CANopen Introduction

CANopen

This article is intended to give engineers a brief overview of the CANopen protocol, which is specified by CiA 301 and maintained by CAN in Automation. For details on the specification itself, please visit CAN in Automation (CiA). For more information on our **CANopen software** package.

CANopen is a real-time protocol used in automation, automotive, and medical equipment. CANopen handles the data link, transport protocol, network management, and application layers. Typically CANopen uses CAN as the physical layer with baud rates ranging from 125 Kbps to 1 Mbps.

Several different application layers exist and are known as device profiles. For example, CiA 401 defines the application layer (i.e. define profile) for a generic I/O device while CiA 402 defines operation for servo drives, frequency inverters and stepper motors. Altogether there are more than 50 separate and defined device profiles.

CANopen Overview

CANopen is a high level communications protocol, which operates on a Controller Area Network (CAN) bus. CiA 301 combined with a device profile (e.g. CiA 401) for a complete specification which specifies exactly how information (e.g. servo motor RPM) is exchanged between electronic control units (ECUs).

The standard goes on to define many other aspects, including message timeouts, how large messages are fragmented and reassembled, the network speed, the physical layer, and the node ID.

CANopen Object Dictionary

At the heart of CANopen is the Object Dictionary (OD), which all nodes must contain. An OD contains all setup and configuration items for a node. The OD is implemented as an array of objects. Each object has an assigned 16-bit index and 8-bit sub-index. This allows for up to 255 sub-entries at each index.

All data is stored as entries in pre-defined locations of the object dictionary. An entries index and sub-index is listed in CiA 301 and it's corresponding device profile. An OD entry has an index, object name, description, type, attribute, and optional field.

Process Data Object (PDO)

A Process Data Object protocol is used to transmit information in an efficient manner with little overhead. Since PDOs are lightweight, CANopen uses them to transmit real-time diagnostic and control information. All PDOs are single frame, which means that a node can transfer up to 8 bytes of data per PDO. However, a single PDO can contain multiple data objects. Multiple objects inside of a single PDO are mapped via the object dictionary.

There are two kinds of PDOs: receive and transmit PDOs (RPDO and TPDO). A RPDO is used when a node needs to receive data (i.e. read data) and a TPDO is used when a node needs to transmit data (i.e. write data). CiA 301 defines eight identifiers, four for RPDOs and four for TPDOs.

PDOs are transmitted periodically, by an external event, or when used the nodes receives a SYNC message. PDOs that are transmitted on the reception of a SYNC PDO are known as synchronous PDOs, while periodic PDOs are known as asynchronous.

Service Data Object (SDO)

An SDO, service data object, is used to read and write values inside of an object dictionary. CANopen defines a write into an OD as an SDO download and a read from an OD as an SDO upload. Since SDOs can be larger than a single frame (i.e. 8 bytes), SDO implementations require fragmentation and reassembly (i.e. transport protocol).

An example of an SDO read is when a node requests the manufacturer device name and/or device type of another node.

Simma Software, Inc. specializes in real-time embedded software for the automotive industry.

Products and services include protocol stacks, bootloaders, device drivers, training, and consultation on the following technologies:

J1939, CAN, J1587, J1708, J2497, J1922, ISO 15765, CANopen, UDS, XCP, LIN.

Simma Software, Inc
5940 South Ernest Drive
Terre Haute, IN 47802
Toll Free: 888-256-3828
Fax: 208-445-2913

www.simmasoftware.com