Improving and testing CiA 401 for the next generation of I/O devices

Bruce M. Decker, P.E., Schneider Electric, American Global Products

CANopen continues to gain acceptance as a robust protocol for use in a variety of industries and applications. As this expansion occurs into new applications, the general-purpose I/O devices required by these applications are starting to push up against the limits of the CANopen Device Profile that describes them – CiA 401.

In this paper, the author proposes some improvements to this venerable standard which will allow for a whole new generation of I/O devices to take full advantage of the CANopen protocol, without being unduly constrained by vestiges of the first generation which may no longer be needed. Changes to default PDO mapping, harmonizing the current behavior differences between analog and digital devices, and making sense of the Device Type object are all examined.

1 Where did the analog field data go?

A controls engineer at a major processing plant is overseeing the commissioning of a new CANopen-based control system. A CANopen system was selected because of its relatively low cost compared to other robust field busses. The electricians have carefully wired several switches and multiple 4-20 mA sensors and actuators to the CiA 401-compliant distributed I/O block. As technicians begin to simulate the process, the control engineer notices that the switch actuations are registering state changes, but that the 4-20 mA sensors are always reporting a value which makes no sense, and never changes. Realizing that this is an unacceptable situation, the engineer directs the technicians to troubleshoot the system. After an hour and a half of investigation, the technicians report that the distributed I/O system is not reporting the values from any of the 4-20 mA sensor devices, although the digital switch information is working just fine. The engineer then calls the distributed I/O manufacturer for support.

Over the next few hours, the engineer’s plea for help is escalated up the vendor’s support organization. Many different analytic procedures are tried, to no avail. Finally, one of the manufacturer’s top engineers is put on the phone to soothe the anger of the increasingly frustrated controls engineer. After spending an hour on the phone, the manufacturer’s engineer finally discerns the problem – the control program simply never set Object 6423h, the Analog Input Global Interrupt Enable, to TRUE. As soon as this is done, the system starts reporting changes in the 4-20 mA sensors. After nearly a day of troubleshooting, the commissioning procedure is finally able to continue.

2 Losing analog precision

Continuing the commissioning process, the controls engineer begins to transmit 16-bit control values to the 4-20 mA actuators. The controls engineer quickly realizes the actuators seem to be exhibiting an unexpected gain of 2. Then, as soon as the commanded value reaches what should be half-scale, the actuator quickly retreats to the 4 mA position, and stays there.

Once again, the controls engineer is on the phone to the I/O manufacturer’s engineer. This time, the problem is diagnosed a little faster: because it is a standard CiA 401 device, all the default analog RPDOs must be configured for signed integer values, meaning the actuators were actually operating on 15-bit, rather than the 16-bit values the controls engineer was expecting. This explains the apparent 2X gain. Since this specific 4-20 mA output device interprets any negative integer value as a minimum current situation, it set the output to 4 mA for all control values over 8000h. In order for the actuators to work on unsigned 16 bit integer values, the controls engineer must map the RPDOs into the manufacturer-specific area of the device's object dictionary, where these registers are defined.

The controls engineer is now quite frustrated that company management selected CANopen as the fieldbus. All of this was supposed to be easy.

3 What does the device type object mean?

By now, the controls engineer has obtained a copy of CiA 401 and a diagnostic tool which allows direct interrogation of the device object dictionary. The engineer decides to start at learning the basics of CANopen, and uses the diagnostic tool to request the value of the device type register, object 1000h.

The value returned is confusing, to say the least – it is 401d. Looking at CiA 401, the engineer believes some of the function bits should be set. After all, the device has digital inputs, digital outputs, analog inputs, and analog outputs. Shouldn't bits 16 through 19 of object 1000h be set? If these are not set, then what is the meaning of the device type object?

The controls engineer is now really questioning the decision to deploy a CANopen system.

4 Re-visiting CiA 401

Even though the preceding travails of our controls engineer are fictional, they are based upon real, recent events of which the author is aware. It is these types of real-world issues which cause many potential customers to think twice about deploying a CANopen – based system.

It is apparent the original authors of CiA 401 were trying to write specifications that were reasonable and usable for the types of devices which existed at the time. They greatly succeeded at this. Some devices, especially CANopen master devices, were fairly limited in their capabilities a few years ago. CANopen was new, and not widely deployed. While the author of this paper is sure they hoped for wide acceptance and deployment of CANopen devices and systems, they could not possibly have foreseen the issues that the customer would or could encounter. To that end, the author believes it is time for the CiA to re-visit the venerable CiA 401 standard, and to improve the standard so that it does not become a hindrance to its own success.

5 Harmonizing analog and digital behavior

Our control engineer's unfortunate first scenario, where the engineer assumed analog and digital input data had the same basic behavior, has occurred more than once in the last few years as CiA 401 generic I/O devices are being deployed. The author's experience is that many end users of CANopen devices do not want, nor do they have the time, to become CANopen experts. They want their device to work "out of the box" for their particular application. When you consider that CiA 401 treats analog and digital inputs differently, it is really not surprising that this confusing scenario is occurring time and time again.

When CiA 401 was written, there was a very real concern that default-configured analog input sources would flood the CANopen bus when the bus was set to the operational state, because the analog input modules would immediately transmit a TPDO for every single least significant bit (LSB) change – possibly just noise in the signal. For digital data, there was no similar concern. As a result, TPDOs for digital inputs are enabled as soon as the bus is set to operational, while analog TPDOs must be explicitly enabled by setting the conditionally required object 6423h to TRUE. There is an optional object (6005h) which may be used to globally enable or disable the sending of 8-bit digital TPDOs. Even if this object is implemented, the default value would be set to TRUE, so the behavior of a default configured system would still be the same.

In retrospect, perhaps it would have been better if CiA 401 had used a different mechanism for analog TPDOs. For example, the TPDO Inhibit Time default value could have been set to 100 milliseconds, insuring that the analog TPDOs did not flood the bus. However, this was not done because Inhibit Time is optional for PDOs.

If there were no CiA 401 compliant units deployed, it would be a simple matter to make the analog and digital input behaviors identical, and to come up with a better solution to the bus flooding issue. However, there are many CiA 401 devices deployed, and making the behaviors consistent in the standard would create a massive and unacceptable backwards compatibility nightmare. However, there is a simple solution.

The underlying problem for our controls engineer was more subtle. Even if the engineer knew *a priori* that he/she needed to set the state of object 6423h to TRUE, the engineer still has no way of knowing that it actually occurred! This is even more serious a situation when a “minor” modification is made to an existing application program which accidentally deletes the setting of the object. If the analog inputs are being used as feedback values for the process control, such as might occur in a simple Proportional-Integral-Derivative (PID) loop, the process can be actually out of control until some other (digital) indicator sets an alarm. Clearly, this is not acceptable. The user needs to know immediately if the analog TPDOs are inhibited when the CANopen device is set to its operational state.

6 A simple compromise solution

The simple solution to this problem is to modify CiA 401 to require all analog input devices to send an EMCY message (the value to be determined) immediately upon transitioning to the operational state if object 6423h is set to FALSE. This allows backwards compatibility with existing CiA 401 devices, and provides a means for immediate feedback to the user. Since the user may actually want the analog PDOs off for some reason, this condition cannot be considered a device fault, and thus the device will not automatically transition to the pre-operational state when this occurs. The user can choose to ignore this EMCY message at their own risk, or they can execute an application - appropriate procedure to prevent a catastrophe.

7 The default PDO mappings

Our controls engineer’s second headache was caused by a misunderstanding of the CiA 401 default mappings. Once again, the original authors of CiA 401 were trying to achieve a level of uniformity between devices, given the capabilities of the CANopen masters and slaves at the time. But again, the capabilities of modern automation devices are turning these attempts at simplification into a hindrance. To set the philosophical groundwork for the next proposal, let us take a quick look at another scientific discipline: object-oriented (OO) software programming.

Two of the foundational principles of OO fit together like a hand in a glove – abstraction and encapsulation. These two principles combine to give rise to the concept of an object class, which describes a standard interface to the object, as well as the behaviors of the object when the interface is accessed by another object. The major tenet of encapsulation is that the object class itself determines its own behavior. This principle means that a member of a particular object class is in control of what it does, and another object cannot make it do something it is not capable of doing.

A third foundational principle of OO is inheritance. Any object class can be “refined” by deriving a child class from it. The child class “inherits” the interface and behavior of the parent object, but is free to modify the particulars to suit the needs of the application program.

Let us now come back to CiA 401 and apply these OO principles to compliant devices. Consider: if a device is designed to handle 4-20 mA devices, where negative data values are meaningless, why should it be forced to default to mapping signed integer values? If we apply the OO principles described, we can conclude that the device should tell the CANopen master what type of data it handles, and not the other way around. After all, if we consider that the 4-20 mA output device is a child class of the analog output parent class, should it be required to support what is at best, sub-optimal data types? In the author's opinion, this should not be the case.

Let us take another example. CiA requires digital input or output devices to support 8-bit data words by default. But what if a manufacturer needs to make a 16-bit cohesive digital module for a particular market need? There are several issues here. First, CiA 401 does not prescribe how to map 16 bit data onto 8 bit objects, so it is conceivable that different vendors will map it different ways, creating an interoperability (or rather, a non-interoperability) headache. In addition, the manufacturer is now burdened with double mapping of the object. If the device is a 16 bit input device, it makes sense to use object 6100h for the data value. But the standard requires it to simultaneously be mapped to object 6000h for 8 bit compatibility. Needless to say, this is a disincentive to manufacturing anything other than a 8 bit digital I/O device.

8 Let the device determine the default PDO mappings

As applications for CANopen devices proliferate, the author believes the newer devices should determine what objects are mapped into the PDOs, based upon what makes sense for the device. A 16 bit digital input device could map object 6100h into the first TPDO. And if the device has multiple digital channels, and does not support analog inputs, it could map several sub-indices of object 6100h into TPDO2 through TPDO4!

A little more work must be done to satisfy the 4-20 mA requirements. CiA 401 does not currently specify standard objects for unsigned analog data, only for 8, 16, and 32 bit signed integer data plus floating point values. These new objects would need to first be defined and then they could be mapped into any of the first four PDOs. An alternative to this is to add a sub-index to the existing objects which specifies the default data type. But this approach is not consistent with the way digital objects are handled.

Of course, this puts more responsibility on the future CANopen masters. They must know, as part of their configuration time, to read all the default PDOs, and how to interpret the data based upon the objects being mapped. Since modern CANopen masters have more processing capability, this should be no issue.

In order to insure compatibility, as well as signal the master that the device conforms to the new version of the specification, there must be an indicator mechanism defined. One of the currently undefined bits in object 1000h, for example bit #22, could be utilized for this purpose. See figure #1 for the current structure of object 1000h in a CiA 401 device.

9 Making object 1000h more meaningful

Our controls engineer's final conundrum came from looking at object 1000h. A value of 401d was certainly a legal value. The specification clearly states: "Any combination of digital/analogue inputs and outputs is allowed", which presumably means no functionality bits being set is also allowed. But the engineer had to wonder, why weren't the capability bits reporting useful information about the device?

Of course, the manufacturer may have had a good reason to not set the bits. However, it is hard to say the device is truly CiA 401 compliant if the upper 16 bits of object 1000h are all zeros. These upper 16 bits are supposed to indicate the existence of the specified mappings and objects for each type of capability.

The CiA Conformance Testing Task Force, of which the author is a member, has examined this issue. Many of the members are of the opinion that a generic I/O device cannot be considered CiA 401 compliant unless at least one of the bits is set, (or a special function, e.g. joystick, is indicated) and the required objects and mappings for the indicated capabilities are conformant with the specification.

Specific Functionality	I/O Functionality	Device Profile Number = 401d
31 24	23 16	15 0

I/O Functionality:

Bit 16 – Digital Input

Bit 17 – Digital Output

Bit 18 – Analog Input

Bit 19 – Analog Output

Bits 20 to 23 – Reserved

Specific Functionality Code:

0 – No specific functionality

1 – Joystick

2 to FFh - Reserved

Figure 1: CiA 401 Object 1000h

10 Device profile conformance testing

Although some people may not realize it, the discussion of the function bits in object 1000h raises another question: how will CiA 401 devices be tested for conformance?

At the time of this writing, no official CiA 401 conformance test exists, although the charter of the CiA Conformance Testing Task Force (TF) is to develop one, as well as a test for CiA 402. Object 1000h will be the key to the test.

The current thinking of the task force is that a manufacturer desiring conformance testing shall first pass the EDS and CiA 301 conformance tests before proceeding to the device profile test.

The first object to be interrogated in the device profile test will be object 1000h. If the value of object 1000h is 401d, the test will halt, because nothing else can be tested. The device cannot really be considered CiA 401 compliant, although it still may be CiA 301 compliant.

For the current version of CiA 401 devices, the conformance test will continue based upon the bits set in object 1000h. If the digital input bit is set, the test will look at the mapping of the first TPDO to insure it is default mapped to object 6000h. It will also verify that all the required objects for the digital input function exist and have their proper default values.

Assuming no failures, the test will examine the CiA 401 required objects and default values for digital outputs, and the analog input and outputs based upon the bits set in object 1000h.

If any of the function bits are cleared, the conformance test will make sure that none of the standard objects for that function exist, and that no PDOs are mapped to these non-existent standard objects.

For the next generation of CiA 401 devices, object 1000h will still be the key to proper conformance testing. The major difference will be to check that the default PDO mappings are made to standard objects which are legal for the function, and that they match the entries in the EDS file. For example, if a new generation device reports analog output functionality, the PDOs must map to any of the existing analog output objects, or any of the newly defined unsigned analog output objects.

11 Summary

The preceding suggestions for improving CiA 401 have been formally submitted to the CiA for consideration. The intent is to make sure the standard does not become a hindrance to its adoption for the next generation of CANopen I/O devices. As the suggestions are considered by the CiA, many other good suggestions are likely to be brought forward. It is the author's hope that the next version of CiA 401 will be flexible enough to allow common-sense adaptations of devices so device manufacturers may solve their customers diverse requirements.

12 Acknowledgements

The author would like to thank Messrs. Jochen Weiland and Günther Wenzel of Schneider Electric and Mr. Roland Rauch of Berger Lahr for their help in developing the suggestions set forth in this paper.

References

[1] CiA Draft Standard 401, *CANopen Device Profile for Generic I/O Modules*, Version. 2.1, 17 May 02, CAN in Automation e. V.