esd electronics, inc.

HARDWARE • SOFTWARE • ENGINEERING

Controller Area Network

CAN

A Serial Bus System - Not Just For Vehicles

The need for serial communication in vehicles

Many vehicles already have a large number of electronic control systems. The growth of automotive electronics is the result partly of the customer‘s wish for better safety and greater comfort and partly of the government‘s requirements for improved emission control and reduced fuel consumption. Control devices that meet these requirements have been in use for some time in the area of engine timing, gearbox and carburetor throttle control and in anti-block systems (ABS) and acceleration skid control (ASC).

The complexity of the functions implemented in these systems necessitates an exchange of data between them. With conventional systems, data is exchanged by means of dedicated signal lines, but this is becoming increasingly difficult and expensive as control functions become ever more complex. In the case of complex control systems (such as Motronic) in particular, the number of connections cannot be increased much further.

Moreover, a number of systems are being developed which implement functions covering more than one control device. For instance, ASC requires the interplay of engine timing and carburetor control in order to reduce torque when drive wheel slippage occurs. Another example of functions spanning more than one control unit is electronic gearbox control, where ease of gear changing can be improved by a brief adjustment to ignition timing.

If we also consider future developments aimed at overall vehicle optimization, it becomes necessary to overcome the limitations of conventional control device linkage. This can only be done by networking the system components using a serial data bus system. Bosch developed a system for this purpose, the “Controller Area Network” (CAN), which has since been standardized internationally (ISO 11898) and has been “cast in silicon” by several semiconductor manufacturers.

Using CAN, peer stations (controllers, sensors and actuators) are connected via a serial bus. The bus itself is a symmetric or asymmetric two wire circuit, which can be either screened or unscreened. The electrical parameters of the physical transmission are also specified in ISO 11898. Suitable bus driver chips are available from a number of manufacturers.

The CAN protocol, which corresponds to the data link layer in the ISO/OSI reference model, meets the real-time requirements of automotive applications. Unlike cable trees, the network protocol detects and corrects transmission errors caused by electromagnetic interference. Additional advantages of such a network are the easy configurability of the overall system and the possibility of central diagnosis.

The purpose of using CAN in vehicles is to enable any station to communicate with any other without putting too great a load on the controller computer.

Use of the CAN network in vehicles

There are four main applications for serial communication in vehicles, each having different requirements and objectives.

- Networking controllers for engine timing, transmission, chassis and brakes. The data rates are in the range - typical of real-time systems of 200 kbit/s to 1 Mbit/s.
- Networking components of chassis electronics and electronics which make the vehicle more comfortable. Examples of such multiplex applications are lighting control, air-conditioning, and central locking and seat and mirror adjustment. Particular importance has to be attached here to the cost of the components and wiring requirements. Typical data rates are around 50 kbit/s.
- In the near future, serial communication will also be used in the field of mobile communication in order to link components such as car radios, car telephones, navigation aids etc. to a central, ergonomically designed control panel. The functions defined in the

Prometheus project, such as vehicle-to-vehicle and vehicle-to-infrastructure communication will depend to a large extent on serial communication.

- At present, CAN is used for the first three applications, but for diagnosis the preferred solution is an interface according to ISO 9141.

Industrial applications of the CAN network

A comparison of the requirements for vehicle bus systems and industrial field bus systems shows amazing similarities: low cost, operability in a harsh electrical environment, high real-time capabilities and ease of use are equally desirable in both sectors.

The standard use of CAN in Mercedes-Benz‘s ”S” Class and the adoption of CAN by US commercial vehicle manufacturers for fast transmissions (up to 1 Mbit/s) has made industrial users prick up their ears. Not only manufacturers of mobile and stationary agricultural and nautical machinery and equipment have chosen to use CAN, it has also been the choice of manufacturers of medical apparatus, textile machines, and special-purpose machinery and elevator controls. The serial bus system is particularly well suited to networking”intelligent” I/O devices as well as sensors and actuators within a machine or plant.

The textile machinery industry is one of the pioneers of CAN. One manufacturer equipped his looms with modular control systems communicating in real time via CAN networks as early as 1990. In the meantime several textile machinery manufacturers have joined together to form the ”CAN Textile Users Group”, which in turn is a member of the international users and manufacturers group ”CAN in Automation”. Similar requirements to those of the textile machinery are to be found in packaging machinery and machinery for paper manufacture and processing.

In the USA a number of enterprises are using CAN in production lines and machine tools as an internal bus system for networking sensors and actuators within the line or machine. Some users, such as the medical engineering sector, decided in favour of CAN because they had particularly stringent safety requirements. Similar problems are faced by other manufacturers of machinery and equipment with particular requirements with respect to safety (e.g. robots and transport systems).

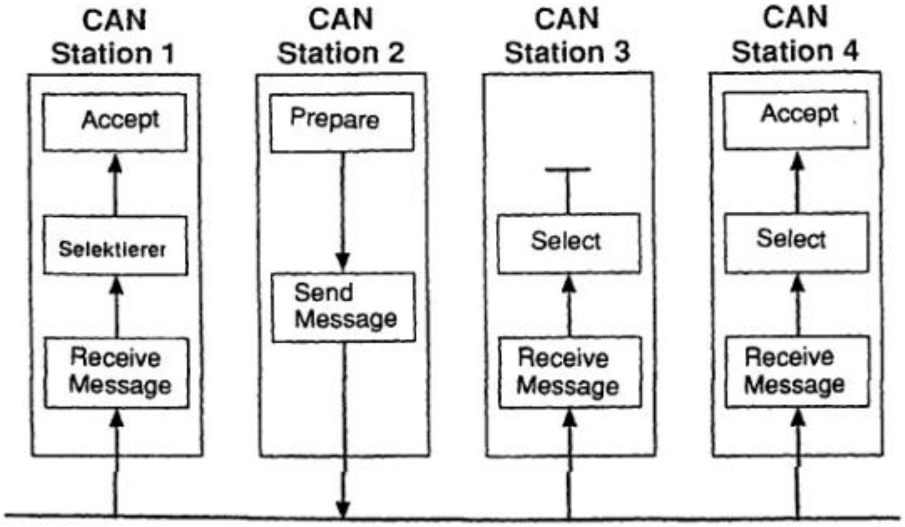
Apart from the high transmission reliability, the low connection costs per station are a further decisive argument for CAN. In applications where price is critical it is of essential importance that CAN chips be available from a variety of manufacturers. The compactness of other controller chips is also an important argument, for instance in the field of low-voltage switchgear.

How the CAN network functions

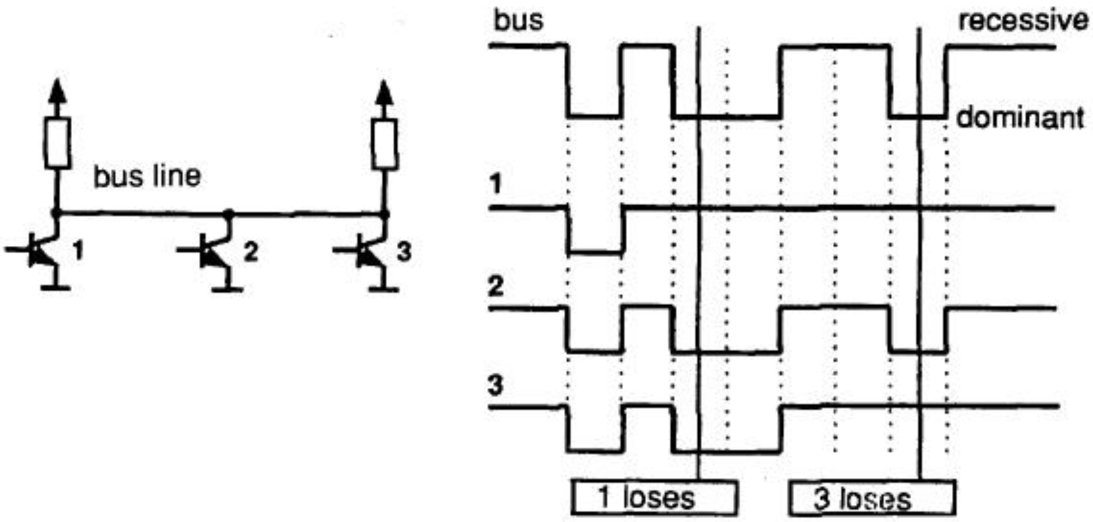
Principles of data exchange:

When data are transmitted by CAN, no stations are addressed, but instead, the content of the message (e.g. rpm or engine temperature) is designated by an identifier that is unique throughout the network. The identifier defines not only the content but also the priority of the message. This is important for bus allocation when several stations are competing for bus access.

If the CPU of a given station wishes to send a message to one or more stations, it passes the data to be transmitted and their identifiers to the assigned CAN chip (”Make ready”). This is all the CPU has to do to initiate data exchange. The message is constructed and transmitted by the CAN chip. As soon as the CAN chip receives the bus allocation (”Send Message”) all other stations on the CAN network become receivers of this message (”Receive Message”). Each station in the CAN network, having received the message correctly, performs an acceptance test to determine whether the data received are relevant for that station (”Select”). If the data are of significance for the station concerned they are processed (”Accept”), otherwise they are ignored. A high degree of system and configuration flexibility is achieved as a result of the content-oriented addressing scheme. It is very easy to add stations to the existing CAN network without making any hardware or software modifications to the existing stations, provided that the new stations are purely receivers. Because the data transmission protocol does not require physical destination addresses for the individual components, it supports the concept of modular electronics and also permits multiple reception (broadcast, multicast) and the synchronization of distributed processes: measurements needed as information by several controllers can be transmitted via the network, in such a way that it is unnecessary for each controller to have its own sensor.



Broadcast transmission and acceptance filtering by CAN nodes



Principle of non-destructive bitwise arbitration

Non-destructive bitwise arbitration:

For the data to be processed in real time they must be transmitted rapidly. This not only requires a physical data transfer path with up to 1 Mbit/s but also calls for rapid bus allocation when several stations wish to send messages simultaneously.

In real-time processing the urgency of messages to be exchanged over the network can differ greatly: a rapidly changing dimension (e.g. engine load) has to be transmitted more frequently and therefore with fewer delays than other dimensions (e.g. engine temperature) which change relatively slowly. The priority at which a message is transmitted compared with another less urgent message is specified by the identifier of the message concerned. The priorities are laid down during system design in the form of corresponding binary values and cannot be changed dynamically. The identifier with the lowest binary number has the highest priority.

Bus access conflicts are resolved by bitwise arbitration on the identifiers involved by each station observing the bus level bit for bit. In accordance with the “wired and” mechanism, by which the dominant state (logical 0) overwrites the recessive state (logical 1), the competition for bus allocation is lost by all those stations with recessive transmission and dominant observation. All "losers” automatically become receivers of the message with the highest priority and do not reattempt transmission until the bus is available again.

Efficiency of bus allocation:

The efficiency of the bus allocation system is determined mainly by the possible application for a serial bus system. In order to judge as simply as possibly which bus systems are suitable for which applications the literature includes a method of classifying bus allocation procedures. Generally we distinguish between the following classes:

- *Allocation on a fixed time schedule.*
  Allocation is made sequentially to each participant for a maximum duration regardless of whether this participant needs the bus at this moment or not (examples: token slot or token passing).
- *Bus allocation on the basis of need.*
  The bus is allocated to one participant on the basis of transmission requests outstanding, i.e. the allocation system only considers participants wishing to transmit (examples: CSMA, CSMA/CD, flying master, round robin or bitwise arbitration). For CAN, bus allocation is negotiated purely among the messages waiting to be transmitted. This means that the procedure specified by CAN is classified as allocation on the basis of need.

Another means of assessing the efficiency of bus arbitration systems is the bus access method:

- *Non-destructive bus access.*
  With methods of this type the bus is allocated to one and only one station either immediately or within a specified time following a single bus access (by one or more stations). This ensures that each bus access by one or more stations leads to an unambiguous bus allocation (examples: token slot, token passing, round robin, bitwise arbitration)
- *Destructive bus allocation.*
  Simultaneous bus access by more than one station causes all transmission attempts to be aborted and therefore there is no successful bus allocation. More than one bus access may be necessary in order to allocate the bus at all, the number of attempts before bus allocation is successful being a purely statistical quantity (examples: CSMA/CD, Ethernet).

In order to process all transmission requests of a CAN network while complying with latency constraints at as low a data transfer rate as possible, the CAN protocol must implement a bus allocation method that guarantees that there is always unambiguous bus allocation even when there are simultaneous bus accesses from different stations.

The method of bitwise arbitration using the identifier of the messages to be transmitted uniquely resolves any collision between a number of stations wanting to transmit, and it does this at the latest within 13 (standard format) or 33 (extended format) bit periods for any bus access period. Unlike the message-wise arbitration employed by the CSMA/CD method this nondestructive method of conflict resolution ensures that no bus capacity is used without transmitting useful information.

Even in situations where the bus is overloaded the linkage of the bus access priority to the content of the message proves to be a beneficial system attribute compared with existing CSMA/CD or token protocols: in spite of the insufficient bus transport capacity, all outstanding transmission requests are processed in order of their importance to the overall system (as determined by the message priority).

The available transmission capacity is utilized efficiently for the transmission of useful data since "gaps" in bus allocation are kept very small. The collapse of the whole transmission system due to overload, as can occur with the CSMA/CD protocol, is not possible with CAN. Thus, CAN permits implementation of fast, traffic-dependent bus access which is non-destructive because of bitwise arbitration based on the message priority employed.

Non-destructive bus access can be further classified into:

- Centralized bus access control and
- Decentralized bus access control

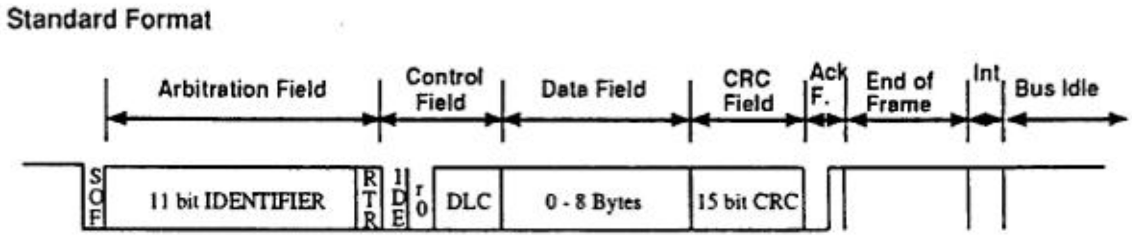
depending on whether the control mechanisms are present in the system only once (centralized) or more than once (decentralized).

A communication system with a designated station (inter alia for centralized bus access control) must provide a strategy to take effect in the event of a failure of the master station. This concept has the disadvantage that the strategy for failure management is difficult and costly to implement and also that the takeover of the central station by a redundant station can be very time-consuming.

For these reasons and to circumvent the problem of the reliability of the master station (and thus of the whole communication system), the CAN protocol implements decentralized bus control. All major communication mechanisms, including bus access control, are implemented several times in the system, because this is the only way to fulfill the high requirements for the availability of the communication system.

In summary it can be said that CAN implements a traffic-dependent bus allocation system that permits, by means of a non-destructive bus access with decentralized bus access control, a high useful data rate at the lowest possible bus data rate in terms of the bus busy rate for all stations. The efficiency of the bus arbitration procedure is increased by the fact that the bus is utilized only by those stations with pending transmission requests. These requests are handled in the order of the importance of the messages for the system as a whole. This proves especially advantageous in overload situations.

Since bus access is prioritized on the basis of the messages, it is possible to guarantee low individual latency times in real-time systems.



Message frame for standard format (CAN Specification 2.0A)

Message frame formats:

The CAN protocol supports two message frame formats, the only essential difference being in the length of the identifier (ID). In the standard format the length of the ID is 11 bits and in the extended format the length is 29 bits. The message frame for transmitting messages on the bus comprises seven main fields.

A message in the standard format begins with the start bit ”start of frame”, this is followed by the ”arbitration field”, which contains the identifier and the ”RTR” (remote transmission request) bit, which indicates whether it is a data frame or a request frame without any data bytes (remote frame).

The "control field” contains the IDE (identifier extension) bit, which indicates either standard format or extended format, a bit reserved for future extensions and - in the last 4 bits - a count of the data bytes in the data field.

The "data field” ranges from 0 to 8 bytes in length and is followed by the "CRC field”, which is used as a frame security check for detecting bit errors.

The "ACK field”, comprises the ACK slot (1 bit) and the ACK delimiter (1 recessive bit). The bit in the ACK slot is sent as a recessive bit and is overwritten as a dominant bit by those receivers which have at this time received the data correctly (positive acknowledgement). Correct messages are acknowledged by the receivers regardless of the result of the acceptance test. The end of the message is indicated by "end of frame”. ”Intermission” is the minimum number of bit periods separating consecutive messages. If there is no following bus access by any station, the bus remains idle (”bus idle”).

Detecting and signaling errors:

Unlike other bus systems, the CAN protocol does not use acknowledgement messages but instead signals any errors that occur. For error detection the CAN protocol implements three mechanisms at the message level:

- *Cyclic Redundancy Check (CRC)*
  The CRC safeguards the information in the frame by adding redundant check bits at the transmission end. At the receiver end these bits are re-computed and tested against the received bits. If they do not agree there has been a CRC error.
- *Frame check*
  This mechanism verifies the structure of the transmitted frame by checking the bit fields against the fixed format and the frame size. Errors detected by frame checks are designated "format errors”.
- *ACK errors*
  As mentioned above, frames received are acknowledged by all recipients through positive acknowledgement. If no acknowledgement is received by the transmitter of the message (ACK error) this may mean that there is a transmission error which has been detected only by the recipients, that the ACK field has been corrupted or that there are no receivers.

The CAN protocol also implements two mechanisms for error detection at the bit level.

- *Monitoring*
  The ability of the transmitter to detect errors is based on the monitoring of bus signals: each node which transmits also observes the bus level and thus detects differences between the bit sent and the bit received. This permits reliable detection of all global errors and errors local to the transmitter.
- *Bit stuffing*
  The coding of the individual bits is tested at bit level. The bit representation used by CAN is NRZ (non-return-to-zero) coding, which guarantees maximum efficiency in bit coding. The synchronization edges are generated by means of bit stuffing, i.e. after five consecutive equal bits the sender inserts into the bit stream a stuff bit with the complementary value, which is removed by the receivers. The code check is limited to checking adherence to the stuffing rule.

If one or more errors are discovered by at least one station (any station) using the above mechanisms, the current transmission is aborted by sending an "error flag”. This prevents other stations accepting the message and thus ensures the consistency of data throughout the network.

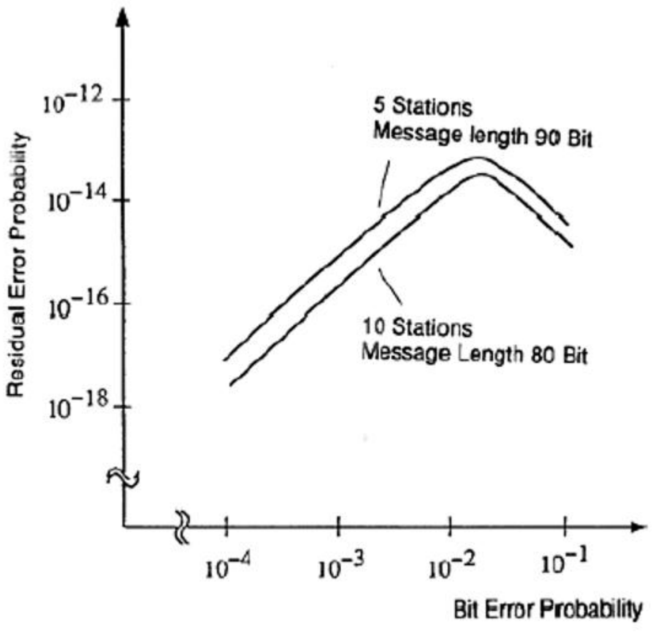
After transmission of an erroneous message has been aborted, the sender automatically re-attempts transmission (automatic repeat request). There may again be competition for bus allocation. As a rule, retransmission will be begun within 23 bit periods after error detection; in special cases the system recovery time is 31 bit periods.

However effective and efficient the method described may be, in the event of a defective station it might lead to all messages (including correct ones) being aborted, thus blocking the bus system if no measures for self-monitoring were taken. The CAN protocol therefore provides a mechanism for distinguishing sporadic errors from permanent errors and localizing station failures (fault confinement). This is done by statistical assessment of station error situations with the aim of recognizing a station‘s own defects and possibly entering an operating mode where the rest of the CAN network is not negatively affected. This may go as far as the station switching itself off to prevent messages erroneously recognized as incorrect from being aborted.

Data reliability of the CAN protocol:

The introduction of safety-related systems in automobiles brought with it high requirements for the reliability of data transmission. The objective is frequently formulated as not permitting any dangerous situations for the driver to occur as a result of data exchange throughout the whole life of a vehicle.

This goal is achieved if the reliability of the data is sufficiently high or the residual error probability is sufficiently low. In the context of bus systems data, reliability is understood as the capability to identify data corrupted by transmission faults. The residual error probability is a statistical measure of the impairment of data reliability: it specifies the probability that data will be corrupted and that this corruption will remain undetected. The residual error probability should be so small that on average no corrupted data will go undetected throughout the whole life of a system.



Residual error probability as a function of bit error probability

Calculation of the residual error probability requires that the errors which occur be classified and that the whole transmission path be described by a model. If we determine the residual error probability of CAN as a function of the bit error probability for message lengths of 80 to 90 bits, for system configurations of, for instance, five or ten nodes and with an error rate of 1/1000 (an error in one message in every thousand), then maximum bit error probability is approximately 0.02 – in the order of 10^{-13}. Based on this it is possible to calculate the maximum number of undetectable errors for a given CAN network.

For example, if a CAN network operates at a data rate of 1 Mbit/s, at an average bus capacity utilization of 50 percent, for a total operating life of 4000 hours and with an average message length of 80 bits, then the total number of messages transmitted is 9 x 10^{10}. The statistical number of undetected transmission errors during the operating life is thus in the order of less than 10^{-2}. Or to put it another way, with an operating time of eight hours per day on 365 days per year and an error rate of 0.7 s, one undetected error occurs every thousand years (statistical average).

Extended format CAN messages

The SAE "Truck and Bus" subcommittee standardized signals and messages as well as data transmission protocols for various data rates. It became apparent that standardization of this kind is easier to implement when a longer identification field is available.

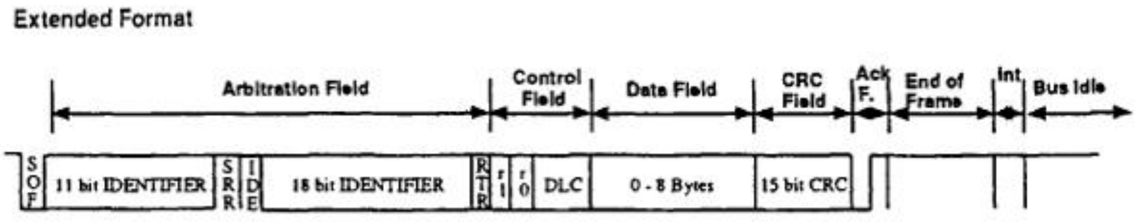
To support these efforts, the CAN protocol was extended by the introduction of a 29-bit identifier. This identifier is made up of the existing 11-bit identifier (base ID) and an 18-bit extension (ID extension). Thus the CAN protocol allows the use of two message formats: StandardCAN (Version 2.0A) and ExtendedCAN (Version 2.0B). As the two formats have to coexist on one bus it is laid down which message has higher priority on the bus in the case of bus access collisions with dithering formats and the same base identifier: the message in standard always has priority over the message in extended format.

CAN controllers which support the messages in extended format can also send and receive messages in standard format. Only messages in standard format can be transmitted on the entire network if CAN controllers that only cover the standard format (Version 2.0A) are used on that network. Messages in extended format would be misunderstood. However there are CAN controllers which only support standard format but recognize messages in extended format and ignore them (Version 2.0B passive).

The distinction between standard format and extended format is made using the IDE bit (Identifier Extension Bit) which is transmitted as dominant in the case of a frame in standard format. For frames in extended format it is recessive.

The RTR bit is transmitted dominant or recessive depending on whether data are being transmitted or whether a specific message is being requested from a station. In place of the RTR bit in standard format the SRR (substitute remote request) bit is transmitted for frames with extended ID. The SRR bit is always transmitted as recessive to ensure that in the case of arbitration the standard frame always has priority bus allocation over an extended frame when both messages have the same base identifier. Unlike the standard format, in the extended format the IDE bit is followed by the 18-bit ID extension, the RTR bit and a reserved bit (r1).

All the following fields are identical with standard format. Conformity between the two formats is ensured by the fact that the CAN controllers which support the extended format can also communicate in standard format.



Message frame for extended format (CAN Specification 2.0A)

Implementations of the CAN protocol

Communication is identical for all implementations of the CAN protocol. There are differences, however, with regard to the extent to which the implementation takes over message transmission from the microcontrollers which follow it in the circuit.

CAN controller with intermediate buffer:

CAN controllers with intermediate buffer (formerly called basicCAN chips) have implemented as hardware the logic necessary to create and verify the bitstream according to protocol. However, the administration of data sets to be sent and received, acceptance filtering in particular is carried out to only a limited extent by the CAN controller.

Typically, CAN controllers with intermediate buffer have two reception and one transmission buffer. The 8-bit code and mask registers allow a limited acceptance filtering (8 MSB of the identifier). Suitable choice of these register values allows groups of identifiers or in borderline cases all ID‘s to be selected. If more than the 8 ID-MSB‘s are necessary to differentiate between messages then the microcontroller following the CAN controller in the circuit must complement acceptance filtering by software.

CAN controllers with intermediate buffer may place a strain on the microcontroller with the acceptance filtering, but they require only a small chip area and can therefore be produced at lower cost. In principle they can accept all objects in a CAN network.

CAN controller with object storage:

CAN objects consist mainly of three components: identifier, data length code and the actual useful data. CAN controllers with object storage (formerly called fullCAN) function like CAN controllers with intermediate buffers, but also administer certain objects. Where there are several simultaneous requests they determine, for example, which object is to be transmitted first. They also carry out acceptance filtering for incoming objects. The interface to the following microcontroller corresponds to a RAM. Data to be transmitted are written into the appropriate RAM area, data received are read out correspondingly. The microcontroller has to administer only a few bits (e.g. transmission request).

CAN controllers with object storage are designed to take as much strain as possible off the local microcontroller. These CAN controllers require a greater chip area, however, and are therefore more expensive. In addition to this, they can only administer a limited number of chips. CAN controllers are now available which combine both principles of implementation. They have object storage, at least one of which is designed as an intermediate buffer. For this reason there is no longer any point in differentiating between basicCAN and fullCAN.

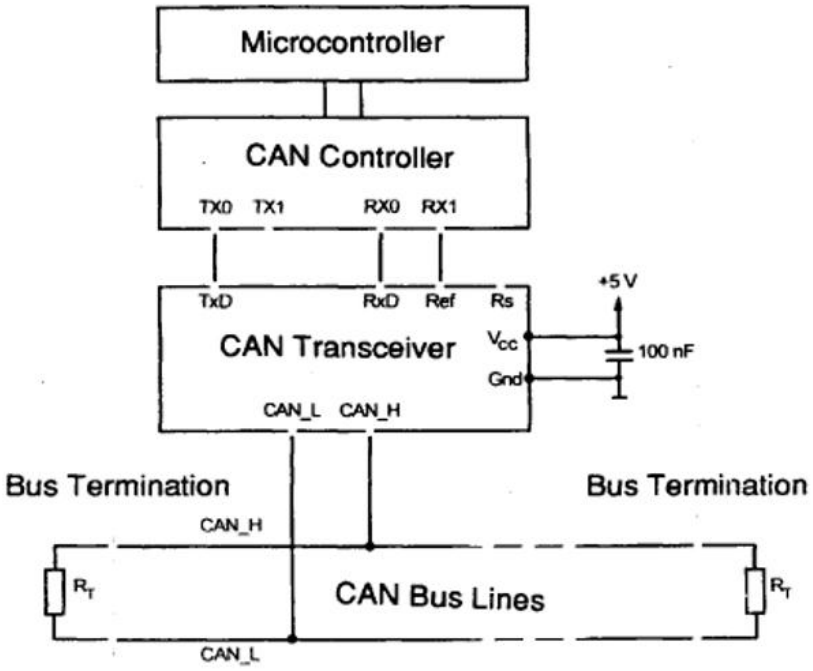
CAN slave controllers for I/O functions:

As well as CAN controllers which support all functions of the CAN protocol there are also CAN chips which do not require a following microcontroller. These CAN chips are called SLIO (serial link I/O). CAN chips are CAN slaves and have to be administered by a CAN master.

Physical CAN connection

The data rates (up to 1 Mbit/s) necessitate a sufficiently steep pulse slope, which can be implemented only by using power elements. A number of physical connections are basically possible. However, the users and manufacturers group CAN in Automation recommends the use of driver circuits in accordance with ISO 11898.

Integrated driver chips in accordance with ISO 11898 are available from several companies (Bosch, Philips, Siliconix and Texas Instruments). The international users and manufacturers group (CiA) also specifies several mechanical connections (cable and connectors).



Physical CAN Connection according to ISO 11898

Source Proof:

This text was kindly provided for us by the CiA (CAN in Automation) nonprofit organization formed to provide technical, product and marketing information for the CAN protocol.