CAN – CONTROLLER AREA NETWORK

Abstract[1]***. Modern automation systems demand high degrees of stability, reliability and tolerance to counter external interference. This document describes how these goals could be achieved by using CAN networks. It describes how CAN networks work and what are the benefits of using CAN. Distinct CAN features, such as error handling and confinement are reviewed in detail. It also reviews the history of CAN and lists specific areas, where CAN networks are irreplaceable.***

1. WHAT IS CAN

Controller Area Network (CAN) is a broadcast, differential serial bus standard. It was developed in the 1980s by a German company, Robert Bosch and was initially intended, that CAN would be used in the automotive industry, as car manufacturers required an easier way to connect a constantly increasing amount of electronic devices fitted on modern cars. The reason for such a standard to emerge was the increasing complexity of electronic and mechanical systems used in cars. The manufactures struggled to satisfy required safety and reliability requirements due to the amount of wiring needed.

Additionally new safety standards, emerging in the automotive industry required higher reliability and safety of vehicles. CAN was developed specifically to accommodate those issues by providing a more reliable way of interconnecting different system in vehicles and industrial machinery. CAN was also able to reduce the costs of implementing wiring, as less complexity was introduced into the system, thus reducing time spent on engineering the systems.

In the 1990s the need to implement higher-level protocols was increasing, thus developments were made to build protocols on top of CAN. In 1991 CAN Kingdom was introduced by Kvaser. DeviceNET was introduced in 1994 by Allen-Bradley and was another higher-level protocol utilizing CAN. CANopen was introduced in 1995 by CAN In Automation (CiA is a users and manufacturers group, established in 1992).

The CAN standard was developed in such a way, that it could operate normally even in environments of high interference. CAN uses a comprehensive error handling mechanism, which allows it to work in very harsh environments. Because of this, today, CAN is used in medical equipment, agriculture, large optical telescopes and in many more areas.

---

[1] Artikkel on valminud Tallinna Tehnikaülikooli arvutitehnika instituudi aine IAF0030 – Arvutitehnika erikursus: Veakindlad arvutisüsteemid – raames. Juhendaja Gert Jervan.

The default CAN standard supported data rates of up-to 1 Mbps (Mega-bits per second) and can operate using a normal twisted pair wire (shielded or unshielded). The maximum rate, however, depends on the quality and length of the cables used in a particular network.

As an example of a CAN network, we could review different systems in a modern car. Needless to say, having dedicated wiring for every system in today's automotive industry is not acceptable. Implementing reliable operation of airbags, air conditioning, ABS, stability control, electric windows, central locking, electrical seats and so forth would not be possible without CAN. The amount of dedicated wiring would exceed any reasonable amount and reliability of such a system would be dramatically decreased. Maintenance costs would be unacceptably high. On the other hand, if a CAN network is used to tie these systems together, the amount of wiring needed would be significantly decreased.

2. HOW CAN WORKS

PRINCIPLE

Every CAN network consists of nodes, which transmit or receive data. There are no addresses used on a CAN network. The transmitting node sends the message to everyone and only the nodes, which actually require this information process it.

Each message, sent within a CAN network, has it's own unique identifier. These identifiers are programmed into the receiving nodes, that way, they know what messages they need to process and which ones to discard.

The unique identifier is also a priority identifier. The lower the identifier numeric value, the higher the priority. This priority system is used with a non-destructive arbitration, which ensures, that no data loss will occur if 2 or more nodes happen to send messages simultaneously. Since each message has it's priorities and non-destructive arbitration is used, no message will be lost.

BIT ENCODING

CAN networks use a NRZ (Non Return to Zero) mechanism for better error handling and isolation. It is used in conjunction with bit stuffing. Basically this mechanism ensures, that there are enough signal transitions in the line and provides high resilience to external disturbance.

Bit Timimg

CAN Nodes have their own clocks, which they use to determine when the data should be transmitted. Synchronization works by dividing each frame into segments. The segments used are: Synchronization, Propagation, Phase 1 and Phase 2. Each segment has an adjustable length.

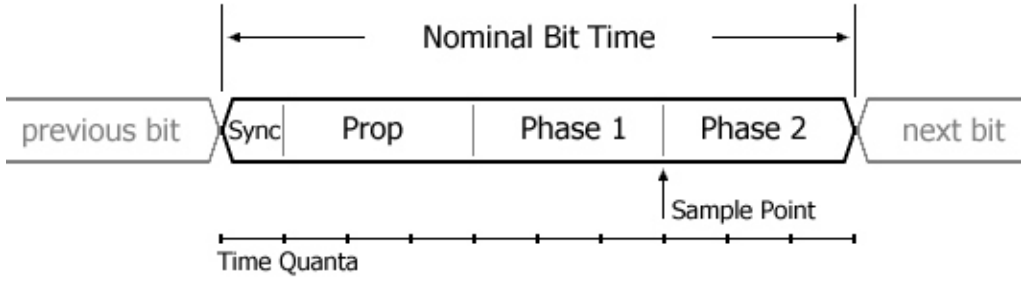

Figure 1. CAN Frame

ROBUSTNESS

CAN networks also feature robust mechanisms and will continue to work in unexpected circumstances. The ISO 11898 standard recommends, that CAN networks continue to operate even if the physical bus (the hardware carrier, e.g. Twisted pair) is damaged. According to the ISO standard the network should continue to operate if

- Either of the two wires in the bus is broken
- Either wire is shorted to power
- Either wire is shorted to ground

CAN NETWORK EXPANSION

CAN network are extremely easy to expand and they are in general very flexible. This is mainly due to the absence of addresses on the network. The sending node sends messages to all the nodes and newly added nodes (the ones added after the network was designed) can simply read the required data, when it gets to them, without any need to reconfigure the sending nodes or any other node for that matter.

CAN BUS ARBITRATIONS SYSTEM

In any system some parameters are more important, than others. This is also true for automotive systems and therefore CAN features a priority system to distinguish between different types of messages. For instance, the revelations of the engine change more frequently than the temperature of the engine coolant. So the user (driver) needs to get the RPM information more frequently.

The more frequently changing parameters are sent more frequently and are always given higher priority (lower identification number).

To decide which message has which priority CAN uses a method known as Carrier Sense, Multiple Access with Collision Detection (CSMA/CD), but it is combined with non destructive bitwise arbitration. This provides collision resolution, instead of just dropping data messages, which have lower priority.

NON DESTRUCTIVE BITWISE ARBITRATION

As it was mentioned previously, the priorities of messages are defined by their unique identifier numeric representation. This is defined while the system is being designed. Every sending node will need to be configured to send messages with a special unique identifier and other nodes cannot use this identifier for their messages once that number has been assigned. Any potential bus conflicts are resolved by bitwise arbitration in accordance with the wired-and mechanism, by which a dominant state (logic 0) overwrites a recessive state (logic 1).

This method of arbitration provides bus allocation on the basis of need and it offers significant efficiency benefits while compared to a destructive bus allocation on an Ethernet network for example. Only the capacity of the bus is limiting the speed for a CAN network. Even when this limit is reached the network will not collapse and messages will still be sent properly and in accordance with their priorities (unlike with an non-switched Ethernet, where such a situation would decrease the throughput significantly and wait times will increase because of destructive arbitration).

CAN MESSAGE FORMAT

Message Frames

All data in the CAN network is transmitted by using Message Frames. In the case of a network without addresses it would look like one transmitting node sends a message, which all nodes receive, but only the ones, to which this message was relevant, process it.

There are 2 types of CAN protocols. The difference between them is basically the length of the unique identifier, that can be used. The longer the identifier, the more nodes you can potentially put on the network.

The first is the standard CAN protocol (version 2.0A) and it has 11 bit identifiers.

The second is the extended CAN protocol (version 2.0B) and it has 29 bit identifiers (but also supports 11 bit identifiers for backward compatibility).

The 2.0A Message Format

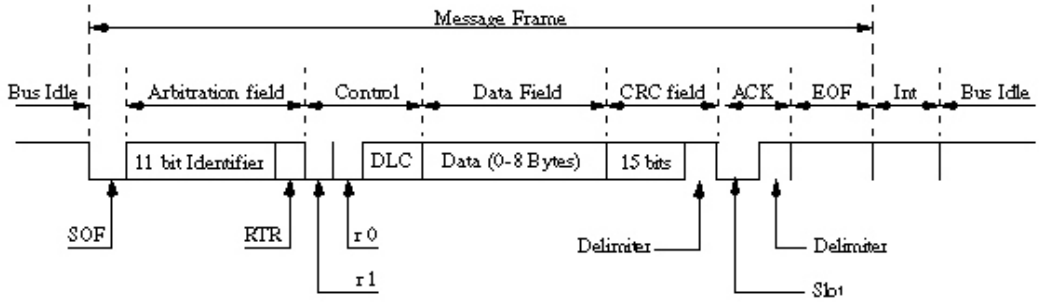

Fig-

CAN messages consist of the following parts:

- A Start of Frame (SOF) field – this shows the start of the message. It is a single dominant bit, which is used to synchronize the nodes after the bus has been in the idle state.
- The identifier is the standard CAN 11 bit identifier, that indicates a priority of the message. In a CAN network, the lower the binary value of the identifier, the higher the priority.
- The RTR is the single Remote Transmission Request. If it is set to 1, then the information is requested from a node. All nodes, in the network receive this request, however the identifier determines which node is to reply. Once the node replies, all nodes receive this reply, but only those, assigned to receive this reply actually process the received data.
- R1 (also called IDE in some sources) is a single identifier extension. It is used to indicate whether a standard or a non-standard (an extended) identifier is used.
- R0 is a reserved bit, which will possibly be used in future protocol revisions.
- DLC (Data Length Code) is a 4 bit digit, containing the number of bytes of data being transmitted.
- A Data Field, containing from zero to eight bytes.
- The CRC is a standard CRC that is optimized for bit counts less than 127 (BHC code). It is generated and decoded in hardware, making it a very fast error checking method for the CAN frame. In order to carry out the CRC calculation the polynomial to be divided is defined as the polynomial, the coefficients of which are given by the destuffed bit stream consisting of START OF FRAME, ARBITRATION FIELD, CONTROL FIELD, DATA FIELD (if present) and, for the 15 lowest coefficients, by 0. This polynomial is divided (the coefficients are calculated modulo-2) by the generator-polynomial: X15 + X14 + X10 + X8 + X7 + X4 + X3 + 1. Following the CRC is a single recessive bit (CRC delimiter) as well as an ACK bit and an ACK delimiter.
- Every node receiving an accurate message overwrites this recessive bit in the original message with a dominate bit, indicating an error-free message has been sent. Should a receiving node detect an error and leaves this bit recessive, it discards the message and

the sending node repeats the message after rearbitration. In this way each node acknowledges (ACK) the integrity of its data. ACK is 2 bits, one is the acknowledgement bit and the second is a delimiter. After the message has been transmitted the bus can either be idle or already have another message. So the idle time may be zero or non-zero and the network would still operate normally.

- The EOF is the end of frame. A seven bit field marks the end of the frame and disables bit stuffing. If it is dominant, it means, that a stuffing error has occurred. When 5 bits of the same logic level occur in succession during normal operation, a bit of the opposite logic level is stuffed into the data.

The 2.0B Message Format

The 2.0B format is an evolution of the standard CAN network. It was designed to meet modern demands, as the previous format was getting outdated.

The formats are completely interoperable. Nodes from different message formats can work together, however there are several restrictions to how they behave.

Basically the nodes that support message 2.0A format can simply ignore the messages sent by 2.0B transmitting nodes.

The number of unique identifiers available to users, on a single 2.0A network, is 2,032 (2^{11}-2^4).

Leaving aside the use for compatibility purposes with 2.0B, the number of unique identifiers available on a 2.0B network is in excess of 500 million!

3. IMPLEMENTATIONS OF CAN

CAN nodes could be designed differently depending on what the requirements for the system are. There are 2 implementations of CAN and the complexity and cost of the nodes depends on each one of them.

Basic CAN

In a basic CAN there is a link between the microcontroller and the CAN controller. The microcontroller in a CAN node does absolutely all the processing. So for example if a CAN node is doing some processing and it received a new message, then the microcontroller would be interrupted to process the incoming message. In some cases this would not be acceptable, especially if high data rate needs to be achieved and full CAN should be used.

Full CAN

In full CAN there is a special piece of hardware integrated into each node, which handles all the networking without interrupting the microprocessor. In this type of node, receiving and sending of data is done independently of what the microprocessor is doing. This allows reaching better quality and speed on the network.

Data Rate and Bus Length

In CAN networks the data rate is closely linked to the bus length (the length of the transmission media). The standard maximum rate is 1 Mbps and it can be achieved if the bus length is 40 metres or below. Below are some sample figures for Data Rate and Bus Length

- 500 Kbps at 100 metres.
- 250 Kbps at 200 metres.
- 125 Kbps 500 metres.

CAN APPLICATION LAYERS

As was described in the previous sections CAN message data payload is 8 bytes. This is not enough for modern applications and new standards are built on top of CAN to fulfil the requirements of those systems. These basically represent an Application Layer (from OSI model), which is built on CAN (CAN is only physical and data-link layers, so it is possible to stack more levels on top).

- CAL (CAN Application Layer)
  Was developed by Philips medical systems and currently maintained by CAN in Automation (CiA). This protocol is royalty free.
- CANopen
  This is an implementation of CAL, but it was defined by the CANopen Communications Profile in CiA.
  "You might also want to get hold of a copy of "Embedded Networking with CAN and CANopen" by Olaf Pfeiffer, Andrew Ayre and Christian Keydel. Published by RTC Books. ISBN: 0-929392-78-7."
- DeviceNet
  This protocol is used in industrial automotive industry.
- SDS (Smart Distributed System)
  SDS is also a CiA-approved application layer. Developed by Honeywell, one of the main uses of SDS is for machine control applications.

4. ERROR HANDLING

CAN networks don't only support error detection, but they also have error confinement capabilities. This means, that errors are accounted for and if necessary the node, sending messages with errors will be shut down.

It must be said, that error detection capabilities in the CAN network are very impressive. Global errors, which occur in every node, are 100% detectable. The probability of and error missed by the CRC check is only $3x10^{-5}$ and with all other mechanisms working in conjunction with CRC it is 10^{-11}.

CRC – Cyclic Redundancy Check

Every message sent on the network has a 15 bit CRC code inside (please refer to 2.0A Message Format Section for more information). All receiving nodes verify the CRC and can detect if the message contains an error or not.

Frame Check

In every message frame that is sent on the CAN network, there are some reserved bits, that are defined by the protocol and are not changeable if the content of the message is changed. The receiving nodes check for those reserved bits and can determine whether the frame had errors.

Bit Stuffing

Bit stuffing is technique used to check communication integrity. In CAN, it is defined, that only 5 consecutive bits can be of the same levels (either 0 or 1). So if the sending node has to send data, that needs more, than 5 of bits to be of the same level, it automatically inverts the 6-th bit. The receiving node knows that and also inverts it to get the message. If an error during transmission has occurred, then the CRC will not match and the message would be considered invalid.

5. ERROR CONFINEMENT

The error confinement mechanism of a CAN network is considered to be unique. The network can not only detect failures, but it can distinguish between temporary and permanent failures. If temporary failures can occur from unexpected environment changed like voltage spikes, permanent failures are caused by corrupt cabling or even defective circuitry.

Error Counts

In every CAN controller, there are registers specifically designed for counting the errors, that occur in the network. Whenever an error is detected it is given a numeric priority identifier (receive errors have priority of 1 and sending errors have priority of 8) and saved to the register.

Basically the idea is: if an error message is flagged, then the counter increments, if the message is correct, then the counter decrements. So if it was a temporary error (or even many), then the counter would decrement, as future messages will be correct. If it is a permanent error, it would not decrement.

Error Active Mode

This is a normal status of the node. So the node is working as it normally would. Here, the value of the error count register is less, than 127.

Error Passive Mode

The node switches to this state, if the error count register value exceeds the one in “Error Active Mode” (127). Now, the node can still transmit data, but it cannot flag any messages it received as having errors. This is also known as an alert state.

Bus Off Mode

If the error count continues to increase and reaches 255, then the node cannot transmit or receive any more data (to prevent damage to the network).

6. IMPROVED CAN STANDARDS

Several updated CAN standards have been designed since the original standard was developer in 1980s. The new standards are fully based on the original, but offer some improvements and were generally designed for some specific needs. For example, for some applications better fault tolerance is required, but higher speed is not needed, for the others it’s vice versa. The most significant standards are listed below.

- **ISO 11898-2**: CAN high-speed
- **ISO 11898-3**: CAN fault-tolerant (low-speed)
- **ISO 11992-1**: CAN fault-tolerant for truck/trailer communication
- **ISO 11783-2**: 250 kbit/s, Agricultural Standard
- **SAE J1939-11**: 250 kbit/s, Shielded Twisted Pair (STP)
- **SAE J1939-15**: 250 kbit/s, UnShielded Twisted Pair (UTP) (reduced layer)
- **SAE J2411**: Single-wire CAN (SWC)

7. CAN APPLICATIONS

VOLVO PASSENGER CAR (XC90)

The image below shows the distributed control architecture of the Volvo XC90. The blocks represent ECUs and the thick lines represent networks. The ECU classes shown on this diagram are powertrain, chassis, infotainment and body electronics. The acronyms for the ECUs are explained below the diagram. The networks are used to connect different ECUs together.There are two CAN buses. The leftmost network in the diagram is a CAN for power train and chassis subsystems. It connects for example engine and brake control (TCM, ECM, BCM, etc.) and has a communication rate of 500 kbps. The other CAN connects body electronics such as door and climate control (DDM, PDM, CCM, etc.) and has a communication rate of 125 kbps. The central electronic module (CEM) is an ECU that acts as a gateway between the two CAN buses. A media oriented system transport (MOST) network defines networking for infotainment and telematics subsystems. It consequently connects ECUs for multimedia, phone, and antenna. Finally, local interconnect networks (LINs) are used to connect slave nodes into a subsystem and are denoted by dashed lines in the block diagram. The maximum configuration for the vehicle contains about 40 ECUs.

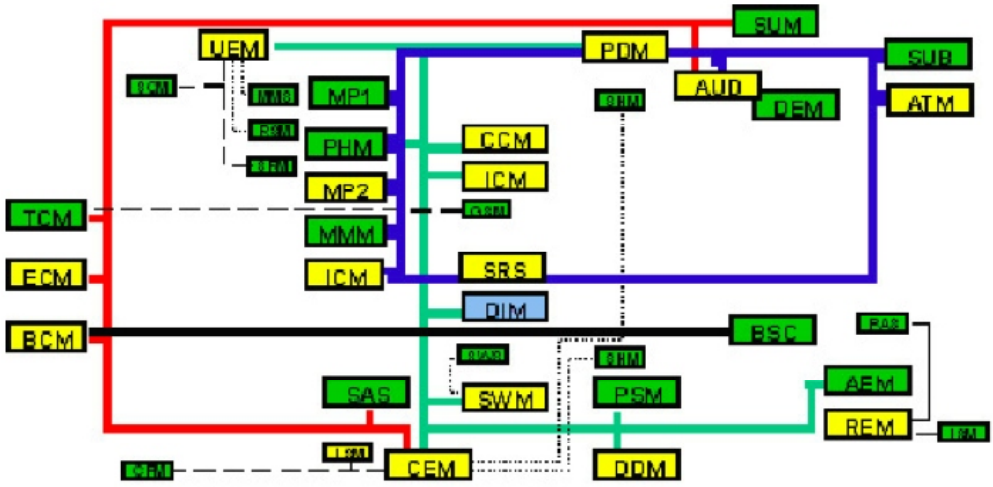

Figure 3. Control architecture of the Volvo XC90

Powertrain and chassis Body electronics: TCM Transmission control module, CEM Central electronic module, ECM Engine control module, SWM Steering wheel module, BCM Brake control module, DDM Driver door module, BSC Body sensor cluster, REM Rear electronic module, SAS Steering angle sensor, SWM Steering wheel module, SUM Suspension module, DDM Driver door module, AUD Audio module, PDM Passenger door module, REM Rear electronic module.

Infotainment/Telematics: CCM Climate control module, MP1,2 Media players 1 and 2, ICM Infotainment control, PHM Phone module, UEM Upper electronic module, MMM Multimedia module, DIM Driver information module, SUB Subwoofer, AEM Auxiliary electronic, ATM Antenna tuner module.

8. CONCLUSION

CAN Networks are designed to work in very harsh environments. They have exceptional error handling and confinement abilities, which makes them an ideal use for heavy-duty equipment, such as agricultural machinery and systems, that must operate in environments with high level of interference. CAN is still widely used today and one of the best standards for reliable and fault tolerant networks.

9. REFERENCES

1. Wikipedia reference for CAN (http://en.wikipedia.org/wiki/Controller_Area_Network)
2. http://hem.bredband.net/stafni/developer/frames.htm
3. http://www.semiconductors.bosch.de/en/20/can/index.asp
4. http://www.s3.kth.se/~kallej/papers/can_necs_handbook05.pdf
5. http://focus.ti.com/lit/an/sloa101/sloa101.pdf
6. http://www.can-cia.org/

KOKKUVÕTE

Tänapäevased automaatsüsteemid peavad olema väga stabiilised, töökindlad ja vastupidavad väliste mõjutuste suhtes. Käesolev artikkel kirjeldab, kuidas neid eesmärke on võimalik saavutada, kasutades andmesideks CAN-võrke. Kirjeldatakse CAN protokolli tööpõhimõtteid ja eeliseid. Detailsemalt peatutakse CANi eriomadustel, mis puudutavad vigadega toimetulemist. Antakse ülevaade ka CANi ajaloost ning selle kasutusvaldkondadest.

Deniss Nikiforov
Tallinna Tehnikaülikool
Infotehnoloogia teaduskond