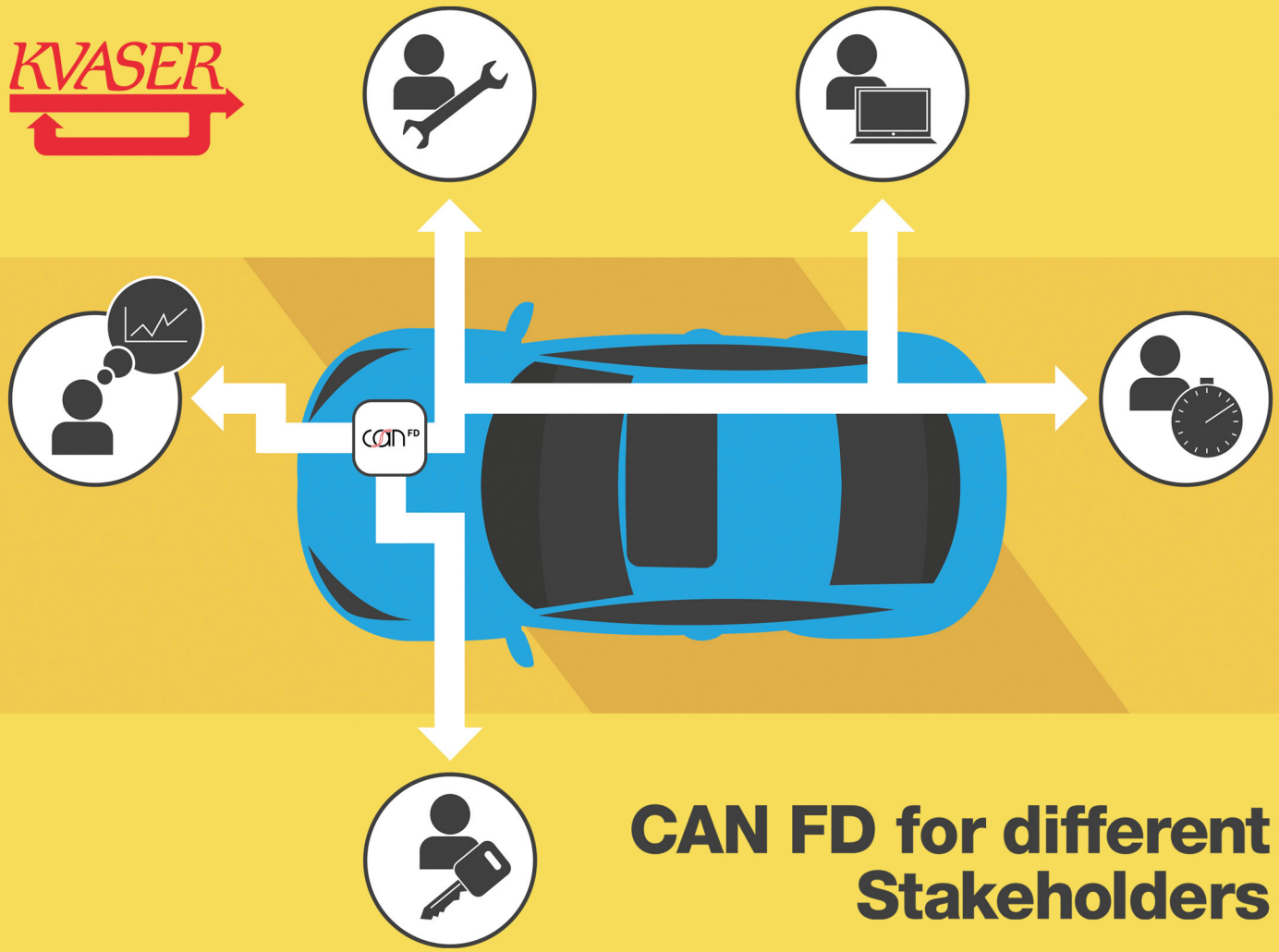
KVASER



CAN FD for different Stakeholders

Table of Contents

CAN FD considerations for different stakeholders

This is an overview of CAN FD as it relates to different stakeholders. The CAN FD knowledge you need depends very much on your position within the organization. This paper outlines the impact of CAN FD on seven different types of stakeholder:

1. The management
2. The CAN software application engineer
3. The real-time expert
4. The CAN driver software engineer
5. The data communication security engineer
6. The data communication integrity engineer
7. The physical layer engineer

The management

The primary rationale for a move from Classical CAN to CAN FD is to increase performance by taking incremental steps. CAN FD communication can be achieved by simply replacing the CAN controllers, with no impact on EMC and environmental performance. And, if for any reason the CAN FD communication proves problematic, it is easy to switch back to Classical CAN until a solution has been found.

If you have a working CAN solution, there is no justification for an immediate switch to CAN FD. The additional functionality in your future products will require more software, more communication and more data to be sent through the system, where CAN FD's higher bandwidth may prove useful. For this reason, it would be worthwhile using CAN FD controllers in new products, even if your intention is to keep using the Classical CAN format. This will ensure your products can cope with the higher bandwidth of CAN FD should it be required.

A key challenge for CAN FD is that all products with Classical CAN controllers need to be replaced with CAN FD controllers before CAN FD frames can be used on the CAN bus. A solution to this is to use two CAN buses, one with Classical CAN frames and one with CAN FD frames, with a router or filter between the two. Kvaser has described a filter that can be placed between the CAN driver and the CAN controller or between two CAN buses that prevents CAN FD frames from reaching the Classical CAN controllers.

The CAN software application engineer

The only big software difference between CAN and CAN FD is the ability to send more than 8 bytes of data. In CAN FD, up to 64 bytes of data can be sent, but messages are limited to a few different lengths as defined by the DLC and shown in table 1.

Table 1 DLC to number of data converter:																
DLC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Max data byte	0	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64

In many cases it is possible to write much simpler code if the data structures are larger than 8 bytes because all data can be sent in just one package, instead of being fragmented into several messages.

Note that if your data structures do not exactly match the data bytes in the CAN frame, it is a good idea to fill the structure with dummy data. If the data structure is not filled with dummy data, random data stored in the CAN buffer will be sent in the CAN frame. One suggestion is to fill up the unused data bytes with 0xCC or 0x55, because this produces plenty of edges that in turn help the receivers to stay in sync with the bits in the CAN frame.

You will get some resistance from the real-time experts when requesting CAN frames above 8 bytes.

CAN FD provides a higher data bandwidth, but without modification of the cables, connectors, filters and CAN drivers, you should not expect more than 2 to 4 times an increase in bandwidth. To limit the time a CAN frame can occupy the CAN bus, you could expect some restrictions when using CAN frames above 16 bytes. If there is low real-time demand, it is possible to use any data length without any restrictions. A typical scenario is when the system is downloading a software update.

The CAN real-time expert

CAN FD could change your system performance dramatically, both for better and for worse.

CAN FD will change the real time performance in three ways:

1. CAN FD allows data to be sent at a higher bit-rate. This makes all the CAN frames much shorter in time and thus, lowers the response time.
2. It is possible to increase the data load above 8 byte, up to 64 byte. In this way, CAN frames are produced that are almost 8 times longer than Classical CAN frames. The downside is that the increased data load causes longer delays between CAN frames, though this can be compensated for by using a higher bit-rate.
3. There are more overhead bits in CAN FD so that even with less than 8 bytes of data, CAN FD frames are longer than Classical CAN frames.

To keep real-time performance unchanged, it is necessary to tightly control package length. Data packages longer than 8 byte can significantly increase latency (up to 8 times per CAN frame when using 64 byte CAN frames). On the other hand, it is possible to lower latency by increasing the bit-rate to make all CAN frames much shorter. By increasing bit-rate, even a 64 byte CAN FD-frame can be made shorter than a Classical-CAN-frame with 8 bytes. However, this increase in bit-rate doesn't come for free because the degree to which it is possible to increase the bit-rate depends on the design of the physical layer.

Your department will be squeezed between software engineers demanding more data in the CAN frames without increasing the latency, and the electrical engineers trying to make the physical layer run at the higher bit-rate. To simplify the software application, in most cases it is easier to send 64 byte packages, to reduce the numbers of interrupts and data handlings. This will demand rescheduling of the CAN frames or increasing the bit rate to reduce the length of the CAN FD frame.

In most cases it won't be a problem to double the bit-rate. When increasing bit-rate by more than that, it will be necessary to test the physical layer to ensure that EMC performance remains the same.

The CAN driver software engineer

There are three design challenges that you need to consider carefully:

1. The increase of data bytes from 8 to 64 byte.
2. Fitting the data-structures within the restricted number of data-lengths supported in CAN FD.
3. Handling two different bit-rates.

Table 2 DLC to number of data converter:

DLC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Max data byte	0	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64

In CAN driver software, typically a fixed data structure of 8 bytes is used to store 0 to 8 bytes of data because the space saved by using dynamic data is very little. To have a fixed data-structure with 64 bytes will be a significant loss of data space if most CAN-frames are less than 8 bytes. The size of the data structure also makes it worthwhile to keep the data structure in one location and only transfer a pointer to this storage.

It is also necessary to define whether the CAN driver software is allowed to use data-lengths that do not match the exact number of data bytes corresponding to the DLC coding. As seen from table 2, there is a direct correlation between the DLC value and the number of data bytes in the CAN frame in the 0 to 8 range. For the DLC values between the 9 to 15 range, it is necessary to convert this number into the actual number of bytes that will fit the CAN frame. This matching of data to the data length in the CAN FD frame is something that must be defined.

The last issue within the driver is to secure the bit-rate settings. Firstly, CAN FD makes it possible to have more time quanta. In the old standard, the number of Time Quanta (TQ) in a nominal bit was in the range of 4 to 25 TQ. In CAN FD, you can use any number between 4 and 161 TQ for the nominal bits. The data-bits at the high bit-rate continue to use a lower TQ number i.e. 4 to 25 TQ. Even when the number of parameters remains the same, the size of each parameter is increased

for the nominal bit. To handle both bit-rates it is necessary to set both bit-parameters: one for the nominal bit and one for the data-bits. It is to be expected that different CAN controllers have different sized parameters and even different interpretations of the parameters. Some CAN controllers will have two bit-rate prescalers, one for each bit-rate, but others will use the same prescaler for both nominal- and data-bits.

CAN FD demands stricter configuration of the nominal bit. Please consult your Physical Layer expert for the correct configuration of both the nominal and data-bits

The data communication security engineer

In most cases, data integrity and security are combined, but I have decided to split them in this discussion. Even if they have much in common, their purpose is different. For me, data security protects data from being manipulated within the system as CAN bus is used for machine control, not for sending secret documents or information on financial transactions. Whilst there is information within the CAN communication that could be sensitive, such as GPS-position, speed, time etc., that information is simple to obtain by other means and protection of the CAN bus will not solve the problem. The main challenge for system security is to protect the CAN communication from intruders sending wrong information into the system or taking control of the system. Systems are at their most vulnerable when electronic units are updated with new software. So firstly we would like to protect the software from reverse engineering and secondly from being replaced by malicious software.

CAN FD allows for more data in each CAN frame, which simplifies the transfer of files and long keys. However, CAN FD does not add any other security features that aren't present on Classical CAN.

The data communication integrity engineer

In most cases, data integrity and security are combined, but I have decided to split them in this discussion. Even if they have much in common, their purpose is different. For me, data integrity refers to the task of protecting the data from being modified by disturbance between the sender and the receiver. This is normally of limited concern in real-time control systems, because even if you get a bad value, it will be replaced with a new value within a few milliseconds.

The CRC check sum is done in a different way in CAN FD and as a result, CAN FD has better protection against undetected errors in the data. To get undetected errors demands CAN communication with a high Error-Frame content. The best way to protect the communication is to improve the physical layer to remove any occurrence of Error-Frames.

If it is not possible to reduce the rate of Error-frames to a safe level, CAN FD will decrease the risk of undetected data errors.

The physical layer engineer

For physical layer engineers, CAN FD is identical to Classical CAN as long as there are no demands to increase the bit rate in the data section.

If there is demand for an increase to bit rate, life gets more complicated. It's best to look at the CAN bus signal with an oscilloscope. If the signal is nice and square, there's no reason not to increase the bit-rate by 2 to 4 times that of Classical CAN. Notably, it is necessary to perform measurements at different locations along the CAN bus wires. The signal can be nice and square in one location and have a serious ring at another location. The arbitration is relatively demanding and if it works with a certain nominal bit-rate, it's possible to increase the data-rate by 2 to 3 times the arbitration bit-rate without any problems. Beyond 4 Mbit/s, the CAN drivers in use today won't perform correctly because the dominant signal is 'sticky'. If the CAN driver sends 5 dominant bits in a row, it will stick to the dominant level and the following recessive bit will be shortened.

There are three limiting factors to the bit-rate;

1. The need to have continuous impedance throughout the CAN wires.
2. CAN drivers with fast slew-rate.
3. CAN driver with symmetric bits.

Factoring in Impedance

It is easy to obtain a cable with continuous impedance. As long as a cable does not change its physical dimension in the cross-section and uses isolation with a homogeneous dielectric constant, there will be no change in its impedance. The problem is that the connectors will cause an impendence mismatch at the end of the cable. This is solved in USB and Ethernet by using dedicated connectors that have been designed to match the cables. When using higher bit-rates as in CAN FD, every cable split, connector and drop line will pose a challenge because of impedance variations that cause signal reflections. These combined energy reflections will result in oscillation at the edge of the bit.

The CAN physical wire driver

CAN drivers are available that are suitable for higher bit-rates. Those CAN drivers will work up to 10 Mbit/s in the lab. The bit-rate achievable in a real installation over the expected temperature range requires validation by test in a real installation under all working conditions.