# CANOPEN – HIGHER LAYER PROTOCOL BASED ON CONTROLLER AREA NETWORK (CAN) SUPPORTS DEVICE PROFILES FOR I/O MODULES, MOTION CONTROL

**Wilfred Voss**
**esd electronics, Inc.**
**Greenfield, MA**

**http://www.esd-electronics.us**

***Abstract*** *- CANopen is an industrial networking technology based on the serial bus system “Controller Area Network” (CAN), which connects a master device with numerous slave devices such as digital I/O, analog I/O, motion controllers, encoders, sensors, actuators, etc. Developed originally for passenger cars, the CAN two-wire bus system is already in use in millions of industrial devices all over the world. In contrast to other field-bus technologies, CAN/CANopen was specifically designed for real-time control and maximum reliability, which made it very suitable for time critical tasks, for instance, motion control. The ultimate goal of CANopen is to provide OEMs and end users a certain level of manufacturer independence by providing standard device profiles for a continuously increasing number of controls. This paper will provide an overview of the CANopen specification (DS-301, DSP-302) and will explore in more detail the device profiles for generic I/O modules (DS-401), drives and motion control (DSP-402), and IEC 61131-3 programmable devices (DS-405).*

## I. INTRODUCTION

CANopen networks can be used as an embedded communication system for microcontrollers as well as an open communication system for intelligent devices. Some users, for example in the field of medical engineering, opt for CANopen, because they have to meet particularly stringent safety requirements. Similar requirements have to be considered by manufacturers of other equipment with very high safety or reliability requirements (e. g. robots, lifts and transportation systems).

Gatekeeper of the CAN and CANopen standard, including device profiles, is CAN-in-Automation (CiA), a non-profit organization that is only controlled by its numerous member companies all over the world.

## II. WHAT IS CAN ?

Controller Area Network (CAN), the underlying physical layer to CANopen, is a serial network technology that was originally designed for the automotive industry, especially for European cars, but has also become a popular bus in industrial automation as well as other applications. The CAN bus is primarily used in embedded systems, and as its name implies, is a network technology that provides fast communication among microcontrollers up to real-time requirements, eliminating the need for the much more expensive and complex technology of a Dual-Ported RAM.

CAN is a two-wire, half duplex, high-speed network system, that is far superior to conventional serial technologies such as RS232 in regards to functionality and reliability and yet CAN implementations are more cost effective. While, for instance, TCP/IP is designed for the transport of large data amounts, CAN is designed for real-time requirements and with its 1 MBit/sec baud rate can easily beat a 100 MBit/sec TCP/IP connection when it comes to short reaction times, timely error detection, quick error recovery and error repair.

Many major semiconductor manufacturers, such as Motorola, Philips, Intel, Infineon, and many more, sell CAN chips, and the fact that millions of them are used in automobiles guarantees low chip prices and long-term availability. Most semiconductor manufacturers who usually integrated a UART with their microprocessor design, in order to support serial communication for RS 232/485, nowadays tend to integrate CAN instead.

The use of CAN in most of European passenger cars and the decision by truck and off-road vehicle manufacturers for CAN led to the availability of CAN chips since 1987. Other high volume markets, like domestic appliances and industrial control, also increase the CAN sales figures and guarantee the availability for the future. [1]

**Controller Area Network**

- Is a high-integrity serial data communications bus for real-time applications
- Is more cost effective than any other serial bus system including RS232 and TCP/IP
- Provides better ease of use than any other serial bus system
- Operates at data rates of up to 1 Megabits per second
- Has excellent error detection and fault confinement capabilities
- Has the ability to function in difficult electrical environments
- Is now being used in many other industrial automation and control applications
- Is an international standard: ISO 11898

**Speed, Reliability, Error-Resistance**

CAN provides the ability to function in difficult electrical environments, a high degree of real time capability and ease of use. The real time capabilities are supported by extremely short arbitration times in the range of microseconds, limited data length and extremely short error recovery times, again, in the range of microseconds.

The reliability and error resistance of CAN has been calculated in a mathematical model. Here is an example using the following parameters and conditions:

- 1 Bit error every 0.7 sec
- Baud rate of 500 kBit/sec
- Operation of 8 hours/day and 365 days/year

According to the mathematical model the *Residual Error Probability* will be *1 undetected error in 1000 years*. [1]

## III. THE CANOPEN HIGHER LAYER PROTOCOL

Even though extremely effective in automobiles and small applications, CAN alone is not suitable for machine automation, since its communication between devices is limited to only 8 bytes per message. Also, CAN is a multi-master system, i.e.

it does not support a master/slave configuration. As a consequence, higher layer protocols such as CANopen for machine control, DeviceNet for factory automation and J1939 for vehicles were designed to provide a networking technology that, among many other networking features, supports messages of unlimited length and allows a master/slave configuration.

The ISO/OSI Reference Model specifies 7 levels beginning with the physical connection up to the actual user application, i.e. the Application Layer.

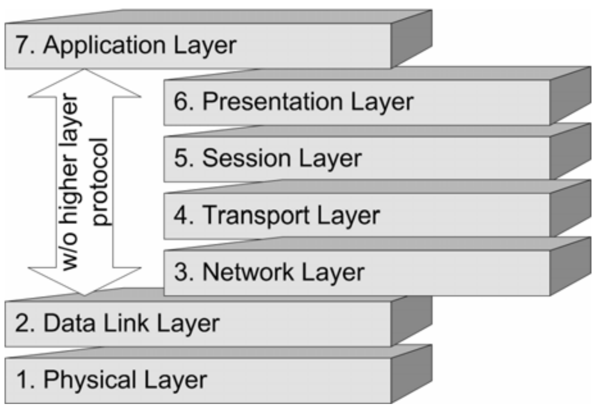


**Figure #1, ISO 7-Layer Reference Model**

The standard CAN implementation bypasses the connection between the Data Link Layer and the Application Layer in order to save on valuable memory resources by minimizing the overhead and, as a result, gaining performance as needed for embedded solutions with limited resources. After all, all layers above the Data Link Layer require additional software resources (higher layer protocols). [1]

The Physical Layer and Data Link Layer are actually integrated into silicon, i.e. the CAN controller including the protocol firmware; they represent a standard CAN application. The CANopen Higher Layer Protocol partially implements the additional layers 3 to 6 as shown in figure #1.

## IV. CAN-IN-AUTOMATION DRAFT STANDARDS

As was mentioned earlier, the gatekeeper of the CANopen standard, including the device profiles, is CAN-in-Automation (CiA), a non-profit organization that is only controlled by its numerous member companies all over the world. In 1995 Can-in-Automation published the CANopen standard, which is represented by two *Draft Standards*, DS 301 to define the *CANopen Application Layer and Communication Profile* and DSP 302 (DSP – Draft Standard Proposal) for the *Framework for Programmable CANopen Devices*.

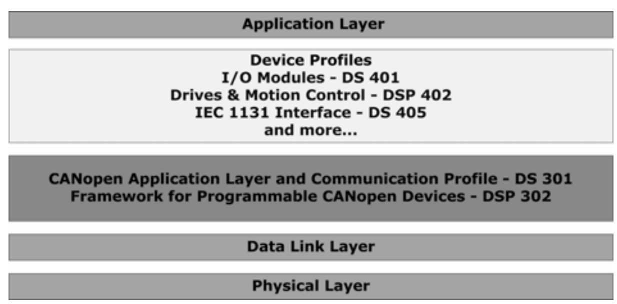


**Figure #2, CANopen Reference Model**

DS 301 and DSP 302 partially cover layers 3 through 6 of the ISO 7-Layer Reference Model as shown in Figure #1 (Compare to Figure #2 above).

According to DS 301 and DS 302 CANopen provides:

- Data Transport of more than 8 bytes per message
- Communication model based on Master/Slave configuration

- Network Management (Network Start-Up, Node Monitoring, Node Synchronization, etc.)
- Notification of Device Errors (Emergency functionality to signal failures of application or communication)
- System-wide Synchronization of Processes
- System-wide time reference

The special tasks of a CANopen Master/Manager are:

- Controlling the network boot-up process
- Verification and supervision of system consistency
- Download of configuration data to new devices
- Controlling the communication status of a device
- Node-Guarding (Master/Slave Monitoring)
- Heartbeat (Device status is transmitted as broadcast information, each device can monitor other devices)

## V. CANOPEN DEVICE PROFILES

In addition to the CANopen standard, CAN-in-Automation has published and is working on various CANopen Device Profiles, which provide a high level of standardization between devices of the same nature.

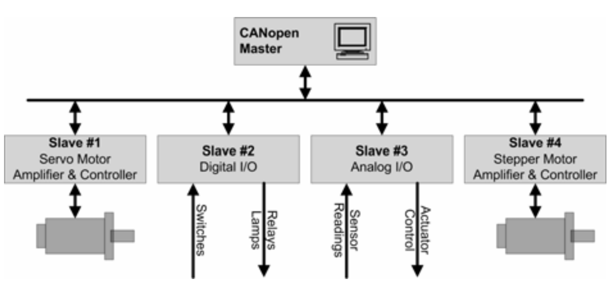


**Figure #3, CANopen Network Example**

Figure #3 shows an example of a CANopen network, where all slave devices, regardless of their individual function, and the master "speak the same language". Each device has its own Object Dictionary that describes the device's functionality. The Object Dictionary is basically a list of objects accessible through the network by means of standardized methods. Each object within the dictionary is addressed through means of a 16-bit Index as shown in figure #4.

| Index (hex) | Object |
|---|---|
| 0000 | Not used |
| 0001 – 001F | Static Data Types |
| 0020 – 003F | Complex Data Types |
| 0040 – 005F | Manufacturer Specific Complex Data Types |
| 0060 – 007F | Device Profile Specific Static Data Types |
| 0080 – 009F | Device Profile Specific Complex Data Types |
| 00A0 – 0FFF | Reserved for future use |
| 1000 – 1FFF | Communication Profile Area |
| 2000 – 5FFF | Manufacturer Specific Profile Area |
| 6000 – 9FFF | Standardized Device Profile Area |
| A000 – BFFF | Standardized Interface Profile Area |
| C000 – FFFF | Reserved for future use |

**Figure #4, Object Dictionary Mapping**

The Device Profile objects in the dictionary may represent process commands, process data, and executable functions. These standardized objects guarantee full compatibility between devices of the same nature, e.g. motion controllers or I/O devices, regardless of the manufacturer of the device, therefore (theoretically) guaranteeing full interchangeability. The *Standardized Device Profile Area* in the range from 6000h through 9FFFh contains all data objects common to a class of devices that can be read or written via the network. The device profiles also use entries in the range of 1000h through 1FFFh (Communications Objects) to describe the device parameters and functionality. For instance, the predefined object at index 1000h of the object dictionary describes the type of device and its functionality. The object at index 1001h contains the error code. Further objects contain the manufacturer device name, hardware version, and software version. [2] [3]

Some of the Device Profiles, developed by CAN-in-Automation, are:

- DS 401 - Generic I/O Modules
- DSP 402 – Drives and Motion Control
- DS 405 – IEC 61131-3 Programmable Devices
- DS 406 – Encoders
- DS 412 – Medical Devices
- …and many more…

## VI. DS 401 – GENERIC I/O MODULES

The CiA Draft Standard 401 describes the functionality of analog and digital input/output modules, which can be used to connect, for instance, switches and signal lamps (digital I/O) or sensors and actuators (analog I/O) to a CANopen network (See also figure #3). The predefined object at index 1000h of the object dictionary (Communication objects range between 1000h and 1FFFh) describes the type of device and its functionality:

| Additional Information | | General information |
|---|---|---|
| Specific functionality | I/O functionality | Device profile number |
| 31 MSB … 24 | 23 … 16 | 15 … 0 LSB |

**Figure #5, Device Type Object**

**General information**

Device profile number: 401d

**Additional information**

1 = Function is implemented

0 = Function is not implemented

| | | |
|---|---|---|
| I/O functionality: | Bit 16: | Digital Input |
| | Bit 17: | Digital Output |
| | Bit 18: | Analog Input |
| | Bit 19: | Analog Output |

Any combination of digital or analog inputs and outputs is allowed.

Further communication objects allow:

- Writing up to 64 digital outputs
- Reading up to 64 digital inputs
- Writing up to 12 16-bit analog outputs
- Reading up to 12 16-bit analog inputs

Each input, digital or analog, can be polled, or read frequently according to a user-definable timer setting, or event driven (change of input).

Another communication object, the error register at index 1001h, provides error codes, which may be:

- Current at outputs too high (overload)
- Short circuit at outputs
- Load dump at outputs
- Input voltage too high
- Input voltage too low
- Internal voltage too high
- Internal voltage too low
- Output voltage too high
- Output voltage too low

[4]

## VII. DSP 402 – DRIVES AND MOTION CONTROL

The CiA Draft Standard Proposal 402 describes the standardized CANopen device profile for motion control products such as servo controllers, frequency converters or stepper motor controllers. The document refers to two major subjects, the device control state machine and the modes of operation.

The device control state machine executes start, stop and further motion mode specific commands. The modes of operation define the behavior of the drive/amplifier. They are:

- Homing Mode
- Profile Position Mode
- Interpolated Position Mode
- Profile Velocity Mode
- Profile Torque Mode
- Velocity Mode

According to the CANopen standard, the *Standardized Device Profile Area* in the range from 6000h through 9FFFh contains all data objects common to a class of devices that can be read or written via the network. The drives' profiles use entries in this area to describe the drives' parameters and functionality. The range of the standardized device profile area allows the utilization of up to 8 motion axes plus the option of implementing I/O modules compliant to DS 401 in place of a motion axis.

The predefined object at index 1000h of the object dictionary (Communication objects range between 1000h and 1FFFh) describes the type of device and its functionality (See also figure #5).

**General information**

Device profile number: 402d

**Additional information**

The additional information is divided into a Type section and Mode bits. They define the drive type(s):

- Frequency Converter
- Servo Drive
- Stepper Motor
- Multiple Device Module

The number of all available objects according to DSP 402 is vast, since motion control can be a complex technology, and a detailed description is most certainly not in the scope of this paper. Most device profiles are available as a download from the CiA web site (http://www.can-cia.org), however, some documents are reserved to CiA members only. [5]

## VIII. DS 405 – IEC 61131-3 PROGRAMMABLE DEVICES

IEC 61131 is a programming language standard that originates from Programmable Logic Controllers (PLCs), but has been adopted very widely in the Industrial world. IEC 61131-3 supports the following programming standards:

- Instruction List
- Sequential Function Chart
- Function Block Diagram
- Structured Text (similar to PASCAL)
- Ladder Diagram

For further details on this programming language refer to http://www.oacg.co.uk/plcopen.pdf.

DS 405 describes the access to a CANopen communication system from within an IEC 61131-3 program based on variables, i.e. access to elementary variable objects and calls to function blocks. It also explains utility functions for debugging, monitoring and network management and the interface between CANopen tools and the programming environment (See figure #6).

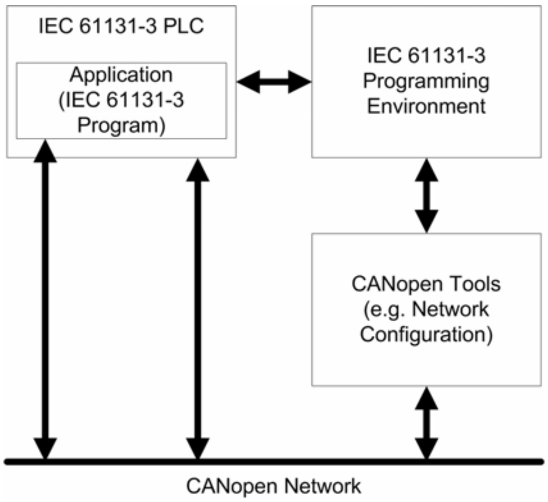


**Figure #6, IEC 61131-3 CANopen Interface**

According to the CANopen standard the object at index 1000h ("Device Type") identifies the device profile of the node (See also figure #5). In this case the profile number is set to 405. The additional information is set to 0 (zero) and is reserved for future use by the CiA.

As mentioned earlier, DS 405 covers two methods of accessing the CANopen network:

1. Variable-based access, accessing individual variables from within the IEC 61131-3 program according to the Network Variables of DSP 302.
2. Function block access, calling function blocks to write and read data.
3. Additional management functions for processing Emergency, Layer Setting Services (LSS) and network state information.

All objects in the range of A000h through AFFFh of the object dictionary are visible as variables to an application programmed within IEC 61131-3. DS 405 also defines a list of standard function blocks, which, in the sense of the CANopen standardization, are optional. [6]

## IX. CONCLUSION

Standardized device profiles are very effective supplements to a higher layer protocol such as CANopen. The standardization effort by CAN-in-Automation is one initiative to come closer the ultimate goal of providing OEMs and end users a high level of manufacturer independence when it comes to the implementation of off-the-shelf electronic devices into their design. Ultimately, CANopen helps to drastically simplify system integration.

## X. REFERENCES


[1] Voss, Wilfred, "A Comprehensible Guide To Controller Area Network", ISBN 0-9765116-0-6, Published by Copperhill Technologies Corporation.

[2] CAN in Automation e.V., "CANopen Application Layer and Communication Profile, CiA Draft Standard 301", Version 4.02, Feb. 13, 2002.

[3] CAN in Automation e.V., "CANopen Framework for CANopen Managers and Programmable CANopen Devices, CiA Draft Standard Proposal 302", Version 3.3.0, Oct. 10, 2003.

[4] CAN in Automation e.V., "CiA Draft Standard 401, CANopen Device Profile for Generic I/O Modules", Version 2.1, May 17, 2002.

[5] CAN in Automation e.V., "CiA Draft Standard Proposal 402, CANopen Device Profile Drives and Motion Control", Version 2.0, Jul. 26, 2002.

[6] CAN in Automation e.V., "CiA Draft Standard 405, CANopen Interface and Device Profile for IEC 61131-3 Programmable Devices", Version 2.0, May 21, 2002.