CAN протокол (версия 1.0)

Идентификатор:

CAN идентификатор — базовый (11 бит)

10	9	8	7	6	5	4	3	2	1	0
fb	dir	dev						srv		

Биты 0...2 ***srv*** (service):

всегда 000 — для совместимости со старым протоколом где это значение «unused»
После полного отказа от старого протокола возможно использование для других целей

Биты 3...8 ***dev*** (device type):

код типа устройства, отправившего пакет

0 - **PC21**
1 - **8AI/DI**
2 - **4DO**
3 - **RS**

Бит 9 ***dir*** (direction):

направление пакета

0 - широковещательный
1 - к конкретному устройству

Бит 10 ***fb*** (feedback):

требование ответного подтверждения пакета

0 - без подтверждения
1 - требуется подтверждение (не используется для широковещательных пакетов)

Типы данных:

Помимо состояния входов/выходов и служебной информации каждое устройство может хранить собственный набор однобайтных данных, набор двухбайтных данных и массив конфигурации. Карта памяти данных определяется типом устройства (dev) и приводится далее в описании соответствующих команд.

Пакет данных:

8 байт:

1 байт — адрес устройства

Если *dir* равен 0 (широковещательный пакет) то указывается адрес устройства, отправившего пакет. В противном случае адрес устройства к которому адресован пакет.

Если старший бит равен 0 то адресуется устройство внутри кластера. Диапазон адресов от 0 до 127.

7	6	5	4	3	2	1	0
0	**адрес внутри кластера**						

В противном случае содержит адрес кластера и адрес устройства. Если устройство предполагает межкластерную передачу данных его адрес внутри кластера должен быть в диапазоне 0...15

7	6	5	4	3	2	1	0
1	**адрес кластера**			**адрес внутри кластера**			

2 байт — код команды

0 — *heartbeat* (оповещение что устройство в сети)
1 – *response* (подтверждение на пакет с битом feedback)
2 – *get info* (запрос информации по устройству)
3 – *do state* (состояние дискретных выходов)
4 – *do write* (установить выход)
5 – *ai state* (состояние аналоговых входов)
6 – *write byte* (запись байта данных)
7 – *write conf* (запись конфигурации)
8 – *regs state with validation* (состояние регистров с требованием возврата принятых данных)
9 – *regs validation* (ответ на команду *regs state with validation*)

3 ... 8 байты определяются командами

Команды:

(содержание байтов 3...8 пакета)

heartbeat (код 0)
Каждое устройство с некоторой периодичностью (минимум каждые 0.5 с) отправляет эту команду в сеть, позволяет контролировать другим устройствам наличие модуля в сети

байт 3 — счётчик (инкрементируется при каждом отправленном модулем пакете heartbeat)
байты 4 — 8 могут использоваться модулями для отправки информации о своём состоянии

PC21	не используются
8AI/DI	не используются
4DO	байт 4 — младшие 4 бита передают текущее состояние выходов. бит 0 — первый выход, бит 3 — четвёртый выход
RS	Байт 4 бит 0 - флаг получения модулем настроек от контроллера: 1 — настройки получены, 0 — настройки не получены

тип пакета - широковещательный

response (код 1)

Если устройство получило пакет, адресованный ему (не широковещательный *dir=1*) с битом *feedback* то оно в ответ посылает пакет *response*

байт 3 — третий байт принятого пакета (первый байт данных, может использоваться как идентификатор пакета)
байт 4 — код команды принятого пакета
байт 5 — код устройства принятого пакета (поле *dev* идентификатора)

тип пакета - широковещательный

get info (код 2)

Запрос текущего состояния другого устройства (актуальное состояние входов/выходов и т. п.)

байты 3 … 8 не используются

Идентификатор:

тип пакета – к конкретному устройству (dir = 1)

do state (код 3)

Текущее состояние выходов

байт 3 — номер стартового выхода N
байт 4 — состояние выходов, начиная со стартового
каждому выходу соответствует 1 бит:
0 — выключен
1 — включен

бит 0 задаёт состояние выхода N (задан в байте 3)
бит 1 — состояние выхода N+1 и т.д.

байт 5 — состояние ошибок выходов
каждому выходу соответствует 1 бит:
0 — ошибок нет
1 — выход неисправен

тип пакета - широковещательный

do write (код 4)

Изменить состояние выходов устройства

байт 3 — номер стартового выхода N
байт 4 — состояние выходов, начиная со стартового

каждому выходу соответствует 1 бит:
0 — выключен
1 — включен

бит 0 задаёт состояние выхода N (задан в байте 3)
бит 1 — состояние выхода N+1 и т. д.

байт 5 — адрес устройства, пославшего пакет

тип пакета – к конкретному устройству (dir = 1)

ai state (код 5)

передаёт двухбайтное значение аналогового входа в мВ или в мкА
(мкА для токового типа входа, мВ — в противном случае)

байт 3 — номер входа
байт 4 — младший байт значения
байт 5 — старший байт значения

тип пакета - широковещательный

write byte (код 6)

передача байта данных устройству по заданному адресу
назначение байта определяется типом устройства к которому адресован пакет и адресом байта

байт 3 — адрес байта данных
байт 4 — значение байта данных
байт 5 — адрес отправителя
байт 6 — тип устройства, которому предназначены данные
используется для дополнительного контроля, если устройство получает пакет и тип устройства в 6 байте не совпадает с его собственным, пакет игнорируется

карта памяти байтов для разных типов устройств:

PC21	не используется
8AI/DI	адрес 0 — тип входов каждый бит соответствует входу. Если бит равен 1 тип входа токовый, если 0 то тип входа — напряжение. Младший бит (бит 0) связан с первым входом, старший (бит 7) с восьмым
4DO	не используется
RS	не используется

тип пакета – к конкретному устройству (dir = 1)

write conf (код 7)

Предусматривает передачу буфера конфигурации, разбитого на пакеты. Число пакетов может быть до 256 включительно. Первый пакет (заголовок) содержит информацию о буфере. Остальные пакеты содержат данные. Максимальная длина буфера 1020 байт (255 пакетов по 4 байта).

байт 3 — адрес отправителя
байт 4 — номер пакета
пакет номер 0 является заголовком

байты 5...8 — данные пакета

Пакет 0 (заголовок):

байт 5 — число пакетов для передачи (не считая заголовок)
байт 6 — длина буфера конфигурации (старший байт)
байт 7 — длина буфера конфигурации (младший байт)
В длине буфера конфигурации учитываются только данные для передачи (без служебных байт и заголовка)
байт 8 — тип устройства, которому адресована конфигурация
(для дополнительного контроля помимо адреса)

Содержание буфера конфигурации определяется типом устройства к которому адресован пакет.

Массив конфигурация для типа устройств RS:

RS модуль опрашивает датчики по протоколу Modbus RTU и информацию о их состоянии транслирует в кан сеть. Массив конфигурации определяет количество датчиков и их тип.

номер байта	**назначение**
1	Контрольная сумма типов опрашиваемых датчиков(старший байт)[1]
2	Контрольная сумма типов опрашиваемых датчиков (младший байт)
3	Число датчиков
4	Сетевой адрес 1-го датчика
5	Тип и приоритет 1-го датчика[2]
6	Сетевой адрес 2-го датчика
7	Тип и приоритет 2-го датчика
8	Сетевой адрес 3-го датчика
9	Тип и приоритет 3-го датчика
10	Сетевой адрес 4-го датчика

1 Для используемых в конфигурации типов датчиков считается контрольная сумма, которая зависит от типа опрашиваемых регистров, их количества, адресов, необходимости контролировать значение регистров. RS модуль тоже рассчитывает контрольную сумму, полагаясь на предустановленный в нём список стандартных датчиков. Несовпадение контрольных сумм означает что устройство, передавшее конфигурацию использует неизвестный модулю RS набор датчиков. При контрольной сумме, равной 0, проверка отключается

2 Старший бит определяет приоритет опроса датчика (1 — высокий, 0 — низкий). Типы датчиков приведены в документации на модуль RS

номер байта	назначение
11	Тип и приоритет 4-го датчика
12	Сетевой адрес 5-го датчика
13	Тип и приоритет 5-го датчика
14	Сетевой адрес 6-го датчика
15	Тип и приоритет 6-го датчика
16	Сетевой адрес 7-го датчика
17	Тип и приоритет 7-го датчика
18	Сетевой адрес 8-го датчика
19	Тип и приоритет 8-го датчика
20	Сетевой адрес 9-го датчика
21	Тип и приоритет 9-го датчика
22	Сетевой адрес 10-го датчика
23	Тип и приоритет 10-го датчика
24	Зарезервирован для конфигурации COM порта

тип пакета – к конкретному устройству (dir = 1)

regs state with validation (код 8)

Транслирует состояние двухбайтных регистров устройства в сеть. Но при этом подразумевает что устройство для которого эти данные важны (например контроллер, видящий регистры от используемых им модулей) вернёт не просто подтверждение что получен пакет (как в случае с feedback), а вернёт и сами значения принятых регистров для контроля их актуальности.

байт 3 - адрес стартового регистра (старший байт)
байт 4 - адрес стартового регистра (младший байт)
байт 5 — значение стартового регистра (старший байт)
байт 6 — значение стартового регистра (младший байт)
байт 7 - значение следуещего регистра (старший байт)
байт 8 - значение следуещего регистра (младший байт)

Если передаётся только один регистр то байты 7 и 8 не используются

тип пакета – широковещательный (dir = 0) или к конкретному устройству (dir = 1)

regs validation (код 9)

Ответ на принятый пакет *regs state with validation* если устройству важно поддерживать актуальность полученных данных.

байты 3...8 полностью повторяют байты принятого пакета

тип пакета - к конкретному устройству (dir = 1)