ТЕПЛОВОДОХРАН

ПУЛЬСАР

для разработчиков

Спецификация протокола «ПульсарМ»

Дата: 23.05.19

Версия: 1.0

1 Канальный уровень

Интерфейс связи в качестве полезных данных должен оперировать октетами (далее байтами), возможна работа в полудуплексном режиме. Ниже приведена диаграмма одной транзакции (приемо-передачи).

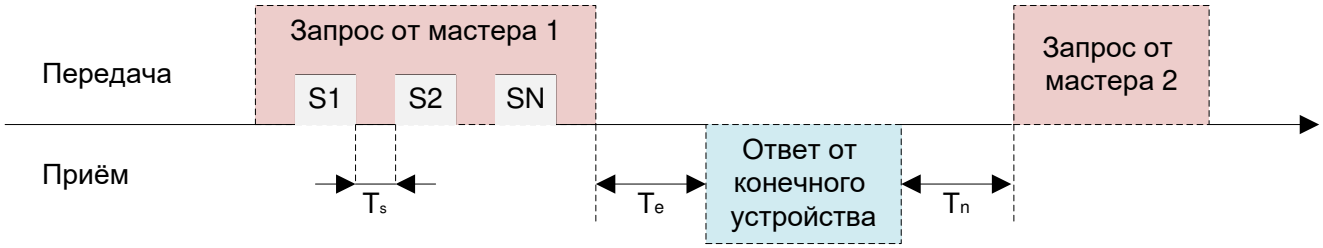


Рисунок 1. Диаграмма приёмо-передачи с указанием временных задержек

- S1, S2…SN – передаваемые символы в рамках одного фрейма.
- T_s – максимальное время задержки между символами. Зависит от реализуемого интерфейса связи. При превышении этого времени устройство считает, что фрейм закончен.
- T_e – время обработки запроса, $T_s \geq T_e$. Максимальное время обработки может варьировать в зависимости от конкретного устройства и запроса. По умолчанию не должна превышать 5 секунд.
- T_n – минимальное время между передачей ответа до нового запроса. Зависит реализуемого интерфейса связи.

1.1 Интерфейс RS-232/RS-485/UART

Параметры по умолчанию:

- скорость 9600, 8 бит данных (минимум), 1 стоп-бит, без проверки четности.
- T_s – 4.5 символа.
- T_n – 1.5 символа.

1.2 Интерфейс TCP/IP

Параметры по умолчанию:

- T_s – 30 мС (обычно конфигурируемо).
- T_n – отсутстсвует.

2 Транспортный уровень

Данные передаются фреймами. Протокол является адресным, с поддержкой широковещательных запросов и проверкой соответствия запроса ответу.

2. 1 Формат фрейма

Максимальный размер фрейма – 255 байт, общий размер технических полей фрейма – 10 байт. Таким образом, максимальный размер полезных данных – до 245 байт.

Поле	**ADDRESS**	**FN**	**LEN**	**PAYLOAD**	**ID**	**CRC**
Формат	BCD MSB	UINT8	UINT8	BLOB	UINT16	UINT16
Размер	4	1	1	≤ 245	2	2



Для многобайтовых полей по умолчанию (если не указано явно) используется прямой порядок байт (LSB). Поля FLOAT32 и FLOAT64 являются числами IEEE.754 одинарной и двойной точности соответственно.

Описание полей:

- **ADDRESS** – адрес конечного устройства. Значение 0x00000000 является широковещательным адресом и указывается только в запросе. Ответ на широковещательный запрос будет содержать адрес конкретный адрес конечного устройства.
- **FN** – функция (далее код запроса). Значение 0x00 зарезервировано под ответ с кодом ошибки (см. пункт 3.9).
- **LEN** – общая длинна фрейма в байтах (включая сам байт длинны). Допустимые значения [10…255].
- **PAYLOAD** – полезные данные, зависят от указанного кода запроса.
- **ID** – идентификатор запроса. Псевдослучайное число, генерируемое мастером при формировании запроса. Конечное устройство должно ответить тем же значением.
- **CRC** – контрольная сумма фрейма. При формировании фрейма считается от всех данных, исключая само поле контрольной суммы.

2.2 Параметры и расчет контрольной суммы

Характеристики:

- Стандарт: CRC-16-IBM.
- Полином: 0xA001.
- Начальное значение: 0xFFFF.
- Начинать с: LSB.
- XOR в конце: нет.

Пример расчет на языке C89

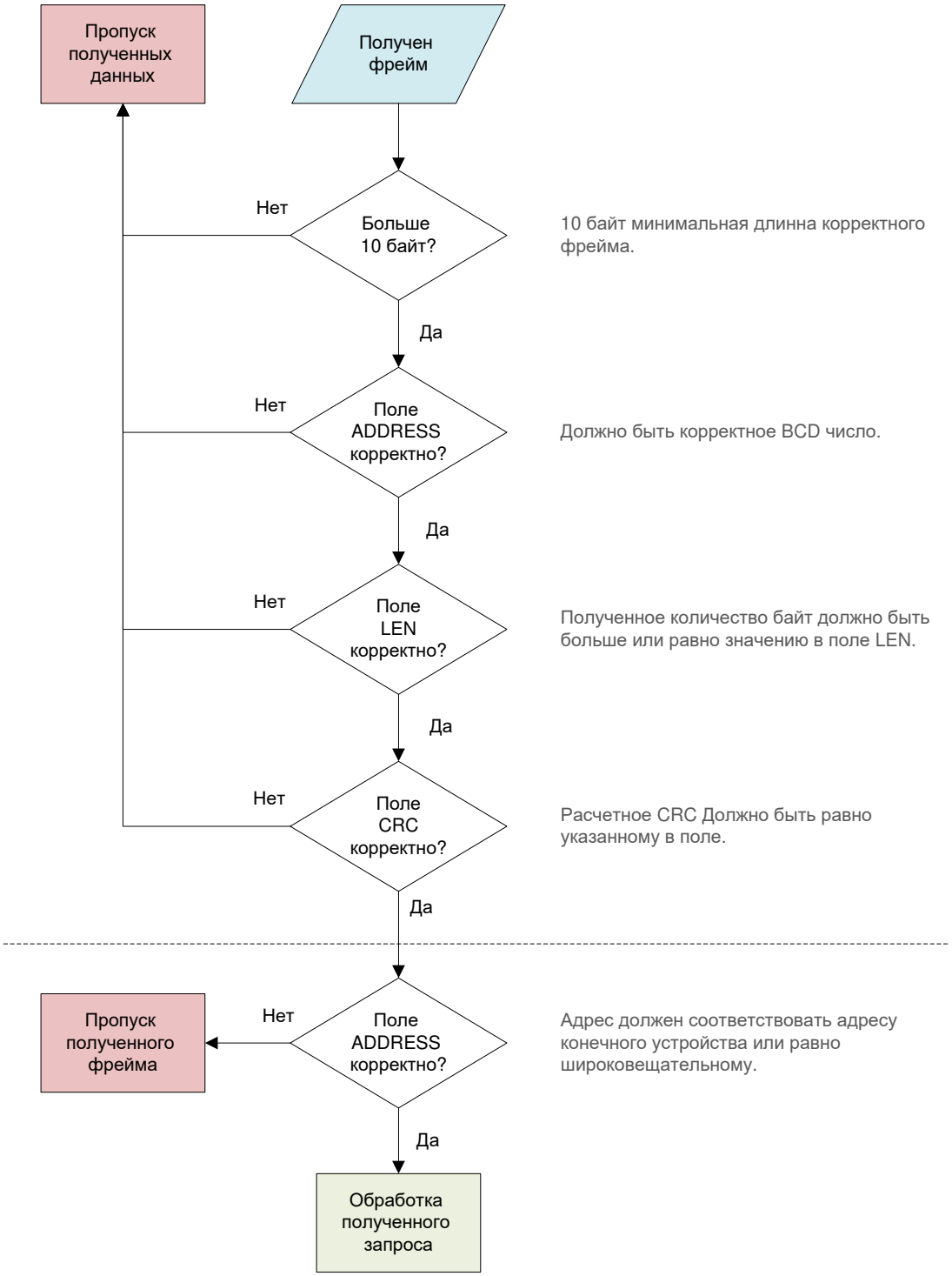
```c
#include <stddef.h>
#include <stdint.h>

uint16_t crc(void const *data, size_t size)
{
    uint8_t i;
    uint16_t result = 0xFFFF;
    uint8_t const *buf = (uint8_t const *)data;
    while (size--)
    {
        result ^= *buf++;
        for (i = 0; i < 8; i++)
            result = (result & 1) ? (result >> 1) ^ 0xA001 : result >> 1;
    }
    return result;
}
```

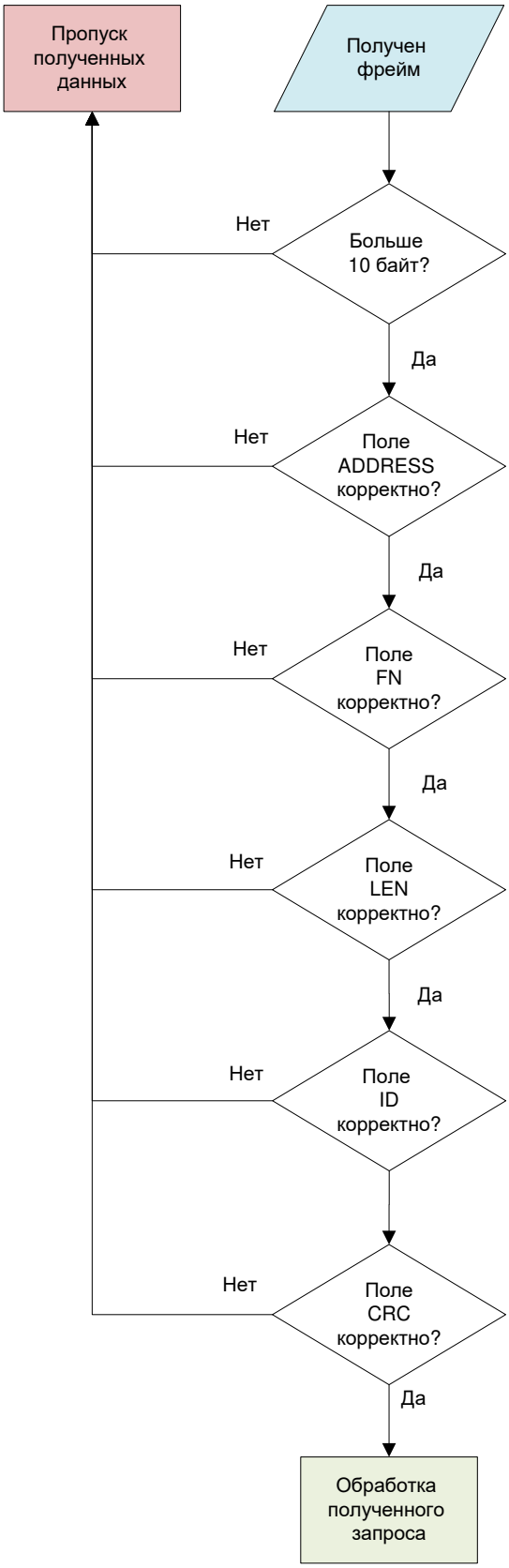


При валидации входящего фрейма удобно считать контрольную сумму от всех байт, включая само поле контрольной суммы. В случае верной контрольной суммы, результат должен быть равен 0x0000.

2.3 Алгоритм валидации входящего фрейма для конечного устройства



2.4 Алгоритм валидации входящего фрейма для мастера



10 байт минимальная длинна корректного фрейма.

Адрес должен соответствовать запрашиваемому, но отличным от широковещательного в случае запроса по широковещательному адресу.

Полученный код запроса должен соответствовать запрашиваемому или быть равным 0X00 в случае ошибки.

Количество полученных байт должно быть больше или равно значению в поле LEN.

Запрашиваемый идентификатор должен быть равен указанному в поле ID.

Расчетное CRC Должно быть равно указанному в поле.

3 Прикладной уровень

В данном разделе описаны форматы стандартных функций. Описывается только тело функции (без фрейма). В зависимости от модели конечного устройства некоторые из стандартных функций могут не поддерживаться.

3.1 Структура даты/времени

Имя структуры: DATETIME.

Поле	**YEAR**	**MONTH**	**DAY**	**HOUR**	**MINUTE**	**SECOND**
Формат	UINT8	UINT8	UINT8	UINT8	UINT8	UINT8
Размер	1	1	1	1	1	1

Описание полей:

- **YEAR** – год, отсчитывается от 2000, допустимые значения [0…99].
- **MONTH** – месяц, допустимые значения [1…12].
- **DAY** – день месяца, допустимые значения [1…31].
- **HOUR** – час, допустимые значения [0…23].
- **MINUTE** – минута, допустимые значения [0…59].
- **SECOND** – секунда, допустимые значения [0…59].

Примечания:

- Если все байты равны 0xFF – считать что дата/время отсутствует.

3.2 Структура макси каналов

Имя структуры: CHMASK.

Поле	**MASK**
Формат	UINT32
Размер	4

Описание полей:

- **MASK** – битовая маска каналов. Например, если установлен младший бит – значит используется канал №1 (считать с 1).

3.3 Чтение текущей даты/времени

Функция: 0x04.

Запрос: данные отсутствуют.

Ответ:

Поле	**DATETIME**
Формат	DATETIME
Размер	6

3.4 Запись текущей даты/времени

Функция: 0x05

Запрос:

Поле	**DATETIME**
Формат	DATETIME
Размер	6

Ответ:

Поле	**STATUS**	**Z1**	**Z2**	**Z3**
Формат	UINT8	UINT8	UINT8	UINT8
Размер	1	1	1	1

Описание полей:

- **STATUS** – результат записи. 0x00 – ошибка, 0x01 – успешно.
- **Z1, Z2, Z3** – пады (всегда нули).

3.5 Чтение каналов

Функция: 0x01

Запрос:

Поле	**MASK**
Формат	CHMASK
Размер	4

Описание полей:

- **MASK** – битовая маска считываемых каналов.

Ответ:

Поле	**CHANNEL1**	**CHANNEL2**	…	**CHANNELN**
Формат	BLOB	BLOB		BLOB
Размер	?	?		?

Описание полей:

- **CHANNEL1, CHANNEL2, CHANNELN** – показания по каналам, отсортированным по возрастанию номера канала. Формат, размер варьируются и описаны в таблице каналов модели конечного устройства.



Размер показаний по каналам является одинаковым, поэтому можно определить размер показания по каналам разделив полученое количество байт полезных данных на количество запрашиваемых каналов.

3.6 Запись каналов

Функция: 0x02

Запрос:

Поле	**MASK**	**CHANNEL1**	**CHANNEL2**		**CHANNELN**
Формат	CHMASK	BLOB	BLOB	…	BLOB
Размер	4	?	?		?

Описание полей:

- **MASK** – битовая маска записываемых каналов.
- **CHANNEL1, CHANNEL2, CHANNELN** – новые показания по каналам, отсортированным по возрастанию номера канала. Формат, размер варьируются и описаны в таблице каналов модели конечного устройства.

Ответ:

Поле	**MASK**
Формат	CHMASK
Размер	4

Описание полей:

- **MASK** – битовая маска успешно записанных каналов.

3.7 Чтение архива по каналам

Функция: 0x06

Запрос:

Поле	**MASK**	**TYPE**	**DATE_START**	**DATE_END**
Формат	CHMASK	UINT16	DATETIME	DATETIME
Размер	4	2	6	6

Описание полей:

- **MASK** – битовая маска считываемого канала. Важно! Должен быть установлен только один бит маски, т.е. запрашивать архив можно только по одному каналу за запрос.
- **TYPE** – тип архива. 0x0001 – часовой, 0x0002 – суточный, 0x0003 – месячный. 0x0004 – получасовой поддерживается не у всех моделей конечных устройств.
- **DATE_START**, **DATE_END** – начальная и конечная дата архивных показаний. Конечное устройство вправе передать меньше показаний при этом обязательно начиная с начальной даты.

Ответ:

Поле	**MASK**	**DATE_START**	**VALUE1**	**VALUE2**		**VALUEN**
Формат	CHMASK	DATETIME	BLOB	BLOB	…	BLOB
Размер	4	6	?	?		?

Описание полей:

- **MASK** – битовая маска зачитываемого канала.
- **DATE_START** – начальная дата архивных показаний, такая же как и в запросе.
- **VALUE1**, **VALUE2**, **VALUEN** – архивные показания по зачитываемому каналу. Формат, размер варьируются и описаны в таблице каналов модели конечного устройства.



В случае, если у полученного архивного показания все биты установлены – это признак отсутствия данных.

3.7 Чтение параметра

Функция: 0x0A

Запрос:

Поле	**INDEX**
Формат	UINT16
Размер	2

Описание полей:

- **INDEX** – индекс считываемого параметра.

Ответ:

Поле	**VALUE**
Формат	BLOB
Размер	8

Описание полей:

- **VALUE** – значение считываемого параметра. Размер поля фиксирован и ограничен 8 байтами. Формат и реальный размер значения параметра варьируются и описаны в таблице параметров модели конечного устройства.

3.8 Запись параметра

Функция: 0x0B

Запрос:

Поле	**INDEX**	**VALUE**
Формат	UINT16	BLOB
Размер	2	8

Описание полей:

- **INDEX** – индекс считываемого параметра.
- **VALUE** – новое значение записываемого параметра. Размер поля фиксирован и ограничен 8 байтами. Формат и реальный размер значения параметра варьируются и описаны в таблице параметров модели конечного устройства. В случае, если реальный размер значения параметра менее 8 байт – оставшиеся байты заполняются 0x00.

Ответ:

Поле	**STATUS**
Формат	UINT16
Размер	2

Описание полей:

- **STATUS** – результат записи параметра, всегда 0x0000 (успешно).

3.9 Ответ с ошибкой

Функция: 0x00

Ответ:

Поле	**CORE**
Формат	UINT8
Размер	1

Описание полей:

- **CODE** – код ошибки

Стандартные коды ошибок:

- 0x01 – отсутствует запрашиваемый код функции.
- 0x02 – ошибка в битовой маске запроса.
- 0x03 – ошибочная длинна запроса.
- 0x04 – отсутствует параметр.
- 0x05 – запись заблокирована, требуется авторизация.
- 0x06 – записываемое значение (параметр) находится вне заданного диапазона.
- 0x07 – отсутствует запрашиваемый тип архива.
- 0x08 – превышение максимального количества архивных значений за один пакет.



Ответ с ошибкой может возникнуть на любой запрос. Поэтому при получении ответа необходимо всегда проверять функцию в ответе.

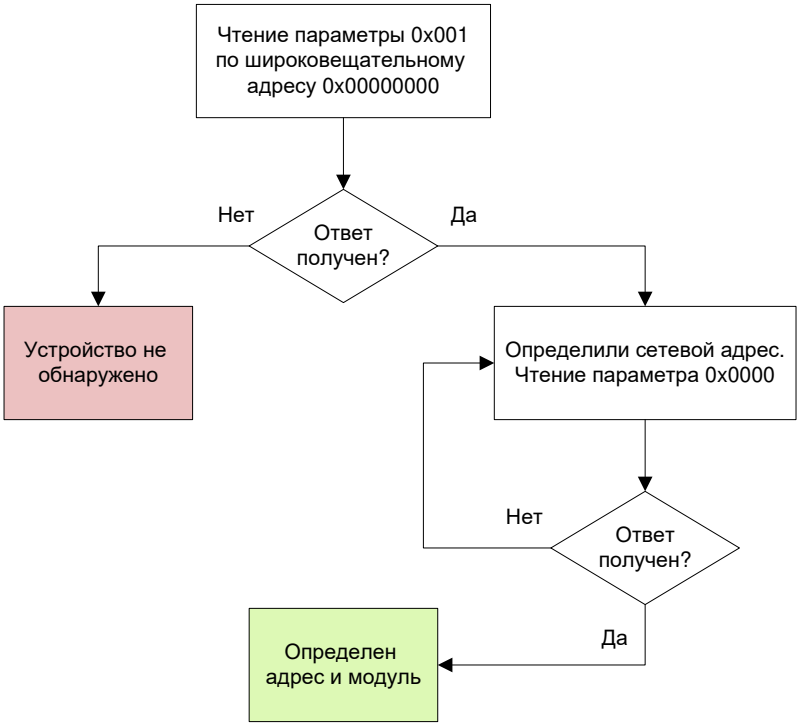
4 Приложение

4.1 Таблица фиксированных (зарезервированных) номеров параметров

Параметр	Индекс	Доступ	Тип данных	Формат
Идентификатор прибора	0x0000	**R**	UINT16	-
Сетевой адрес	0x0001	**R/W**	UINT32	[1…99999999]
Пароль (старый | новый)	0xE000	**W**	UINT64	UINT32(H) | UINT32(L)
Версия ПО	0x0002	**R**	UINT64	см. ниже
Номер прошивки	Байт 0	-	UINT16	[1…999]
	Байт 1			
Аппаратная версия	Байт 2		UINT16	[1…99]
	Байт 3			
Программная версия	Байт 4		UINT16	[1…999]
	Байт 5			
Номер ревизии	Байт 6		UINT16	[0…99]
Модификация	Байт 7		UINT16	[0…99]

4.2 Алгоритм определения сетевого адреса и типа конечного устройства

Алгоритм применяется в случае, если на шине одно, не известен его адрес.



4.3 Алгоритм определения всех устройств на шине

Алгоритм применяется в случае, если на шине несколько устройств с разными сетевыми адресами. Поддерживается не всеми моделями конечных устройств.

