**University of Ulster**

**COM 342: Networks and Data Communications**

**Can-Bus Networking**

**Student Name: Leo Groza**

**Student Number:**

Contents

[Introduction 2](#_Toc498692540)

[Industrial Application 3](#_Toc498692541)

[Can-Bus Networking 5](#_Toc498692542)

[Layers 6](#_Toc498692543)

[Message Transfer 7](#_Toc498692544)

[Error Detection 7](#_Toc498692545)

[Bit Timing 8](#_Toc498692546)

[Clock Synchronization 8](#_Toc498692547)

[Security 9](#_Toc498692548)

[Impact 10](#_Toc498692549)

[Conclusions 11](#_Toc498692550)

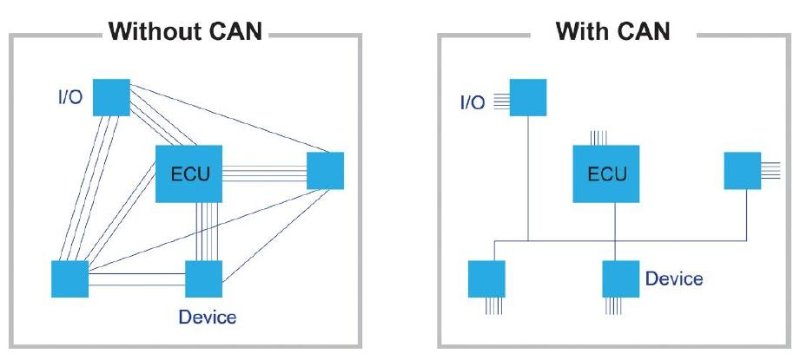
[Bibliography 12](#_Toc498692551)

# Introduction

The Controller Area Network (CAN-bus), also known as CAN, represents a real-time communication network, at speed up to 1 Mb/s, over a two-wires serial network. It supports real-time distributed control with a high security level. (GmbH, 1991)

CAN has been created in the late 70’ by the need of replacing a conventional point to point wiring method that provides a discrete interconnection among systems. CAN has been developed in Europe, in the late 80’, as a replacement for the simple electrical car systems as the complexity of automotive electronics has grown considerably.

As engine management systems have gown in complexity, there was a need to keep each device as an integrated part of an entire system. The solution was to replace the complexity, wiring and tens of connectors, with an intersystem communication between devices with a low cost digital network. Due to the noise of the electrical environment, the network had to have high noise immunity and be capable of retransmitting failed packets and detect/handle errors. BOSCH has begun the development of the Controller Area Network, to be used in automobiles development, and by being robust, CAN networks are replacing, in the present days, point to point communications in different large industry areas such as industrial trains, medical systems, household appliances, control systems and automations.

(Anon., n.d.)

Controller Area Network become an international standard with ISO 11898 and ISO 11519-2. (Catsoulis, 2002) CAN is an affordable resolution for embedded control systems mainly due to light management protocol, build in error detection, error correction features and because of the low cost.

# Industrial Application

Can-bus interface chips and controllers have small dimensions. The parts operate at high and real-time speed and in different environments. All the components are available off the shelf and at low cost and they are implemented in a wide range of products.

CAN has a bus topology, meaning one long cable, is the backbone linking all devices on the network. The nodes are connected to the bus by drop lines. A drop line represents the connection between the main cable and the device.

Using Can-bus networks controllers, the reliability of the product increases and the design time drops down due to the availability and multi sourced components.

CAN network protocol can provide communications in highly complex distributed systems with the named capabilities:

- capability of setting priority to messages;

- multicast communication having bit-oriented synchronization;

- consistency of the data throughout the system;

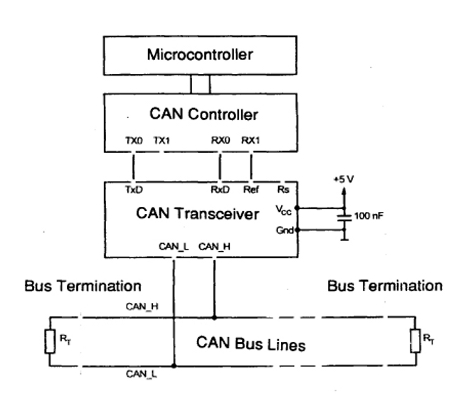
- detection of errors;

- corrupted messages automatic retransmission;

- detection and automatic switching off from defective nodes. (Di Natale, 2008)

Every existing device from a network provided by a Can-Bus network has a CAN controller that makes the device intelligent. As an advantage, the electronic control unit (ECU), can function with a single CAN interface instead of the digital and analogue input connecting every device in the network.

An important aspect in CAN’s specifications represents the Cyclic Redundancy Check. CAN performs an error check on the content of each frame. The frames containing errors are disregarded by each node and a message will be transmitted to signal the error existing in the network, this being an important aspect in the industry. As a very important fact, in this specific structure, local or global errors are separated by the controller and an individual node can disconnect by itself from the network to stop transmitting any more errors. (National Instruments, 2014)

 (esd-electronics-usa.com, n.d.)

# Can-Bus Networking

“*CAN bus is a well-designed network, based on techniques learned from computer networks. It is a serial connected bus*, *where all nodes have access to the network, and collisions between nodes are detected within a very short time. This allows devices to have a relatively equal share of the bandwidth of the bus. The CAN bus is a rugged bus which copes well with errors, and also devices which are not operating correctly”* (Buchanan, 2000)*.*

Controller Area Network represents a bus standard created to give microcontrollers and devices access to communicate with together within a network, with or without a host computer (Buchanan, 2014). Nodes may be added at any time even if the network is functional. When data is transmitted by CAN, the content of the message, is designated by a unique identifier in the network. The specific identifier, besides defining the content of the data, also defines the importance of the data.

­­­CAN protocol characterizes two of seven-layers in the OSI model: the data link and the physical layer. It does not contain any specification for protocols of higher level as: node addresses, transport of large packets, flow control.

The above-named protocols are implemented by a higher-level protocol which:

- creates a logic addresse for nodes;

- formats messages;

- handles system-errors;

- organizes a start-up process.

To acquire a better understanding about the construction of a can-bus network, the definition of a “node” in networking relates to a connection point, a communication end-point or a distribution point. Node’s definition is different from a network to another and from protocol to protocol (Oracle Corporation, 2002).

## Layers

The CAN BUS divides into three main layers:

1. Object layer;

2. Transfer layer;

3. Physical layer.

The Object layer is responsible to provide an interface for the application layer, for finding the message to be transmitted and to regulate what data received by the transport layer is to be used. To be specific, the object layer implements one part of the Data Link layer.

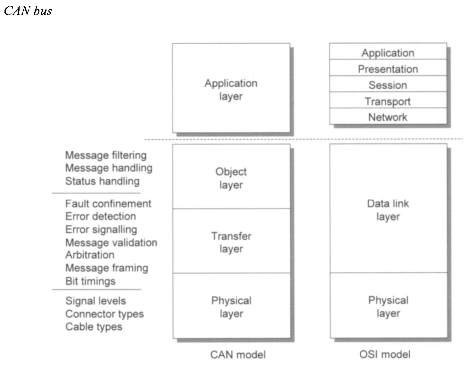
The Transfer layer is responsible for error checking and signaling of errors, frame control and arbitration. The Transport layer provides the timing and it decides when to start a new transmission if the bus is free. The second part of the Data Link layer is taken by the Transfer layer.

The Physical layer relates to the definition of the mechanical and electrical aspects.

Defines the connectors, cables and characterizes the representation of bits and signal levels.

In the Physical layer a bit can either be *dominan*t, or a logical 0, or *recessive*, logical 1. Each bit corresponds to an electrical level implemented in the layer used. In general, modules are “AND wired” connected to the bus, denoting that if one node sets the line to a dominant level, the whole line will acquire the similar state disregarding the separate levels on the line. The *dominant* bit will always lead over the *recessive* bit, and if all the levels are *recessive* the result level shall be *recessive* as well.

As presented at the beginning of the paper, the maximum speed of a CAN-bus network is 1MB/s. This maximum speed implies that the maximum cable length must be 40m, length that implies certain disadvantages.



The CAN bus node architecture depends on the environment and its purpose. Engineers continuously develop products for rough environments and try to reduce the costs of the whole systems.

## Message Transfer

A message transfer inside a CAN-bus network implies four diverse types of frames.

*- Data frame* will contain the data from the transmitter to the receivers.

*- Remote frame* is the data transmitted by a unit to request a data frame transmission having the same identifier.

*- Error frame* is transmitted by a unit regarding an error on the bus.

*- Overload frames* provide extra time between the transfer of data packets.

Remote frames and data frames are separated from following frames with a *interframe space*, so that no station can transmit data for a defined time period.

## Error Detection

The errors inside a CAN-bus can be detected through several methods:

- by monitoring the bit levels;

- CRC;

- bit stuffing;

- finally, message frame check.

Throughout the mentioned methods the system can detect all the global errors, all the local errors, burst errors and random distributed errors.

Considering errors, a node can have three states:

- can be error active – meaning it can send active error flags at the detection of an error;

- can be error passive – meaning it will not send active error flags, only passive flags;

- buss off – the nodes will not be allowed to transmit or have any influence on the bus.

Nodes at initialization are *error active* and transmit active error flags to signal error detection. Once the error count exceeds 128 (times) a node will become *passive* and will only transmit passive flags. A *buss off* node will not transmit anything on the bus.

## Bit Timing

For timing purposes, each bit on the CAN-bus is divided into 4 *quanta.*

Quanta is divided logically in four segments:

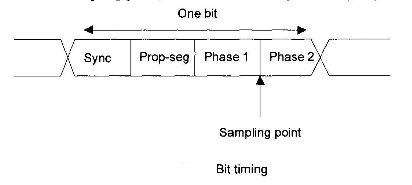
- segment synchronization – used for clock synchronization;

- segment propagation – to compensate the bus lines delay;

- segment 1 phase – used to keep the clocks in synchronization;

- segment 2 phase – is the second part of phase 1.

CAN controllers can also provide options to sample three times during a bit.



## Clock Synchronization

To adjust to a bus clock, CAN controller can lengthen a bit or shorten a bit by a segment.

Two types of clock synchronization can be named:

- Hard synchronization – transpired at the start bit with the recessive to dominant transition. The restart of the bit time is striking from that edge;

- Resynchronization – present when a bit edge is not created at the synchronization segment in a message.

# Security

Security is a topic with increasing importance in industrial applications. In the present days, system communication via CAN-bus is often insecure. Highly suitable concept security algorithms are available for CAN-bus networks but distribution of symmetric cryptographic keys among the nodes is in continuous change.

A very important fact in CAN-bus network security is to make sure that even if one device in the network has been attacked, the impact on the rest of the connected devices can be kept to a minimum.

Security will play an important part in the successful implementation and widespread of connected systems. A major challenge in the security, is the distribution and management of the cryptographic keys.

The threat of CAN-bus communication includes spoofing, unauthorized access, eavesdropping.

A device physically connected to the CAN-bus network can access all the data over the network. To avoid eavesdropping the information passed on the bus needs to be encrypted. The frame contains the destination identifier. A malware device can sniff the network to read the messages passed through the network and find the ID of the victim device. In past works, researchers have been able to obtain unauthorized access to a device by reading the content of a memory block implemented in the network and gained access to the device by reverse engineering. (IEEE Xplore DIgital Library, 2017)

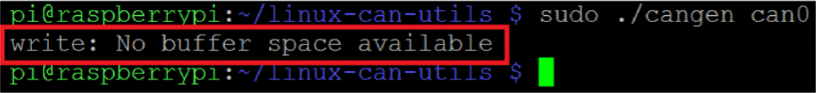


Fig. Screenshot demonstrating buffer overflow

As the next step in security, the general idea implies creating larger frameworks that will include suitable protocols to synchronize frame, transitions between frames and implement group security keys of arbitrary lengths. (Mueller & Lothspeich, 2015)

# Impact

CAN-bus networks are used in a wide range of industrial and automation control systems.

The main applications are:

1. Passenger cars: CAN-bus networks are used as the “in-vehicle network” (IVN).

Majority of European car manufacturers use IVNs based on CAN-bus networks.

2. Factory automation: CAN-bus is used for interconnecting sub-systems, control processes and machinery. Typical application includes production data recordings.

3. Maritime Electronics: CAN-bus networks are used in boats, vessels, integrating and connecting sub-systems. CAN-based ship automation system can be found in this application. Ship automation systems may be composed by several CAN-bus networks.

4. Aircraft electronics: CAN -bus network is the backbone network in aviation, used in data analysis, cockpit interfaces and engine-control systems.

5. Medical equipment: used as networks in several devices such as X-Ray machines. Operating rooms are equipped with CAN-bus networks which manage every function necessary.

6. Building automation: CAN-bus networks manage air conditioning, heating, doors and video and audio control.

7. Non-industrial machine control: gambling machines, bank terminals, printer machines and copy machines all implement an embedded CAN-bus network.

8. Industrial machine control: large printing machines, packaging machines or textile machines, all are embedded by CAN-bus networks. Some applications are motion control oriented.

9. Escalators: all devices in a lift are connected one with other via CAN-bus networks. (CAN 2020 webinar, 2017)­

# Conclusions

Integrated CAN-bus communications allot several advantages to industrial PC users:

Speed: multiple messages can be sent to all connected devices in the same time;

Flexibility: considering its two-wire structure, CAN-bus offers maintenance flexibility and enhanced installation;

Cost: low hardware costs make CAN ideal for applications which require multi-processor communications on a budget;

Reliability: not being sensitive to interference, CAN-bus require less cables and connectors.

CAN-bus is used widely as part of a distributed control system, as the continuous growth of the Internet of Things, the next vital step is to ensure installation longevity, compatibility and expandability. CAN-bus networking is set to be implemented in a wide range of future technologies and modern buildings automation installations. (I/O Hub , 2010)

CAN is a message based system and not an address based system. It is well suited when the data is needed by more than one location and data consistency is mandatory throughout the system.

# Bibliography

1.Anon., 2010. *CANBus.pl.* [Online]   
Available at: http://www.canbus.pl/index.php?id=2&lang=en  
[Accessed 12 11 2017].

2.Anon., n.d. *SEO.* [Online]   
Available at: http://keywordsuggest.org/gallery/1338417.html  
[Accessed 11 11 2017].

3.Buchanan, B., 2014. *The Handbook of Data Communications and Networks: Volume 1. Volume 2.* Paperback ed. s.l.:s.n.

4.Buchanan, W., 2000. *Computer Busses.* s.l.:Elsevier Science.

5.CAN 2020 webinar, 2017. *CAN lower- and higher-layer protocols.* [Online]   
Available at: https://www.can-cia.org/can-knowledge/canopen/cia417/?l=3%23login%23login%23login%23login%23login%23top%23top%23top%23login%23login%23login%23login%23login%23login%23login%23login%23login%23login%23login&cHash=1dc70abb8a5c2095bae13cfd8f940e34  
[Accessed 12 11 2017].

6.Catsoulis, J., 2002. *Designing Embedded Hardware.* s.l.:O'Reilly Media, Inc.

7.Di Natale, M., 2008. *Understanding and using the Controller Area Network,* s.l.: s.n.

8.esd-electronics-usa.com, n.d. *PhysicalCANConnectionISO11898.jpg.* [Online]   
Available at: http://www.esd-electronics-usa.com/Images/CAN%20Images%20Info%20Papers/PhysicalCANConnectionISO11898.jpg  
[Accessed 12 11 2017].

9.GmbH, R. B., 1991. *CAN Specifications.* D-70442 Stuttgart, Patent No. Postfach 30 02 40.

10.I/O Hub , 2010. *Why all the fuss about CAN-BUS?.* [Online]   
Available at: https://www.logicsupply.com/explore/io-hub/fuss-can-bus/  
[Accessed 12 11 2017].

11.IEEE Xplore DIgital Library, 2017. *Secure communication over CANBus.* [Online]   
Available at: http://ieeexplore.ieee.org/document/8053160/  
[Accessed 12 11 2017].

12.Mueller, A. & Lothspeich, T., 2015. *Plug and Secure Communications for CAN.* [Online]   
Available at: http://www.bosch-semiconductors.de/media/automotive\_electronics/pdf\_2/canliteratur\_1/iCC\_2015\_Plug\_and\_Secure\_Communication\_for\_CAN.pdf  
[Accessed 12 11 2017].

13.National Instruments, 2014. *Controller Area Network (CAN) Overview.* [Online]   
Available at: http://www.ni.com/white-paper/2732/en/  
[Accessed 12 11 2017].

14.Oracle Corporation, 2002. *Node Networks.* [Online]   
Available at: https://docs.oracle.com/cd/B10191\_01/calendar.903/b10093/networks.htm  
[Accessed 12 11 2017].