CAN-FD protocol and its impacts on CANopen

Holger Zeltwanger

Bosch has introduced the CAN-FD protocol. This CAN backwards compatible data link layer protocol overcomes two CAN protocol limitations: The data-rate limit of 1 Mbit/s and the 8-byte limit of the data-field length. Using the CANopen application layer on a CAN-FD network would allow faster transmissions as well as longer Process Data Objects (PDO) and Emergency messages (EMCY). Using longer data-fields would also increase the throughput for software downloads, for example. But this would require new SDO protocols.

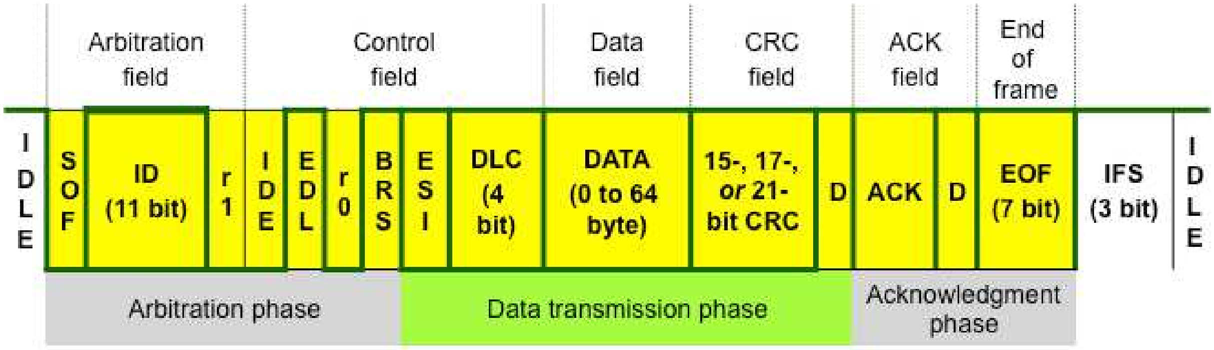
The CAN-FD protocol as proposed by Bosch is backwards compatible to the CAN protocol. This means you can run both protocols on the very same network, if all connected nodes understand the CAN-FD frames. The CAN-FD protocol allows increasing the transmission rate, when only one node is allowed to send dominant state bits. This is true after the arbitration phase and before the Acknowledgement phase. Optionally the data field is longer than 8 byte – up to 64 byte.

The CAN-FD data frame looks similar as the CAN data frame, but introduces some new bits. If the EDL (extended data length) bit following the IDE (ID extension) bit is recessive, the data frame is interpreted as CAN-FD message. After the BRS (bit rate switch) bit, the transmission rate is optionally increased. At the 13th iCC (international CAN Conference) Bosch and Vector demonstrated a CAN-FD prototype network running at 15 Mbit/s. Depending on the network topology (e.g. for star topologies) this bit-rate can't be achieved. A CiA working group is going to make some research on this and will publish some physical layer recommendations for different network topologies. In order to make the CAN-FD devices interoperable, it is also necessary to standardize the bit-rates in the data transmission phase.

The ESI (error state indicator) bit informs the receivers, in which bus error state the transmitter is (Error active or Error passive). This is a new feature helpful for the system design. This information is not available in normal CAN data frames. The Error passive state is already critical from the system point of view, because network-wide data consistency is not more provided, when one node is in Error passive state.

The DLC (data length code) has been extended. The not used bit-combinations are now used to indicate longer data-fields. Depending on the length of the data-field, the CAN-FD protocol uses different CRC polynomials. For data-fields up to 8 byte, the known 15-bit CRC sequence is used. For length between 9 and 16 byte the 17-bit CRC-sequence is mandatory, and for 17-byte and longer data-fields the 21-bit CRC-sequence has to be used. All CRC polynomials provide a Hamming distance of six, meaning up to five randomly distributed bit-failures will be indicate.

The rest of the frame is the very same as the CAN data frame. The Acknowledgement slot bit is already transmitted with the same (lower) bit-rate as in the arbitration phase. By the way, remote frames are not supported at all in the CAN-FD protocol!



SOF = Start of frame (bit is always of dominant state)
ID = Identifier (frame priority *and* content indication)
RTR = Remote transmission request (dominant, if data frame)
IDE = ID extension (dominant for base frame format)
EDL = Extended data length (recessive, if data field is longer than 8 byte)
r0/1 = reserved bit
BRS = Bit rate switch (recessive, if switched to alternate bit-rate)
ESI = Error state indicator (recessive, if transmitting node is in error passive state)
DLC = Data length code (indicates the length of the following data field)
CRC = Cyclic redundancy check (15-bit, 17-bit, or 21-bit)
D = Delimiter of CRC/ACK field (bit is always of recessive state)
ACK = Acknowledgment slot (correctly receiving node sends a dominant bit)
EOF = End of frame (all bits are always of recessive state)
IFS = Inter-frame space (the first two bits are always of recessive state)

Figure 1: CAN-FD base data frame supports a data-field length of up to 64 byte

The CAN-FD networks can use the same physical network as normal CAN networks. This includes transceiver, connectors, and cables. Of course, the network length still limits the bit-rate in the arbitration and the acknowledgement phase. The higher speed during the data transmission phase is not limited by the network length.

Reference	DLC 3	DLC 2	DLC 1	DLC 0	No of bytes
ISO 11898-1	0	0	0	0	0
	0	0	0	1	1
	0	0	1	0	2
	0	0	1	1	3
	0	1	0	0	4
	0	1	0	1	5
	0	1	1	0	6
	0	1	1	1	7
	1	0	0	0	8
CAN-FD	1	0	0	1	12
	1	0	1	0	16
	1	0	1	1	20
	1	1	0	0	24
	1	1	0	1	32
	1	1	1	0	48
	1	1	1	1	64

Figure 2: DLC coding

The factor between the short bit-time in the data transmission phase and the data-rate in the arbitration/ACK phase decides how much the frames are speeded up. This factor has two limits. The first is the speed of the transceivers: Bits that are too short cannot be decoded. The second is the time resolution of the CAN synchronization mechanism: After switching to the short bit time, a phase error of one time quantum in the standard bit time needs to be compensated.

In CAN-FD, two bit-rates need to be configured. The first is the very same as for the normal CAN communication used in the arbitration and ACK phase. The second bit-timing parameters are used for the data transmission phase. Both, the length of the t_q and the number of t_q in one bit-time may be different in the two configurations. The two configurations may be identical, but the bit time in the data transmission phase may not be longer than in the arbitration/ACK phase. The two bits where the switch happens are of intermediate

length, since the configurations are switched at the sample point.
Current CAN transceivers may have, according to ISO 11898-2/5, a loop delay (from the CAN_Tx pin to the CAN_Rx pin) of up to 255 ns. Since transmitters are required to check for errors in their transmitted bits, this would set a lower limit for the bit time in the data transmission phase, if the check needs to be done at the bit's sample point.
Measurements at Bosch have shown that existing CAN transceivers are able to transmit and receive bits that are shorter than their loop delay. In this case the check for bit errors needs to be delayed until the bit value, which is transmitted at the CAN_Tx output is looped back to the CAN_Rx input. This is the purpose of the optional CAN-FD Transceiver Delay Compensation mechanism. Receivers do not need this mechanism. Transmitters apply it in the data transmission phase.
The point-in-time, when the looped back bit value is checked is named the secondary sample point (SSP). The actual loop delay is not a static value, it depends apart from transceiver chip parameters mainly on the operating temperature.
The CAN-FD protocol controller is able to perform a delay measurement to find the optimum position for the SSP. Within each CAN-FD frame, the transmitter measures the delay between the data transmitted at the CAN_Tx output and the data received at the CAN_Rx input. The measurement is performed, when the arbitration is decided but before the bit rate is switched, at the edge from the EDL bit to the r0 bit. A counter measures the delay (in system clock periods). This measurement starts at the beginning of the bit r0 at CAN_Tx and stops when the edge is seen at CAN_Rx. The result is a node-specific value. It does not depend on signal propagation times on the CAN bus line. A configurable offset is added to the measured delay value to place the SSP into the middle of the bits seen at CAN_Rx.
Bosch has achieved with its FPGA implementation of the CAN-FD protocol and normal CAN high-speed transceiver chips (e.g. NXP TJA1040) in a 42-m bus-line topology network with seven nodes a maximum speed of 500 kbit/s in the arbitration/ACK phase and a maximum bit-rate of 15 Mbit/s in the data transmission phase. The test system runs in laboratory environment at room temperature.

Longer PDOs

One of the limits of todays CANopen networks, is the length of Process Data Objects (PDO). They are mapped to a single CAN data frame and can't be segmented. In some application, it is necessary to send more than 8 byte of process data, which belong together. The work-around is to transmit several synchronized PDOs and reassemble them on the application level. This may be done by using unique transmission numbers. In particular, in complex motion control applications, it is required to send multiple target values for position in one shot. Another example is the mode-specific interpretation of PDOs as specified in the CiA 452 motion control profile for PLCopen Motion. Using CAN-FD data frames providing longer data-fields would allow transmitting the control-word, the mode-command, and the necessary target values for position and velocity in one single PDO. This would simplify device implementations as well as system designs.
For this purpose the PDO mapping parameter needs not to be extended, if only a byte-wise mapping is allowed. Todays PDO mapping parameters allow already to map up to 64 process data. Of course, bit-wise mapping is limited to an 8-byte payload.

CiA 401 compliant I/O can benefit from a larger PDO payload. While for digital I/O in most cases it will be sufficient to have 8 byte corresponding to say 64 I/O signals, for analog I/O modules this is not the case.
The second to fourth PDO mappings according to CiA 401 contain four 2-byte analog values. In industrial applications the number of analog signals on a single modular CANopen devices typically is higher. Such modular devices will benefit from the larger payload. Assuming only a 32-byte PDO payload, would allow to transmit in a single PDO up to 16 analog channels instead of 12 values using three PDOs today. If the devices support dynamic mapping, it will even be possible to have 64 digital I/O signals and 12 analog channels in one PDO freely configurable. Assuming 4-byte float values, the benefit is even bigger.

New SDO protocols

There are two ways to improve SDO communication. One is to use the higher bit-rate to increase the throughput, especially if software is downloaded or domains of diagnostic are uploaded. Introducing new SDO services with 64-byte segments would not only increase the throughput, but also to address array or records with a single expedited (limited to a 63-byte payload) or normal SDO access.
On the other hand, SDO protocols should be as easy as possible to be implemented. That is why all SDO messages have today an 8-byte data-length. In order to keep the implementation simple, new optional protocols need to be specified for those applications, which really benefit from the new opportunities. For SDO block transfer, mainly used for software download purposes, the higher payload may be an advantage and accelerate the down- and upload of large amount of data. When specifying SDO protocols with larger payloads, it should be paid attention to the unused bytes. Otherwise it may happen that the benefit of a higher transmission rate is eaten by wasting bytes in the data-field.
The other CANopen communication protocols will not benefit from higher payloads. Except the Emergency message, which may contain more detailed diagnostic information. Also the SRDO (Safety-Related Data Objects) protocol may benefit from longer data-fields as the PDOs are doing. The increase of throughput is also an advantage for safety-related communication. Todays limitation of 64 SRDOs may be overcome and the increased payload may also allow to transmit more safety-related process data in a single network segment.

Conclusion

Adapting the CANopen communication profile to the CAN-FD data link layer needs surprisingly little effort. Adopting the available protocol stacks will not be too difficult. Besides the additional configuration of the second bit-rate parameters, it requires a smart handling of all the frame buffers with the maximum increased frame size of up to 32 or 64 byte, which still can be important in embedded devices with small memory resources.
CiA plans to release the advanced CiA 301 specification, when CAN-FD controllers have been implemented in micro-controllers by more than one chipmaker. ST Microelectronics, Infineon, Freescale, and Fujitsu have indicating their interest to implement the new protocol in the next generation of micro-controllers.

Artykuł na zasadzie przedruku – nie podlega recenzowaniu.