



for developers

# Protocol Specification

## PulsarM 10 (16)



## 1 Data Link Layer

The communication interface must process payload represented by octets (hereinafter referred to as bytes) and it can operate in half-duplex mode. A diagram of a single transaction (reception/transmission) is given below.

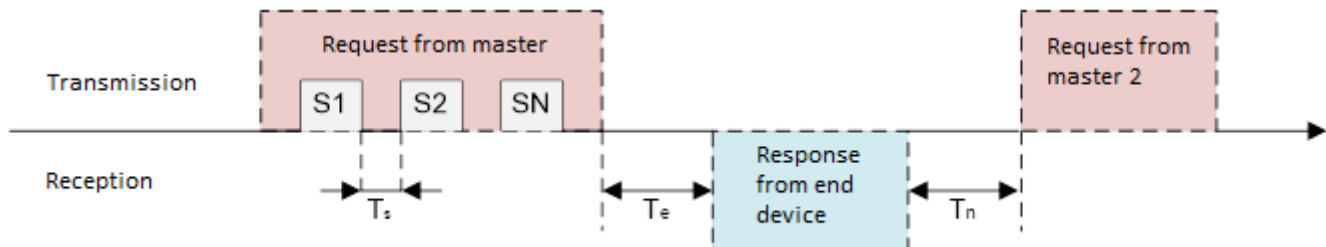


Figure 1. Reception/Transmission Diagram Showing Time Delays

- S1, S2...SN — characters transmitted within one frame.
- $T_s$  — maximum delay between characters. Depends on the implemented communication interface. If this time is exceeded, the device considers that the frame is over.
- $T_e$  — request processing time,  $T_s \geq T_e$ . The maximum processing time may vary on device-by-device basis, as well as depend on the request. By default, it must not exceed 5 seconds.
- $T_n$  — minimum time between transmission of a response and the following new request. Depends on the implemented communication interface.

### 1.1 RS-232/RS-485/UART Interface

Default settings:

- 9600 baud, 8 data bits (minimum), 1 stop bit, no parity
- $T_s$  — 10.0 characters.
- $T_n$  — 1.5 characters.

### 1.2 TCP/IP interface

Default settings:

- $T_s$  — 30 ms (usually configurable).
- $T_n$  — none.

## 2 Transport Layer

The data is transmitted in frames. The protocol is addressable.

### 2.1 Frame Format

The maximum frame size is 255 bytes and the total size of technical fields of the frame is 10 bytes. Thus, the maximum size of the payload is up to 245 bytes.

Field	ADDRESS	FN	LEN	PAYLOAD	ID	CRC
Format	BCD MSB	UINT8	UINT8	BLOB	UINT16	UINT16
Size	4	1	1	≤ 245	2	2



For multibyte fields, little-endian (LSB) is used by default (unless explicitly specified). FLOAT32 and FLOAT64 fields are IEEE.754 single- and double-precision numbers, respectively.

Description of fields:

- **ADDRESS** — device's address. The value 0x00000000 is a broadcast address and is specified only in the request. The response to the broadcast request will contain a specific address of the end device.
- **FN** — function (hereinafter referred to as the request code). The value 0x00 is reserved for the response containing the error code (see subsection 3.9).
- **LEN** — total length of the frame in bytes (including the length byte itself). Valid values — [10...255].
- **PAYLOAD** — payload depending on the specified request code.
- **ID** — request ID. A pseudorandom number generated by the master when creating the request. The end device must respond with the same value.
- **CRC** — frame checksum. When a frame is being created, the checksum calculation involves all data, excluding the checksum field itself.

## 2.2 Parameters and Checksum Calculation

Parameters:

- Standard: [CRC-16-IBM](#).
- Polynomial: [0xA001](#).
- Initial value: [0xFFFF](#).
- Begin with: [LSB](#).
- XOR at the end: [no](#).

Example of calculation in C89

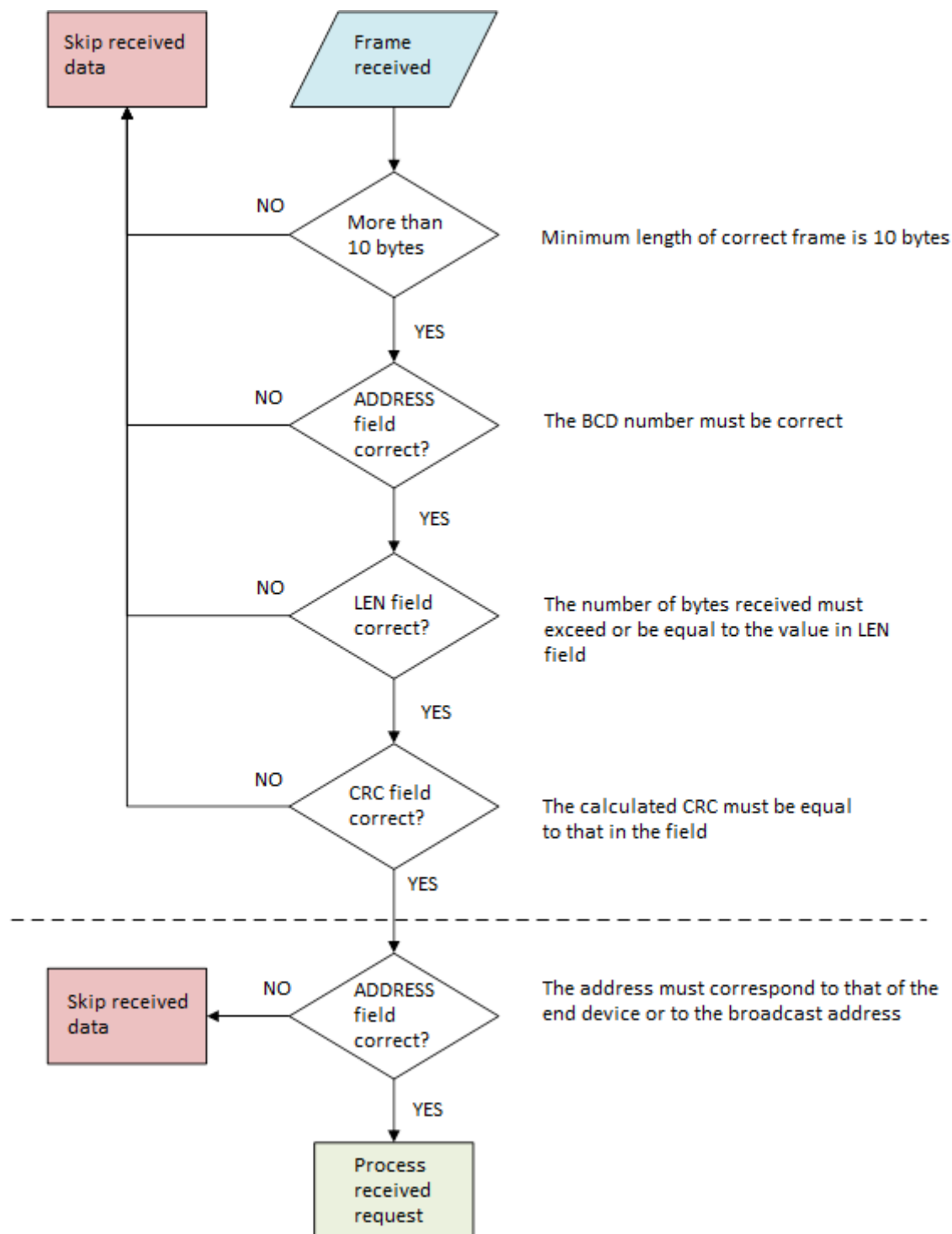
```
#include <stddef.h>
#include <stdint.h>

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;
}
```

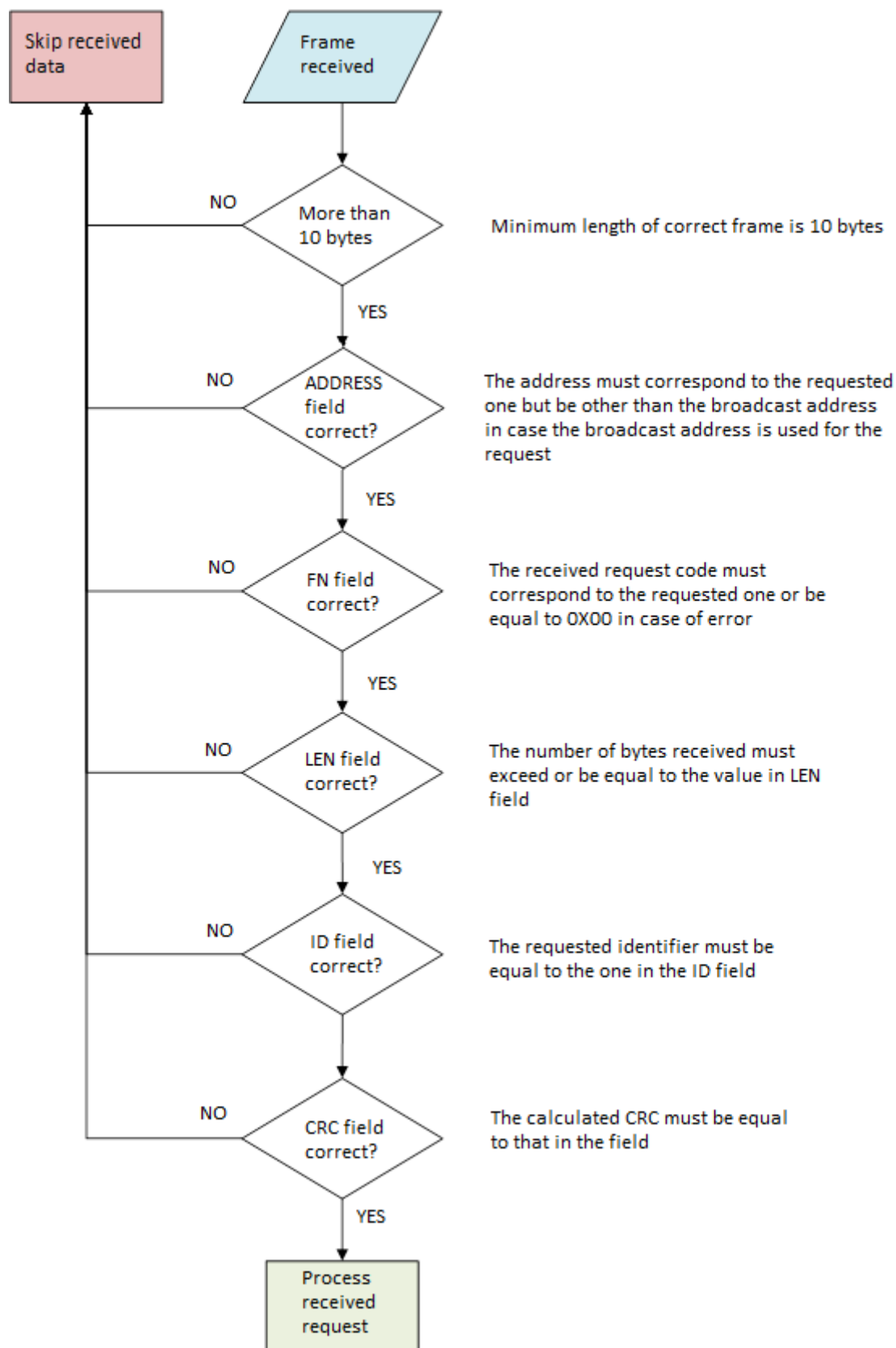


In case of validation of incoming frames, it is convenient to calculate the checksum using all bytes, including the checksum field itself. If the checksum is correct, the result must equal [0x0000](#).

## 2.3 Validation algorithm of Incoming Frames for the End Device



## 2.4 Validation algorithm of Incoming Frames for the Master



### 3 Application Layer

This section describes the formats of standard functions. Only the function body (without the frame) is described. Some end device models may not support all standard features.

#### 3.1 Date/Time Structure

Structure name: DATETIME.

Field	YEAR	MONTH	DAY	HOUR	MINUTE	SECOND
Format	UINT8	UINT8	UINT8	UINT8	UINT8	UINT8
Size	1	1	1	1	1	1

Description of fields:

- **YEAR** — year, counted from 2000, valid values [0...99].
- **MONTH** — month, valid values [1...12].
- **DAY** — day of month, valid values [1...31].
- **HOUR** — hour, valid values [0...23].
- **MINUTE** — minute, valid values [0...59].
- **SECOND** — second, valid values [0...59].

Notes:

- If all bytes are equal to 0xFF, the date/time is considered to be missing.

#### 3.2 Channel Mask Structure

Structure name: CHMASK.

Field	MASK
Format	UINT32
Size	4

Description of fields:

- **MASK** — bitmask of channels. For example, if the least significant bit is set, then channel No. 1 is used (count from 1).



### 3.3 Reading Current Date/Time

Function: 0x04.

Request: no data available.

Response:

Field	<b>DATETIME</b>
Format	DATETIME
Size	6

### 3.4 Writing Current Date/Time

Function: 0x05

Request:

Field	<b>DATETIME</b>
Format	DATETIME
Size	6

Response:

Field	<b>STATUS</b>	<b>Z1</b>	<b>Z2</b>	<b>Z3</b>
Format	UINT8	UINT8	UINT8	UINT8
Size	1	1	1	1

Description of fields:

- **STATUS** — result of writing. 0x00 — error, 0x01 — successful.
- **Z1, Z2, Z3** — pads (always zeros).

### 3.5 Reading Channels

Function: 0x01

Request:

Field	<b>MASK</b>
Format	CHMASK
Size	4

Description of fields:

- **MASK** — bitmask of channels being read.

Response:

Field	<b>CHANNEL1</b>	<b>CHANNEL2</b>	...	<b>CHANNELN</b>
Format	FLOAT64	FLOAT64		FLOAT64
Size	8	8		8

Description of fields:

- **CHANNEL1, CHANNEL2, CHANNELN** — channel readouts sorted by channel number in ascending order. IEEE 754(float64\_t) format.



The size of the channel readouts is the same, so you can determine the size of the channel readout by dividing the number of received bytes by the number of channels requested.

### 3.6 Channel Writing

Function: 0x02

Request:

Field	<b>MASK</b>	<b>CHANNEL</b>
Format	CHMASK	FLOAT64
Size	4	8

Description of fields:

- **MASK** — bitmask of the channel being written.
- **CHANNEL** — new readout for the channel. IEEE 754(float64\_t) format.

Response:

Field	<b>MASK</b>
Format	CHMASK
Size	4

Description of fields:

- **MASK** — bitmask of a successfully written channel.

### 3.7 Reading Pulse's scale factor by Channels

Function: 0x07

Request:

Field	<b>MASK</b>
Format	CHMASK
Size	4

Description of fields:

- **MASK** — bitmask of pulse scale factor of channels being read.

Response:

Field	<b>CHANNEL1</b>	<b>CHANNEL2</b>	...	<b>CHANNELN</b>
Format	FLOAT32	FLOAT32		FLOAT32
Size	4	4		4

Description of fields:

- **CHANNEL1, CHANNEL2, CHANNELN** — pulse scale factor readouts for channels sorted by channel number in ascending order. IEEE 754(float32\_t) format.

### 3.8 Writing Channel Pulse scale factor

Function: 0x08

Request:

Field	<b>MASK</b>	<b>CHANNEL</b>
Format	CHMASK	FLOAT32
Size	4	4

Description of fields:

- **MASK** — bitmask of the channel with the pulse scale factor being written.
- **CHANNEL** — new pulse weight readout for the channel. IEEE 754(float32\_t) format.

Response:

Field	<b>MASK</b>
Format	MASK
Size	4

Description of fields:

- **MASK** — bitmask of the channel with the successfully written pulse scale factor.

### 3.9 Reading Status of the Inputs

Function: [0x19](#)

Request:

Field	<b>MASK</b>
Format	MASK
Size	4

Description of fields:

- **MASK** — bitmask of channels being read.
- Response:

Field	<b>MASK</b>
Format	UINT32
Size	4

Description of fields:

- **MASK** — bitmask of channels being read. The set bits correspond to the open state of the sensors, while the reset bits correspond to the closed state.

### 3.10 Reading the History Data by Channels

Function: [0x06](#)

Request:

Field	MASK	TYPE	DATE_START	DATE_END
Format	CHMASK	UINT16	DATETIME	DATETIME
Size	4	2	6	6

Description of fields:

- **MASK** — bitmask of the channel being read. Attention! Only one mask bit must be set, i.e. the history can only be requested through one channel per request.
- **TYPE** — history type. [0x0001](#) — hourly, [0x0002](#) — daily, [0x0003](#) — monthly.
- **DATE\_START, DATE\_END** — start and end date of the history readouts. The device has the right to transmit less readouts but it must start from the start date.

Response:

Field	MASK	DATE_START	VALUE1	VALUE2	...	VALUEN
Format	CHMASK	DATETIME	FLOAT32	FLOAT32		FLOAT32
Size	4	6	4	4		4

Description of fields:

- **MASK** — bitmask of the channel being read.
- **DATE\_START** — initial date of history readouts, the same as in the request.
- **VALUE1, VALUE2, VALUEN** — history readouts on the read-out channel. IEEE 754(float32\_t) format.



If the received history readout is equal to 0xFFFFFFFF0 then there seems to be no data.

### 3.11 Reading the Parameter

Function: 0x0A

Request:

Field	INDEX
Format	UINT16
Size	2

Description of fields:

- **INDEX** — index of the parameter being read.

Response:

Field	VALUE
Format	BLOB
Size	8

Description of fields:

- **VALUE** — value of the parameter being read. The field size is fixed and limited to 8 bytes. The format and actual size of the parameter value vary and are described in the end device model parameter table.

### 3.12 Writing the Parameters

Function: [0x0B](#)

Request:

Field	INDEX	VALUE
Format	UINT16	BLOB
Size	2	8

Description of fields:

- **INDEX** — index of the parameter being read.
- **VALUE** — new value of the parameter being written. The field size is fixed and limited to 8 bytes. The format and actual size of the parameter value vary and are described in the end device model parameter table. If the actual parameter value is less than 8 bytes, the remaining bytes are filled with [0x00](#).

Response:

Field	STATUS
Format	UINT16
Size	2

Description of fields:

- **STATUS** — result of parameter writing, always [0x0000](#) (successful).

### 3.13 Response Error

Function: [0x00](#)

Response:

Field	CORE
Format	UINT8
Size	1

Description of fields:

- **CODE** — error code



Standard error codes:

- 0x01 — requested function code missing.
- 0x02 — request bitmask error.
- 0x03 — invalid request length.
- 0x04 — missing parameter.
- 0x05 — writing locked, authorization required.
- 0x06 — written value (parameter) outside the specified range.
- 0x07 — requested history type missing.
- 0x08 — maximum number of operations or data per batch exceeded.



Response errors can occur after any request. Therefore, when receiving a response, you should always check the function in the response. For older firmware versions, 0x00 and ID=0x0000 correspond to an unknown error type.

## 4 Appendix

### 4.1 Table of Fixed Parameter Values

Parameter	Index	Access	Data type	Format
Auto-switching to daylight saving time	0x0001	<b>R/W</b>	UINT16	0 — off. 1 — on.
Pulse width	0x0003	<b>R/W</b>	FLOAT32	ms (10..1999)
Pause time	0x0004	<b>R/W</b>	FLOAT32	ms (10..1999)
Firmware version	0x0005	<b>R</b>	UINT16	[1...65535]
Diagnostics	0x0006	<b>R</b>	UINT64	bit2 — memory error bit0 — character error in the channel(s)