

# *Интерфейс связи CAN*

# История развития



Specified late 80s by  
Robert Bosch, Germany



ISO/OSI Standard  
SAE Standard

- 1991 CAN
- 1992 CANKingdom
- 1993 ISO11898
- 1994 DeviceNet
- 1994 SDS
- 1995 29-bit (CAN2.0B)
- 1996 CANopen

# CAN-интерфейс

CAN-протокол был разработан фирмой Robert Bosch GmbH для использования в автомобильной электронике, отличается повышенной помехоустойчивостью, надежностью и обладает следующими возможностями:

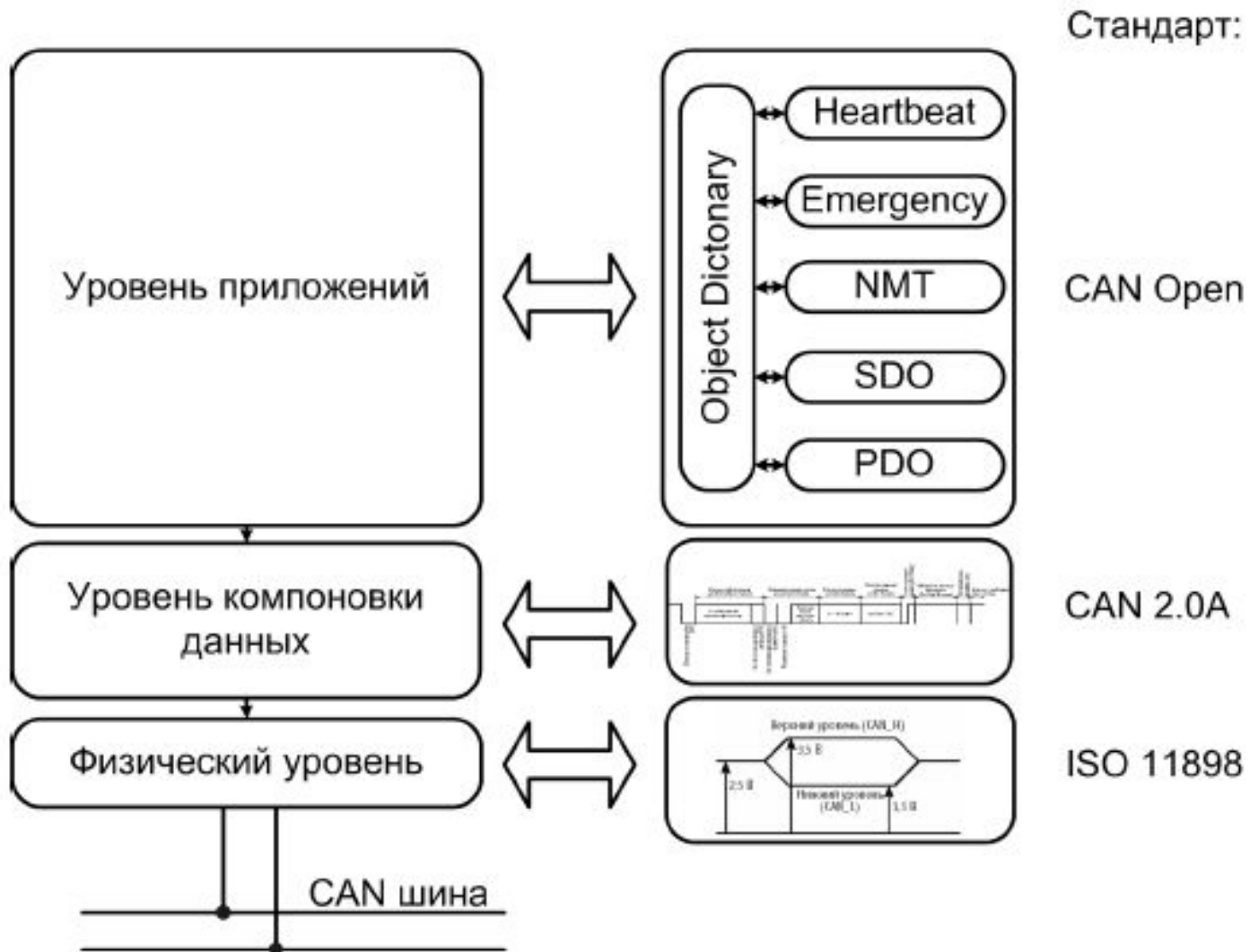
- работает по витой паре на расстоянии до 1 км.
- конфигурационная гибкость,
- получение сообщений всеми узлами с синхронизацией по времени,
- неразрушающий арбитраж доступа к шине,
- режим мультимастер,
- обнаружение ошибок и передача сигналов об ошибках,
- автоматическая передача сбойных сообщений при получении возможности повторного доступа к шине,
- различие между случайными ошибками и постоянными отказами узлов с возможностью выключения дефектных узлов,

# CAN-интерфейс

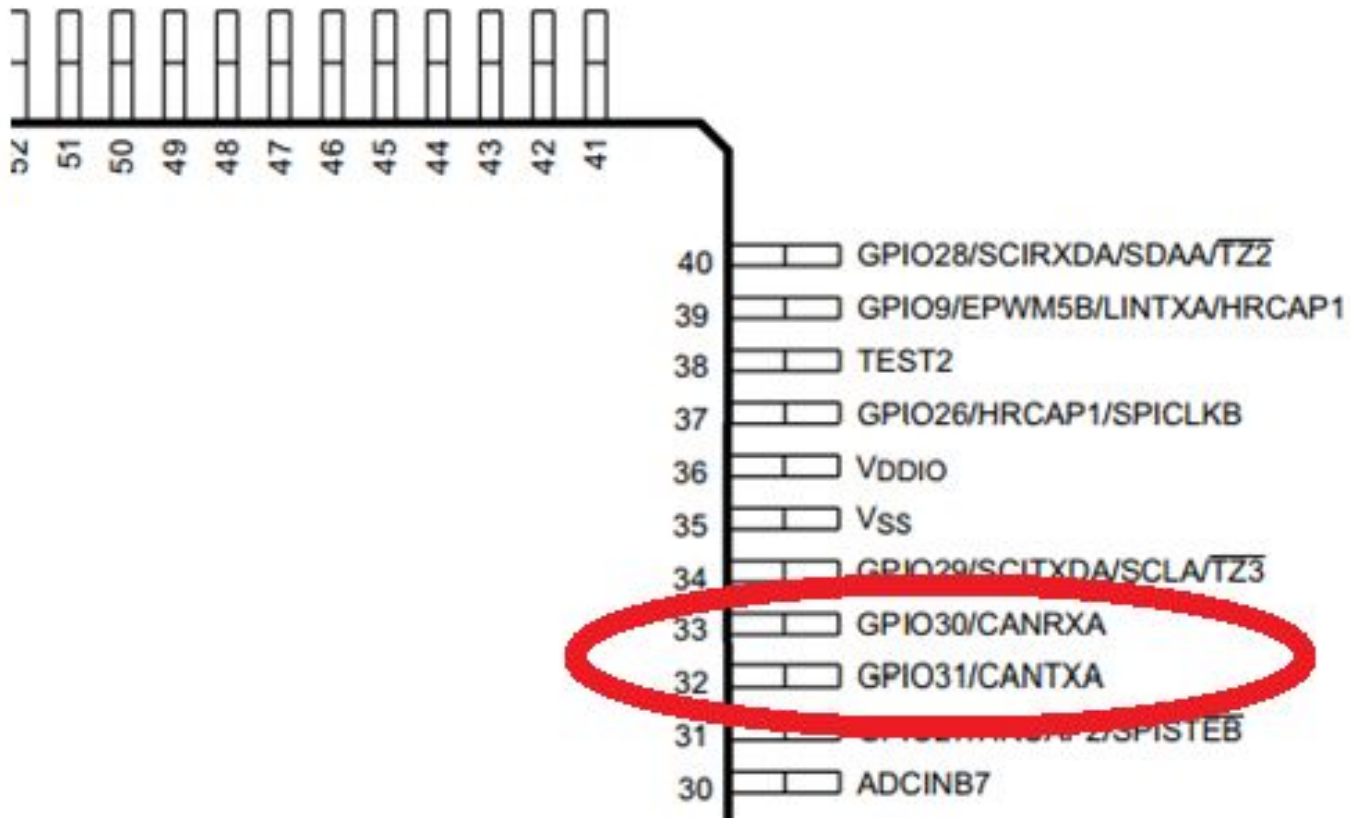
CAN-протокол распространяется на следующие уровни:

- Физический уровень определяет, как именно будут передаваться сигналы, их электрические уровни и скорость передачи.
- Транспортный уровень представляет собой ядро CAN-протокола. Он отвечает за синхронизацию, арбитраж, доступ к шине, разделение посылок на фреймы, определение и передачу ошибок и минимизацию неисправностей.
- Объектный уровень обеспечивает фильтрацию сообщений и обработку сообщений и состояний.

# CAN-интерфейс

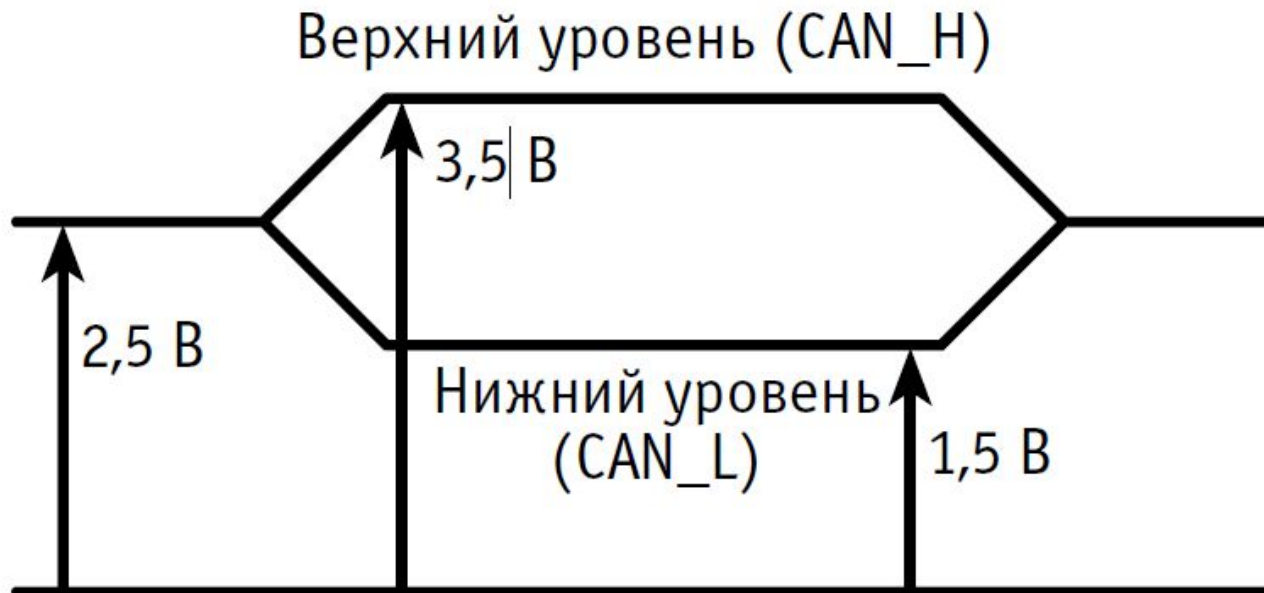


# Физический уровень



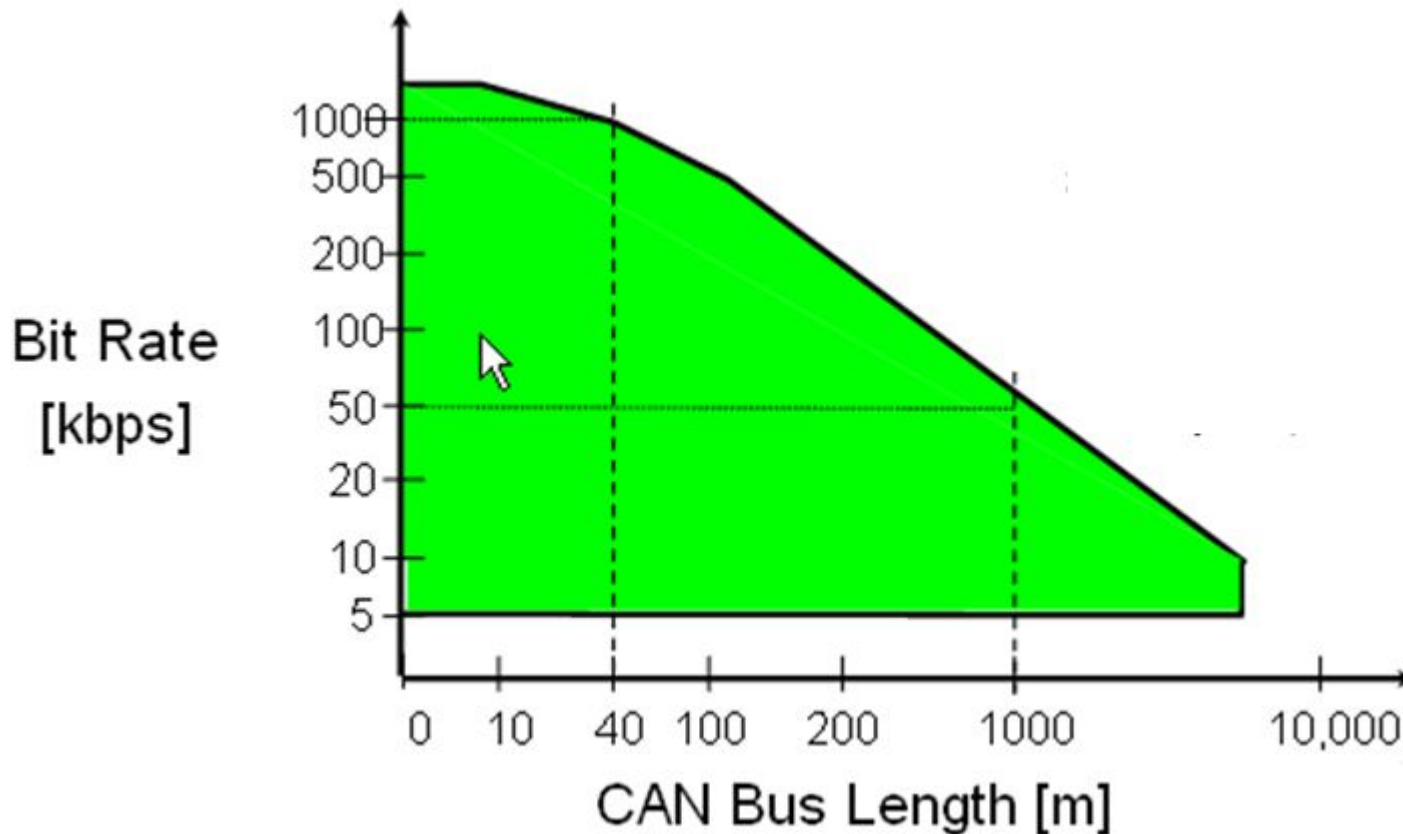
# Физический уровень

- Дифференциальное включение приемопередатчиков обеспечивает подавление синфазной помехи.
- Уровень сигналов составляет 1/3 от значения напряжения питания.
- Напряжение питания не определяется жестко.
- Максимальное расстояние между узлами — до 1 км.
- Скорость обмена до 1 Мбит/с при длине линии 60 м.



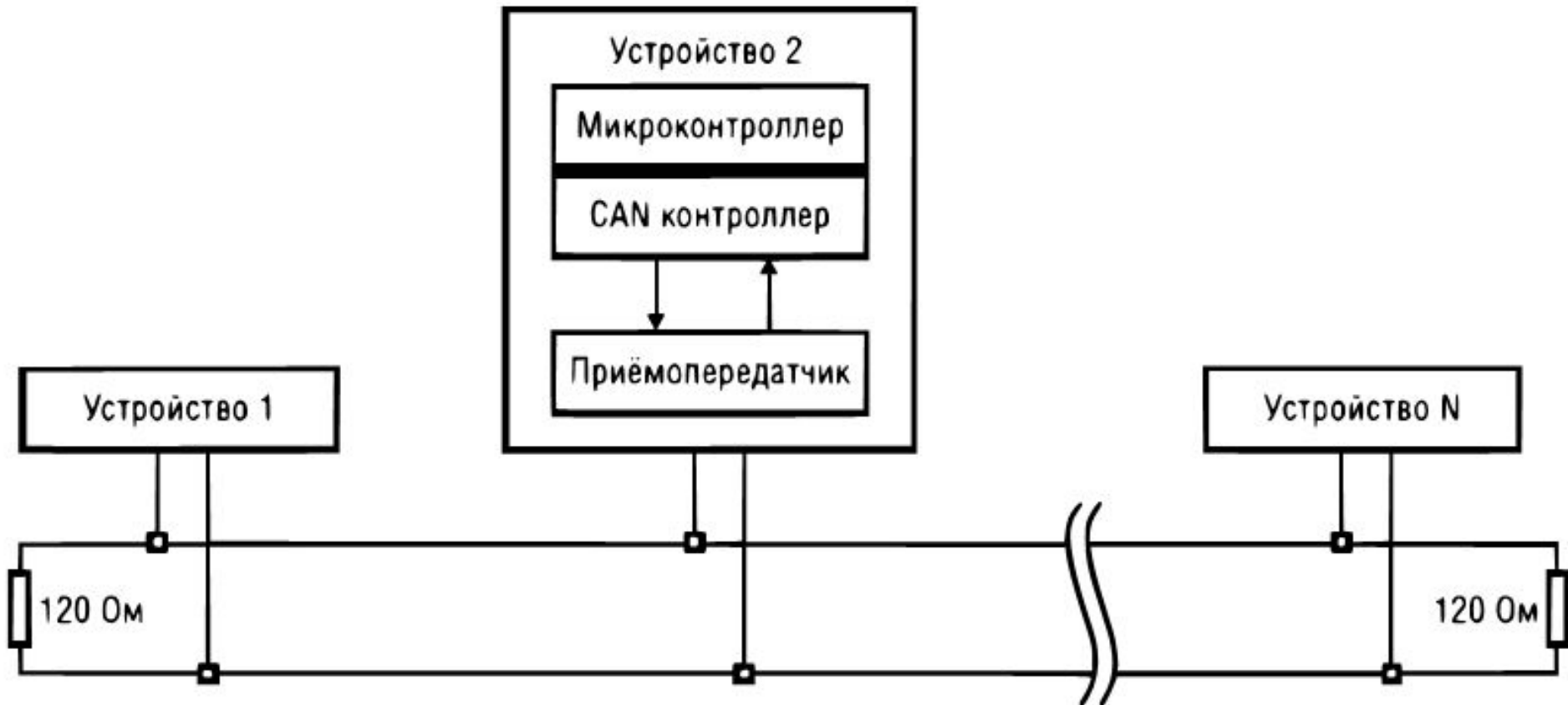
# Физический уровень

Скорость передачи, Кбит/с	Длина шины, м
1000	40
500	100
125	500
10	5000



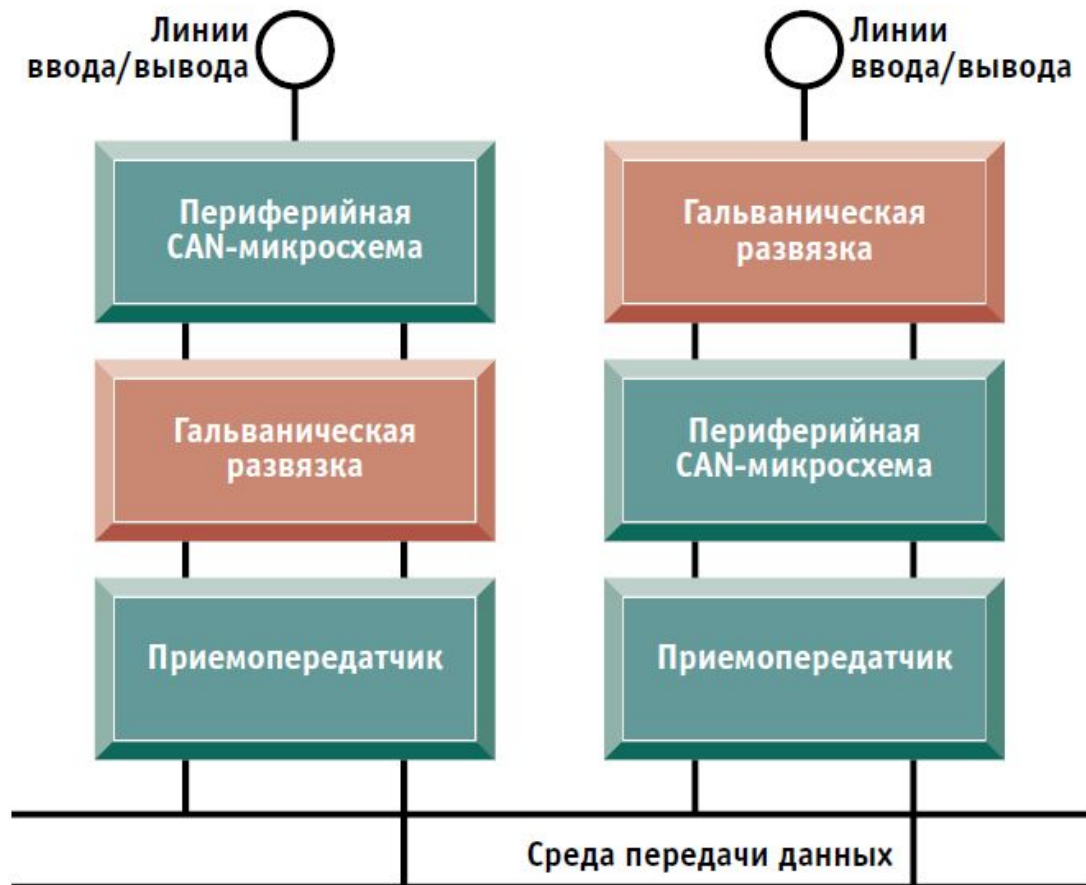


# Физический уровень



# Физический уровень

Возможность применения гальванической развязки, причем гальваническая развязка может устанавливаться либо между приемопередающим буфером и микросхемой, обеспечивающей функции CAN, либо между микросхемой и остальной системой.



# Типы фреймов в CAN протоколе

В CAN протоколе определены следующие типы фреймов:

- фрейм данных перемещает данные с передатчика на приемник (приемники);
- удаленный фрейм запрашивает передачу фрейма данных, связанного с определенным идентификатором;
- фрейм ошибки выражает, какой узел обнаружил ошибку шины/сети;
- фрейм перегрузки обеспечивает задержку между передачей фреймов, чтобы управлять потоком данных.

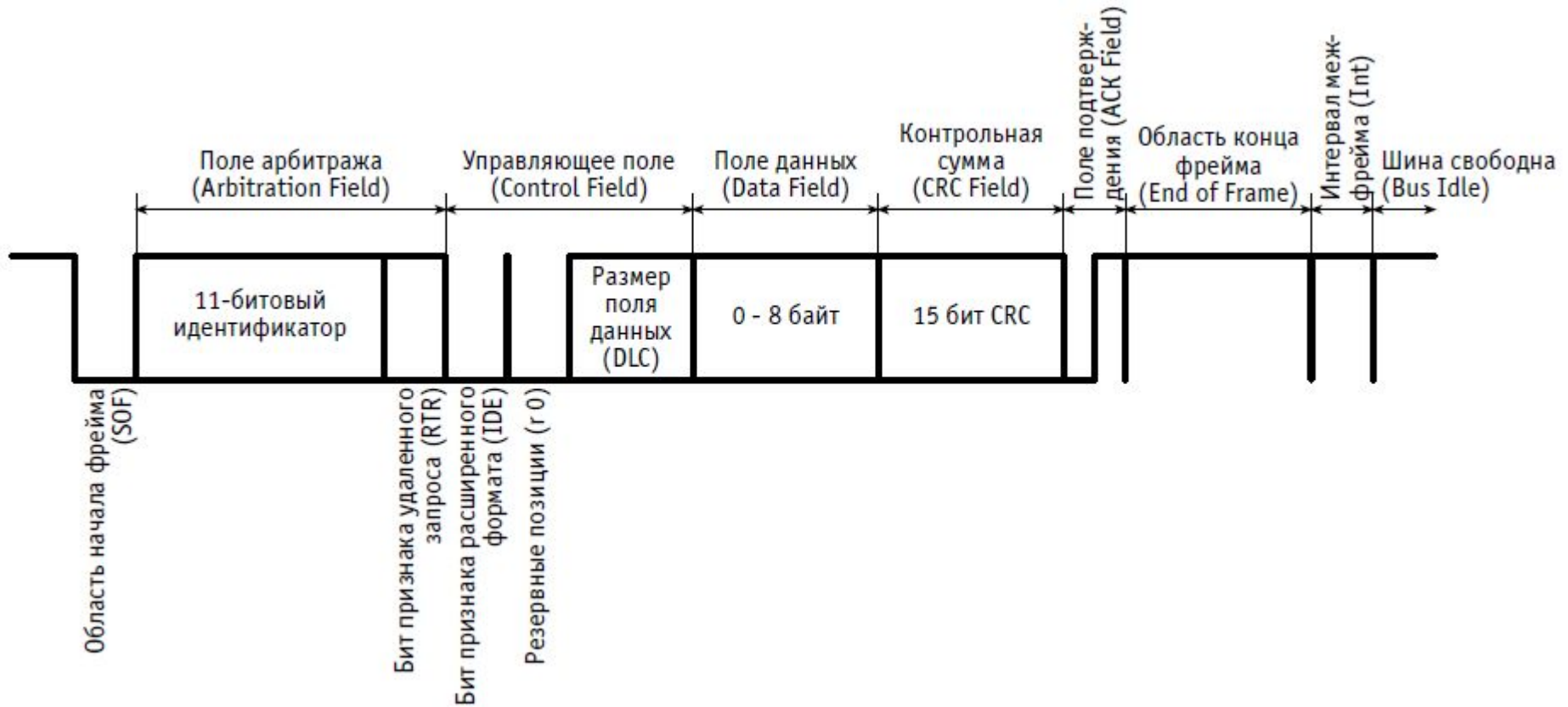
# Формат фреймов в CAN-протоколе

## Стандартный фрейм

Поле	Длина, бит	Описание
Начало кадра (SOF-Start Off Frame)	1	Сигнализирует начало передачи кадра. Доминантный бит
Идентификатор (ID-Identifier)	11	Уникальный идентификатор
Запрос на передачу (RTR, Remote Transmission Request)	1	Должен быть доминантным
Бит расширения идентификатора (IDE- Identifier Extension)	1	Должен быть доминантным
Нулевой резервный бит (RBO-Reserved Bit Zero)	1	Резервный
Длина данных (DLC- Data Length Code)	4	Длина поля данных в байтах (0–8)
Поле данных, байт	0–8	Передаваемые данные (длина в поле DLC)
Контрольная сумма (CRC-Cyclic Redundancy Code)	15	Контрольная сумма всего кадра
Разграничитель контрольной суммы	1	Должен быть рецессивным
Бит подтверждения (ACK)	1	Передатчик передаёт рецессивный бит, а приёмник вставляет доминантный бит
Разграничитель подтверждения	1	Должен быть рецессивным
Конец кадра (EOF)	7	Должен быть рецессивным

# Формат фреймов в CAN-протоколе

## Стандартный фрейм



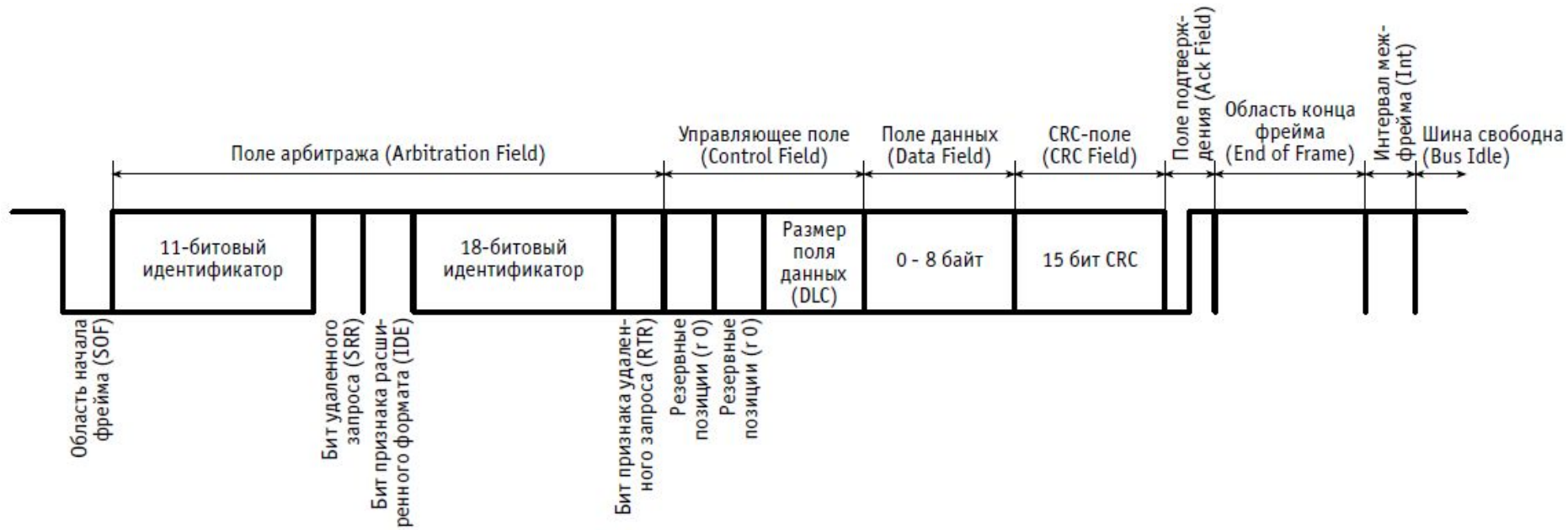
# Типы фреймов в CAN-протоколе

## Расширенный фрейм фрейм

Поле	Длина, бит	Описание
Начало кадра (SOF-Start Off Frame)	1	Сигнализирует начало передачи кадра. Доминантный бит
Идентификатор А	11	Первая часть идентификатора
Подмена запроса на передачу (SRR)	1	Должен быть рецессивным
Бит расширения идентификатора (IDE- Identifier Extension)	1	Должен быть рецессивным
Идентификатор В	18	Вторая часть идентификатора
Запрос на передачу (RTR)	1	Должен быть доминантным
Резервные биты (RB1 и RB0)	2	Резерв
Длина данных (DLC)	4	Длина поля данных в байтах (0–8)
Поле данных, байт	0–8	Передаваемые данные (длина в поле DLC)
Контрольная сумма (CRC)	15	Контрольная сумма всего кадра
Разграничитель контрольной суммы	1	Должен быть рецессивным
Промежуток подтверждения (ACK)	1	Передатчик передаёт рецессивный бит, а приёмник вставляет доминантный бит
Разграничитель подтверждения	1	Должен быть рецессивным
Конец кадра (EOF)	7	Должен быть рецессивным

# Формат фреймов в CAN-протоколе

## Расширенный фрейм



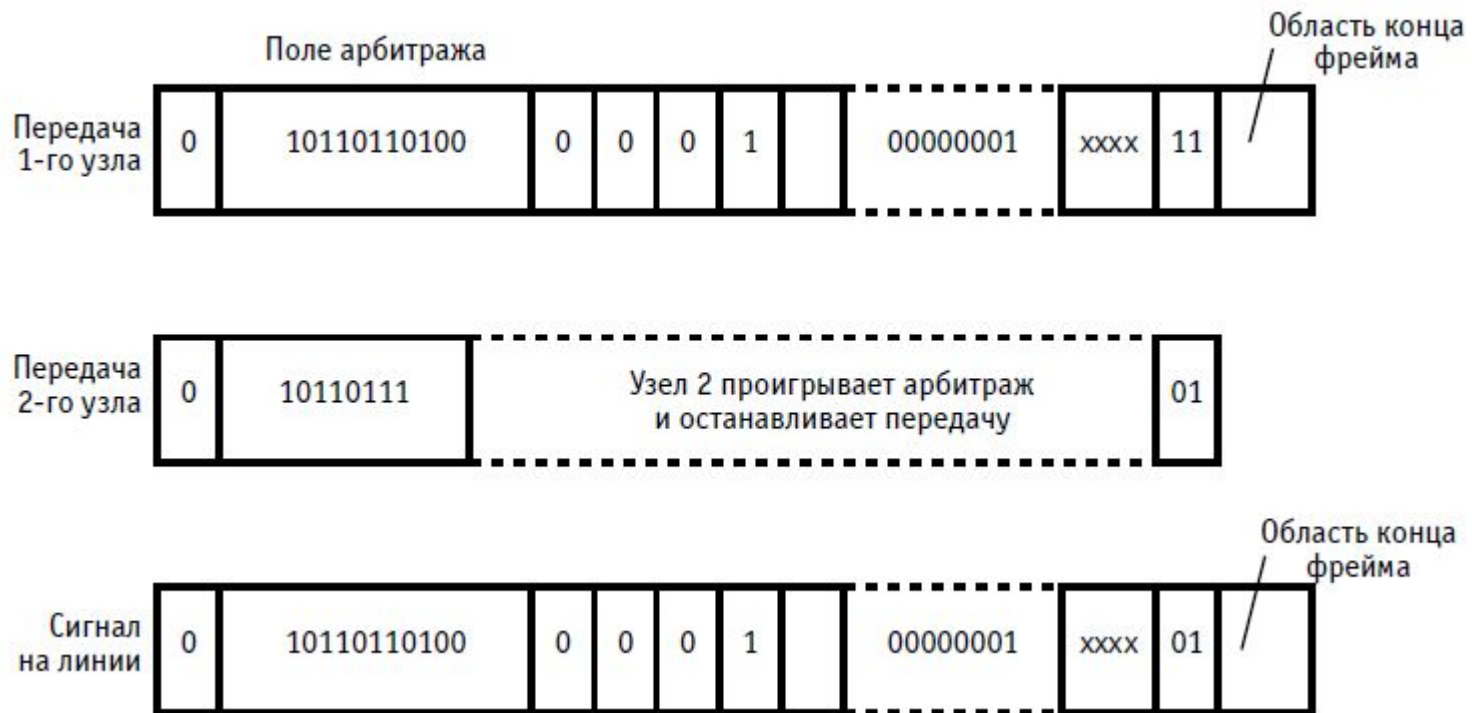
# Формат фреймов в CAN-протоколе

- Поле SOF (Start of Frame) находится в начале фрейма данных и удаленного фрейма и содержит один доминирующий бит.
- Поле арбитража Arbitration Field содержит 11битовый идентификатор и RTRбит, показывающий, является данный фрейм фреймом данных или удаленным фреймом.
- Идентификатор предназначен для адресации сообщений и используется механизмом арбитража.
- Управляющее поле Control Field содержит 6 битов, из которых 4 бита (DLC0DLC4) составляют поле Data Length Code, показывающее количество байтов данных, которое будет передаваться в поле данных; два других бита зарезервированы для следующих редакций протокола.
- Поле данных Data Field содержит передаваемые данные, причем количество передаваемых байтов указывается в поле Control Field и не может превышать 8.
- Поле CRC обеспечивает механизм избыточного контроля по четности передаваемых данных.
- Поле подтверждения ACK Field содержит участки ACK Slot и ACK Delimiter и выполняет следующую функцию: передающий узел посылает по одному рецессивному биту на каждом из участков, а приемник, если он принял сообщение без сбоев, устанавливает на линии доминирующий бит в поле ACK Slot. При наложении рецессивного и доминирующего уровней на линии устанавливается доминирующий, и это событие сигнализирует передающему узлу о том, что передача прошла нормально и повтор не требуется.
- Поле конца фрейма EOF содержится в фрейме данных и удаленном фрейме и состоит из семи рецессивных битов.



# Арбитраж CAN-протоколе

При одновременной передаче фрейма на линию,  
узел с идентификатором с самым низким значением выигрывает арбитраж.



# Ошибки в CAN-протоколе

CAN-протокол обеспечивает механизмы обнаружения следующих типов ошибок:

- Разрядная ошибка
- Ошибка подтверждения
- Ошибка заполнения
- CRC-ошибка
- Ошибка формы

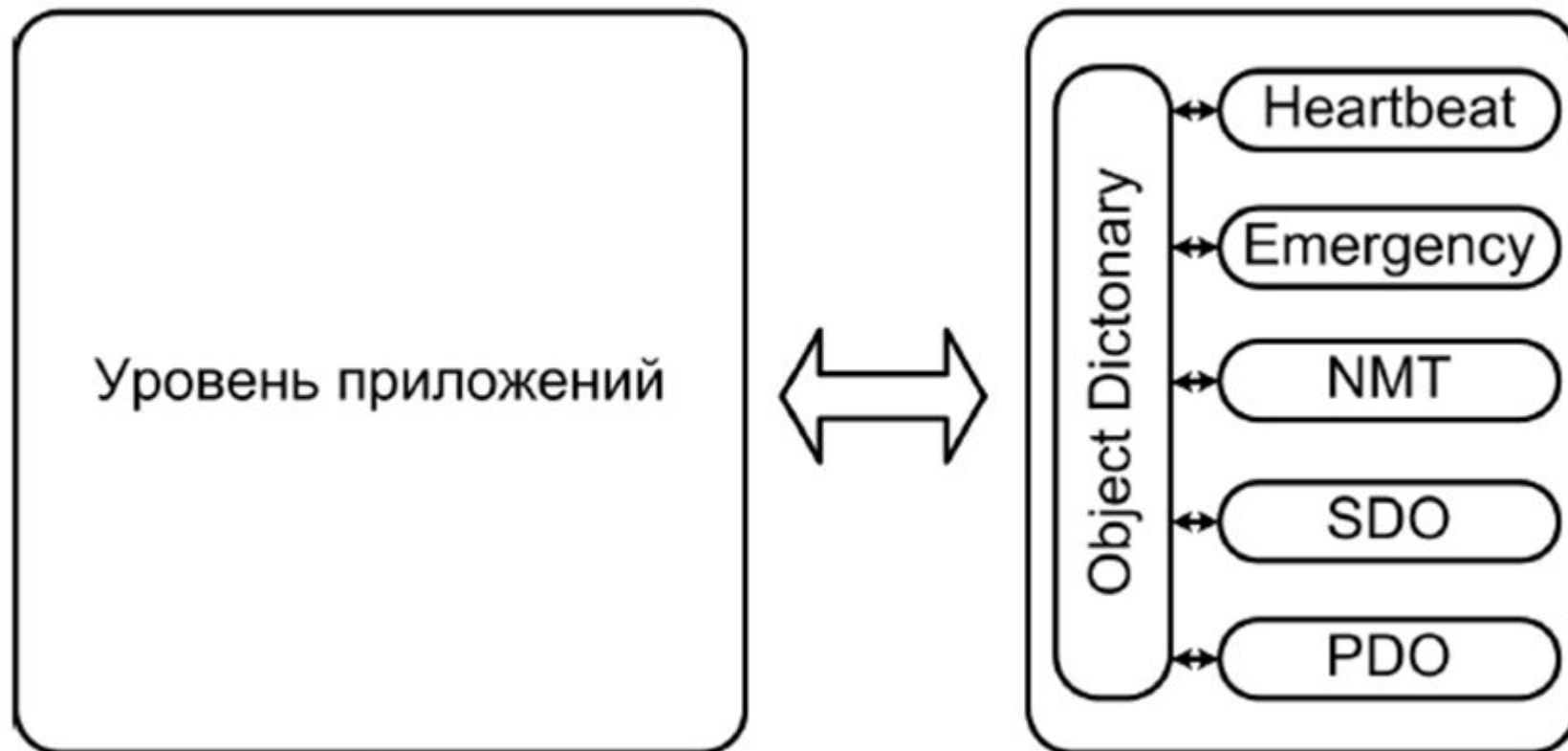
# Прерывания в CAN-протоколе



Количество байтов данных	0	1	2	3	4	5	6	7	8
Количество битов в CAN-фрейме	47	55	63	71	79	87	95	103	111
Интервал между прерываниями на скорости передачи 125 кбит/с (период 1 бита = 8 мс)	376	440	504	568	632	696	760	824	888
Интервал между прерываниями на скорости передачи 250 кбит/с (период 1 бита = 4 мс)	188	220	252	284	316	348	380	412	444
Интервал между прерываниями на скорости передачи 500 кбит/с (период 1 бита = 2 мс)	94	110	126	142	158	174	190	206	222

# *Протокол CANopen*

# Стандартные сервисы CANopen



# Словарь CANopen

Основной функциональной единицей протокола CANopen является объект. Под объектом может пониматься набор данных, несущих информацию о параметрах (например, показания датчика температуры), конфигурации узла или сети, возникших ошибках и т.п. Поэтому для устройства (узла) необходимым условием работы в сети является наличие словаря, представляющего собой группу доступных в определенном порядке объектов.

Адрес (индекс)	Объекты
0001h – 001Fh	Стандартные типы данных
0020h – 003Fh	Комплексные типы данных
0040h – 005Fh	Комплексные типы данных производителя
0060h – 007Fh	Типы специфичные для профиля оборудования (простые)
0080h – 009Fh	Типы специфичные для профиля оборудования (комплексные)
00A0h – 0FFFh	Зарезервировано
1000h – 1FFFh	Область коммуникационного профиля
2000h – 5FFFh	Область профиля производителя
6000h – 9FFFh	Область профиля стандартных устройств (стандартизованных)
A000h – BFFFh	Стандартизованная интерфейсная область
C000h – FFFFh	Зарезервировано

# Словарь CANopen

№	Индекс	Поды	Длинный текст	Короткий текст	Английский текст	Формат	Адрес
485	5000h	00h	Задатчик интенсивности	ЗИ	Ramp	Корневой	
486	5000h	01h	T разгона до номинала	rmpT	T разгона до номинала	R.Q-тип.132 8,24 м.к.№09h	rmp.T
487	5000h	02h	Вход ЗИ	ВходЗИ	Ramp input	R.Q-тип.132 8,24 м.к.№14h	rmp.input
488	5000h	03h	Выход ЗИ	ВыходЗИ	Ramp output	R.Q-тип.132 8,24 м.к.№14h	rmp.output
489	5100h	00h	ПИ П регистр тока Id	Per_id	ПИ П регистр тока Id	Корневой	

Объект имеет следующие атрибуты:

- Индекс
  - Подындекс
    - Текст
      - Формат
        - Адрес
          - Группа
            - По умолчанию
              - Мин./Макс.
                - CallBack

# Heartbeat-сервис

Сервис реализует контрольное тактирование в виде периодической передачи специального широковещательного heartbeat-сообщения (heartbeat – «сердцебиение»), обеспечивая слежение за состоянием устройств в сети.

Сообщение отсылается короткой однобайтовой посылкой:

Поле	ID (идентификатор сообщения)	DLC (число байт данных)	Данные: D0
Значение	700h + № узла	1	N

Где «N» – код сетевого состояния устройства:

Состояние	Описание
0 – BOOTUP	Сообщение с данным кодом посылается всего один раз, после начальной инициализации устройства, после чего оно переходит в состояние «PREOPERATIONAL».
4 – STOPPED	Функционируют только сервисы Heartbeat и NMT.
5 – OPERATIONAL	Номинальный режим работы. Функционируют все сервисы: Heartbeat, NMT, SDO, PDO.
127 – PREOPERATIONAL	В этом состоянии возможна настройка (модификация параметров) устройства извне с помощью SDO-сервиса. В данном состоянии функционируют сервисы: Heartbeat, NMT, SDO.



# NMT-сервис

NMT-сервис (network management) служит для управления состояниями узлов сети со стороны мастера. Сервис позволяет удалённо изменять сетевое состояние устройства.

## *Параметры вызова:*

- `ppc` – указатель на структуру используемого модуля CAN драйвера `CANopen`.
- `nmtCommand` – код NMT-команды:
  - 1 – перевести в состояние `OPERATIONAL`;
  - 2 – перевести в состояние `STOPPED`;
  - 128 – перевести в состояние `PREOPERATIONAL`;
  - 129 – перезапустить устройство (путём программного сброса процессора);
  - 130 – перезапустить коммуникационный модуль CAN (включая перенастройку PDO).

# SDO и PDO-сервисы

Для медленного обмена данными, например, для изменения настроек конечных устройств, просмотра банков событий и т.д. используется SDO-сервис (service data objects), работающий по принципу «клиент-сервер» («запрос-ответ»).

*Протокол делится на два типа взаимодействия: это запрос параметра от сервера (SDO upload protocol) и запись параметров в сервер (SDO download protocol).*

Оперативный (realtime) обмен данными между узлами сети, как правило, осуществляется с помощью PDO-сервиса (process data objects), обеспечивающего периодическую или событийную (по требованию пользователя) отправку запакованных широковещательных сообщений длиной до 64 бит (передача сообщений является неподтверждаемой).

PDO условно делятся на входящие (RPDO) и исходящие (TPDO). Устройства, в конкретный момент времени выдающие PDO на шину, называются производителями (producer), а принимающие эти PDO – потребителями (consumer).

*Сообщение PDO содержит в себе данные, набранные из данных других объектов словаря CANopen. Другими словами, отправляет значения выбранных пользователем объектов словаря в автоматическом режиме. Отображение объектов в PDO описывается структурой, называемой PDO-отображением (PDO-mapping). Это отображение связывает принимаемые/отправляемые данные с конкретными объектами словаря.*