CAN XL for real-time control and efficient file transfer

Many engineers are wondering why they should use CAN XL, when 10Base-T1S Ethernet is around the corner. This article provides some detailed background information, which helps to answer this.

CAN XL and Ethernet specify data link layer frames that can be sent over a physical media, so in that sense, there is no difference between them. Everybody dreams of implementing a sole solution to solve all problems: why not just use Ethernet to handle all in-vehicle networks? The reality is that every solution has its pros and cons. The difference between CAN XL, Ethernet or any other layer-2 approach is in the details:

- How is access to the media secured?
- How is the start of the frame detected and synchronized to the receiver?
- How is the bit rate defined?
- How is it possible to find the start of the user data?
- How is it possible to find the end of the user data?
- How is the frame transmission protected from errors?

Classical CAN was designed to provide autonomous communication that could function with limited support from higher-layer protocols. CAN has never claimed to be generic: it was developed to handle real-time information in a simple, safe, and robust way at low cost. To optimize the communication for this real-time task, it was necessary to accept certain constraints and extensions beyond the generic OSI layer model:

- Predictable access to the network; arbitration in CAN.
- Short delays to transmit important data; priority in CAN.
- Short frames to limit delays; 8-byte data frames in classical CAN.
- All nodes evaluate the same information at the same time; information is broadcasted in CAN networks.

The last demand (last bullet point) is that all consumers receive the data frame at the same instance in time. The simple solution to this problem is to send and receive all information on one single common media, and in order to deliver that functionality, CAN is limited to simplex communication. With full duplex communication, any node can send at any time because the communication media is always free and available. 100Base-T1 solves this by using point-to-point (P2P) communication with one single media whereby the transmitter can receive a signal by first subtracting the sent energy. This requires relatively complex filtering in the transceiver and as such, is overly complicated when adding just one more unit onto this common ▷

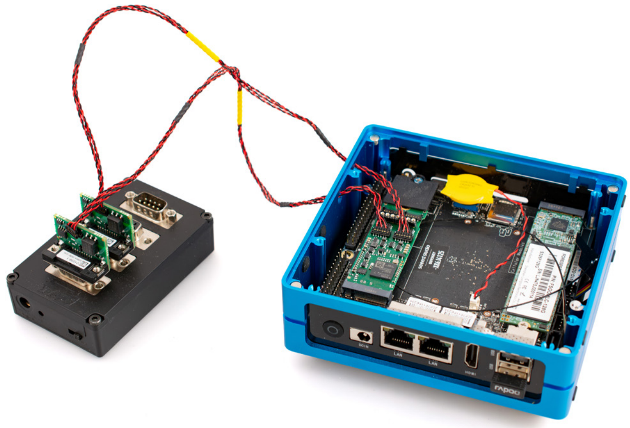

Figure 1: Linux computer with Kvaser's M.2 interface and the two CAN interfaces connected to a short CAN network at the left (Source: Kvaser)

media. There is always a delay in the path between the sampling of the information and the use of the information by the consumer. If this delay has a secured limit, it is possible to use P2P and switches, as in Ethernet today, which spread the data to all consumers. Such solutions are more complex, and the switches add costs. CAN provides a common database with low timing jitter for all installed devices intrinsically.

The least but not last demand in the bullet list, to get frames with limited extensions in time, can be solved in two ways: have frames with few bits and keep the bit-length short (high bit rate). The most common solution to this in the computer world is to increase the clock-rate (bit rate). The cable itself can handle 500 Mbit/s, but cable quality, connectors, and drop-lines disrupt continuity and cause ringing, which limits the usable bit rate. Added to this, CAN XL uses arbitration for bus access, which demands a lower bit rate during the first phase of the package transfer. At the outset, 10Base2 Ethernet was also based on a multidrop network, but to reduce cable complexity, all solutions with more than 10 Mbit/s use P2P in combination with switches and bridges. To lower cable complexity and cost of CAN, it is necessary to use a bit rate as low as possible. CAN XL makes it possible to select a bit rate from 0,125 Mbit/s to 20 Mbit/s, enabling you to achieve the highest possible performance that your cable budget allows. When it comes to overhead, there is no big difference between 10Base-T1S and CAN XL, but the slower arbitration of CAN XL will increase the length in time of the frame transmission. When longer cables are used in CAN XL this same concept requires the use of a slower bit rate during the arbitration phase.

The second demand is how to get fast access to the communication media. The best solution would be to employ a media where anybody can start sending without considering other senders. This problem is partly solved by 100Base-T1, which provides full duplex between the node and the switch, but this just moves the access problem to the switch. If two nodes start to send a frame to one receiving node, the switch must store one of the frames, because it cannot send two frames concurrently. The fundamental Ethernet solution to this problem is CSMA/CD (Carrier Sense Multiple Access/Collision Detection), which means "if the media is free, anybody can start sending and if a collision is detected, stop sending and wait a random time before returning to CSMA/CD". This works fine for the office LAN and Internet because rare and intermittent delays are no problem as long as they don't significantly lengthen download times. In a real-time system, all delays must be limited to a defined length in time over the lifetime. To solve this problem, it is necessary to define rules to ensure that all units transmit in an order that guarantees that all frames are received within a certain time limit. One such set of rules is defined in the TSN (Time Sensitive Network) standard. CAN XL solves this using CSMA/CR (Collision Resolution), with the only difference being that the collision is not destructive. In this way, the package with the highest priority wins the media and all other senders will receive that frame and recycle the CSMA/CR process.

The first demand is how to resolve a collision without harming the communication package. As described in the second bullet point, this is solved by replacing CSMA/CD as used in Ethernet with CSMA/CR. No transmitter starts sending a frame, if there is already a frame on the communication media (CS). If the media is free however, anybody can start sending a frame. There is therefore the probability that two or more units start sending a data frame at the very same time.

CAN solves collisions one bit at a time so that if a unit is sending a recessive bit "1" and it reads back a dominant bit "0", it will back off and stop sending bits. This is what's known as bit-wise arbitration: several bits into the CAN data frame, just one sender is left and that winning unit completes transmission of the complete CAN data frame. This nice feature does not come for free because the CAN bit requires time for the signal to stabilize before all nodes read back and check the sampled bit-value. The shortest CAN bit is equal to the longest delay, multiplied by two, plus some phase margin for clock variations and noise. At 1 Mbit/s, the time budget in the bit is 1000 ns. To protect the sample point from random phase noise, it is recommended that a 10-% phase margin is used, which reduces the available bit time to 800 ns. A 0,5-% oscillator tolerance in the nodes could cause a phase offset as large as 0,5 %/bit x 10 bit x 2 = 10 %, which could further reduce available bit time to ▷

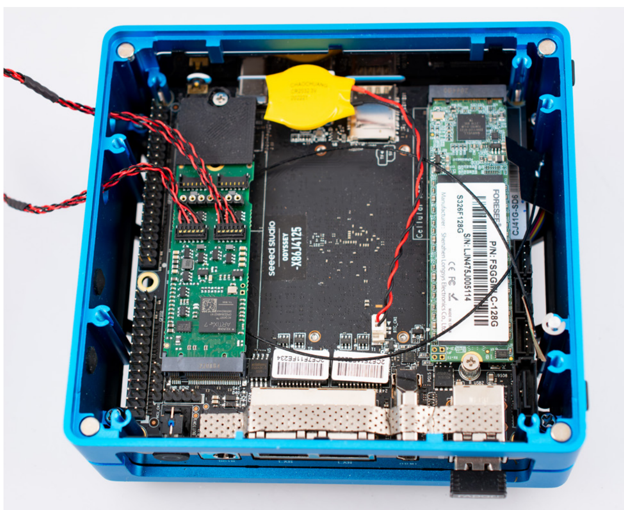

Figure 2: The M.2 interface including the CAN XL IP core to the left; the second M.2 slot to the right holds the SSD (Source: Kvaser)

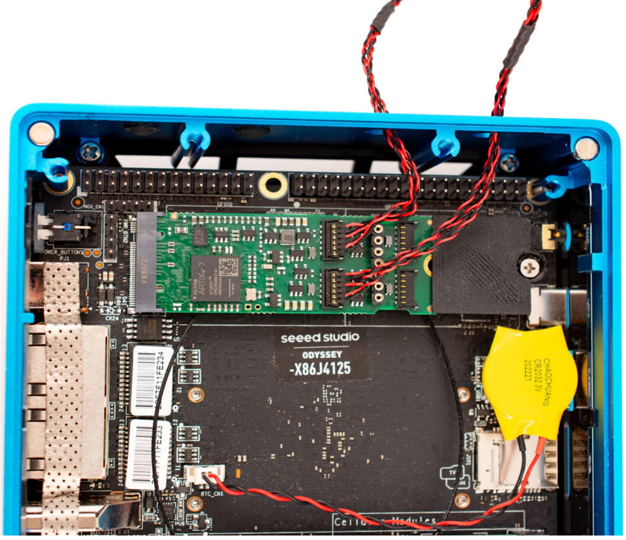

Figure 3: The CAN transceivers are located on a separate module and red/black wires are the connection between the M.2 module and the CAN transceiver (Source: Kvaser)

600 ns. The CAN transceivers typically have a delay below 200 ns, reducing the available bit time to 200 ns. With a cable delay at 5 ns/m, the cable must be limited too; 200 ns / (2 x 5 ns/m) = 20 m. Modern transceivers could reduce delay to 50 ns, which combined with good oscillators would reduce the total phase margin to 10 % of the CAN bit length. However, to increase the bit rate, it is necessary to reduce delays more and the only major option is to reduce cable length.

To combine higher bit rate and longer cables in CAN networks, a different approach is needed. As seen from the description above, a low bit rate is only necessary during arbitration. Once one sender has won, the bit rate is only limited by the CAN-transceiver slew-rate and the cable layout (impedance variations). Cleverly, CAN XL starts out with a lower bit rate that matches the cable length (delays) during the arbitration phase and once there is only one sender, it switches to a higher bit rate. During the CiA-organized plugfest in Detroit we ran CAN XL at 20 Mbit/s over 30 m. It should be possible to achieve speeds of more than 20 Mbit/s with a carefully designed cable layout.

Where Ethernet wins over!

The main reason for using Ethernet is when communicating with a normal computer over LAN and Internet, because it is possible to use usual TCP/IP without any modification. If you only need to transfer information over a serial communication, Ethernet is the obvious choice. Windows, Linux, and many other OS (operating systems) have software support such as TCP/IP and UDP included, providing a fast track for software development.

Real-time video transmission necessitates the use of high bandwidth communication channels (0,1 Gbit/s to 5 Gbit/s). MIPI is used today, which is essentially normal Serdes LVDS (low-voltage differential signaling) and was designed to connect the camera to the computer inside a housing.

Ethernet provides a more robust technology that is designed to survive in a relatively rough environment. If Ethernet is used for cameras and other devices that ▷

Kvaser and CAN XL

Kvaser has added CAN XL functionality to its existing CAN/CAN-FD IP core. This IP core is mostly used in FPGA-based products, but it is also used for the MCU for at least one major player. It forms the base for Kvaser's CAN XL developments. A challenging aspect of the design is the increased package size, which demands CAN buffers with 8 byte, 64 byte, and up to 2048 byte. The other big concern is how to efficiently handle a burst of short frames with few bytes mixed with long frames with 2048 byte.

The first step was to confirm that the IP still matched the Bosch CAN/CAN FD reference model, after which CAN XL functionality could be added to the existing IP core. Compared to adding CAN FD to classical CAN, it was relatively easy to add CAN XL functionality as it includes just more CRC blocks, bit-timing registers, PWM (pulse-width modulation), and some additional slightly more complex functionality. Despite the absence of a reference model for CAN XL, Kvaser experienced no problems in communicating with all the hardware that showed up at the CiA plugfest in Detroit. The CAN XL development board by C&S does not have fault injection, so it was not possible to test all fault conditions error handling. The error handling is relatively complex and to have everything 100 % correct without a reference model is not to be expected. Without a reference model it will be complicated to secure a compatible CAN XL IP.

need high bandwidth, it is tempting to also use it for low-bandwidth signals that today run over CAN. Even though audio and video are both time critical, they have some major differences with most other real-time signaling:

- The signaling is asymmetric, with high bandwidth from the camera and almost none to the camera. The same is true for the video display but in the opposite direction.
- The bandwidth is constant and with a fixed frame rate.

It would be possible to expand the time schedule used for video to also include other real-time signals. The problem is that in a car there is also a huge number of parameters that occur very rarely but still must be treated as real-time data (e.g. headlight switching, turn light). This is the main reason, that there have seen solutions such as TT-CAN and Flexray. Ethernet as such is not intended for real-time communication (even if it is fast) and to provide predictable signaling in Ethernet TSN is placed on top of the Ethernet to provide scheduled traffic with a guaranteed maximum delay for all critical data.

To decrease latency there are Ethernet transceivers designed to support PHY-level Collision Avoidance (PLCA), which is a fixed order transmission with dynamic slot lengths. An overview and some simulations can be found in the paper entitled 'PLCA Clause 148 Overview'. As described in this paper, PLCA is very efficient for frames with more than 60 byte, but the latency for 8 nodes is 4 ms to 8 ms, compared to 0,25 ms for CAN at 1 Mbit/s for any number of nodes for the package with the highest priority. For each package with a lower priority, 0,125 ms should be added to give us 15 high-priority CAN frames sent within the first 2 ms. This equates to 120 byte sent from 15 nodes, during which time, two 10Base-T1S nodes have transferred 2500 byte. However, it should be remembered that another 6 ms are needed before all eight nodes have had a chance to send information.

The case for CAN/CAN XL

Serial communication in a real-time control system is more complicated than just downloading a file. CAN is optimized for real-time control even with low demand on component tolerance, limited software and use of low-cost cabling. When CAN was invented in 1983, the MCUs (micro-controller units) have had 4-KiB ROMs, 128-byte RAMs, and 1,5-% oscillator tolerance. To run Ethernet, 50-ppm (parts per million) oscillators are needed. TCP/IP software size is about 6 KiB for the micro-IPs and 23 KiB for lightweight-IPs, while at least a 2-KiB RAM allows for small Ethernet frames only. The main justification for using LIN and CAN FD Light is to reduce the hardware cost further by making components without any software. The use of a product without or with limited software also makes it simpler to secure functional safety standards like ISO 26262.

With the CAN XL technology it is possible to optimize the bandwidth and delays to the installed cable layout to within the range of 0,125 Mbit/s to 20 Mbit/s. Thus, CAN XL provides an excellent compromise between small software, flexibility, cost, and performance. ◀

Author

Kent Lennartsson
Kvaser
kent@kvaser.com
www.kvaser.com