CAN - Controller Area Network

CAN – An Overview

EE-Meeting, 21.11.2005

H.Brand@gsi.de

References:

- http://de.wikipedia.org/wiki/Controller_Area_Network
- http://en.wikipedia.org/wiki/Controller_Area_Network
- http://www.kvaser.com/can/protocol/index.htm
- http://www.can-cia.org/canopen/

GSI

CAN - Controller Area Network

- multicast shared serial bus standard
- Originally developed in the 1980s by Robert Bosch GmbH, for connecting electronic control units (ECUs)
- Specifically designed to be robust in noisy environments and can utilize a differential balanced line like RS-485. It can be even more robust against noise if twisted pair wire is used.
- Initially created for automotive purposes
  - Also for embedded control applications (e.g., industrial) that may be subject to noise
- Bit rates up to 1 Mbit/s below 40 m.
- CAN data link layer protocol is standardized in ISO 11898-1 (2003)
  - logical link control (LLC) sublayer and the Media Access Control (MAC) sublayer

CAN – Tranfer Methode I

- CSMA/CA
  Carrier **S**ense **M**ultiple **A**ccess / **C**ollision **A**voidance
  - Nicht zu verwechseln mit CSMA/CD wie beim Ethernet
  - Dabei werden Kollisionen beim Buszugriff durch die Arbitrierung oder Bit-Arbitrierung vermieden.
  - Die Daten sind NRZ-L codiert. Datensicherung durch **CRC-Verfahren**
  - Synchronisierung der Busteilnehmer durch **Bit-Stuffing**
    - In CAN frames a bit of opposite polarity is inserted after five consecutive bits of the same polarity
    - (111111 or 000000) are considered an error
- Kupferleitungen oder Glasfaser.

CAN - Tranfer Methode II

- Im Falle von Kupferleitungen arbeitet der CAN-Bus mit Differenzsignalen.
  - Er wird normalerweise mit 3 Leitungen ausgeführt: CAN_HIGH, CAN_LOW und CAN_GND (Masse). CAN_LOW enthält den komplementären Pegel von CAN_HIGH gegen Masse. Dadurch können Gleichtaktstörungen unterdrückt werden, da ja die Differenz gleich bleibt.
  - Recommendation: ISO11898-2 (High-Speed Medium Access Unit) Twisted-Pair-Kabel with **120** (108...132) Ohm impedance.
- Die Übertragung der Daten erfolgt so, dass ein Bit, je nach Zustand, entweder **dominant** oder **rezessiv** auf den Busleitungen wirkt. Ein dominantes überschreibt dabei ein rezessives.

CAN - Data Transmission

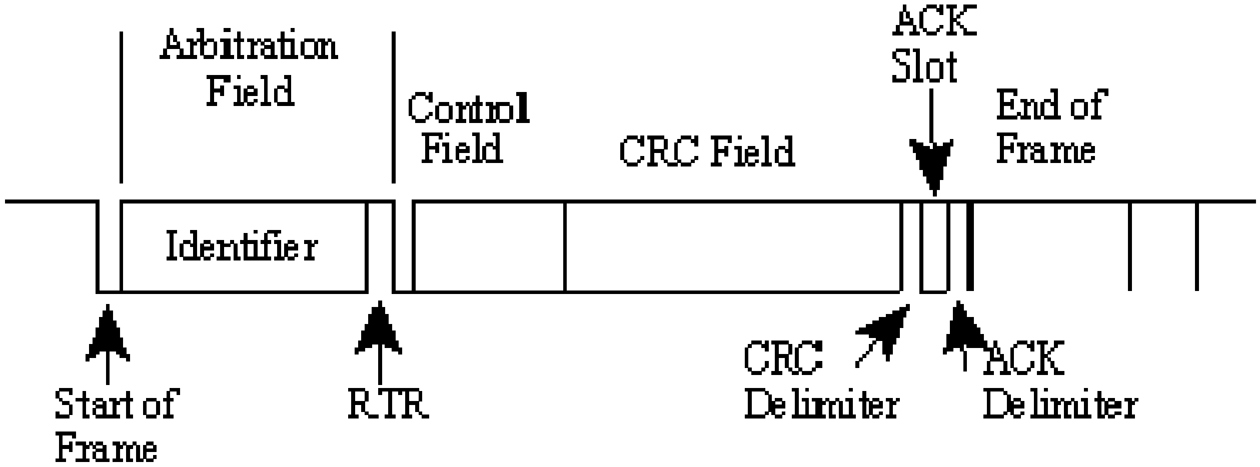
- CAN transmits data through a binary model of
  - "dominant" bits, logical 0 and
  - "recessive" bits, logical 1.
- If one node transmits a dominant bit and another node transmits a recessive bit then the dominant bit "wins"
  - a logical AND between the two).
- A dominant bit is asserted by creating a voltage across the wires while a recessive bit is simply not asserted on the bus.
  - If anyone sets a voltage difference, everyone sees it, hence, dominant.
- Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA)
  - If two or more devices start transmitting at the same time, there is a priority based arbitration scheme to decide which one will be granted permission to continue transmitting.

CAN – Frames I

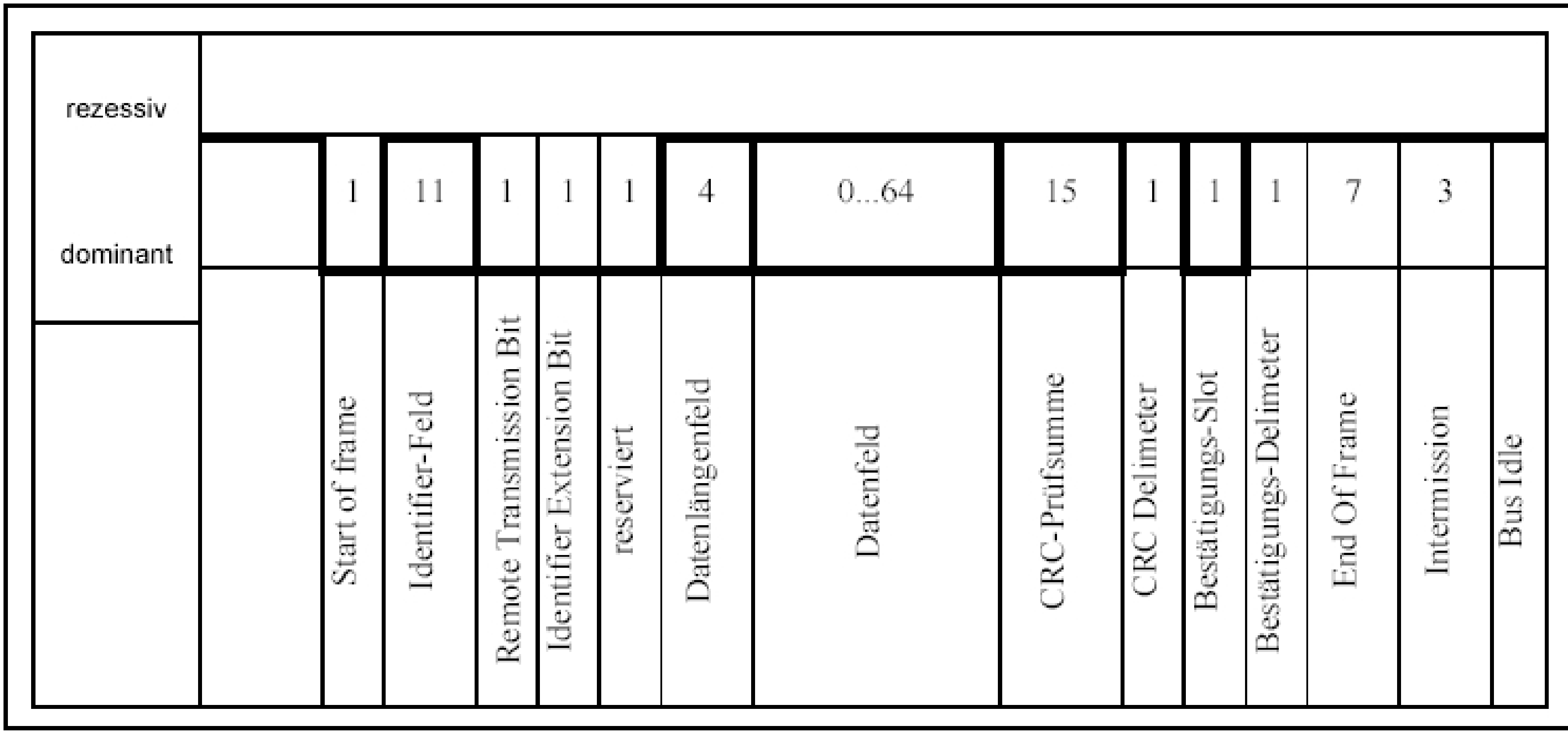
- Data frame
  The data frame is the only frame for actual data transmission. There are two message formats:
    - Base frame format: with 11 identifier bits
    - Extended frame format: with 29 identifier bits
    - The CAN standard requires the implementation *must* accept the base frame format and *may* accept the extended frame format. But *must* tolerate the extended frame format.
    - One restriction placed on the identifier is that the first 7 bits cannot be all recessive bits.
- Remote Frame
- Error Frame
- Overload Frame

CAN – Data Frame

- Object Identifier
  - –Identifies message content, **not** a device
  - –Receiver decides whether a data frame is relevant for him or not.
  - –In principle one device can send and receive an arbitrary number of identifier, but
  - –Unique sender requiered for each identifier in a system
  - –Priority



CAN – Data Frame Details

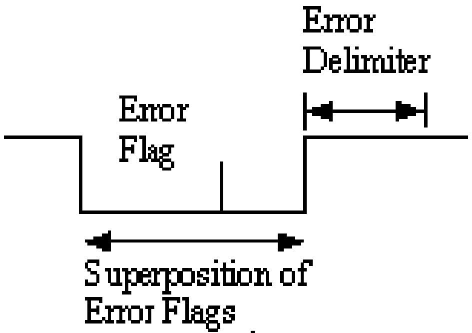


CAN – Frames II

- Remote Frame

  The remote frame is identical to the data frame except:

  - –RTR bit set to recessive
  - –data length contains the number of bytes that are required from the data frame
  - –The response is a seperate frame.
    - •Data is not inserted in the current remote frame



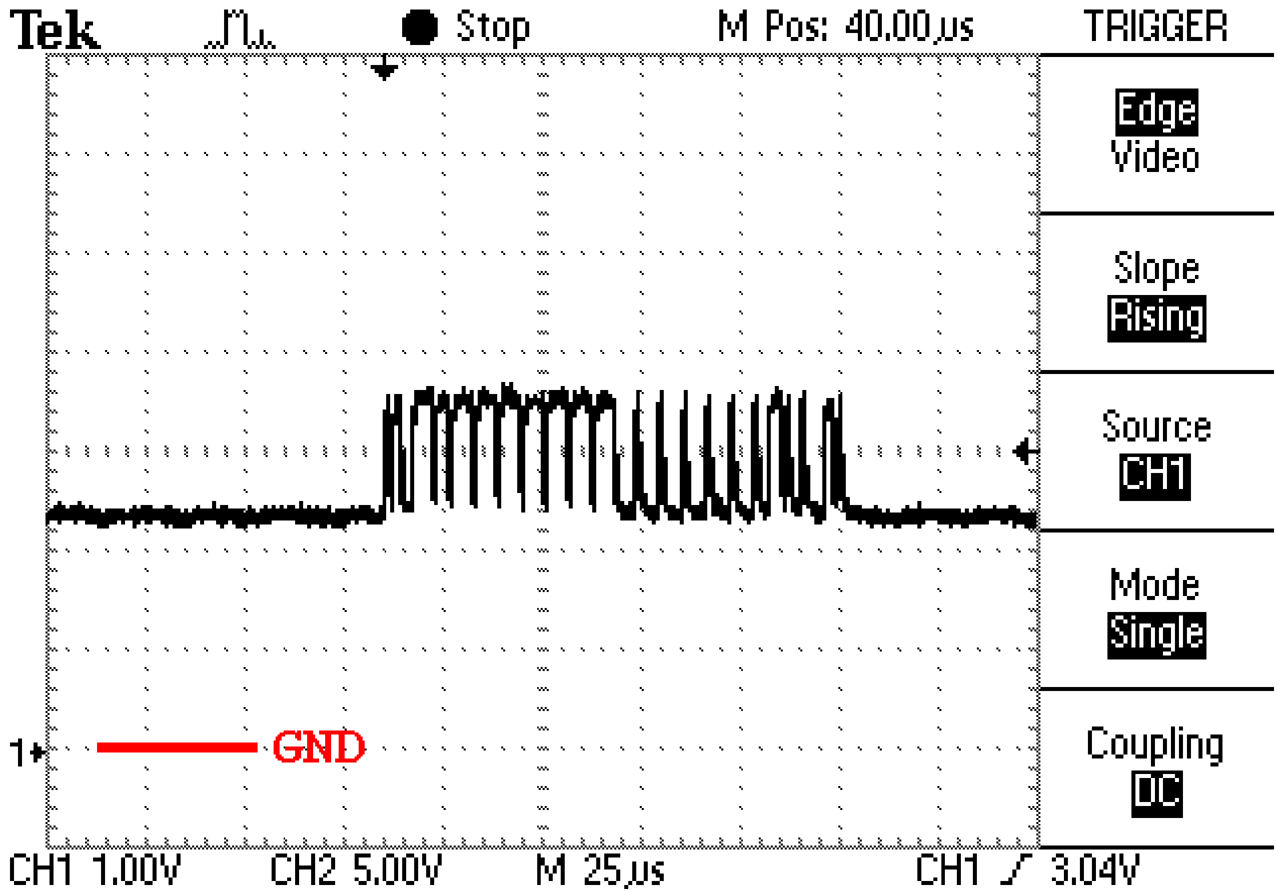
- Error Frame

  There are two types of error frames

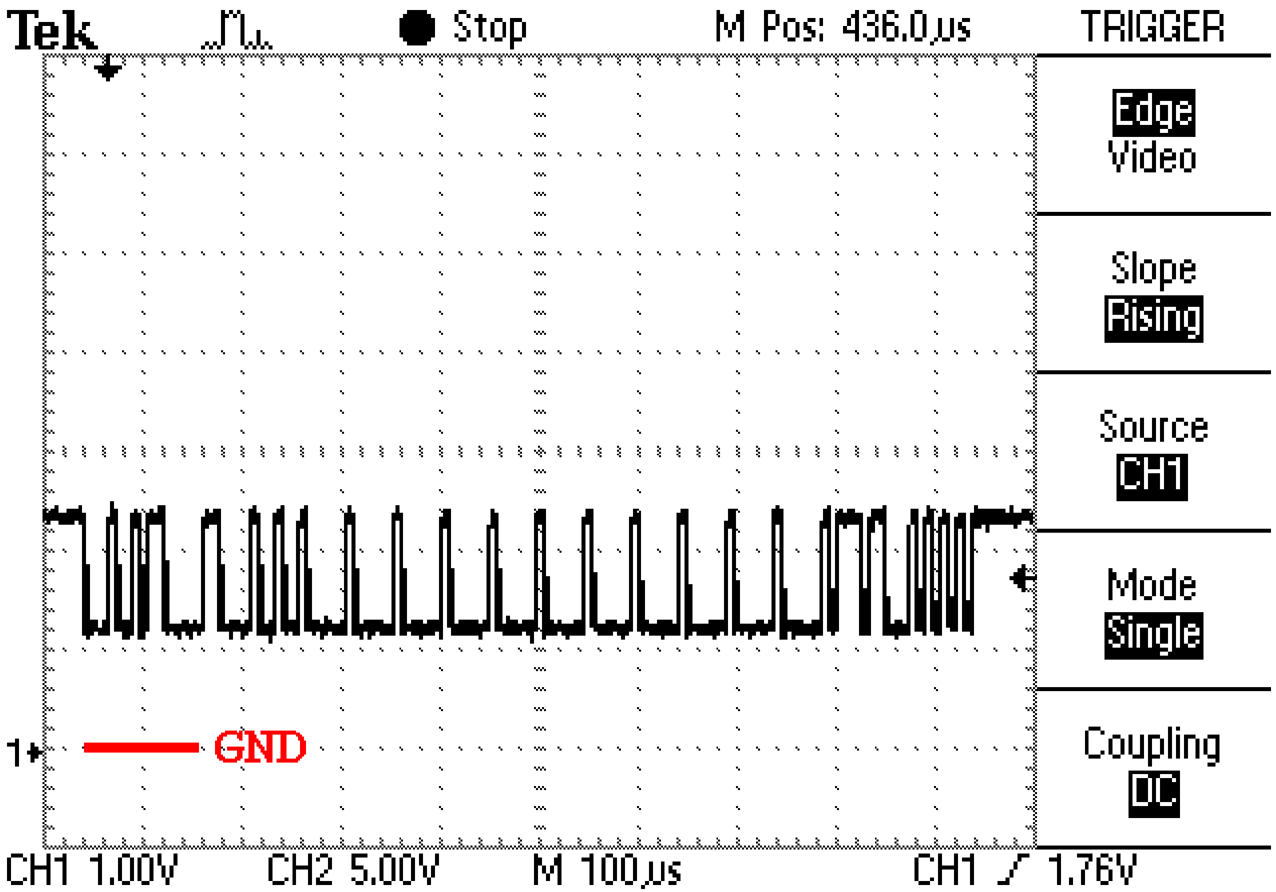
  - –Active Error Frame: Transmitted by a node detecting an error on the network that is in error state "error active".
  - –Passive Error Frame: Transmitted by a node detecting an active error frame on the network that is in error state "error passive".
  - –6 bits of the same value (thus violating the bit-stuffing rule) and an Error Delimiter, which is 8 recessive bits

•Overload Frame (not used any more, all hardware is depricated)

CAN – Timing 1 MBit/s



CAN – Timing 125 kBit/s

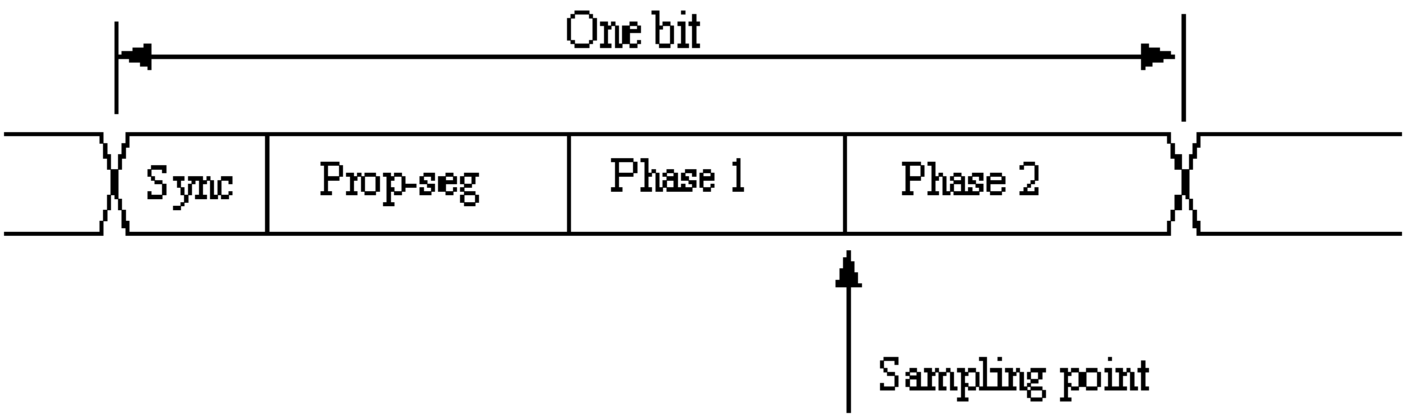


CAN – Bit Timing

- **The Layout of a Bit**

Each bit on the CAN bus is, for timing purposes, divided into at least 4 *quanta*. The quanta are logically divided into four groups or *segments*

  - –Synchronization Segment
  - –Propagation Segment
  - –Phase Segment 1
  - –Phase Segment 2



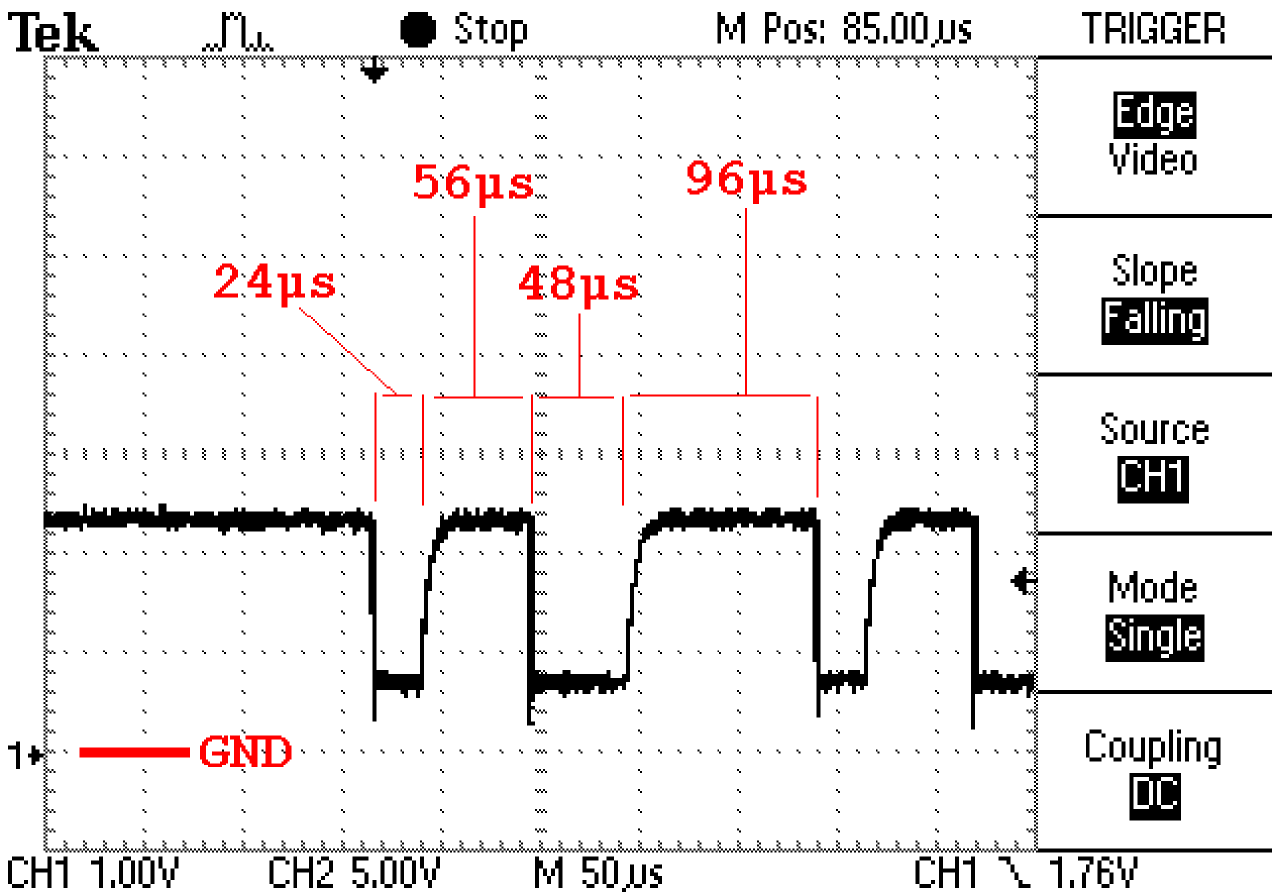
CAN – Error Handling I

- Transmit Error Counter
- Receive Error Counter.
- Several rules how these counters are incremented and/or decremented
- In essence, a transmitter detecting a fault increments its Transmit Error Counter faster than the listening nodes will increment their Receive Error Counter.
- Error Detection Mechanisms
  - Bit Monitoring.
  - Bit Stuffing.
  - Frame Check.
  - Acknowledgement Check.
  - Cyclic Redundancy Check.

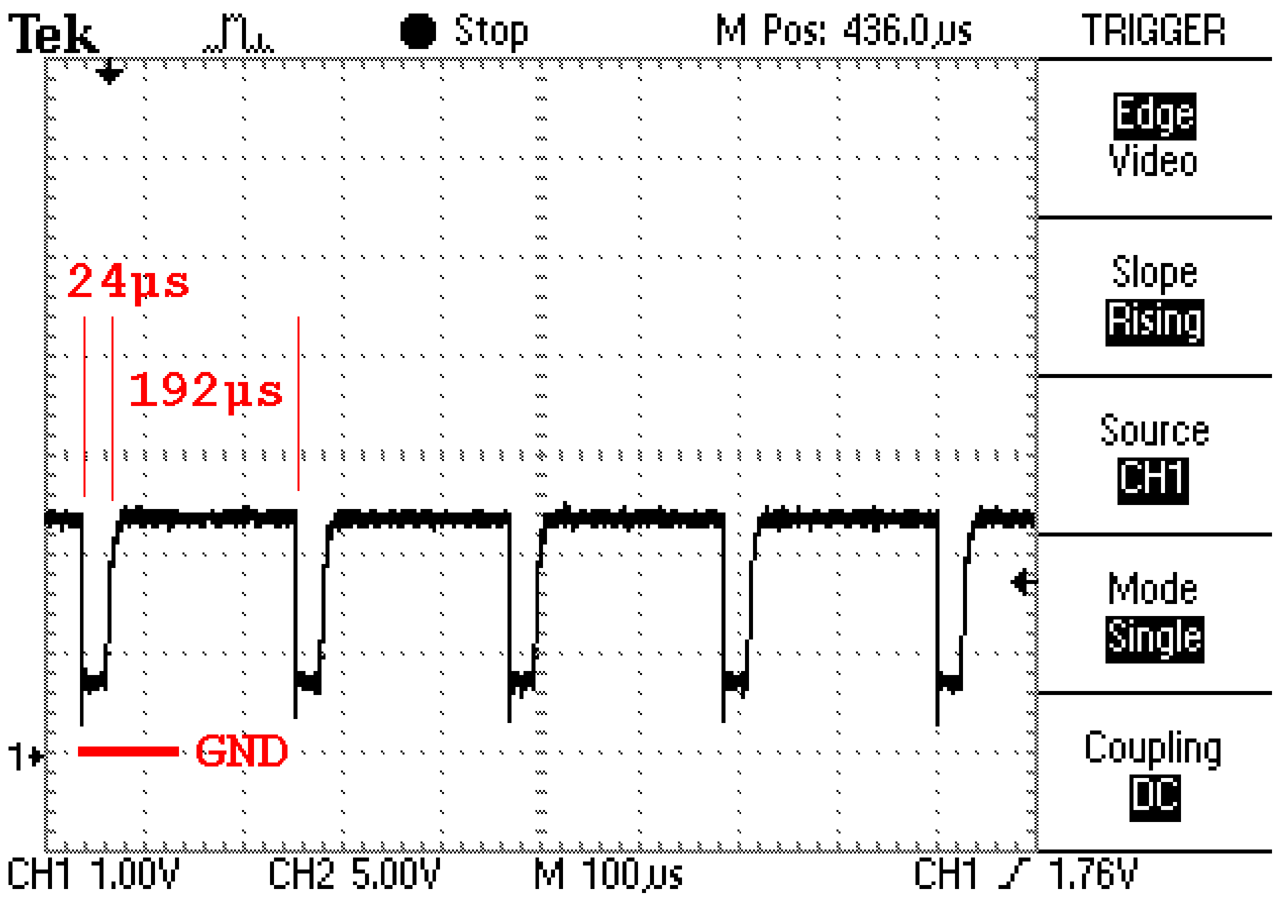
CAN – Error Handling 2

- A node starts out in ***Error Active*** mode.
  When any one of the two Error Counters raises above 127, the node will enter a state known as ***Error Passive*** and when the Transmit Error Counter raises above 255, the node will enter the ***Bus Off*** state.
- An ***Error Active*** node will transmit *Active Error Flags* when it detects errors.
- An ***Error Passive*** node will transmit *Passive Error Flags* when it detects errors.
- A node which is ***Bus Off*** will not transmit *anything* on the bus at all.
- The rules for increasing and decreasing the error counters are somewhat complex, but the principle is simple: transmit errors give 8 error points, and receive errors give 1 error point. Correctly transmitted and/or received messages causes the counter(s) to decrease.

CAN – No Bus Termination Error Active



CAN – No Bus Termination Error Passive



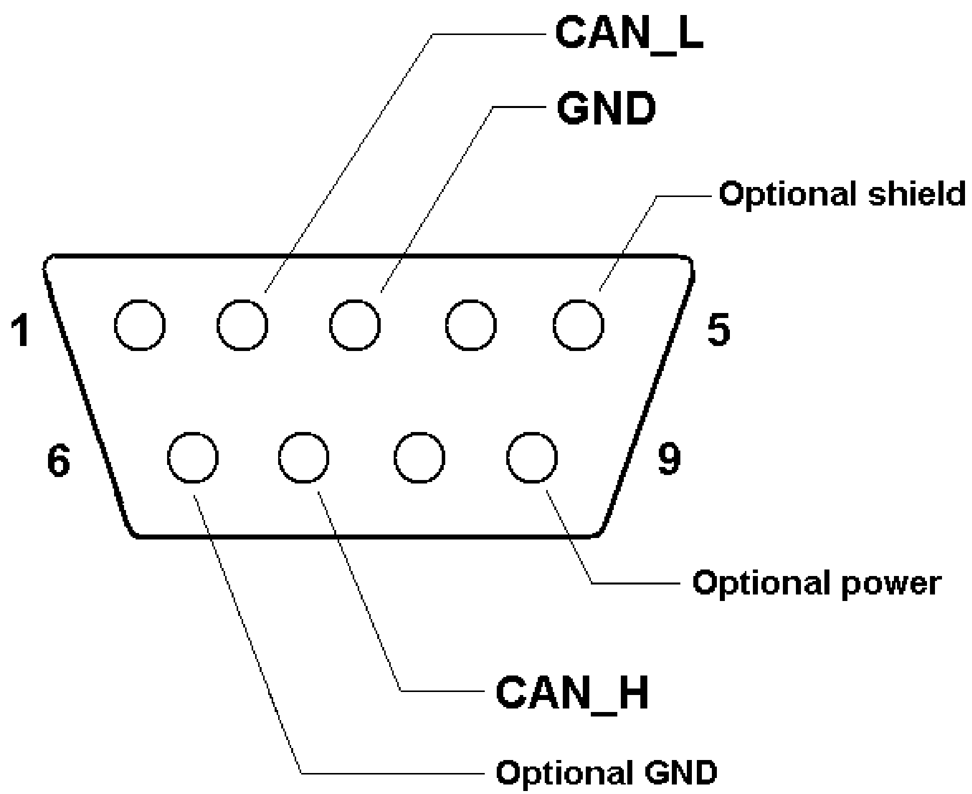
CAN – Bus Failure Modes I

- The ISO 11898 standard enumerates several failure modes of the CAN bus cable:
  1. CAN_H interrupted
  2. CAN_L interrupted
  3. CAN_H shorted to battery voltage
  4. CAN_L shorted to ground
  5. CAN_H shorted to ground
  6. CAN_L shorted to battery voltage
  7. CAN_L shorted to CAN_H wire
  8. CAN_H and CAN_L interrupted at the same location
  9. Loss of connection to termination network
- For failures 1-6 and 9, it is "recommended" that the bus survives with a reduced S/N ratio, and in case of failure 8, that the resulting subsystem survives. For failure 7, it is "optional" to survive with a reduced S/N ratio.

CAN – Bus Failure Modes II

- In practice, a CAN system using 82C250-type transceivers will not survive failures 1-7, and may or may not survive failures 8-9.
- There are "fault-tolerant" drivers, like the TJA1053, that can handle all failures though. Normally you pay for this fault tolerance with a restricted maximum speed; for the TJA1053 it is 125 kbit/s.

CAN - Connector



This is a male connector viewed from the connector side, or a female connector viewed from the soldering side.

CAN – ISO Standards

- There are several CAN physical layer standards:
  - **ISO 11898-2**: CAN high-speed
  - **ISO 11898-3**: CAN fault-tolerant (low-speed)
  - **ISO 11992-1**: CAN fault-tolerant for truck/trailer communication
  - **SAE J2411**: Single-wire CAN (SWC)
- **ISO 11898-2** uses a two-wire balanced signaling scheme. It is the most used physical layer in car powertrain applications and industrial control networks.
- The ISO 11898-4 standard defines the time-triggered communication on CAN (TTCAN). It is based on the CAN data link layer protocol providing a system clock for the scheduling of messages.

CAN – Application Layer

- As the CAN standard does not include tasks of application layer protocols
  - flow control,
  - device addressing,
  - transportation of data blocks larger than one message
- Many implementations of higher layer protocols were created.
  - **DeviceNet**
  - **CANopen**
  - **SDS**
  - **J1939**
  - **CAN Kingdom**

Performance Problems

Basic CAN vs. Full CAN

- The terms "Basic CAN" and "Full CAN" originate from the childhood of CAN. Once upon a time there was the Intel 82526 CAN controller which provided a DPRAM-style interface to the programmer.
- Then came along Philips with the 82C200 which used a FIFO- (queue-) oriented programming model and limited filtering abilities.
- To distinguish between the two programming models, people for some reason termed the Intel way as "Full CAN" and the Philips way as "Basic CAN".
- Today, most CAN controllers allow for both programming models, so there is no reason to use the terms "Full CAN" and "Basic CAN" - in fact, these terms can cause confusion and should be avoided. Of course, a "Full CAN" controller can communicate with a "Basic CAN" controller and vice versa. There are no compatibility problems.