New methods for the analysis of the physical layer of CAN networks and possibilities for robustness improvement

Tobias Frey, Thomas Waggershauser (IXXAT Automation)

With CAN-based systems being used in manifold applications that require continuous operability, also under harsh environmental conditions and extended service-cycles, the verification of CAN operability is essential. Especially, the direct detection of failure sources enabling thorough maintenance before systems may fail.

This presentation shows how the use of enhanced CAN specific test tools in combination with common measurement devices like Digital Signal Analyzers is key to solve typical failures in CAN systems. Often these failures require significant time and effort to solve: e.g. detecting faulty configured devices leading to multiple bit-rates or use of same CAN-IDs by several nodes in a network or the how to detect the device that destroys any messages by its primary error-flag.

Furthermore various possibilities will be discussed which can be used to improve the robustness of CAN networks. These possibilities can help harden a network against failures on the physical layer and misbehavior of devices in order to avoid a breakdown of the complete network.

1 Introduction

With CAN being used in more and more diverse applications with requirements regarding continuous operability, pre-failure maintenance and quick complete error-detection become more and more important and widespread.

Hence, common measurement equipment can be used for many problems; and for the ease of use, CAN-specific maintenance equipment has been developed over the years and is in common use.

Nevertheless, there are still basic CAN-failures that cannot yet be easily detected by CAN-specific maintenance equipment or by common measurement equipment like oscilloscopes.

However, combined usage of common measurement equipment and improved CAN-test-tools provides insight into typical questions including:

- How to detect nodes using the wrong bit-rate?
- How to figure out which CAN-node is causing an error frame?
- How to detect missing CAN-nodes?

2 Wrong setting of CAN bit-rates

Systems with wrong bit-rates settings are a very common problem which may lead to non-useable CAN-networks and require much time to detect.

Problems with bit-rates include:

- Wrong bit-rate in global network
- Different bit-rates used by several CAN-nodes or bus-segments.
- Drifting devices

Often bit-rate problems are due to installation problems – e.g. when devices are installed or replaced and the bit-rate setting was misconfigured or simply forgotten. It may also be due to faulty configurations on devices using soft-configuration, e.g. using CANopen LSS services.

Detecting a globally misconfigured bit-rate can easily be done using most CAN-monitoring tools with included bit-rate auto-detection or an oscilloscope.

However, common CAN-monitoring tools fail when several bit-rates are used in a single network, as these only check for a valid bit-rate. As soon as one valid bit-rate is detected, these tools normally stop auto-detection and provide the first found bit-rate as the correct one.

Therefore the question is: Do CAN-systems operate with several different bit-rate settings and are the different bit-rates detectable using standard CAN-hardware? Three different scenarios have to be considered when discussing wrong bit-rate settings.

2.1.1 One device with different bit-rate

A single device is set to another bit-rate whilst all other devices are using the defined bit-rate. Depending on system design, higher layer protocols devices will attempt to start operation and transmit their boot-up messages. Depending on several factors, including bus-load, location of nodes on the bus media and difference of the bit-rates used, the CAN-network might work or may also fail immediately or after specific operation duration.

At least the node with faulty set bit-rate will not be able to communicate with other nodes. Correspondingly, it will not be visible to the other nodes. Therefore, this node is effectively missing even though it is attached to the network.

As an example: In a network with a limited number of nodes, low busload (less than 20%) and significantly different bitrates, the main system (the nodes operating at the correct bit rate) will work. Assuming an existing system controller does not stop the system as one device is missing or the application software stops node due to missing data. All devices operating at the same bit-rate will work. But the single device with a faulty bit-rate will either:

- Enter error passive state due to the absence of an acknowledgement – it continuously repeats this message until it either gets an acknowledgement or until it goes into bus-off due to other errors.
- Enter bus-off state as its and other CAN-frames are destroyed due to the different bitrates.

For networks with high busloads, significantly more CAN-frames will be destroyed; therefore the probability of restarting CAN-nodes is high [1, 2].

2.1.2 Multiple devices with different bit-rates

In this case, several nodes operate at mismatched bit-rates. This leads to several different bit-rates in the network.

If we use the same assumptions as above (low bus-load, significantly different bit-rates, limited amount of nodes and no main system controller stopping the system) the system might work – at least the nodes with same bit-rate will be able to communicate with each other.

Nevertheless, there will be a significantly high number of error-frames.

2.1.3 Devices with drifting bit-rate

Even if devices are correctly configured, it might happen that devices show a wide-drifting range of their bit-rate. This also leads to temporary different bit-rates and may show similar behavior as described in the case of a single device or several devices using faulty-set bit-rates.

2.2 Detection of different bit-rates

In regards to the detection of different bit-rates, we will focus on a more user-related view: How to easily detect if different bit-rates are used and which nodes are faultily configured?

To detect the different bit-rates, several tools may be used:

- Oscilloscope
- Standard CAN service, diagnostic and monitoring tools
- CAN-based system controllers

When using an oscilloscope, a very detailed analysis is possible and also very small bit-rate variations can be measured. This is often the only way of identifying the CAN-node with a faulty set bit-rate. However, an oscilloscope is more expensive than the other tools and analysis requires significantly more CAN-know-how and effort if a basic oscilloscope without CAN-trigger and CAN-decode functionality is used.

CAN-service tools, whether hand-held stand-alone tools or PC-based solutions, e.g., PC-CAN interface with CAN-monitoring-software, are commonly used to check the basic operation parameters of CAN-networks like bus-load, active CAN-nodes and CAN-identifiers. But these tools may also allow a very detailed analysis of the data communication. Some of these CAN-monitoring and CAN-test tools provide automatic bit-rate detection which sets the CAN-controller to different standard bit-rates and selects the bit-rate which provided valid CAN-frames. This allows for the possibility to detect different bit-rates.

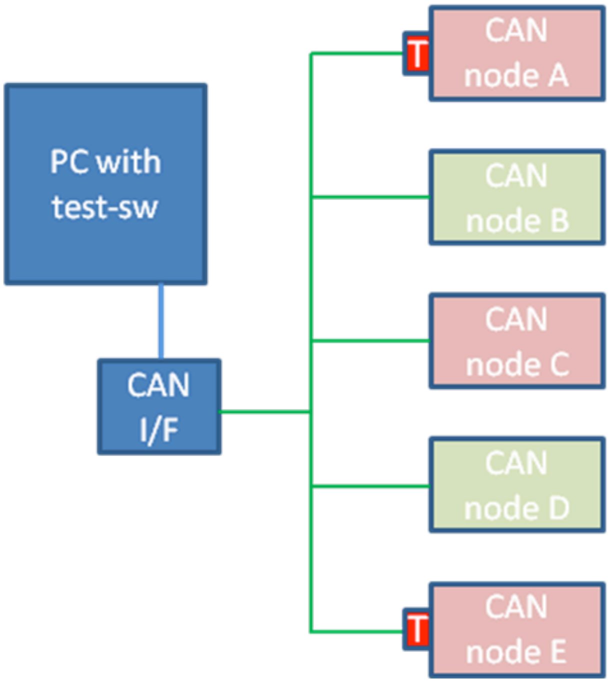
CAN-based system controllers can also be equipped with this bit-rate scan mechanism.

As we will focus on the user-view, we will only explain the possibilities when using CAN test-tools and will omit the usage of oscilloscopes. We will also omit system-controllers as the results are identical to the results when using CAN test-tools.

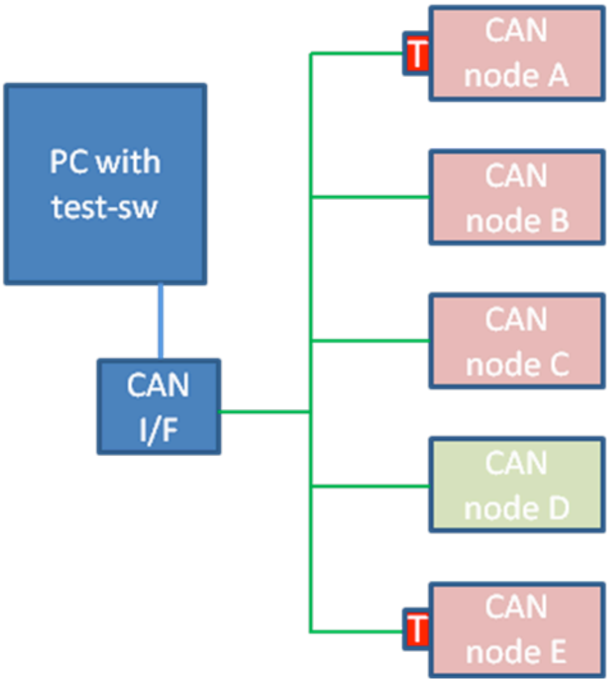
2.3 Detection of different bit-rates using CAN-test tools

With some modifications by the tool providers, it is possible to enable checking for several simultaneously used standard bit-rates in a CAN-network. To verify the operation and reliability of this idea, IXXAT generated a prototype-test-software. The automatic bit-rate detection of the IXXAT CANopen Device Manager was used as a basis for this test-software. The test-software interacts with a standard CAN-controller scanning for common bit-rates, including the CIA bit-rates as specified for CANopen Networks. Scanning is done by setting the CAN-controller to a bit-rate and checking if valid CAN-frames are received within a pre-defined check-time. If valid CAN-frames are received, the selected bit-rate is included in the list of active bit-rates. After expiration of the check-time, the CAN-controller is set to the next bit-rate to be tested. To make sure that the check-time does not fall into the restart-time of a CAN-node going into bus-off, the complete scan-procedure was repeated.

A standard CAN-PC interface using a standard NXP SJA1000 CAN controller was used. In addition, the test-setup included the modified monitoring tool with included bit-rate-detection and a CAN-network consisting of five CAN-nodes as shown in schematic 1. CAN-Termination is attached to nodes A and E



Schematic 1: Multiple devices with different bit-rates.



Schematic 2: One device with different bit-rate.

2.3.1 Multiple devices with different bit-rates

As shown in schematic 1, nodes A, C and E use bit-rate 125kbit/s whilst nodes B and D use bit-rate 250kbit/s. All nodes are pure CAN-nodes with no additional master device required for node operation.

Results: After running the automatic bit-rate detection, the test-software shows both used bit-rates. Several standard CAN-test tools used for comparison show only the first found bit-rate depending on the implementation of the bit-rate detection.

Using a canAnalyser set to the found bit-rates provides information on which nodes are using which bit-rate. This allows detecting the faulty configured devices.

2.3.2 One device with different bit-rate

In this test-setup, nodes A, B, C and E use bit-rate 125kbit/s whilst only node D uses bit-rate 250kbit/s as shown in schematic 2. All nodes are pure CAN-nodes with no additional master device required for node operation.

Results: After running the automatic bit-rate detection, the test-software shows both used bit-rates.
Several standard CAN-test tools used for comparison show only the first found bit-rate depending on implementation of the bit-rate detection.

Using a canAnalyser set to the found bit-rates provides information on which nodes are using which bit-rate. This allows detecting the faulty configured device.

2.3.3 One device with drifting bit-rate

In this test-setup, nodes A, B, C, D and E use bit-rate 125kbit/s. Node D was modified in a way to achieve a wide-range drift of the bit-rate. All nodes are pure CAN-nodes with no additional master device required for node operation. The drifting was manually controlled during test-operation.

Results: After running the automatic bit-rate detection, the test-software shows bit-rate 125kbit/s. However, the error-rate shown in a simultaneously running canAnalyser was lower than expected.

The explanation for this is that the drift was not big enough to get close enough to other standard-bit rates – this was proven using an oscilloscope. Therefore, the test-software would need to use all bit-rates supported by the CAN-controller then canAnalyser set to all active bit-rates would show node D using several bit-rates.

3 Detecting the sources of error-frames

The CAN protocol is focused on providing robust communication independent from external influences. Therefore, CAN makes use of advanced error detection, error notification and error containment mechanisms which are included in the protocol-engine of each node.

The only way to get more information on the node which started the error-flag is using an oscilloscope. However, even with this, it is often not possible to identify the causing CAN-node.

If passive Error-flags are visible on the oscilloscope, then the node transmitting this current CAN-frame is destroying it. Therefore the CAN-identifier can be used to select the causing device. However, in certain cases this does not help, e.g. if the CAN-frame is a RTR frame.

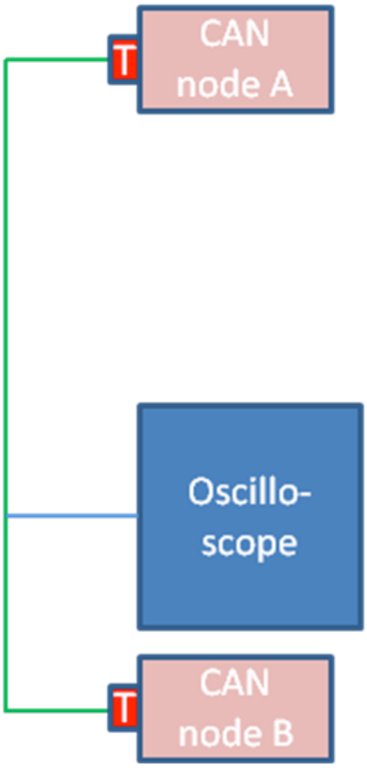
For active error-flags or in case of not possible detection using the above way, the always improving oscilloscopes can help detect the culprit. With oscilloscopes offering better bandwidth, higher sample rates and easy-to use mask-tests, it is also possible to detect the CAN-node causing the error by only the starting edge of the error flag.

The allocation of a message due to a single signal edge is only possible if the signals from the different nodes differ to a certain extend. The signal difference is due to:

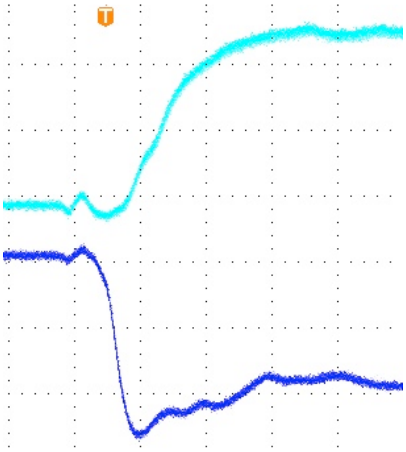
- Different layout and components used in devices, notably protection circuits have major impact on the signal form.
- Variances in components even in identical built nodes can cause the signal form to differ. Differences in resistors and capacitors lead to different signal levels (e.g. due to changing power-supply of CAN-transceiver, changing capacitance/impedance, etc.).
- Differences in voltage-supply and local EMI. If the power-supply of the node is affected and offers inconsistent voltage levels, this can have effects on the CAN-node (depending on node-design), same is true for disturbances that are on the voltage line and might affect devices via this way.
- Position on network cable also influence the signals, the signal form of distant devices differs from nodes close to the measurement device, even if the devices would have identical signal-forms if connected directly to the measurement device.
- Position regarding other CAN-nodes in a CAN-network also affects the signal-form significantly [3]. CAN-side local EMI effects do influence the signals significantly – and these EMI might also be due to specific CAN-nodes, e.g., high-power inverters.

3.1 Significance of signal characteristics

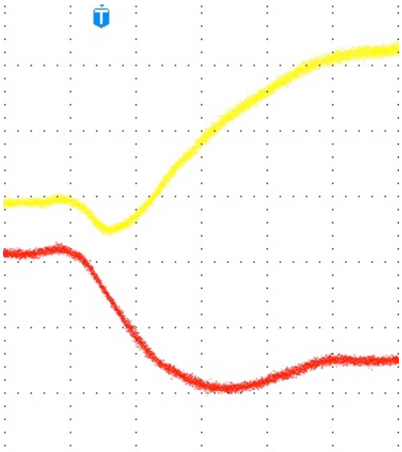
To show the difference in signals, two nodes of the same making in an optimized laboratory network (total cable-length 10 meters, minimum external signal distortions and distance to oscilloscope 1 meter for node B and 9 meters for node A) are measured. The test setup is done as shown schematic 3. It is easily possible to distinguish messages from different nodes by only one single signal-edge as shown in graphs 1 to 3. To achieve this view a good scope was used, with an external trigger on the recessive-dominant signal-edge. Each graph shows a timeline of 60ns at a bit-rate of 500 kBit/s.



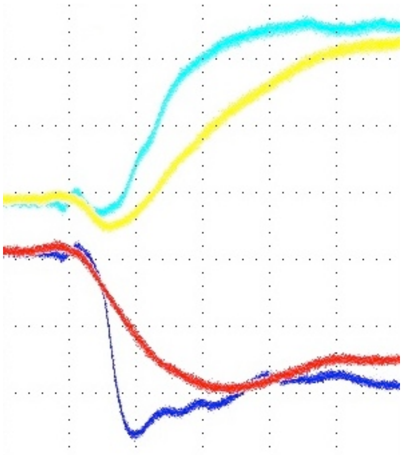
Schematic 3: Test-setup for node detection according to signal characteristics



Graph 1: Node 1 (purple: CAN-low; blue: CAN-high)



Graph 2: Node 2 (red: CAN-low; yellow: CAN-high)



Graph 3: Node 1 (purple: CAN-low; blue: CAN-high) and node 2 (red: CAN-low; yellow: CAN-high) in overlay

3.2 Generate signal-database

First, it is necessary to get each node's signal measured. Notably, the signal edge recessive-dominant is important. To get good results, the network should show the same behavior as during standard operation otherwise signals will look too different for good allocation to the different nodes.

When measuring, the Oscilloscope needs to be triggered to the specific messages from the different nodes or it is necessary to verify that only the specific node to be measured is transmitting. Note that detaching other nodes from the network is not good as this will also influence the CAN-signals.

Therefore a good oscilloscope with internal or external CAN-trigger capabilities should be used. Whether this signal is measured and stored by an oscilloscope or a PC-based tool with external sampling hardware is not relevant to the measurements. Either way, it is recommended that the used oscilloscope or sampling hardware should provide a bandwidth and sampling rate of >500 MHz. For basic CAN-Analysis a lower-performance Oscilloscope is suitable, but due to the fact that only a single edge needs to be analyzed in detail, a limited sampling performance will give poor results and it will be hard to identify the error-generating CAN-node. In addition, oscilloscopes with integrated Mask-generation and Mask-tests will ease the comparison of the different signals measured.

The easiest way to get this special signal-edge from all nodes is to measure all messages in normal operation mode for a certain time. The oscilloscope should decode the Identifier of the messages and store the signal information to generate a kind of “signal database” for the checked CAN-network.

With all nodes being measured, signal information can be stored in the oscilloscope’s memory.

This stored information now allows determining all messages transmitted by one single node by comparing the sampled signals. By verification of this message, signal-to-node assignment, the user can also check the quality of scan.

In addition, the questioned error-flag should be sampled using the same oscilloscope and same settings. Notably, the signal edge recessive-dominant is important.

Now this sample is to be used to generate a signal-mask, and by reloading the single-node signal samples, it is possible to determine the best fitting signal. As this best fitting signal is calculated from the similarity of the signal masks, the quality of this solution could be calculated in fitting percent.

A) High percent fitting

If the fitting percentage is high, then the sender of the error flag seems to be found.

B) Low percent fitting

If the fitting percentage is low or if the measurement system is not able to find a node that fits, then the following might be cause:

- The error flag was sent by several nodes at the same time due to detection of message data errors.
- Other physical effects cause a global CAN-failure which results in all nodes starting the error-flag.

In this case the transmitting node as well as the physical characteristics of the CAN-network need be examined in detail.

The available signal-samples and the complete sampled CAN-error message will also help to find the reason of the error. To examine the node transmitting the destroyed message either by checking the CAN-ID, or if the CAN-ID is destroyed or possibly used by several nodes (e.g. for RTR-frames) by using the sampled signal-edge of the data field for comparison with the already available “signal database”.

4 Detection of missing CAN nodes

Problems caused by missing CAN nodes can often be found in systems where single CAN nodes are not configured correctly or have been changed and the preset Identifier is not correct. Other reasons for missing devices include defective nodes, defective CAN-communication path of devices, broken CAN-cabling/connectors and other.

4.1 Missing nodes due to wrong CAN-ID

If CAN-nodes are misconfigured and are using the wrong CANopen node ID, several results are possible:

- A CAN system with two or more nodes transmitting data-messages using the same identifiers.
- A CAN system with one or more nodes transmitting messages with undefined identifiers.

If several nodes use the same identifiers and no system-master detects missing devices or defective boot-up messages, it is quite difficult to find this error. In this case it is possible to use the signal information of the messages as described in section “Detecting culprits of error-frames“. When the bus is scanned by the used measurement-tools, it is possible to identify two nodes that send messages with the identical Identifier as a data-message with a specific identifier will show significantly different signal-edge quality.

In addition, by transmitting confirmed messages (e.g. specific network management messages if available or an SDO check for serial number of a device) using a CAN-test tool, error message will appear on the bus as all nodes using this specific CAN-identifier will respond immediately and will transmit their data which should be different (e.g. for device-serial numbers). This will then result in error frames – using an oscilloscope the faulty set CAN-Identifier can be detected.

If only one or more nodes are using wrong identifiers which are not occupied in the system, the system will show undefined messages or nodes. Using a test-tool or the system controller scanning for network-devices will show missing nodes but will also show the undefined messages. In this case, the missing nodes are faulty set and can be detected directly and reconfigured.

4.2 Missing nodes due physical failure

If CAN-nodes are missing due to physical failures, CAN-system controller or standard CAN-test tools for logical and physical layer can be used. As the nodes are simply missing, the system-controllers and the CAN-test tools providing node-scan-functionality will show a list of nodes with the defective nodes missing. A simply comparison to system-manuals or previous measurements will show the defective nodes.

In addition, a physical network check might be useful as depending on failure cause other nodes or complete network segments might also be damaged , e.g., if node-failure is due to overvoltage or other global external influences.

5 Improvement of network robustness

With networks growing in length, number of nodes and increasing requirements on data-throughput, low cycle-times and the frequent requirement to keep systems modular and flexible the requirements on CAN-system designs get more and more complex. System designers need to adhere the CAN-network to the CAN-specifications to achieve reliable networks. There is also a tendency to deviate from the CAN-specifications to fulfill the requirements of system flexibility.

And the more complex a CAN-system gets, the more challenging it is to keep it within the CAN-specifications:

- Added physical bus-load due to number of nodes, connectors, cables, stub-lines, EMI…
- Increasing communication bus-load due to number of messages, error-frames...

From years of experience in testing CAN-systems showing communication problems, one good rule is to keep the network as simple as possible. This can also be achieved by splitting a complex network in several segments:

- Physical split: This is good if a network shows problems due to physical problems

- Logical split: This may help if the communication is at its limits

5.1 Split on the physical layer

By splitting a network on the physical layer, several objectives can be achieved:

- Increased EMI robustness by additional filtering, CAN-signal amplification and reduction of the cable-lengths which act as “antennas”
- Increased flexibility by enabling stubs, different signal levels and network media (e.g. glass-fiber)
- Increased number of nodes in a logical network, especially if nodes with extensive protection circuits are used

This physical splitting is done by using so-called CAN-repeaters [4]. These consist of simple logic and at least two CAN-transceivers for the required physical bus-attachment.

5.2 Split on the logical layer

By splitting a network on the logical layer, several objectives can be achieved:

- Increased network robustness reduced bus-load and optional filtering of errors and messages
- Extension of CAN-networks by integrating different bit-rates into one CAN-system or by interconnection of several CAN-systems via another communication path, e.g. Ethernet
- Combination of several CAN-messages into one CAN-message or modification of CAN-IDs and / or CAN-message data

This logical splitting is often done by using CAN-bridges or gateways that allow the interconnection to other systems like Ethernet, Bluetooth or Industrial Ethernet. Normally, these bridges or gateways also split the network on the physical level; therefore the benefits of repeaters are also valid here.

5.3 Fundamentals for use of topology components

The main issue of network topology components including CAN-repeaters, bridges and gateways is that these units add latency to the overall network. The latency is dependent on the complexity of a device –a simple CAN-repeater only adds a few nanoseconds, while a complex gateway might add several milliseconds. As long as the overall latency does not affect the application, these components will improve the system robustness.

In addition, topology components help to extend the capability of CAN-networks. However, these units cannot eliminate basic system-design flaws. To achieve good results the very basics like grounding, network-layout, shielding and power-supply that are part of the physical network design need to be observed. In addition, software design flaws or communication design errors can only be minimized but not extinguished using additional network components [5].

6 Conclusions

Even with highly evolved CAN-test tools and standard measurement systems offering dedicated but flexible test-assistance-software, there are several common problems in CAN-networks which can only be detected and solved by using the available equipment in innovative ways. With “real-life” applications, which might significantly differ from laboratory networks, it is often necessary to provide a bit more insight into network-details. Using available test-tools to limit the failure reason to a manageable amount is key to success. As for the service engineer, the perfect test tool that immediately identifies the exact failure is never available onsite when attempting to troubleshoot a CAN-system.

In addition, many problems can be eliminated by using special topology-components, as long as the main rules of system design are taken into account.

References

[1] Etschberger, K.: Controller Area Network (CAN) Grundlagen, Protokolle, Bausteine, Anwendungen. Dritte Auflage. Hanser Verlag, 2002

[2] ISO-IS 11898-1:2003, Controller area network - Part 1: Data link layer and physical signaling; ISO-IS-11898-2:2003, Controller area network - Part 2: High-speed medium access unit

[3] Steve Corrigan: Minimum distance between CAN nodes. CiA Newsletter 4/2011 pages 12ff

[4] Frank Pastors and others: Introduction CAN-Repeater, http://www.ixxat.com/introduction-repeater_en.html

[5] Thomas Waggershauser, Tobias Frey: CAN network analysis seminar, http://www.ixxat.com/can-analyzing-seminar_en.html

Tobias Frey
IXXAT Automation GmbH
Leibnizstr. 15, D-88250 Weingarten
Phone +49 (751) 561 46 – 0
Phone +49 (751) 561 46 – 29
frey@ixxat.de
www.ixxat.de

Thomas Waggershauser
IXXAT Automation GmbH
Leibnizstr. 15, D-88250 Weingarten
Phone +49 (751) 561 46 – 0
Phone +49 (751) 561 46 – 29
waggershauser@ixxat.de
www.ixxat.de