I had my first real contact with CANopen in the year 2000. Our small embedded system consulting firm was just a few weeks old, we had some experience with CAN, and sure enough, a client found us: “I have this inherited Z80 assembly project here to which we need to add CAN RTR PDO communication, actually CANopen. I can provide a C compiler and provide you hooks from our own scheduler. Can you help us?” Today I know that there were multiple red flags in this request. Can you spot them all?

- “inherited project”: No one has a clue how this works.
- “assembly project”: You will never know what it really does.
- “RTR” (remote transmission request): Why do you use something that asks for trouble?
- “can provide”: Eventually (a day before the deadline), you might get something that vaguely does what you desperately needed to do your work.
- “own scheduler”: A long table with call instructions. If you add a call, it might get executed at any time and screw up all timing behavior for sure.

What can I say, we were young and needed the money and we were still adventurous: we took the job. We helped the client select a commercial CANopen source code, configured it, and built a library to include to the assembly project. Despite all the red flag issues we got it working eventually and by the time we did, we were true CANopen experts: any and all traps that such a project can contain, we had found them all.

The RTR mode is something that even today still pops up once in a while. Due to a specification gap, chip manufacturers have implemented it differently – in a non-compatible way. That is why RTR was removed from the CANopen specifications and the CiA published a white paper more than ten years ago that asked implementers to stop using it. But you cannot kill a technology easily, which is why some Zombie devices still use it.

Nevertheless, when the next call came, we felt much more confident about our ability to handle it: “We have this commercial CANopen stack here and the performance on our 8-bit controller is horrible, can you have a look?” At that point, they had already spent a few weeks with technical
Insert – Ready – Cloud Solution!
Simply connect CANopen to the Cloud with our IoT Chip

- plug-in computer core for a direct cloud connection (IoT Chip)
- freely programmable by users
- combines sensor2cloud requirements with data preprocessing features ranging from time-critical signals to efficient control tasks
- libraries and protocols such as CANopen, MQTT or Modbus pre-installed

Test our IoT Chip at the “embedded world” trade fair from 14 to 16 March in Nuremberg.
support to narrow it down, but as there were several software providers involved, everyone simply blamed the other party. Once on-site (eight hours of travel and a flight to get there), it took not much more than half an hour to solve the problem. Somewhere “hidden” (just a few function calls away) in the timer interrupt, a display timer was maintained that used floating point arithmetic... on an 8-bit controller... running at some 20 MHz... This was a leftover from the CANopen stack demo. I had been contracted for a minimum of eight hours and I don’t really remember what I did during the remaining seven and half hours. Possibly this was the client where I always ended up in that nice Irish Pub just around the corner?

On the next day back at the office, we received a call from the CEO of the CANopen stack provider. My, he sounded upset. How could we go in there as external consultants and fix his code in only 30 minutes? What the --? Was he really complaining that we had not informed him first? We were contracted by his client to look at his code. Well, as a consultant you cannot satisfy everyone all the time; just make sure you keep the paying party happy. In the end, we have to be thankful because these early experiences with multiple commercial CANopen stacks encouraged us that in regards to 8-bit micro-controllers we can do better. That was the moment of birth of our own CANopen stack, optimized towards 8-bit systems.

However, when it comes to “shortest consulting job”, then the job mentioned above was topped later on. We received a call that said: “You have been recommended to us. Our CAN communication does not work at all, can you come in here and help?” In the CAN technical support world, the first thing you learn is to verify all physical settings first: cabling, termination resistors, and bit-rate. The client assured us that all of this had been checked over and over. Again, it involved multiple hours of traveling and a minimum of eight hours of consulting for us to go on the journey. It was a military application and for me to get access, some of the equipment had to be covered in linen, so that I could not see it. I felt more like being in some old attic than in a lab. Well, they gave me access to the CAN cable and what did I do first? I checked the multimeter. Do I need to say more? The termination resistance was about half of the expected 60 Ω, which is what you get with two termination resistors. Those were my best-paid ten minutes as a consultant ever: plugging in a missing termination resistor.

Speaking of military applications: those can be really strange. I once made it to the lobby of a major US military supplier and was asked: “What is your clearance level?” My answer was: „Well, I am German and not aware that I have any“. So the engineering team came out, joined me in the lobby, and for about an hour we discussed hypothetical CAN issues in the lobby. But in regards to technical support, military customers can be great. We really had a conversation along the following lines:

• Client calls in: “My CANopen communication based on your stack does not work.”
• Support: “How have you configured it?”
• Client: “I can not tell you, that is a secret.”
• Support: “Weeeell, are you seeing any CAN messages on the bus?”
• Client: “Yes.”
• Support: “So which messages are you seeing?”
• Client: “I cannot tell you, that is a secret.”
• Support: “Ooookay, what can you tell me about your CANopen communication?”
• Client: “It does not work.”

Typically that about ends the conversation and you never hear from them again. Over the years, the “weird” clients and applications have became less, but possibly they simply do not pass the red flag filter we have now: if a first-time customer request raises more than one red flag, we now leave it to the competitors to call them back first.

Interestingly, the decisions you have to make as a CANopen consultant can also involve the dark side, and I am not talking about the marketing department here, but something even darker. Years after we completed a project for a client, their legal department contacted us. They were in a US patent fight with a competitor and were in need of CAN and CANopen experts. It was unbelievable: the amounts that had been made available for the consultancy by far exceeded our usual rates. For this case, it would have been a multiple of the total engineering rate we ever charged to that company. We were tempted for a few (micro)seconds to take the money – but as an engineer; going into legal patent battles; in support of the side with the patent; potentially facing the CiA and Holger on the opposing side? Nahh...

Author
Anonymous
Look what we have for you!

An absolute must-have!

ecomatmobile BasicController relay: modular mini controller with powerful relay outputs with diagnostic capabilities for mobile applications. Direct connection of sensors and actuators without further external wiring, powerful CAN interfaces, programmable to IEC 61131-3 with CODESYS. Can be extended with other components, such as the BasicDisplay. With E1 type approval of course.

www.ifm.com/gb/ecomatmobile_basic
Phone: +49 0800 16 16 16 4