Comparing CAN, CAN FD, and Ethernet

This analysis compares Classical CAN, CAN FD, and and Ethernet communication with focus on a decentralized battery management system.

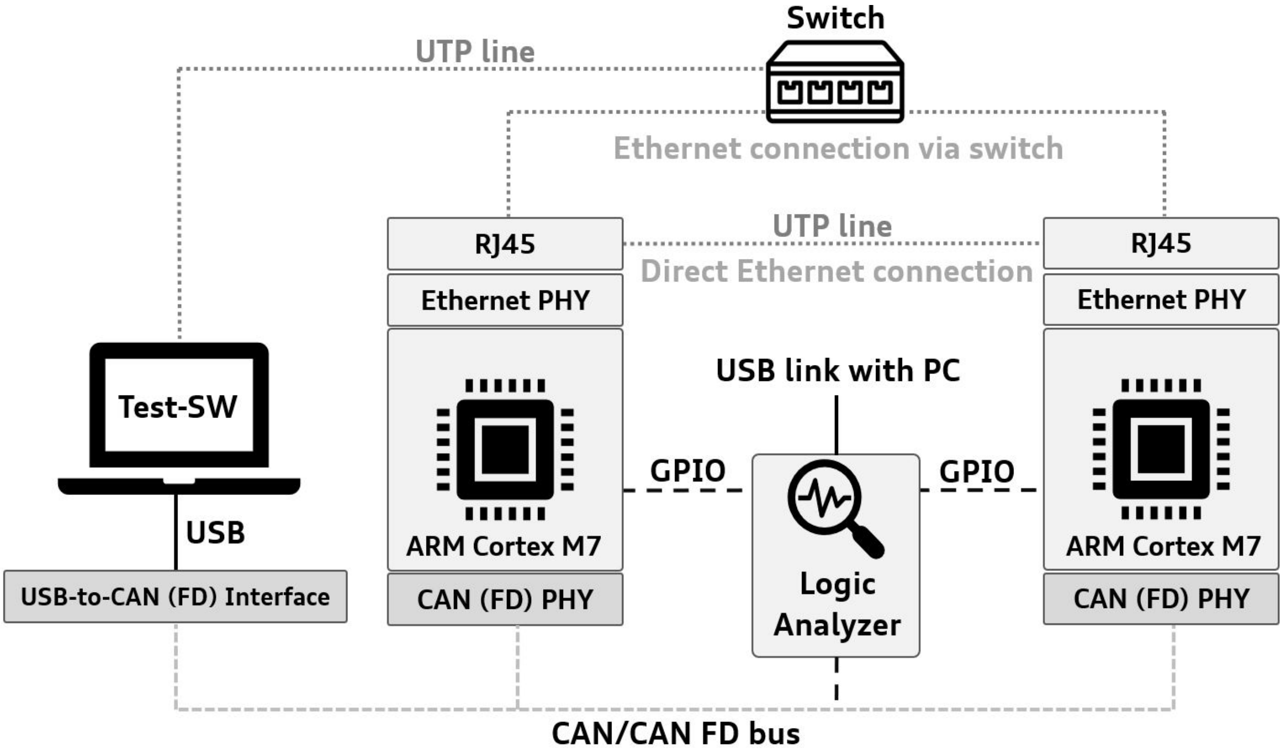

Figure 1: Test setup for comparative analysis of CAN, CAN FD, and Ethernet (Source: OTH Regensburg)

Networked control systems such as battery management systems, smart grids, or vehicular systems, consist of sensors, actuators, and controllers linked via a common communication line. The system control can be distributed among several nodes thus building a decentralized control system enabling a communication-based coordination of the control tasks. Nodes can be added or removed even after an initial installation, which offers the required flexibility for different applications.

A communication network can cause unpredictable delays which can affect the system control. For CAN communication, e.g., only the latency of the highest prioritized data frame can be determined. For the remaining data frames, the delay depends on the situation on the network and is not predictable. These network-induced delays may increase the time jitter of the control loop, which consequently can lead to instability. In addition, possible data or information loss or data manipulation, endanger the control coordination. Therefore, the data rate and the reliability of the underlying communication network are key factors of the networked control system. In addition, the processor load caused ▷

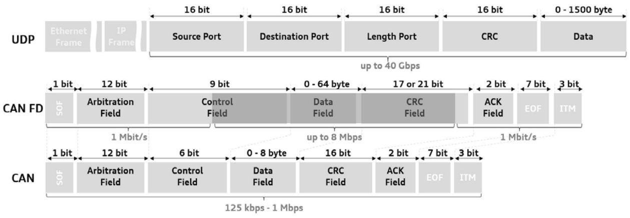

Figure 2: Structure and size of the UDP, CAN FD, and CAN frames and the maximum transmission rates (Source: OTH Regensburg)

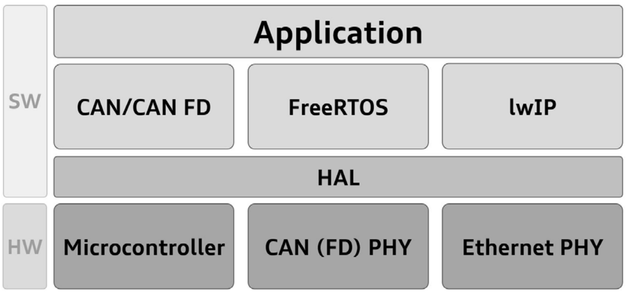

Figure 3: Software components and the interface between them and the hardware (Source: OTH Regensburg)

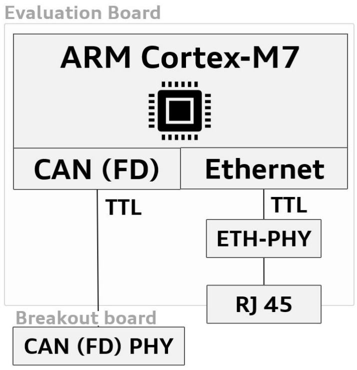

Figure 4: Hardware components used to implement CAN, CAN FD, and Ethernet communication (Source: OTH Regensburg)

by the communication is relevant as it affects the calculation of system states and the setting of control parameters.

Energy required for the communication network influences the system efficiency in a respective application. Energy efficiency, small latency, and reliability are the key features for the communication networks.

The decentralized battery management system (DBMS) is an example application for networked controlled systems [1].

The DBMS consists of renewable energy generators, a variable number of different batteries, and varying loads. For battery control, battery-specific condition parameters are communicated regularly, which allows to adjust the required charging/discharging power of the batteries. Additionally, current, voltage, and temperature values are measured in millisecond intervals. Each data packet comprising few bytes contains a time stamp and is sent to all participating nodes forming the basis for the collaborative system control. Within the DBMS, it is therefore required to regularly send frames with few data bytes quickly, reliably, and without errors in order to achieve system-wide data consistency.

Examined networks

Classical CAN is widely used in distributed embedded systems. Its limited communication bandwidth (up to 1 Mbit/s) and payload (user data) size (up to 8 byte) restrict the applicability in increasingly complex electronic systems (Figure 2). CAN FD comes with higher data-transfer rate (up to 8 Mbit/s) and larger payload size (up to 64 byte).

Ethernet offers data rates up to the Gbit/s-range via a twisted pair (TP). In embedded systems, 100BASE-TX with 100 Mbit/s is most common, since higher data rates also increase processor and memory requirements. It offers a payload size of up to 1 500 byte and provides low latency but is also more complex compared to Classical CAN and CAN FD. Ethernet-based solutions offer various protocols. In this article the user datagram protocol (UDP) is considered as it is a relatively simple compared to the transmission control protocol (TCP). It avoids confirmation of correct frame reception and thus supports unicast, multicast, and broadcast communication. The size of the UDP message header (8 byte) is significantly reduced compared to the up to 60-byte ▷

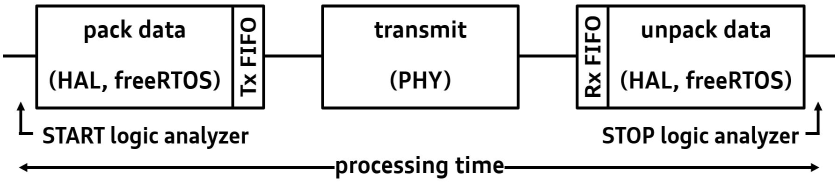

Figure 5: Measurement of frame processing time: The start pin of the logic analyzer is set before the frame is created and the stop pin is set after the frame is completely received (Source: OTH Regensburg)

header of the TCP. In the scope of this article, the UDP/IP protocol stack is examined and is only one of several options.

For the evaluation of the energy efficiency, the maximum power consumption of the communication technologies is measured. To evaluate the transmission reliability, the bit error rates (BER) and the residual error rates (RER) are determined. Furthermore, the frame processing time and the processor load of each communication technology are measured as both influence the system control. Since networked control systems may consist of numerous participants, the behavior of the communication technologies under high network load is investigated.

In this article, theoretical comparisons have been made for a substantiated evaluation of the communication technologies. Data sheets, existing literature, and simulations are referenced. In addition, comparative tests within a hardware test setup enabled practical, realistic results, especially for transmission speed, error susceptibility, and behavior under high frame load.

Test environment setup

The basis for the comparative analysis is the software implementation (Figure 3) of the communication technologies and a test environment (Figure 1). The test setup consists of software and hardware components.

Software implementation: For a direct comparative analysis, the three communication technologies are utilized and measured individually. In addition, combined operation enables direct comparisons under identical conditions. The real-time operating system FreeRTOS [2] is applied in order to support multi-threading (Figure 3). Classical CAN and CAN FD use the same hardware module on the micro-controller and are therefore combined into one thread. To switch between CAN and CAN FD, only a few bits of the control field in the data frame are modified. In the following, CAN (FD) describes both CAN and CAN FD. Ethernet communication requires a software design that supports the protocols from the physical to the transport layer. The TCP/IP stack lightweight IP (lwIP) was developed for embedded systems and offers advantages in efficiency and program scope [3], [4]. The stack supports several application program interfaces (APIs). It provides functions to initialize the UDP module and also handles the Internet Protocol (IP), e.g., setting the IP address, subnet mask, or gateway mask. The UDP thread is created after initialization and manages the sending and receiving of messages. The initialization of the periphery is performed by functions of the hardware abstraction layer (HAL).

Hardware components: For the hardware realization, the evaluation board STM32H743ZI [5] and an external CAN transceiver [6] are used (Figure 4). The evaluation board integrates an ARM Cortex-M7 processor and provides two CAN (FD) modules and one Ethernet module. Furthermore, there is an Ethernet PHY and a RJ45 connector on the board. For CAN (FD) only the logic modules are available, therefore an external CAN transceiver is added.

Test setup: Two micro-controllers with corresponding interfaces and a test PC are connected (Figure 1). The test PC monitors the network and performs the tests. The micro-controllers are connected to the CAN (FD) network via CAN (FD) transceivers. For a connection between the test PC and the CAN (FD) network a CAN-to-USB interface [7] is used. The test PC and the micro-controller integrate an Ethernet PHY and are connected to the Ethernet network via an unshielded twisted pair (UTP) line. The Ethernet network is connected via a switch. For the determination of the Ethernet frame processing time, the micro-controllers are also directly connected to each other via the RJ45 interface. For the frame-processing time measurement, a logic analyzer is used, which is connected to the test PC via USB. Evaluation criteria and acquisition methods

For the comparative analysis, the following evaluation criteria are determined:
- Frame processing time
- Processor workload
- Energy consumption
- Error rate
- Rx-Fifo load

All comparative measurements are performed with a transmission speed of 500 kbit/s for CAN; 500 kbit/s or 1 Mbit/s as well as 4 Mbit/s for CAN FD; and 100 Mbit/s for Ethernet.

Frame processing time: For the comparison of the frame processing time, the same number of user data is transmitted at an identical 1,5-m link between a sender and a receiver. The frame processing time is measured with a logic port that monitors the start and stop pin. It consists mainly of the transmission time and additionally of the computing time for packing/unpacking the frame. The start pin is set to high immediately before the send command, while the stop pin is set after the frame has been completely received (Figure 5).

Processor workload: For system control, operating parameters are recorded, system states are calculated, and the corresponding data are managed by the micro-controller. Communication, in particular the preparation of frames, sending and receiving of frames, also places a load on the micro-controller. For this reason, the processor workload allocated to communication, is determined. The measurement is divided into workload caused by the initialization and the sending or receiving of frames.

▷

References

[1] A. Reindl, H. Meier, M. Niemetz, “Scalable, Decentralized Battery Management System Based on Self-organizing Nodes”, in: Brinkmann A., Karl W., Lankes S., Tomforde S., Pionteck T., Trinitis C. (eds) Architecture of Computing Systems – ARCS 2020. Lecture Notes in Computer Science, vol 12155. Springer, Cham, https://doi.org/10.1007/978-3-030-52794-5 13, 2020.

[2] FreeRTOS Kernel Developer DOcs. https://www.freertos.org/features.html

[3] lwIP 2.1.0 — lightweight IP Stack. http://www.nongnu.org/lwip/2 1 x/index.html

[4] STMicroelectronics: UM1713 User manual — Developing applications on STM32Cube with LwIP TCP/IP stack. 2017. https://www.st.com/resource/en/user manual/dm00103685-developing-applicationson-\\stm32cube-with-lwip-tcpip-stack-stmicroelectronics.pdf

[5] Datasheet, STMicroeletronics: STM32H742xI/G STM32H743xI/G, https://www.st.com/resource/en/datasheet/stm32h743bi.pdf, 2019.

[6] Datasheet, Microchip: MCP2561/2FD, http://ww1.microchip.com/downloads/en/devicedoc/20005167c.pdf, 2014.pdf, KNF Kongress, 2001.

Energy consumption: The energy consumption is determined for each communication technology separately using the Power Consumption Calculator (PCC) from STMicroelectronics. The calculation is based on the data sheet information. The CAN (FD) and Ethernet sequences are analyzed. As Classical CAN and CAN FD use identical peripherals, they are examined collectively in the CAN (FD) sequence. Energy consumption in run, idle, and sleep mode is determined.

Error rate: Correct data transmission is the basis for the effective network control. The considered error rate focuses on errors occurring during transmission or processing of frames. A distinction is made between detected (handled) errors and undetected (not handled) errors. The latter may have damaging consequences. To determine the probability of occurrence, the bit error rate (BER) and the residual error rate (RER) are defined. The BER is the number of bit errors in relation to the total number of bits sent (Equation 1).

$$\mathrm{BER} = \frac{\#\text{Bit errors}}{\#\text{Bits}_{\text{total}}}$$

The RER is the number of undetected, erroneous frames in relation to the total number of frames, whereby the residual package error (RPE) is the number of undetected frames with errors (Equation 2).

$$\mathrm{RER} = \frac{\mathrm{RPE}}{\#\text{Messages}_{\text{total}}}$$

For comparison of error rates, existing literature, and data sheets are referenced.

Rx-Fifo load: A high transmission rate with a large frame volume can lead to information loss. The frames are first stored in the corresponding Rx-Fifos and subsequently processed. If the frames are processed too slowly with constantly new incoming frames, an Rx-Fifo overflow may occur and entries that have not yet been processed are overwritten. The load of the Rx-Fifo is investigated in case of a high frame load. Therefore, the test PC generates a correspondingly high quantity of numbered test frames. ◀

To be continued: This article is split in two parts. In the next issue of the CAN Newsletter magazine, you can read Part 2. This article was originally presented as a paper at the Embedded World Conference 2021 Digital.

Authors

Andrea Reindl, Daniel Wetzel,,Norbert Balbierer, Hans Meier, Michael Niemetz
Faculty of Electrical Engineering and Information Technology - Ostbayerische Technische Hochschule Regensburg
Contact the authors via andrea.reindl@st.oth.regensburg.de
www.oth-regensburg.de
Sangyoung Park
Smart Mobility Systems - Technical University of Berlin
sangyoung.park@tu-berlin.de
www.sms.tu-berlin.de

17th international CAN Conference
iCC

Registration is open

June 14 to 17, 2021

14 15 16 17

Opening

12:30 Holger Zeltwanger CAN in Automation

Session I: Physical layer

Chairperson Holger Zeltwanger CAN in Automation

13:00 **The physical layer in the CAN XL world**
Magnus-Maria Hell Infineon Technologies

13:30 **Characterizing the physical layer of CAN FD**
Johnnie Hancock Keysight Technologies

Session II: CAN XL data link layer

Chairperson Reiner Zitzmann CAN in Automation

14:00 **Introducing CAN XL into CAN networks**
Florian Hartwich Robert Bosch

14:30 **CAN XL error detection capabilities**
Dr. Arthur Mutter Robert Bosch

15:00 **CRC error detection for CAN XL**
Dr. Christian Senger University of Stuttgart

CiA CAN Coffee (C^3)

15:30 **Chat with the speakers**

16:30 **End of day 1**

14 15 16 17

CiA webinars

9:00 **Interoperability of CAN XL & CAN FD** **Register**
Dr. Arthur Mutter Robert Bosch

10:00 **CAN FD a Vitamin Shot for SAE J1939** **Register**
Peter Fellmeth Vector Informatik

11:00 **Migration from classic CANopen to CANopen FD** **Register**
Alexander Philipp emotas embedded communication
Torsten Gedenk emotas embedded communication

Session V: CAN FD lower layers

Chairperson Dr. Frank Deicke Fraunhofer IPMS

12:30 **CAN signal improvement and designing 5-Mbit/s networks**
Tony Adamson NXP Semiconductors Netherlands

13:00 **A leightweight communication bus based on CAN FD for data exchange with small monolithic actuators and sensors**
Fred Rennig ST Microelectronics

13:30 **Improved CAN-driver**
Kent Lennartsson Kvaser

Session VI: Engineering

Chairperson Kent Lennartsson Kvaser

14:00 **Designing a CAN-to-TSN Ethernet gateway**
Nikos Zervas CAST

14:30 **Automated workflow for generation of CANopen system monitoring graphical user interfaces (GUI)**
Dr. Heikki Saha TK Engineering

15:00 **Benchmarking of CAN systems using the physical layer – car, truck, and marine case studies**
Dr. Christopher Quigley Warwick Control Technologies

CiA CAN Coffee (C^3)

15:30 **Chat with the speakers**

16:30 **End of day 3**

From Classical CAN via CAN FD to CAN XL

14 15 16 17

CiA open-house technical group meetings

9:00	**IG J1939** CiA 510, SAE J1939 series, ISO 11992 series, ISO 16844 series, etc.	**SIG contrast media injector** CiA 425 series	**SIG (electrical) drives** CiA 402 series	**SIG special-car add-on devices** CiA 447 series
10:30	**IG profiles** Co-ordination of CiA profiles specifications	**SIG truck gateway** CiA 413 series, DIN 4630, DIN 14704	**IG CANopen FD** CiA 13XX series	**SIG CAN FD Light** CiA 604

Keynote

Chairperson Holger Zeltwanger CAN in Automation

12:30 **Future of CAN from prospective of an OEM**
Carsten Schanze Volkswagen

Session III: CANopen testing

Chairperson Uwe Koppe MicroControl

13:15 **A new approach for simulating and testing of CANopen devices**
Mark Schwager Vector Informatik

13:45 **CANopen FD conformance testing – today and tomorrow**
Oskar Kaplun CAN in Automation

Session IV: CANopen FD

Chairperson Christian Schlegel Christian Schlegel Consulting

14:15 **A simplified classic CANopen-to-CANopen FD migration path using smart bridges**
Christian Keydel Embedded Systems Academy

14:45 **A theoretical approach for node-ID negotiation in CANopen networks**
Alexander Philipp emotas embedded communication

15:15 **CANopen FD devices identification via new layer setting services (LSS)**
Yao Yao CAN in Automation

CiA CAN Coffee (C^3)

15:45 **Chat with the speakers**

CiA open-house technical group meeting

16:45 **SIG subsea**
CiA 443

18:00 **End of day 2**

14 15 16 17

CiA webinars

9:00 **CAN physical layer options** **Register**
Magnus Maria Hell Infineon

10:00 **CAN XL and PWM coding** **Register**
Matthias Muth NXP

11:00 **CAN FD topology simulation** **Register**
Patrick Isensee c&s Group

Session VII: Security

Chairperson Torsten Gedenk emotas embedded communication

12:30 **Embedded security recap**
Thilo Schumann CAN in Automation

13:00 **Achieving multi-level CAN (FD) security by complementing available technologies**
Prof. Dr. Axel Sikora Hochschule Offenburg
Olaf Pfeiffer Embedded Systems Academy

13:30 **CAN XL made secure**
Donjetë Elshani Infineon
Vivin Richards Infineon
Harald Zweck Infineon

Session VIII: CAN XL higher layers

Chairperson Dr. Arthur Mutter Rober Bosch

14:00 **IP concepts on CAN XL**
Peter Decker Vector

14:30 **Multi-PDU concept for heterogeneous backbone networks**
Christian Schlegel Christian Schlegel Consulting

15:00 **Standardized layer-management options for CAN-based networks**
Holger Zeltwanger CAN in Automation

CiA CAN Coffee (C^3)

15:30 **Chat with the speakers & Closing**

16:30 **End of conference**

Sponsors

emotas embedded communication
esd www.esd.eu
GEMAC
PEAK System
VECTOR