fast

Access Technique

- CAN relies on CSMA for access control:
  - When no data is exchanged, level on the bus is recessive
  - Before transmission, nodes observe the state of the network
  - When network is idle, transmission starts immediately
- Collisions are improbable but not impossible
- CAN introduces collision resolution scheme: Bus arbitration

CSMA/CR: Bus Arbitration

Bus arbitration essentially finds out the most urgent frame.

- Level on bus is dominant if one node is sending dominant bit
- Nodes can reliably check level on bus
- On transmission, each node compares level on bus against written value
- If node transmits recessive bit but reads dominant, it backs off

CSMA/CR: Bus Arbitration

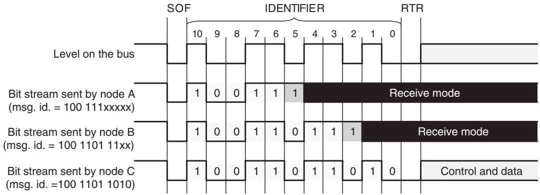

SOF
IDENTIFIER
RTR
10
9
8
7
6
5
4
3
2
1
0
Level on the bus
Bit stream sent by node A
(msg. id. = 100 111xxxxx)
1 0 0 1 1 1
Receive mode
Bit stream sent by node B
(msg. id. = 100 1101 11xx)
1 0 0 1 1 0 1 1 1
Receive mode
Bit stream sent by node C
(msg. id. =100 1101 1010)
1 0 0 1 1 0 1 1 0 1 0
Control and data

Figure: Arbitration phase in CAN

- Nodes transmit message identifier starting with MSB
- Lowest identifier corresponds to highest priority
- Message with highest priority wins contention

Error Management

- Fundamental requirement for CAN is robustness
- Specifications foresee five mechanisms for error detection:
  - 15 bit wide CRC: Discover up to five erroneous bits
  - Frame check: CRC, ACK, EOF delimiters have to be recessive
  - Acknowledgement check: Transmitter checks for set ACK bit
  - Bit monitoring: Transmitter checks level on bus against written value
  - Bit stuffing: Each node verifies if bit stuffing rules have been violated
- Residual probability for undetected corrupt message is $4.7 \cdot 10^{-11}$ times the frame error rate or less

Thanks for your attention!

Questions? Ideas? Suggestions?

Telecommunication
Networks Group