New test concept for improving CANopen device quality

Meeting the demand for plug and play functionality without restriction

Mark Schwager, Vector Informatik

In many applications ranging from the utility vehicle, industrial and building technology fields to the medical field, automation components work together based on the CANopen standard. It should be easy to couple sensors, actuators, operating units and controllers to one another according to the plug and play principle. To meet this demand, significant testing effort is required on the part of the manufacturer, as is a proper device description file. This article presents a new approach to testing where the virtualization of CANopen devices, a clever abstraction principle and test automation play a central role. The concept reduces the time and effort required for testing while also ensuring both a greater depth of testing and quality of CANopen devices.

Naturally, every CANopen device should pass the conformance test provided by the CiA User Organization (CAN in Automation) before it is released. The conformance test serves to ensure basic CANopen functionality by running through typical communication processes which must be possible according to the device description file. This includes several request-response scenarios for testing of the proper transmission and reception of process data objects (PDOs) and service data objects (SDOs) according to the standard, the testing of network management functions and so on.

The existence of a device description file, the so-called electronic data sheet (EDS) for each CANopen device is a crucial element of the plug and play functionality of CANopen. The object directory of the device description file uses an index and sub-index to precisely document which objects are available, their locations/addresses, the objects which other devices may read and whether they are write-protected or intended for data exchange via PDO. The conformance test also checks whether the specifications in the device description file match those of the actual device – at least to the greatest extent possible.

Test script maintenance: punishment for employees?

The conformance test cannot and is not intended to test actual device functions and root out complex problems which can arise when integrating the application with the CANopen stack into the device firmware. Rather, it is the job of every manufacturer to ensure that their CANopen devices provide the level of performance required by users according to the EDS file. For testing, developers frequently create manual test scripts with which the developments are to prove their production readiness. As is well known, each new software version must be subjected to the full gamut of tests (referred to as regression tests, among other things) during the development process. Should entries and addresses in the object directory have changed, for example, the test scripts must be adjusted accordingly each time. This type of work is tedious, disliked and willingly neglected, with only the most necessary tests being carried out at times. Oftentimes, the hope that everything will just be okay prevails. This is in contrast to systematic testing.

Application test operation reveals hidden flaws

In light of this and similar situations, the question of how CANopen devices can be tested efficiently and systematically arises. Initially, it is desirable to not only test devices in stand-alone operation, but under realistic operating conditions with other bus subscribers as well – that is, in the application together with corresponding receivers. The manufacturer of a CANopen rotary encoder, for example, specifically requires devices which poll the data generated by the encoder in all different ways. This could be the direction of rotation and the angular velocity, in addition to the absolute position. At the same time, other participants communicate on the bus to simulate a certain level of utilization.

You can certainly buy a bunch of suitable CANopen devices from different manufacturers on the market and connect them together within a test network for this purpose. However, this will quickly become an expensive undertaking, and handling a large number of physical devices is inflexible and time-consuming as well. A much more elegant and cost-effective solution seems to be the concept of not having to acquire other physical devices, but simulating them using appropriate software instead. The requirements for this are ideal, as device description files are available in virtually unlimited variety. They can usually be downloaded from the website of the respective manufacturer.

EDS files provide data for virtual CANopen nodes

A generator for virtual CANopen devices now reads in the available device description files and creates virtual counterparts from them. It creates a corresponding variable in the virtualization software – which runs on a Windows PC – for each entry in the object directory. This PC is equipped with interface hardware for CANopen for communication with the system under test and any other physical CANopen devices if applicable. Accordingly, the test arrangement is comprised of one physical and one virtual bus branch, which opens up very interesting testing and simulation options. The virtual CANopen nodes represent a precise map of the device description file and largely behave like their physical counterparts. Once there is a representation in the simulation model for all objects in the description files, read and write access is tailored to the simulated nodes in a similar way to the objects of real devices.

It also makes sense to create a virtual map in the simulation for the device to be checked. A test script is then able to access mirror memory relatively quickly. At the same time, the system handles updating of the object directory in the physical test hardware during write access. The mirror memory noticeably simplifies the writing of test scripts, as it enables the equal treatment of physical and virtual devices during write and read access. The test programmer does not need to think about whether they are currently writing code for a physical or virtual device every single time. This whole arrangement serves to simulate realistic operating situations with any desired amount of additional bus traffic and creates a flexible testing environment which can be quickly adapted to changing tasks.

Working with plain-text names instead of cryptic HEX values

With the intention of simplifying work for the test engineer even further, the demonstrated test concept also provides for the consistent use of plain-text names when accessing CANopen objects. People in charge of the task thus no longer need to remember countless indices and sub-indices, but simply ask about the “encoder position” when formulating a request, for example. Here, the system uses precisely that plain-text name found in the device description file in addition to the data type and access rights etc.

As multiple identical encoders can be used, and to also enable the operation of multiple parallel CANopen buses – something quite common in physical applications – we recommend using universally unambiguous namespaces. This can be implemented with a suitable designation hierarchy, which is placed ahead of the plain-text name. The structure of a variable designation then looks as follows: the keyword CANopen is followed by the network name, then the device name and finally the plain-text parameter name from the object directory. This prevents naming conflicts of any type from occurring.

Carrying out (regression) tests without modifying code

Abstraction of the object index comes along with major advantages with regard to changes in the object directory. Over the course of the development of a CANopen device, the object indices can move, new objects can be added, and other objects may be omitted from time to time. Using a servomotor as an example, the setpoint position might be represented by object index 2000 today, but could be located at index 2200 tomorrow. What this means for test code, where the objects are addressed directly, is obvious: all parts of the test program which point to modified object indices must be manually identified and updated. Not so with plain-text names. In the program code of the test, it's no longer "write value XY to object 2000," but "write value XY to motor setpoint position" instead. The simulation software need only read in the new device description file. In this way, the test programs written once in accordance with the plain-text naming convention can be executed unchanged as many times as desired with no adjustments to the code following each modification of the device firmware.

Get there faster with test automation

In addition to virtual CANopen devices and the use of plain-text names, test automation forms what is essentially the third column of the concept for efficient testing of CANopen developments. Long-term tests, for example, are hardly feasible and can run for months without test automation. Quite a number of CANopen devices are used in safety-critical applications like medical technology and the energy sector. The development and marketing of these types of devices is unimaginable without corresponding long-term tests, often even in conjunction with climate exposure test cabinets for the purpose of artificial aging.

Each testing department fundamentally benefits from flexible test automation. The time-saving effect is enormous and opens up testing possibilities which would not be feasible otherwise. This is why a testing system with test automation which can be scaled to suit increasing requirements is preferable. Initially, you may only want to run through the same test many times, changing configurations and boundary conditions and creating intermediate reports each time automatically with each execution. Later on, several pieces of test hardware will have to prove their production readiness in complex simulation models with many virtual CANopen devices and CANopen buses at the same time. Ideally, the test framework can also be enhanced with hardware modules providing analog and digital I/O signals as well. Voltages, currents and signal characteristics can be generated, which serve to stimulate CANopen sensors. Suitable relay cards can switch supply voltages on and off under the control of scripts, create artificial short-circuits, connect interference voltages and lots more. There's almost no limit to the tester's imagination.

Conclusion

Supplying the customer with top-quality, fault-free CANopen devices should be the top priority of every manufacturer. The test concept demonstrated here facilitates work when testing CANopen developments, saves time and leads to a significantly greater depth of testing. CANopen devices can be realistically tested in an application, test programs can be reused many times without code modifications and complex test scenarios can be run through. By connecting modularly designed digital and analog I/O cards, switching relays and other hardware modules, a fully automated testing environment can be provided.

Mark Schwager
Vector Informatik GmbH
Holderäckerstr. 36
DE-70499 Stuttgart
www.vector.com