Eye diagram analysis for CAN FD

The eye diagram is an analysis method for evaluating the signal quality of transmission networks. Eye diagram analysis can help finding corrupting influences on CAN networks.

CAN networks can transmit information bit by bit from sender to partly distant receivers. The information transfer can, however, be easily corrupted by the network topology, cable length between the participants, line and terminating resistors as well as external electric influences.

Both eye diagram analysis and serial bit mask analysis allow these influences to be identified and corrected early during the configuration phase of the CAN network.

Due to the higher and flexible bit-rate of the data phase, CAN FD is considerably more noise-sensitive than Classical CAN, which has a fixed bit-rate only for the entire frame. The transmission of CAN FD frames always begins with the lower bit-rate of the arbitration phase. A switch to the higher bit-rate of the data phase is made at the sampling point of the bit-rate switch (BRS) bit. At the sampling point of the cyclic redundancy check (CRC) delimiter bit, the bit-rate is switched back to the lower bit-rate. The shorter duration of the bits during the faster data phase has a negative effect on the signal quality, which can best be analyzed with an eye diagram. The signal quality is highly dependent on:
- Design and complexity of the bus topology e.g. line or star topology,
- Adversely selected cable routing and/or bus termination,
- Susceptibility of the system to higher transmission rates typical in vehicle networks,
- Cable properties, such as their impedances or shielding.

Creation of a classical eye diagram

During the network design, a configuration for the phase segments of the bit as well as their sampling points is created separately for the arbitration phase and the data phase. According to the CAN protocol, one bit is time-divided into four segments: a synchronization segment, a segment for compensating time delays, and two phase segments, which compensate the phase errors of the bit edges. The phase segments are often designated TSEG1 and TSEG2. The TSEG 1 summarizes the first phase segment and the compensation segment. Figure 1 shows the time division of a bit into the mentioned segments. The sampling point lies between the phase segments. TSEG2, on the other hand, corresponds to the phase segment which begins after the sampling point. Since the grid of the superimposed bits is displayed directly in the eye diagram, it is immediately apparent how well the superimposed bits fit into this grid.

The bit edges should ideally lie within the synchronization segment. The algorithm used here works like a real CAN controller. Due to the time delay of the bit edges, caused by the CAN transceiver delay and the jitter of the control units, CAN controllers must resynchronize at the transition from the recessive to the dominant bus level on the receiver side. This is a prerequisite for detecting the logic level of a bit at the set sampling point. With the adjustment of the phase segments, the robustness of the synchronization mechanism can be influenced. These...
settings are usually checked using an eye diagram, which allows the user to visualize whether his controller settings are practical and meaningful. Due to the fact that CAN FD has two different bit-rates, it is advisable to create separate eye diagrams for the arbitration phase and the data phase.

With the software tools CANoe and CANalyzer from Vector, the user is able to configure a CAN network and record voltage signals using the Option Scope. After measuring, the user can perform an eye diagram analysis, which analyzes all received frames bit by bit and superimposes them graphically in a fixed time window. In this example, the time window is a percentage of the bit duration (Figure 1). The displayed bit segments show the configured controller settings. Possible deviations in the individual bits can be quickly identified in this view. With a good controller setting, the rising edges of all bits lie in the synchronization segment. If the bit signals also reach their dominant and recessive voltage levels uniformly, i.e. without overshoot, a robust bus topology and a correctly selected bus termination can be assumed. The diagram has an “eye” due to the fact that all the bits on the x-axis are normalized to the theoretical bit-width (reciprocal of the bit-rate), while the voltage values of the bits are plotted on the y-axis. For the described case, the eye would be wide open (Figure 2). In the opposite case, the eye would be closed, which is an indication of errors in the network structure (Figure 3). The data phase is analyzed in Figures 2 and 3 with a sampling point set to 70 % and a data-rate of 2000 kbit/s.

Further refinement of analysis criteria

In order to narrow down possible sources of error, it is recommended to create an eye diagram from different aspects. For this purpose, various filter options are provided:
- Frame type, for example CAN or CAN FD,
- CAN channel number,
- Control unit name,
- Defined bit sequences.

---

**CANlink® wireless**

Economic Wireless CAN Communication via Bluetooth and WLAN

**AS A WIRELESS GATEWAY**
Multi-functional interface for various applications.

**CAN-BRIDGE FUNCTION**
Connect CAN networks wirelessly.

**AS A SERVICE INTERFACE**
Use our dashboard software to display data individually on a computer, smartphone or tablet.

**COMPACT AND ROBUST**
Low-cost wireless CAN interface in a compact housing for easy assembly.
To further refine the criteria for evaluating the eye diagram, it is helpful to create a predefined theoretical eye in the form of a bit mask. For this purpose, the user generates a bit mask as a freely definable polygon, thus defining the "good" area, which may not be crossed by a voltage signal of any bit. It is useful to define a separate bit mask for the arbitration phase and for the data phase. Figure 4 shows an eye diagram with a bit mask, in which some bits violate the defined mask. In the background, the configured segments of the CAN controller are shown.

Serial bitmask analysis

So far, the classical eye diagram, which represents the overlapping of individual bits, has been considered. An alternative visualization is the serial bit analysis. In principle, both analysis procedures are performed identically. Only in the case of the serial bitmask analysis, the bits are displayed in the order sampled by an oscilloscope. Again, it is possible to define bit masks, which are displayed for every dominant and recessive bit. The advantage over the classic eye diagram is that bit errors are assigned directly bit by bit. It is also possible to analyze only part of the bit stream. The configuration possibilities already discussed for the eye diagram can also be applied to the serial bitmask analysis. Figure 5 shows all the bits of a defined analysis area. Each bit has a bit mask with the red mask indicating a violation in the first bit.

Automating the analysis process

Both analysis methods can be automated with the Vector product CANoe. To do this, the user must first define his test cases. For example, one test case is defined to analyze the data phase of all CAN FD frames and another test case for only the arbitration phase. For each test case, a specific bit mask can be used as a test criterion. If the bit mask is violated by the bit signal, the test case result will be negative. Each test case is automatically recorded, evaluated, and stored in a test report, so that the user can understand why a test case failed.

Conclusion

The methods described here for the analysis of a CAN/CAN FD network help to quickly identify and fix design faults or negative external influences. Eye diagram analysis with CANalyzer and CANoe is also available for other bus systems such as Flexray. With CANoe, the user also has the option to perform tests in a reproducible and automated manner via CAPL test sequences. Due to the high degree of automation and with suitable tools, such tests can be repeated with minimal effort.

An alternative approach for detecting protocol errors of a control unit is the analysis of different bit sequences of a frame. With this method, a distinction can for example be made between the bits which are transmitted during the acknowledge phase by all control units and those of a particular control unit.

Author

Mirko Donatzer
Vector Informatik
mirko.donatzer@vector.com
www.vector.com
Driver, Software, and Programming Interfaces

Every PC interface from PEAK-System is delivered with a wide range of drivers, software, and programming interfaces. The scope of supply includes:

- CAN FD interface drivers for Windows 10, 8.1, 7 and Linux
- PCAN-View: Windows software for monitoring CAN and CAN FD busses
- PCAN-Basic API for developing applications with CAN and CAN FD connection for Windows (32/64 bit)
- PCAN-PassThru for using applications that are based on Pass-Thru (SAE J2534) with interfaces from PEAK-System
- Programming interfaces for standardized protocols from the automotive sector like:
  - PCAN-CCP API for the communication with ECUs according to the CAN Calibration Protocol
  - PCAN-XCP API for communication with ECUs according to the Universal Measurement and Calibration Protocol (CAN FD support since version 2)
  - PCAN-ISO-TP API for the transfer of data packages according to ISO-TP (ISO 15765-2)
  - PCAN-UDS API for the communication with ECUs according to UDS (ISO 14229-1)
  - PCAN-OBD-2 API for vehicle diagnostics according to OBD-2 (ISO 15765-4)

PCAN-USB X6

6-Channel CAN FD Interface for USB 2.0

The new PCAN-USB X6 is designed for the use in test benches with hardware-in-the-loop (HIL) simulations, in product line testing, and for working with multiple CAN or CAN FD busses.

- Adapter for USB 2.0 (compatible to USB 1.1 and USB 3.0)
- 6 High-speed CAN channels (ISO 11898-2)
- Complies with CAN specifications 2.0 A/B and FD
- CAN FD support for ISO and Non-ISO standards switchable
- CAN FD bit rates for the data field up to 12 Mbit/s
- CAN bit rates from 25 kbit/s up to 1 Mbit/s
- CAN bus connection via D-Sub, 9-pin (in accordance with CiA 303-1)
- Time stamp resolution 1 µs
- CAN termination at the CAN connection can be activated through solder jumpers, separately for each CAN channel
- Galvanic isolation up to 300 V, separately for each CAN channel on request
- High-speed USB 2.0 downstream port
- Voltage supply from 8 to 30 V
- Extended operating temperature range from -40 to 85 °C
- Measurement of bus load including error frames and overload frames on the physical bus
- Induced error generation for incoming and outgoing CAN messages

Six-Channel CAN FD Interface
16th international
CAN Conference (iCC)

Historical City Hall, Nuremberg (DE), March 7 - 8, 2017

Register before Feb. 06, 2017 to make sure you get the early bird rate.

Session I: Keynotes
Holger Zeltwanger (CiA): A personal review and an outlook
Christian Schlegel (HMS): The role of CAN in the age of Ethernet and IOT

Session II: Application
Bernhard Floeth (Opel): Using an enhanced condensed device configuration file format for CANopen boot-loading and/or device testing
Ana Antunes (Instituto Politecnico de Setúbal): CAN-based modules for A320 flying simulators
Jeremy Lebon (Vives): Remote controllers skid-steer

Session III: Semiconductor
Tony Adamson (NXP): Managing the transition to robust CAN FD
Magnus-Maria Hell (Infineon): The new wake-up pattern for a robust system
Roland Lieder (Renesas): Gateway processor evolution in automotive networks

Session IV: Time synchronization
Florian Hartwich (Bosch): CAN frame time-stamping – supporting AUTOSAR time base synchronization
Hauke Webermann (esd): CAN send and receive with hardware time-stamping
Marc Boyer (Onera): Reducing CAN latencies by use of weak synchronization between stations

Session V: CAN FD
Dr. Marc Schreiner (Daimler): Introduction of CAN FD into the next generation of vehicle E/E architecture
Christoph Wosnitza (C&S Group): Interoperability challenges for CAN-FD/PN transceivers: Lessons learned from CAN high-speed interoperability tests
Uwe Koppe (MicroControl): CAN driver API - migration from classic CAN to CAN FD

Session VI: System design
Torsten Gedenk (Emtas): Use cases and advantages of the XML device description format for CANopen FD devices
Manfred Brill (Schneider Electric): FDT, OPC UA, and CANopen - a compelling combination
Peter Decker (Vector): Automated analysis for vehicle communication

Session VII: CANopen
José A. Pulido (Doga): CANopen, a key factor in motor control systems for seeding applications
Andrew Ayre (Embedded Systems Academy): Automated trace analysis for testing of CANopen devices
Klaus Rupprecht (Sys Tec): CANopen safety development solutions

Session VIII: Security and safety
Bernd Elend (NXP): Security enhancing CAN transceivers
Olaf Pfeiffer (Embedded Systems Academy): Scalable security for CAN, CANopen, and other CAN protocols
Dr. Heikki Saha (TK Engineering): Systematic approach to maintain safety performance in the service of CANopen system

For more details, please, contact the CiA office at headquarters@can-cia.org

www.can-cia.org