

English Version
Die deutsche Version
Карта сайта

Обзор протокола CANopen

Добавил(а) microsin

Поделиться

...
Расширенный поиск

Нашли опечатку?

Пожалуйста, сообщите об этом - просто выделите ошибочное слово или фразу и нажмите **Shift Enter**.



Блог одного
Сумасшествия

Самое общее представление о стандарте **CANopen** можно получить из русской Википедии, однако это не дает полной картины для понимания протокола, а просто объясняет общие термины. CANopen разработан как высокоуровневый сетевой протокол, работающий поверх физического протокола **CAN** (Controller Area Network), что позволяет осуществлять обмен данными между устройствами от разных производителей, и гарантирует взаимозаменяемость устройств. Здесь приведено общее описание стандарта, собранное из разных источников (см. "Ссылки"). Все непонятные термины и сокращения приведены в конце статьи (см. раздел "Словарик"). Здесь приведен перевод документации с сайта canopensolutions.com [2].

Общая информация о протоколе CANopen

[Базовые понятия CANopen]

Физическая структура сети CANopen. Архитектура нижнего уровня CAN определяет физическую структуру сети CANopen. По сути она самая обычная, ничем не отличающаяся от других сетей CAN - используется линейная (шинная) топология, где все устройства параллельно подключены к одной двухпроводной линии связи. Чтобы избежать паразитных отражений сигнала, оба конца сети должны быть нагружены на терминирующую нагрузку (2 резистора по 120 ом). Дополнительно должны быть учтены максимально допустимые длины сегментов сети между отдельными сетевыми узлами - в зависимости от скорости передачи и параметров линии.



Рис. 2. Физическая организация сети CAN при использовании CANopen.

Рекомендуемые допустимые скорости (bit rate) для сети CANopen даны в CiA 301: 10 kbps, 20 kbps, 50 kbps, 125 kbps, 250 kbps, 500 kbps, 800 kbps and 1000 kbps. В CiA 301 также дается рекомендация для конфигурации длительности (тайминга) бит.

Дополнительно для CANopen должны быть соблюдены 2 дополнительных условия. В принципе, это самые обычные требования к сети CAN:

- Все узлы сети должны быть сконфигурированы на одинаковую скорость.
- В сети не должно быть узлов с совпадающими идентификаторами.

К сожалению, нет механизмов для автоматического соблюдения этих условий. Системный интегратор должен проверить и при необходимости привести в соответствие bit rate и node-ID каждого отдельного узла сети при соединении их в общую сеть. Обычно node-ID конфигурируется непосредственно на устройстве через DIP-переключики или шестнадцатеричные роторные переключатели. Также конфигурирование может быть осуществлено через дополнительный внешний интерфейс (UART, USB, JTAG). Альтернативные решения требуют предварительную установку этих параметров через 2 зарезервированных идентификатора CAN с помощью программного обеспечения, при участии так называемой службы настройки слоя "LSS-service" (layer setting service), как это описано в CiA 305.

Логическая структура CANopen. Все стандартное 11-битное адресное пространство низкоуровневого протокола CAN поделено между службами CANopen (см. далее Диаграмму 1): NMT, Sync, TimeStamp, PDO, SDO, Guarding, LSS. Это обеспечивает доступ ко всем объектам протокола CANopen через плоскую адресацию, упрощающую взаимодействие по сети.

Обмен данными, OD и EDS. Одним из самых важных свойств CANopen является стандартизованное описание устройства (описание его функций), которое называется словарем объектов (object dictionary, OD). Это таблица, имеющая одинаковую структуру для всех типов устройств. Таким образом, это дает возможность получить доступ снаружи (т. е. через шину CAN) ко всем важным данным, параметрам и функциям устройства с использованием логической системы адресации (index, subindex).

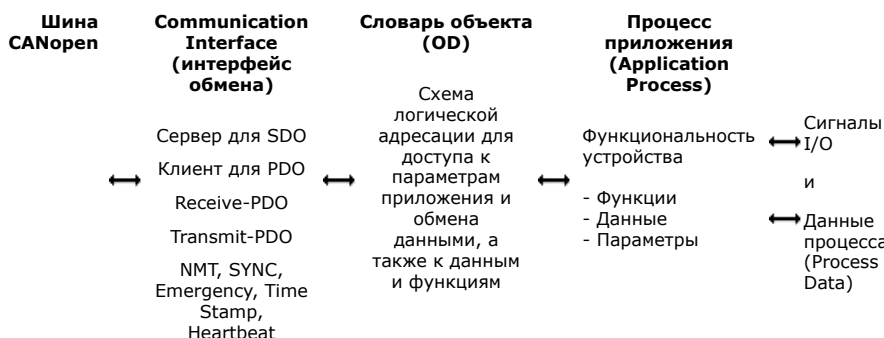


Рис. 3. Информационная схема CANopen.

Каждый элемент словаря (объект) адресуется через 16-битный индекс и 8-битный sub-индекс. В объектах записана информация об узле сети CANopen: какие данные узел принимает или передает, каким способом, текущее состояние узла.

В дополнение к стандартизованному описанию коммуникационных свойств устройств в соответствии с CiA 301 [15], CANopen определяет так называемые профили устройств ("device profiles") для типичных устройств для различных областей применения. Они указывают наиболее важные параметры, данные и функции на каждый тип устройства (например модули ввода/вывода, приводы, энкодеры, и т. д.).

Описание электронного оборудования (electronic data sheet, EDS) содержит тип данных и функции по каждой записи директории OD. Обычно EDS представляет собой файл ASCII, в котором содержатся все данные. Чтобы сделать эти данные более гибкими и расширяемыми в контексте обработки, их формат был изменен на XML.

Передача данных через SDO и PDO. Есть 2 базовых, отличающихся друг от друга способа передачи данных по протоколу CANopen. Способ service data objects (**SDO**) основан на обмене по принципу client-сервер, и позволяет использовать прямую адресацию объекта по индексу и sub-индексу (index и subindex). Он используется для конфигурации устройства, передачи больших блоков данных в обоих направлениях (upload, download), но требует дополнительной нагрузки на протокол. Поэтому SDO медленный (по сравнению с PDO) способ передачи данных. Соединение по принципу SDO осуществляется как точка-точка, с задействованием элементов словаря, и подразумевает парный обмен пакетами с наличием подтверждения получения информации. Некоторую аналогию SDO можно провести с протоколом TCP, общий принцип тот же. С помощью SDO можно передавать произвольный объем данных.



Рис. 4. Обмен SDO.

Способ process data objects (**PDO**) предоставляет эффективную передачу данных по принципу генератор-потребитель (producer-consumer). Длина пакета данных ограничена 8 байтами, однако это не накладывает излишней нагрузки на протокол, как при обмене по принципу SDO. Один PDO может содержать значения более чем одной записи из словаря объектов (OD), но содержимое PDO должно быть определено на этапе инициализации. Каждое устройство может указать до 512 объектов PDO для приема и передачи с учетом ограничений системы (по памяти, вычислительной мощности) или сети (количество доступных идентификаторов CAN).

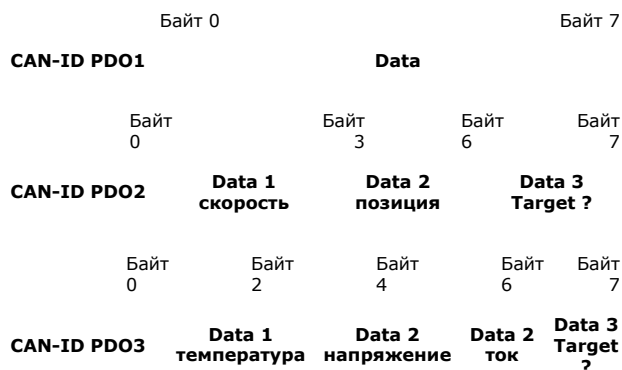


Рис. 5. Обмен PDO.

PDO запускаются либо по запросам remote, либо по внутренним событиям, таким как переполнение таймера, или когда (циклически) приходит синхронное сообщение передачи (synchronous transmission message, SYNC). Все узлы сети могут принять сообщение (потребители сообщений, PDO-Consumer, см. рисунок ниже). Этим PDO похож на сетевой протокол UDP. Для последующей обработки может быть применена фильтрация по CAN-ID только интересующих объектов.

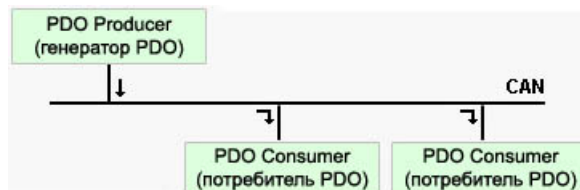


Рис. 6. Широковещание PDO.

Передача по принципу PDO осуществляется посылками, которые содержат не более 8 байт, но зато это быстрые пересылки, не требующие подтверждения которые могут использоваться для обмена данными в реальном времени.

SDO-конфигурация устройства

Процесс PDO-обмена данными

Служба SYNC

Аварийное сообщение (Emergency Message, объект EMCY)

Управление сетью CANopen (network management, NMT)

Мониторинг сети: Guarding и Heartbeat

Функционирование сети это обмен данными. Для понимания функционирования сети CANopen разделим все данные на функциональные и технологические.

Функциональные данные - те данные, которые описывают целевое функционирование системы (температура, величины управляющих воздействий исполнительных механизмов), те данные, которые передавались бы между блоками, даже если бы в качестве связующего звена использовалась линия связи отличная от CAN, например, LIN или USB, или Ethernet, или I2C.

Технологические данные - те, которые обеспечивают функционирование сети в целом, контроль корректной работы всех узлов, конфигурирование частей системы — те данные, появление которых связано с использованием сети CANopen и не зависит непосредственно от задач, решаемых системой.

Документ CiA DS-201 выделяет 4 основных группы подсистем (Fig.3 CiA DS-201):

CMS: передача сообщений. Сюда относятся: обмен функциональными данными, обмен срочными сообщениями, обмен данными по запросу, управление объектным словарём.
NMT: управление сетью, контроль работы устройств сети.
DBT: динамическое распределение идентификаторов.
LMT: управление конфигурированием устройства.

[Использование заранее определенных идентификаторов сообщений для простых систем]

Чтобы уменьшить объем конфигурирования, требуемый для простых структур сети (1: n коммуникационных взаимосвязей между управляющим устройством и несколькими устройствами более низкого уровня), CANopen поддерживает предопределенное назначение идентификаторов сообщений (Predefined Connection Set). Этот набор заранее определенных идентификаторов поддерживает одно emergency message на узел, сообщения синхронизации и меток времени, одно SDO-соединение на устройство, NMT-сообщения для управления узлами и их мониторинга, и до 4 transmit-PDO и 4 receive-PDO на устройство.

Сеть CANopen может дифференцировать между собой максимум до 127 узлов. Эти узлы совместно используют 11-битное пространство идентификаторов.

Сначала делается дифференцирование между сетевыми функциями и функциями, связанным с устройством. Один идентификатор CAN резервируется на каждую связанную с сетью функцию (например управление узлом через NMT), один идентификатор на устройство требуется для функциональности, связанной с устройством (например, аварийные сообщения, сообщения PDO), поскольку должна быть возможность дифференцировать одинаковые функции на разных устройствах. Более важные функции назначаются на более приоритетные идентификаторы COB-ID. Для будущих расширений и по историческим причинам некоторые идентификаторы сообщений не назначены. Таким образом, есть возможность работать системам наличием одного управляющего главного узла и до 127 подчиненных узлов, без необходимости переконфигурирования узлов сети.

Следующая диаграмма показывает результирующее разделение пространства идентификаторов CAN.

Диаграмма 1. Распределение адресного пространства стандартных идентификаторов CAN для элементов протокола CANopen.



Обратите внимание, что идентификаторы распределены между службами таким образом, чтобы выделить им нужный приоритет в соответствии с низкоуровневой работой протокола CAN (чем меньше значение идентификатора, тем приоритет прохождения пакета выше). Таким образом, пакеты NMT будут распространяться с самым высоким приоритетом, а LSS с самым низким.

Следующая таблица показывает выделение идентификаторов Predefined Connection Set:

Таблица 5. Принцип выделения предопределенных идентификаторов CAN.

Объект обмена (Communication object)	COB-ID, hex	Подчиненные узлы (Slave nodes)
Управление узлом NMT (NMT node control)	000	Только прием (receive)
Синхронизация (Sync)	080	Только прием (transmit)
Emergency	080 + NodeID	Передача
TimeStamp	100	Только прием
PDO	180 + NodeID	1. Transmit PDO
	200 + NodeID	1. Receive PDO
	280 + NodeID	2. Transmit PDO
	300 + NodeID	2. Receive PDO
	380 + NodeID	3. Transmit PDO
	400 + NodeID	3. Receive PDO
	480 + NodeID	4. Transmit PDO
SDO	500 + NodeID	4. Receive PDO
	580 + NodeID	Transmit

	600 + NodeID	Receive
Мониторинг узла (NMT node monitoring) по принципу node guarding или heartbeat	700 + NodeID	Transmit
LSS	7E4 7E5	Transmit Receive

SDO и PDO всегда используются парно (например для передачи и для приема), где правило состоит в том, что узел с более низким идентификатором COB-ID (и соответственно с более высоким приоритетом) передает, и с более высоким идентификатором COB-ID (т. е. с более низким приоритетом) принимает.

Как уже упоминалось, Predefined Connection Set дает возможность работать системам с одним главным узлом и до 127 подчиненных узлов, и при этом не требуется переконфигурировать что-либо. Здесь имеется старший управляющий узел, например для передачи данных процесса на узел с node-ID 5 можно использовать сообщения PDO с идентификаторами COB-ID 0x205, 0x305, 0x405 и 0x505; данные процесс от этого узла принимаются через сообщения PDO с идентификаторами COB-ID 0x185, 0x285, 0x385 и 0x485. Таким образом, по умолчанию управляющий узел может обмениваться с подчиненным узлом до 32 байтами процесса входа и 32 байтами процесса выхода. В нашем примере управляющий узел может получить доступ к словарию объекта узла номер 5 запросом SDO с идентификатором COB-ID 0x605, и принять соответствующий SDO-ответ с COB-ID 0x585.

Для более сложных сетевых структур, например с взаимосвязями n: m, или если недостаточно использовать предопределенное количество PDO на устройство, предварительное назначение идентификаторов должно быть изменено путем реконфигурирования выделения идентификаторов (параметр PDO). Для этой цели рекомендуется использовать инструментальный конфигурирования.

[Layer Setting Services (LSS)]

Стандартом четко определено, что должно обязательно быть выполнено 2 условия для взаимодействия устройств CANopen по сети: все устройства должны использовать одинаковую скорость обмена (baudrate), и идентификаторы узлов CANopen (node-ID) должны быть уникальными в пределах сети. Но что делать, если устройства не переключаются в состояние с этими свойствами?

Спецификация CANopen DS-305: Layer Setting Services (LSS) описывает, как можно настроить устройства CANopen с помощью простого протокола.

Описание протокола LSS

[Стандартные профили устройства и приложения]

На основе CiA 301 [15] как фундаментального стандарта CANopen появилось впоследствии большое количество документов, описывающих стандартные устройства или стандартные приложения. В этих дополнительных стандартах (так называемые профили устройства и приложения, device and application profiles) с помощью элементов словаря объекта (OD) определены поведение и параметры стандартизованных устройств или приложений. Цель стандартизации на основе профилей устройства состоит в том, чтобы устройствами одного класса можно было бы обмениваться, что позволяет получить независимость от конкретного производителя устройств (устройства от разных производителей становятся взаимозаменяемыми). Профили приложения должны упростить интеграцию систем, построенных на устройствах от разных вендоров. Профили обычного устройства (Generic device profile) описывают интерфейс одного устройства, в то время как профили приложения описывают все интерфейсы устройств, которые являются частью приложения. Профили устройства также могут содержать дополнительные коды ошибок, скомпилированные типы данных, светодиоды устройства (LED) и многие другие вещи. Профиль устройства обычно определяет отображение по умолчанию для первых 4 receive-PDO и transmit-PDO, представляющих наиболее общие элементы OD, специфические для профиля. Таким образом, устройство со стандартным профилем может непосредственно, прямо "из коробки" использоваться без необходимости настройки его элементов словаря, относящихся к параметрам и обмену данными.

Нельзя полностью описать устройство во всех возможных вариантах. Таким образом, все профили позволяют определять специфические для вендора свойства внутри так называемого диапазона профиля производителя ("vendor-specific profile range"). Таким способом можно описать функции, атрибуты и параметры, которые не содержатся в стандартном профиле.

Стандарт CiA 401 ("Device Profile for I/O Modules") известен как описание наиболее важного профиля устройства. Этот профиль описывает аналоговые и цифровые интерфейсы ввода и вывода, и их возможности по назначению параметров. CiA 401 задает элементы OD для максимум 2040 цифровых входов/выходов и до 255 аналоговых входов/выходов. Помимо стандартизованных элементов словаря для текущих значений есть ряд продолжающихся элементов OD для управления параметрами поведения этих входов и выходов.

Другим очень важным профилем устройства является CiA 402, это профиль устройства для драйверов с интерфейсом CANopen ("Drives and Motion Control"). Этот профиль покрывает сервоконтроллеры, шаговые двигатели и частотные преобразователи. Наподобие CiA 401, этот стандарт также основан на модели для описания поведения привода. Модель привода описывает машину состояний и поддерживает среди разных других параметров работу режима позиционирования, режима ускорения и режима крутящего момента. Здесь два из таких наиболее важных и поэтому обязательных элемента OD: управляющее слово ("control word") и слово состояния ("status word") для установки и получения обратно режима привода и его статуса. Они отображены первыми в каждом PDO по умолчанию.

В следующей таблице перечислены определенные в настоящий момент профили устройства и приложения CANopen:

Таблица 7. Профили устройства CANopen.

Номер профиля	Класс устройства
CiA 401	Обычные модули ввода вывода (Generic I/O Modules)
CiA 402	Управление приводами (Drives and Motion Control)
CiA 404	Измерительные устройства и контроллеры управления с обратной связью (Measuring devices and Closed Loop Controllers)
CiA 405	Программируемые устройства, ПЛК (IEC 61131-3 Programmable Devices)
CiA 406	Круговые и линейные энкодеры (Rotating and Linear Encoders)
CiA 408	Гидравлические приводы и линейные вентили (Hydraulic Drives and Proportional Valves)

CiA 410	Измерители угла наклона (Inclinometers)
CiA 412	Медицинское оборудование (Medical Devices)
CiA 413	Грузовые шлюзы (Truck Gateways)
CiA 414	Текстильное и ткацкое оборудование (Yarn Feeding Units, Weaving Machines)
CiA 415	Машины для дорожного строительства (Road Construction Machinery)
CiA 416	Управления дверьми зданий (Building Door Control)
CiA 417	Управление лифтами (Lift Control Systems)
CiA 418	Батарейные модули (Battery Modules)
CiA 419	Зарядные устройства для батарей (Battery Chargers)
CiA 420	Оборудование штамповки, волочения, гибки, резки (Extruder Downstream Devices)
CiA 422	Городской транспорт (Municipal Vehicles – CleANopen)
CiA 423	Системы управления дизельными машинами ЖД (Railway Diesel Control Systems)
CiA 424	Управление дверями поездов ЖД (Rail Vehicle Door Control Systems)
CiA 425	Дополнительные модули медицинской диагностики (Medical Diagnostic Add-on Modules)
CiA 445	Устройства идентификации (RFID Devices)

Более подробное описание содержимого этих профилей можно найти на сайте can-cia.org [1].

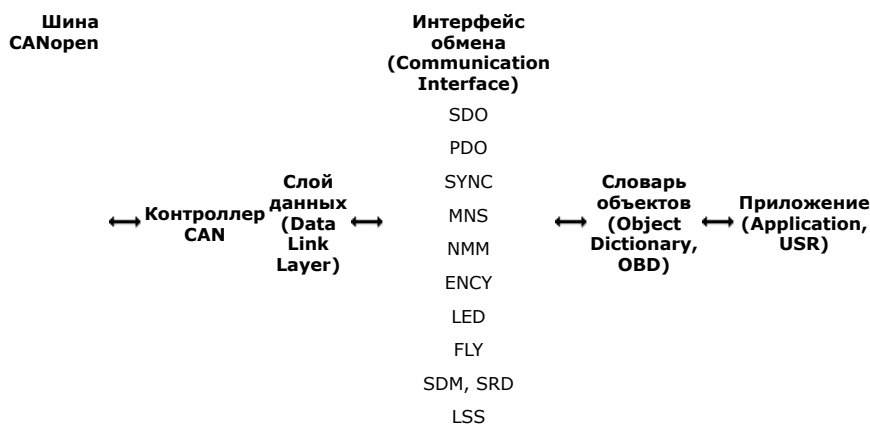
[Реализация CANopen]

Для реализации CANopen во встраиваемых системах используются стеки протоколов, которые обычно предоставляются в форме лицензий компаний-разработчиков. Компания HMS предоставляет под брендом IXXAT всесторонне протестированный стек протоколов CANopen либо в обычном коде языка C, либо в виде готовой адаптации к большому количеству удобных стеков для различных микроконтроллеров. Стек CANopen доступен как "Slave" или "Master/Slave", обкатанный на множестве приложений. Термины "Slave" и "Master" относятся к различной функциональности устройства в контексте управления сетью, что реализовано протоколом CANopen в форме логической взаимосвязи главное/подчиненное устройство (Master/Slave).

Готовый стек CANopen предоставляет разработчику полную функциональность, требуемую для обмена данными, управления сетью, общих системных служб и описания устройства. Таким образом, реализация системы на базе купленного стека протокола заключается в подключении функционала приложения к стеку протокола.

Структура ПО протокола CANopen. Стек IXXAT CANopen от компании HMS разработан в масштабируемой, модульной форме, в соответствии с различными службами CANopen, такими как Network Management (NMT), Process Data Objects (PDO), Service Data Objects (SDO), Emergency (EMCY), Synchronization (SYNC), SDO-Manager (SDM/SDR), Layer Setting Services (LSS) и Object Dictionary (OBD). Все модули имеют доступ к словарю объектов, который представляет центральный экземпляр программного пакета как мост между коммуникационным интерфейсом и данными, параметрами и функциями приложения.

Доступ к контроллеру CAN (для передачи и приема сообщений) осуществляется через интерфейс "Data Link Layer", который интегрирован в отдельный модуль (DLL). Это позволяет быстро и просто адаптировать ПО протокола CANopen к разным реализациям контроллера CAN.



Структура программного обеспечения IXXAT CANopen stack

Рис. 30. Принцип организации стека CANopen компании IXXAT.

Простое конфигурирование и оптимизированные требования к памяти. Функционал стека IXXAT CANopen может быть просто адаптирован под требования приложения. Для этого разработчик просто должен активировать или деактивировать функции/модули как определения ("define") в центральном конфигурационном файле.

Для реализации подчиненного устройства с довольно полезным функциональным набором на 16-разрядном микроконтроллере требуется примерно 13 килобайт ROM и 1 килобайт RAM. Если у разработчика имеется меньше доступных ресурсов, то эти значения могут быть дополнительно уменьшены с помощью специальных модификаций.

HMS предоставляет программное обеспечение протокола CANopen (IXXAT CANopen protocol software) как кросс-платформенный исходный код C, или код, специально адаптированный к применению к определенным моделям микроконтроллеров и контроллеров CAN. Таким образом, разработчик получает большую степень свободы к адаптации и управлению программного обеспечения, как если бы он разрабатывал его сам.

С начала работ по стандартизации CANopen компания HMS активно участвовала в организациях CANopen. Поэтому HMS определенно может реализовать самые последние стандарты на самом раннем этапе их появления. HMS инвестирует значительный объем работ в непрерывное продвижение и обслуживание своих стеков CANopen. Посредством соглашения о поддержке программного обеспечения клиенты HMS могут участвовать в этом непрерывном дальнейшем развитии и извлечь выгоду из поддержки квалифицированных специалистов CANopen. Чтобы предоставить пользователям высококачественные продукты, HMS всегда проверяет свой стек протоколов на совместимость тестовым ПО от CiA [1]. Фактически ПО IXXAT успешно используется во многих областях применения уже много лет, гарантируя пользователям бесперебойное, успешное внедрение CANopen в свои продукты.

Разрабатывать или купить? Снова и снова разработчики решают для себя вопрос - стоит или нет разрабатывать ПО протокола CANopen самостоятельно, потому что есть несколько свободно доступных альтернатив без необходимости платить за лицензию, и даже есть несколько альтернатив с открытым исходным кодом. Конечно, это интересная проблема в плане разработки. Однако с точки зрения это стоит дешево, поскольку зрелое ПО протокола CANopen предлагается сейчас несколькими производителями по вполне привлекательным ценам.

Поскольку разработка достаточно задокументированного протокола стека протокола CANopen займет скорее всего как минимум 4..6 месяцев, даже если не будет реализован полный набор спецификаций стандарта. Однако все еще неизвестно, является ли такая "начальная реализация" достаточно оптимизированным, безотказным продуктом, и корректно ли реализованы тонкости прямой спецификации CANopen. Возможное применение бесплатных или открытых реализаций мало чем тут поможет. Большинство из этих подходов реально интересны только для тренировки или других некоммерческих применений.

Если мы предположим, что собственная разработка стека протокола CANopen может быть реализована за 4 человеко-месяца при почасовой оплате 60 евро, то стоимость такой разработки может быть больше чем 35000 евро. Однако это очень оптимистичная прикидка, и хорошо протестированное и надежное решение с предоставленным исходным кодом может быть получено за часть от этой цены. Таким образом, самостоятельная разработка стека протоколов становится полностью неинтересной с точки зрения экономики, включая время выхода конечного продукта на рынок.

Что в себя включает готовая реализация CANopen? HMS предоставляет свой IXXAT CANopen protocol stack с подробной документацией и примерами программ. Файлы проекта предоставлены вместе с примерами программ, которые позволяют осуществить прямую интеграцию в соответствующее рабочее окружение производителя компилятора. Все примеры программ могут быть непосредственно запущены на базовой платформе (оценочная плата разработчика или интерфейсная карта IXXAT).

Реализация выполнена следующим образом:

- Адаптация аппаратуры к целевой платформе (таймер, система прерываний).
- Создание элементов словаря объектов.
- Адаптация конфигурационного файла.
- Компиляция и тестирование. Для тестирования приложение примера может быть перенесено на целевую платформу. Это можно взять также как базу для собственного приложения.

Ранние реализации CANopen на целевой системе можно было настроить за несколько дней.

Альтернативно реализация может быть осуществлена компанией HMS. В этом случае требования к проекту определяются вместе с заказчиком, и затем реализуются разработчиками HMS. Поскольку работа выполняется квалифицированными инженерами, которые хорошо знакомы с этим программным обеспечением, то это представляет недорогую, быструю в реализации альтернативу для самостоятельной реализации.

Доступно программное обеспечение и инструментарий CANopen. Обзор доступных стеков протоколов IXXAT можно найти во врезке "Стеки протоколов для встраиваемых систем IXXAT", или на сайте IXXAT. Общее описание полезного инструментария для разработки и быстрого старта, как canAnalyser, CANopen Device Manager, CANopen ConfigurationStudio (базовая версия) можно найти во врезке "Инструментарий и интерфейсы IXXAT".

Стеки протоколов для встраиваемых систем IXXAT

Инструментарий и интерфейсы IXXAT

[Другие платные инструменты]

CANopen Magic

[Свободный инструментарий для CANopen]

IXXAT Device Description Explorer for CANopen and POWERLINK. Это свободный от выплат инструмент от компании HMS позволяющий пользователю инспектировать и конвертировать файлы описания устройства CANopen и POWERLINK в соответствии со стандартами CiA 306 и CiA 311, по спецификации EPSG DS 311 XML Device Description.

canAnalyser - Demo. Это мощный и разносторонний инструмент для разработки, тестирования и обслуживания сетей CAN и CANopen. Страница загрузки: [Demos Software & Tools site:ixxat.com](http://DemosSoftware&ToolsSite:ixxat.com).

MicroCANopen. Существует в 3 версиях: MicroCANopen, MicroCANopen Classic и MicroCANopen Plus. Из них бесплатна только первая версия, с сильно урезанными возможностями. Частности, в примерах для ARM среды разработки IAR есть демонстрационные проекты CANopen на основе урезанной версии MicroCANopen (проект basic-microcanopen-project). Подробнее про MicroCANopen см. описание на сайте canopenstore.com, строка для поиска [MicroCANopen site:canopenstore.com](http://MicroCANopenSite:canopenstore.com).

CanFestival. Возможно, это самая продвинутая библиотека из бесплатных, с исходным кодом. Портитована на многие популярные встраиваемые платформы. Хорошее описание протокола CANopen и работы с утилитами CanFestival можно найти по ссылке [10].

CANopen DeviceExplorer (Demo). Утилита для диагностики сети, урезанную версию можно скачать с сайта emtas.de (строка для поиска [CANopen DeviceExplorer site:emtas.de](http://CANopenDeviceExplorerSite:emtas.de)), исполняемый файл инсталлятора будет называться наподобие setup-emtas-cde-2_6_2.exe. Достоинство утилиты в том, что она поддерживает множество заводских адаптеров, в том числе и USB-CAN адаптеры SYSTEC (тип адаптера выбирается на этапе инсталляции). Под Windows после установки запускайте программу под учетной записью администратора.

Вы можете по адресу service@emtas.de послать запрос на получение лицензии, в ответ Вам пришлют 30-дневную бесплатную лицензию (файл наподобие [emtas_XXXXX_Demo_YYYYMMDD_cde_2-x-x.lidat](#)). Полнофункциональная персональная лицензия стоит 780 евро, для её получения также требуется предоставить свое имя и адрес.

AN945, стек CANopen от Microchip [14]. Библиотека для микроконтроллеров PIC18 с низкими требованиями к FLASH и RAM. Исходный код присутствует, при желании можно портировать на другие платформы.

[Словарик]

Application object (прикладной объект). Функциональность устройства, которую оно предоставляет, описывается прикладными объектами. Прикладные объекты могут быть читаемыми или записываемыми параметрами устройства, данными или функциями. К прикладному объекту можно получить доступ через однозначный адрес в OD.

bps, kbps bit per second, kilobit per second. Единица измерения физической скорости следования бит.

CANopen object (объект CANopen) Функциональность CANopen-устройства, видимая через шину, описывается объектами CANopen. Объектами CANopen могут быть данные, параметры или функции устройства. Объект может быть идентифицирован в объектном словаре с помощью 16-битного индекса и 8-битного sub-индекса.

CiA аббревиатура от CAN in Automation, название организации, которая разработала спецификацию CANopen.

CiA-301 коммуникационный профиль CANopen [15]. Обязательная для всех устройств CANopen спецификация коммуникационной модели и структура OD. Начиная с версии 4.0 были включены CMS и NMT, DBT был удалён, и LMT перешел в LSS.

CiA-302 общая спецификация для программируемых CANopen устройств []. Среди прочего содержит предварительные определения для CiA-405.

CiA-401 CANopen-профиль устройства для стандартных модулей ввода/вывода.

CiA-402 CANopen-профиль устройства для приводов.

CiA-405 CANopen-профиль устройства для IEC-1131 программируемых устройств.

CiA-406 CANopen-профиль устройства для датчиков положения.

Client-SDO (клиент SDO). Клиент SDO обозначает инициатора передачи SDO. Оно имеет доступ к записи объектного словаря "сервера SDO". Обычно клиентом выступает мастер шины CANopen, а сервером подчиненное устройство CANopen. Мастер через объекты SDO получает параметры из словаря OD подчиненного устройства.

CMS CAN-based Message Specification, спецификация сообщений, основанная на протоколе CAN. Предоставляет объекты типов Variable, Event и Domain для разработки и указания, как функциональность устройства (узла сети CAN) может быть доступна через интерфейс CAN. Т. е. как выгружать или загружать набор данных ('domain'), длина которых превышает максимальный размер 8 байт элементарного сообщения CAN, включая функцию обрыва передачи ('abort transfer').

COB communication object, объект обмена данными. Это сообщение, которое передается по CANopen. Данные передаются посредством COB.

COB-ID / COBID (идентификатор коммуникационного объекта). COB-ID создает коммуникационное соединение между передаваемым и принимаемыми коммуникационными объектами, и одновременно определяет приоритет сообщения. Это стандартное свойство сети CAN - идентификатор с меньшим номером имеет приоритет выше. Идентификатор 0 с наивысшим приоритетом зарезервирован для сервисов управления сетью (NMT).

COS Change Of State, информация об изменении состояния. Когда объект PDO работает под управлением событий (Event Driven PDO), то он автоматически передает COS. Это обычно происходит, когда имеется новая информация (например, поменялось значение на входе). Чтобы избежать чрезмерного забития шины, может быть задано время запрета (inhibit time).

Communication cycle period (период коммуникационного цикла). Период коммуникационного цикла определяет интервал времени между последовательными объектами синхронизации.

Communication parameters (коммуникационные параметры). Атрибуты PDO описаны в его коммуникационных параметрах. Эти атрибуты включают в себя тип передачи, время подавления (время запрета, inhibit time) и идентификатор COB-ID.

DBT DistriBuTor, динамическое распределение идентификаторов.

DCF Device Configuration File, файл конфигурации устройства. DCF-файл описывает реальное, существующее, настроенное устройство в сети. Структура DCF файла соответствует структуре EDS файла, плюс имеется специфичная для проекта конфигурация этого устройства. Среди прочего конфигурация содержит скорость передачи данных, PDO отображение (PDO mapping), специфичное для проекта название устройства, устанавливает идентификатор узла и параметризацию прикладных объектов.

Device profiles (профили устройств). Функциональность устройства описывается с помощью стандартных функций в области стандартизированного профиля устройства, а для специфичных для производителя функций устройства – в области специфичного для производителя профиля устройства.

Dummy / Dummy entry (фиктивная, или пустая запись). Фиктивное отображение необходимо для заполнения пропусков в отображении принимаемого PDO. Например, если объект PDO имеет длину 8 байт, но в нем имеют отображение полезной нагрузки только последние 2 байта, то тогда первые 6 байт должны быть отображены на фиктивные записи.

Если типы данных (с индексами 1..7) отображены, они обслуживаются как фиктивные. Соответствующие данные в PDO устройством не вычисляются. Эта опциональная функция полезна в тех случаях, когда передаются данные на несколько устройств через один и тот же PDO, но каждое из устройств использует определенную, свою отдельную часть этого PDO. Нельзя создавать фиктивное отображение для TPDO.

EDS electronic data sheet, описание электронного оборудования. EDS описывает функциональность устройства. Этот файл должен быть предоставлен разработчиком / произ-водителем устройства CANopen. EDS-файл содержит общие и специфичные данные устройства, некоторую статистическую информацию о самом файле и детальное, полное описание OD.

Emergency object / EMCY (объект аварии). Высокоприоритетным объектом аварии устройство сигнализирует о наступлении неустраняемой внутренней ошибки устройства или о сбросе одной или всех внутренних ошибок устройства. Поддержка сообщения об ошибке устройства является необязательной. Аварийный код ошибки указывает тип ошибки в соответствии с CiA-301.

Granularity (Гранулярность). Максимально возможное количество объектов, которые могут быть введены в PDO, определяется гранулярностью (длине объекта в битах) прикладных объектов. Максимальный размер поля данных PDO – 8 байт данных. Таким образом при гранулярности 8 не более чем 8 байтовых прикладных объектов может быть отображено на PDO. При гранулярности 1 поддерживается ровно 64 булевых прикладных объекта.

Granularity задается в разделе [DeviceInfo] словаря OD. Если здесь задано 0, то это означает, что отображение модифицировать нельзя.

Guard time (время охраны). Относится к протоколу Guarding, предназначенному для отслеживания работоспособности узлов сети CANopen. NMT-мастер циклически передает запрос к подчиненному устройству, чтобы получить его текущее состояние узла. На данный запрос должен быть дан ответ в течении времени жизни узла (node life time). Время жизни узла является результатом умножения фактора времени жизни (life time factor) на время охраны узла. Подчиненный узел не проводит мониторинг NMT мастера, если время охраны = 0. Тем не менее он отвечает на протокол Guarding. Нарушения охраны узлов описаны в CANopen-спецификации CiA301 [15].

HMS аббревиатура обозначает название компании HMS Industrial Networks, которая предоставляет программно-аппаратные решения для CANopen.

Inhibit time (время подавления, или время запрета). Объект данных процесса (PDO) может быть повторно передан только после истечения этого времени.

IXAAT бренд компании HMS для дистрибуции решений CANopen.

LMT Line Management, управление конфигурированием устройства.

LSB Least Significant Bit, младший значащий бит.

LSS Layer Setting Services, службы CANopen, предназначенные для первоначальной настройки узлов сети.

NCF Network Configuration File, файл конфигурации сети.

NMT Network Management, система обслуживания сети. Сервисный элемент прикладного уровня, который состоит из конфигурации, инициализации и контроля ошибок сети, а также процесса синхронизации во всей сети CANopen. Управление сетью имеет структуру мастер/подчиненный.

Node guarding (охрана узла). Циклический мониторинг узла. Специальный протокол, предназначенный для отслеживания работоспособности узлов сети CANopen.

Node-ID (идентификатор узла) Отдельное устройство CANopen определяется в сети по его номеру узла (между 1 и 127). Этот номер конфигурируется при настройке сети (либо перемычками, либо специальным ПО, либо по протоколу LSS). Идентификатор узла 0 зарезервирован для служб NMT.

OD, OBD object dictionary, словарь объектов. Объектный словарь это структура данных, через которую можно обратиться ко всем объектам CANopen-устройства. OD разделен на область с общей информацией об устройстве, такой как название производителя и т. д., а также область, которая содержит параметры связи, и область, которая описывает специфичную для устройства функциональность. Через записи (объекты) OD прикладные объекты устройства, такие как входные и выходные сигналы, параметры устройства, сервисы устройства или сетевые переменные становятся доступными по сети в стандартизированной форме. OD образует интерфейс между сетью и прикладным ПО.

OD entry (запись объектного словаря). Информационная запись OD.

PDO process data object (объект данных процесса). Специальный объект CANopen, предназначенный для обмена полезными (прикладными) данными по сети CAN. PDO представляет фактические средства транспорта для передачи данных процесса. PDO передается "поставщиком" (producer, продюсер) и может быть получен одним или несколькими "потребителями" (consumer). Данные процесса, передаваемые поставщиком в PDO, могут состоять максимум из 8 байт. PDO передается без подтверждения, и требует идентификатора, однозначно назначенного для PDO. Значение передаваемых данных определяется идентификатором, который они используют, и отображением на PDO (PDO mapping), назначенной для PDO. Приоритет и режим работы PDO определяются параметрами, специфичными для соединения. Для управления PDO и поставщики PDO, и потребители PDO требуют соответствующих структур данных. Данные, необходимые поставщику PDO, организованы в виде так называемых TxPDO / TPDO записей OD; данные, которые будут получены потребителем PDO, организованы в виде так называемых Rx-PDO / RPDO записей OD.

PDO linking (PDO связывание). PDO связывание представляет собой коммуникационное соединение между передаваемым PDO и соответствующими принимаемыми PDO. Коммуникационная связь образуется путем назначения одинакового PDO идентификатора для передачи и приема PDO.

PDO mapping (отображение объектов на PDO). Распределение поля данных PDO (максимум 8 байт) между прикладными объектами определяется PDO отображением. Оно может быть статическим (то есть постоянным) или динамическим (то есть изменяемым).

Predefined connection set (заранее заданное распределение идентификаторов). Предопределенное распределение идентификаторов означает предопределенное назначение CAN, основанное на идентификаторе узла и коде функции. Для следующих коммуникационных объектов предопределенное распределение идентификаторов регулирует значение для COB-ID: охрана узла (Node guarding) /сердцебиение (heartbeat), аварийный объект (EMCY), посылка синхронизации (SYNC), временная метка (TIME), сервер SDO1, RPDO1 .. RPDO4 и TPDO1 .. TPDO4.

RPDO принимаемый объект данных процесса. См. также PDO.

Scan timeout (тайм-аут сканирования). Интервал времени, в течение которого устройство должно ответить, после того как было вызвано: для того, чтобы быть признанным присутствующим в сети CANopen.

SDO service data object - специальный объект CANopen, предназначенный для обслуживания узлов сети CAN. Это коммуникационный объект CANopen, используемый для конфигурирования и параметризации CANopen устройств. Он может использоваться для передачи больших объемов данных (не ограниченных 8 байтами, как в PDO). Записи OD устройства могут быть доступны для чтения или записи через SDO. Необходимая запись объектного словаря адресуется через индекс и sub-индекс. SDO формирует прямой 1:1 канал связи между любыми двумя узлами в стиле клиент-сервер. Клиент запрашивает данные для чтения или записи, а сервер предоставляет данные. Обычно в качестве клиента выступает мастер сети, а сервером подчиненные узлы сети.

SDO timeout (тайм-аут SDO). На SDO-запрос должен быть дан ответ в течение времени тайм-аута. Время тайм-аута задается в миллисекундах.

Server SDO (сервер SDO). Каждое устройство должно поддерживать как минимум один сервер SDO, и таким образом предоставить доступ к записям в его OD. Спецификация объекта сервера SDO требует одного CAN идентификатора, определенного для каждого направления передачи, поскольку он является подтверждаемым сервисом. CAN идентификаторы первого сервера SDO (SDO1) зависят от идентификатора узла, и они строго регламентированы.

Примечание: некоторые простые устройства CANopen, которые не поддерживают конфигурирование, вообще не предоставляют поддержку протокола SDO. В таком случае для их подключения / опознания в сети требуется загрузка файла EDS.

SI unit International System of Units (SI-Units), интернациональная система единиц измерений. Представляет набор физических единиц системы измерения. Стандартизована серией международных спецификаций ISO/IEC 80000. Этот стандарт также определяет международную систему International System of Quantities (ISQ). Система SI-Units является почти глобально адаптированной: только Liberia, Myanmar и США не адаптировали SI-Units как свою официальную систему мер и весов.

Sync object / SYNC (объект синхронизации). Объект синхронизации используется для синхронизированного сбора данных, синхронизированного командного стробирования и циклической передачи данных процесса. Прием объекта синхронизации запускает обновление и передачу синхронных сообщений. Для этого одно устройство (поставщик синхронизации, SYNC producer) циклически передает высокоприоритетные объекты синхронизации. Для полного описания объект синхронизации требует задания параметра периода коммуникационного цикла и параметра длины синхронного окна. Если параметр инициализируется 0, он не имеет никакого эффекта.

Synchronous window length (длина окна синхронизации). Интервал времени после получения объекта синхронизации, когда должен быть отправлен PDO, имеющий тип синхронной передачи.

Timestamp message (сообщение метки времени). Используется для повторной синхронизации локальных таймеров, чтобы обеспечить более высокие требования базиса синхронизации для всех устройств системы.

Transmission type (тип передачи). Режим работы PDO указывается в коммуникационном профиле устройства через параметр "тип передачи". CANopen предоставляет следующие типы передачи для PDO: Synchronous (синхронная) - передача зависит от объекта синхронизации, либо Asyclic (асинхронная) - один раз или циклически, при каждом приеме или после некоторого количества объектов синхронизации, заданного через скорость передачи данных.

Asynchronous (асинхронная) - передача инициируется специфичным для производителя событием или событием, определенным в профиле устройства. Remote (удалённая) - передача происходит только после RTR-запроса другим подписчиком (потребителем PDO).

Transmission rate (скорость передачи). В режиме циклического синхронного PDO это значение представляет число сообщений синхронизации, которое должно быть получено до того, как будет разрешена повторная передача PDO.

TPDO (передаваемый объект данных процесса). Передаваемый PDO (см. PDO).

ПЛК программируемый логический контроллер.

узуфрукт из Википедии: (лат. usus — использование, лат. fructus — доход) — вещное право пользования чужим имуществом с правом присвоения доходов от него, но с условием сохранения его целостности, ценности и хозяйственного назначения. Предметом узуфрукта могут быть вещи, потребление которых возможно без их уничтожения, например, земельные участки, животные и, в классическом римском праве, рабы; 1) денежный капитал не может быть предметом узуфрукта. 2) Устанавливается пожизненно, на определённый срок, или с условием, наступление которого прекращает право узуфруктуария (пользователя имущества).

[Ссылки]

1. CANOpen site:can-cia.org.
2. CANOpen Basics site:canopensolutions.com.
3. CANOpen – The standardized embedded network site:can-cia.org.
4. Free software CANOpen framework site:canfestival.org.
5. CANOpen Bootloader Protocol Stack site:microcontrol.net.
6. CANOpenNode: CANOpen based stack for communication in embeded control systems site:sourceforge.net.
7. Arduino Generic CAN Open Node site:github.com.
8. CANFestivino: Arduino Library Version of CANFestival CANOpen Stack site:github.com.
9. CANOpen MCP2515.
10. Протокол CanOpen site:robot-develop.org.
11. Порядок следования байт (endianness).
12. MCP2515: контроллер шины CAN с интерфейсом SPI.
13. CANOpen high-level protocol for CAN-bus site:nikhef.nl.
14. AN945: стек CANOpen для PIC18 ECAN.
15. CiA301: слой приложения и профиль коммуникации CANOpen.

Комментарии

#1 **Евгений** 10.06.2019 09:03 +1
Есть объект 1800h: Параметры TPDO 1
Индекс 1800h
Название TPDO 1 communication parameter
Тип данных PDO parameter record

Индекс 00h
Название highest sub-index supported
Стандартное значение 02h
Индекс 01h
Название COB-ID used by TPDO
Стандартное значение \$NODEID+180h
Индекс 02h
Название transmission type
Стандартное значение FEh
Индекс 05h
Название event timer
Стандартное значение 01F4h

Объект 1A00h: Отображение TPDO 1
Название TPDO 1 mapping parameter
Тип данных PDO parameter record
Описание элемента
Индекс 00h
Название number of mapped objects
Стандартное значение 10h
Описание элемента
Индекс 01h
Название mapped object 1 – System state
Описание элемента
Индекс 02h
Название mapped object 2 – Active kit code

Как выдать через CAN данные по индексам 01h..10h?

microsin: судя по описанию, максимальный поддерживаемый индекс у этих объектов 02h, поэтому получить данные по индексам 03h..10h не получится.

[Цитировать](#)

[Обновить список комментариев](#)
[RSS лента комментариев этой записи](#)

Добавить комментарий

Имя (обязательное)
 E-Mail (обязательное)
 Сайт



Осталось: 1000 символов

☐ Подписаться на уведомления о новых комментариях



Обновить

Отправить

