CAN Newsletter Online

CAN NEWSLETTER MAGAZINE

CANopen FD use cases

CANopen FD allows efficient and simplified embedded networking. It comes with interesting features added to classic CANopen networking by keeping the robustness and scalability. This article examines why and when choosing CANopen FD.

(Source: Adobe Stock/CiA)

The complete article is published in the June issue of the CAN Newsletter magazine 2021. This is just an excerpt.

Modern applications demand network design flexibility. Even the end user shall have the option to modify the application or network setup. Thus, a dynamic handling of communication coherences between the devices in the network is needed. Additionally, increased requirements for safety-relevant or secure communication demand a high data throughput; not only during configuration and maintenance, but also during system runtime. Furthermore, requirements derived from condition monitoring or IoT (Internet of Things) applications, demand an increased communication bandwidth on embedded level. The new features, added to CANopen FD, allow very efficient embedded networking, today and tomorrow; as illustrated by means of the following examples.

Control of multi-axis systems

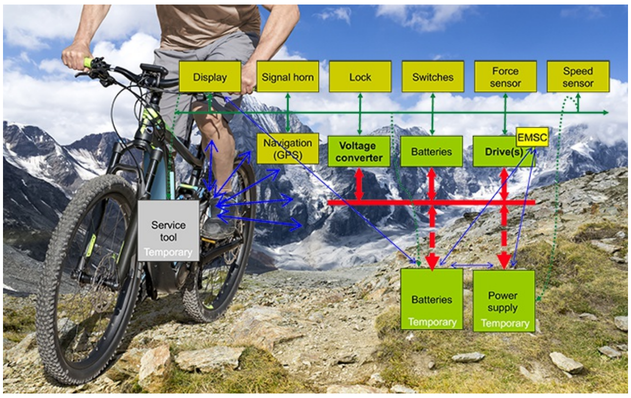
In several applications the synchronization of various tasks is requested. For example, in multi-axis systems some axis shall start with their movements at the very same time. In Classical CAN-based networking, this task can be solved. For starting synchronized movements, the axis that shall operate synchronously for example can use a synchronized time base or a specific global event as trigger. In CANopen FD, extended PDOs provide a simplified solution. A single CAN FD data frame is utilized to transfer the drive commands in the embedded network, to all drives, at the very same time. CiA 402-6 has already specified merging several control words for several axis, in one PDO. Thus, the addressed axis are getting the commands at the very same moment and start operating simultaneously. An additional effort for synchronization is just not needed.

Security and authentication

Distributed applications that handle sensor values, on which an invoice is generated for example, have to proof that the invoice is based on the correctly-measured value. They have to assure not to use some manipulated values. Typically, in such applications, system designers have opted the classic CANopen SDO transfer. This confirmed point-to-point connection, makes sure that the right sensor value is collected from the intended sensor, that is currently in an error-free operating state. To provide all this information in a single, Classical PDO, the size of 8 byte is not sufficient. Transferring the information in several segments, may causes run-time conditions, which have to be detected. The CANopen FD PDO, with a size of up to 64-byte payload, overcomes these limitations and eases the setup of such applications.

End-of-line production

Device manufacturers want to download the very same software, to many devices of the very same type. In classic CANopen, this requires a single SDO access to any device that shall get the latest firmware. With the enhanced functionality of the CANopen FD USDO broadcast service, up to 126 devices can be updated, with a single USDO access. For downloading 16-KiB firmware to a single device, the classic SDO would require about 9 300 CAN data frames for transmitting. Taking into account that every segment is confirmed with an 8-byte Classical CAN data frame. The number of exchanged Classical CAN data frames is even doubled. Additionally, the same effort is required for any device that shall be updated. Consequently, the time that is needed for updating the firmware on the devices is increased, proportional to the number of devices that shall be updated.



Pedelecs are typically based on CAN and can be modified by the end user (Source: Adobe Stock/CiA)

The CANopen FD USDO accelerates this use case significantly. For downloading a 16-KiB firmware to a single device, about 1 200 CAN FD data frames are required. With special regard to the segmented USDO protocol, the number of used frames is also doubled for confirming the reception of every segment. But in contrast to classic SDOs, the USDOs utilize a CAN FD frame for confirmation, which is about half of the size of the Classical CAN data frame, used by the classic SDO. In CANopen FD, the protocols use a CAN FD data field, that has been adapted to the real required payload.

In case such a scenario is executed at a nominal bit rate of 500 kbit/s, by using the CANopen FD segmented USDO protocol without bit rate switching, the firmware update would be handled in less than half of the time, compared to the classic CANopen segmented SDO protocol. But up to now, the real power of the USDO has not been applied. Classic CANopen SDO protocol is only applicable in a unicast session. Therefore, any additional device that requires a new firmware, adds the full time for a single device firmware download to the entire procedure. In CANopen FD, it does just not care how many devices in the network need a firmware update. Even in the worst case that the maximum number of devices (126) need an update of the very same firmware, the time for updating this firm- ware remains unchanged. Thanks to the USDO broadcast services, the time for updating devices' firmware does not depend on the number of devices, in case the devices are of the same type and need the same firmware. Another booster for this scenario, utilizing the CAN FD bit rate switch, has not been considered, yet, and would reduce the time for this scenario additionally.

If you would like to read the full article from Reiner Zitzmann (CAN in Automation), you can download it free of charge or you download the entire magazine.

cw