

# SIEMENS



## FibroLaser™ III

### Communication Manual (LON-Protocol)

Liefermöglichkeiten und technische Änderungen vorbehalten.  
Data and design subject to change without notice. / Supply subject to availability.  
Sous réserve de modifications techniques et de la disponibilité.  
© 2015 Copyright by  
Siemens Switzerland Ltd

Wir behalten uns alle Rechte an diesem Dokument und an dem in ihm dargestellten Gegenstand vor. Der Empfänger anerkennt diese Rechte und wird dieses Dokument nicht ohne unsere vorgängige schriftliche Ermächtigung ganz oder teilweise Dritten zugänglich machen oder außerhalb des Zweckes verwenden, zu dem es ihm übergeben worden ist.

We reserve all rights in this document and in the subject thereof. By acceptance of the document the recipient acknowledges these rights and undertakes not to publish the document nor the subject thereof in full or in part, nor to make them available to any third party without our prior express written authorization, nor to use it for any purpose other than for which it was delivered to him.

Nous nous réservons tous les droits sur ce document, ainsi que sur l'objet y figurant. La partie recevant ce document reconnaît ces droits et elle s'engage à ne pas le rendre accessible à des tiers, même partiellement, sans notre autorisation écrite préalable et à ne pas l'employer à des fins autres que celles pour lesquelles il lui a été remis.

<b>1</b>	<b>Introduction .....</b>	<b>4</b>
<b>2</b>	<b>General Information .....</b>	<b>5</b>
2.1	Physical Interfaces .....	5
2.1.1	Serial Interface .....	5
2.1.2	Ethernet.....	5
2.2	Identifiers and Data Types .....	6
2.3	Telegram Coding.....	7
2.3.1	General Structure.....	7
2.3.2	CRC8 Check Sum.....	7
2.3.3	LON Address Conventions .....	8
2.3.4	LON over TCP/IP connections .....	8
<b>3</b>	<b>Function Codes .....</b>	<b>9</b>
3.1	Overview .....	9
3.2	Alarm Zone Related Data .....	9
3.2.1	Alarm Triggering Locations .....	9
3.2.2	Alarms .....	10
3.2.3	Average Zone Temperature .....	11
3.2.4	Maximum Zone Temperatures .....	12
3.2.5	Minimum Zone Temperatures.....	12
3.2.6	Acknowledge and Reset .....	13
3.3	Data Transmission .....	13
3.3.1	Start Telegram .....	14
3.3.2	Data Telegram .....	14
3.3.3	End Telegram.....	15
3.4	Miscellaneous .....	15
3.4.1	Software Version.....	15
3.4.2	Attendance Check.....	15
3.4.3	Controller Address .....	16
3.4.4	Device Status .....	16
3.4.5	Event Memory.....	18
3.4.6	Date and Time.....	18
3.5	Errors and Notices .....	19
3.5.1	General Telegram Structure .....	19
3.5.2	Error and Notice Codes .....	19
<b>4</b>	<b>References.....</b>	<b>21</b>
<b>A.</b>	<b>Appendices.....</b>	<b>21</b>
A.1	Discontinued Function Codes .....	21
A.2	CRC8 Algorithm .....	23
A.3	TPC access code example .....	24

# 1 Introduction

---

The LON Protocol for the OTS30xx series of controllers has been designed to be compatible with the previous versions of the OTS100 and OTS-X public code protocols. Due to the extended capabilities of the OTS30xx devices compared to previous OTS devices, some function codes had to be added.

The new capabilities are provided with new function codes. Important changes are marked with **OTS3** at the margin.

The LON Protocol definitions for the previous series of OTS controllers are not included in the document. See the following documents for further information about older devices:

- OTS100 controllers with firmware version up to 10.56: Optical temperature measurement system, Single-Channel OTS-System, Communication Protocol and public codes. Version 1.12 of 17 January 2005.
- OTS100 controllers with firmware version up to 10.65, OTS Generation 2 controllers with firmware version 2005 or with newer firmware versions in Version 2005 compatibility mode: Optical temperature measurement system, Single/Multi Channel OTS-System (OTS Generation 2), Software Versions 40xx / 45xx / 50xx / 55xx, Communication Protocol and public codes (Version 2005 Compatible Mode). Version 1.11 of 26 January 2009.
- OTS-X controllers with firmware version 4000 or newer in standard mode: Optical temperature measurement system, Single/Multi Channel OTS-System (OTS Generation 2), Software Versions 40xx / 45xx / 50xx / 55xx, Communication Protocol and public codes. Version 1.11 of 26 January 2009.

## 2 General Information

### 2.1 Physical Interfaces

#### 2.1.1 Serial Interface

The serial interface is a standard RS-232 connection. The wiring using a cable with two SUB-D 9-pin female connectors is depicted in Figure 1. The parameters of the connection are:

- 19200 Baud or 115200 Baud data rate. Please see the FibroManager Manual for details about changing the baud rate of the connection;
- 8 data bits, 1 stop bit;
- no parity;
- RTS/CTS hardware flow control.

**Notes:**

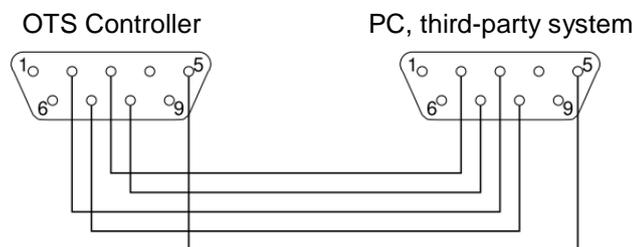


Fig. 1 RS-232 Wiring

- The serial driver on Microsoft Windows systems has some issues in proper handling of the hardware flow control at high data rates. If the OTS reports any CRC errors when using a connection to a Windows system, select the lower data rate of 19200 Baud for the connection and/or disable the send FIFO in the Windows serial driver.
- The typical maximum cable length is 15 m at 19200 Baud and 2 m at 115200 Baud. Somewhat longer lengths maybe realized using a low capacitance (< 2500 pF) cable.
- If the OTS system measures a large number of zones at short measurement times, make sure that the selected data rate is sufficient to transmit all selected data. Omit data from the transmission or use the Ethernet interface (see 2.1.2) if necessary.

#### 2.1.2 Ethernet

The Ethernet interface of the OTS controller complies with the 100 MBit/sec '100BASE-T' Ethernet standard (IEEE 802.3 Clause 25, also known as IEEE 802.3u), using standard Cat5 cables. It is fully compatible with the 10 MBit/sec '10BASE-T' Ethernet standard (IEEE 802.3 Clause 14, also known as IEEE 802.3i) over Cat3 or Cat5 cables.

The Ethernet interface can use auto-negotiation or run at fixed data rates of 10 or 100 MBit/sec with half-duplex or full-duplex data transfer. See the FibroManager Manual for details.

## 2.2 Identifiers and Data Types

The data types comply with the ISO/IEC 9899:1999 and IEEE 1003.1™-2008 standards [2, 3] as defined in table 1.

Date and time are transmitted as 22 x *char* using the format

*\_dd-mmm-yyyy\_HH:MM:SS\_*

unless otherwise noted.

**OTS3**

Time stamps are given in Universal Coordinated Time (UTC) if the OTS controller is configured to synchronize with a NTP server. If the time in the OTS controller is set manually using FibroManager, the OTS will use this time as reference. Daylight saving times are not used. See section 3.4.6 for details reading the current date and time.

Further conventions are:

- All data types longer than 8 bits (1 byte) are transmitted binary in low byte first byte order<sup>1)</sup> 'little endian'.

<sup>1)</sup> This is the native byte order of e.g. Intel based systems. On big endian systems (high byte first, e.g. Power PC, SUN SPARC), the data must be swapped appropriately.

Data type	Definition
<i>int8_t</i>	8 bit two's complement signed integer (1 byte)
<i>char</i>	ASCII character (1 byte)
<i>bool</i>	truth value (1 byte), transmitted as char '0' (48dec, 30hex) representing the value FALSE and '1' (49dec, 31hex) representing the value TRUE, respectively
<i>int16_t</i>	16 bit two's complement signed integer (2 byte)
<i>unit8_t</i>	8 bit unsigned integer (1 byte)
<i>unit16_t</i>	16 bit unsigned integer (2 byte)
<i>unit32_t</i>	32 bit unsigned integer (4 byte)
<i>time_t</i>	date and time according to IEEE 1003.1™-2008, section A.4.15 [2] (4 byte)
<i>float</i>	32 bit floating point value (4 byte according to IEEE Std 754-1985 [1])

Tab. 1 Data types

Field	Meaning
<i>dd</i>	day, using two digits with a leading '0' if necessary. The value must be in the range between 1 and the maximum days within the respective month.
<i>mmm</i>	month as a case sensitive abbreviated string ' <i>Jan</i> ', ' <i>Feb</i> ' etc
<i>yyyy</i>	year
<i>HH, MM, SS</i>	hour, minute and second, each using two digits with a leading '0' if necessary. The values must be in the range from 0 to 23 (hour) or 59 (minute, second), respectively.

Tab. 2 Data fields

- The Fiber numbers run from 0 up to a maximum of 47, depending on the number of channels present. Fiber numbers are transmitted binary unless otherwise noted.
- A request is put using a telegram with one user data byte *char* '?' (63dec, 3Fhex).
- A fiber-specific request is put using a telegram with two user data bytes *char* '?' followed by *uint8\_t* fiber number.

## 2.3 Telegram Coding

### 2.3.1 General Structure

The data transmission between the OTS3 and the FibroManager or a third-party system is based on the exchange of telegrams.

Each telegram consists of

- a header with a size of 6 bytes and
- an optional user data block containing 0 to 214 bytes.

The header contains the sender and recipient addresses as well as the function code (FC) defining the function of the telegram. The user data byte count gives the number of bytes in the user data block (0 to 214). The interpretation of the optional user data block is determined by the function code. For details see section 3.

The whole telegram is checked using a CRC algorithm. Table 3 represents the general structure of the telegrams.

### 2.3.2 CRC8 Check Sum

The correct and undisturbed data transmission is validated using a CRC8 check sum. This check sum, transmitted as first byte in the telegram, is calculated over the whole telegram data starting from the recipient's LON address and including all user data. An example implementation for the CRC8 algorithm can be found in appendix A.2.

Header					Data
<i>uint8_t</i>	<i>uint8_t</i>	<i>uint8_t</i>	<i>uint8_t</i>	<i>uint8_t</i>	0 to 214 x <i>uint8_t</i>
CRC8	LON address recipient	LON address sender	function code (FC)	user data byte count	user data (optional)

Tab. 3 LON telegram structure

### 2.3.3 LON Address Conventions

---

The LON addresses are used to identify the OTS controllers and third-party products in a network structure. Note that each LON address must be unique in each network segment. The address assignments for these systems are as follows:

<b>System type</b>	<b>Valid LON addresses</b>
third-party system	always 0
OTS controller	2 to 255 Factory default settings are calculated from series number as follows: <ul style="list-style-type: none"><li>• OTS 1002 to OTS 1201: LON 2 to 201</li><li>• OTS 1202 to OTS 1401: LON 2 to 201</li><li>• ...</li></ul> Other LON addresses can be assigned to each controller using FibroManager.

**OTS3**

LON address 1, which has been used to identify FibroManager in previous OTS firmware versions, is not used for OTS30xx systems.

### 2.3.4 LON over TCP/IP connections

---

The Ethernet interface of the OTS includes the telegrams described in section 2.3.1 into standard TCP/IP packets according to Internet Standard RFC 793 [4]. The OTS acts as a socket server which can be connected using the standard IEEE 1003.1™-2008 *connect ()* system call [2]. The OTS controller supports up to 8 simultaneous connections.

An example for accessing an OTS controller over a TCP connection is shown in appendix A.3.

## 3 Function Codes

### 3.1 Overview

The overview on all function codes is available in table 4.

Code	Usage	Definition in section	Page
352	Alarm triggering location	3.2.1	9
355	Average zone temperature	3.2.3	11
356	Maximum zone temperature	3.2.4	12
361	Minimum zone temperature	3.2.5	12
371	Data transmission follow-up	3.3.2	14
372	Data transmission finalising	3.3.3	15
374	Data transmission start	3.3.1	14
379	Alarm	3.2.2	10
382	Device status	3.4.4	16
383	Event memory	3.4.5	18
391	Date and time	3.4.6	18
395	Acknowledge and reset	3.2.6	13
1005	Software version	3.4.1	15
1099	Attendance check	3.4.2	15
1800	Controller address	3.4.3	16
19xx	Errors and notices	3.5.2	19

Tab. 4 Function codes overview

### 3.2 Alarm Zone Related Data

The OTS controller can monitor temperatures and temperature changes in zones of each fiber, and it can activate alarms at certain conditions. The telegrams described here transmit all zone related information like alarms, zone average, maximum and minimum temperatures etc.

Please see the FibroManager Manual for further information on the set-up of alarm zones and the selection of information to be transmitted.

#### 3.2.1 Alarm Triggering Locations

##### Function Code 352

**Usage** In case of an alarm (see 3.2.2) function code 352 transmits the alarm triggering locations. The user data of the FC352 telegram contains the meter positions of alarm triggering locations along the sensor fiber. The meter position refers to the optical connector at the OTS or any other configured reference point. Any 'point of return' set in the configuration is not taken into account, thus all data are given linearly. The structure of the user data section of the telegram is

- *uint8\_t* fiber number;
- 1 to 106 x *int16\_t* meter positions of alarm triggering locations along the sensor cable.

Large fires usually trigger a number of locations. In order to reduce the amount of data, adjacent triggering locations are transmitted only as start and end position. In that case, the end position is transmitted as a negative number.

Example: The telegram contains the three user data values 705, 3360, -3492. This represents an isolated triggering location at 705 m and a contiguous trigger range of 132 m length, starting at 3360 m and ending at 3492 m.

If the number of detected alarm positions is larger than the maximum number compatible with the telegram format, only the first 106 positions and/or delimiters are transmitted. In addition, this event triggers notice 1974/D0.

**Query** To enquire the alarm triggering locations detected by the last measurement of a fiber, a telegram with FC352 and two user data bytes

- *char* '?'
- *uint8\_t* fiber number

has to be sent to the controller. This request triggers notice 1964/7X if no measured data is available.

### 3.2.2 Alarms

#### Function Code 379

**OTS3**

**Usage** Function code 379 transmits address points triggered by alarms, fiber breaks, errors or digital input events. The structure of the user data section in the FC379 telegram is

- *int8\_t* fiber number. This value is -1 (FFhex); for non fiber-specific telegrams, e.g. error messages or input event address points.
- 0 up to 48 address points, each having the format

**OTS3**

- *uint16\_t* address point number in the range of 1 to 3100 (fiber address points 1 to 3100) or 1 to 255 (system address point 1 to 255), respectively<sup>2)</sup>;
- <sup>2)</sup> Note that previous OTS firmware versions transmitted fiber address points as fiber number \* 256 + logical address point.

**OTS3**

- *uint8\_t* flags. Note: This value is 1 for a fiber break, error, warning and input event. For alarm or pre-alarm events, the value represents a bit mask encoding active conditions according to the alarm parameter set. At least one bit is set for alarm or pre-alarm events:

Bit #	Alarm or pre-alarm triggering event
1	maximum temperature criterion has triggered the alarm or pre-alarm
2	minimum temperature criterion has triggered the alarm or pre-alarm
3	hot spot criterion has triggered the alarm or pre-alarm
4	first differential criterion has triggered the alarm or pre-alarm
5	second differential criterion has triggered the alarm or pre-alarm
6	the third differential criterion has triggered the alarm or pre-alarm
7	indicates an address point triggered by a simulation command

Tab. 5 Alarm and pre-alarm events

**1.2.0.0**

Starting with Firmware revision 1.2.0.0, alarms and MANUAL RESET pre-alarms which are not 1.2.0.0 active any more are reported as address points with 0 flags until they are reset by a reset operation. Previous firmware revisions do not report

alarms and MANUAL RESET pre-alarms which are not active any more, i.e. in these firmware revisions the flags are never 0 for alarms or pre-alarms.

For a control function event, only bit 0 is used. The flags value is 1 (01hex) if the control function is activated, and 0 (00hex) when the system reports the deassertion of a control function.

If an alarm triggers more than 48 address points, multiple telegrams are transmitted. If no alarm is detected, the controller sends a FC379 telegram without address points after the measurement.

**Query** To enquire the alarm address points from the last measurement, a telegram with FC379 and two user data bytes

- *char* '?'
- *uint8\_t* fiber number

has to be sent to the controller. This request triggers notice 1964/7X if no measured data is available.

**Notes:** Only the latest alarm address points are send on enquiries, but address points of fiber breaks nor system no events.  
Function code 379 replaces function code 354 of previous versions of the protocol.

### 3.2.3 Average Zone Temperature

---

#### Function Code 355

**Usage** Function code 355 transmits average zone temperatures. Each FC355 telegram contains the average zone temperatures of up to 50 zones. For higher numbers of zones per fiber, up to 19 further telegrams (with increasing block numbers) are transmitted. The structure of the user data section of these telegrams is

- *uint8\_t* fiber number;
- *uint8\_t* block number (1 to 20);
- 1 to 50 x *float* average zone temperatures in order of zone numbers. The average zone temperatures are given in °C. For hidden zones and zones completely located behind a fiber break, the returned value is -1000.0. For zones containing a fiber break, the average zone temperature is calculated only from locations before the fiber break position.

**Query** To enquire the average zone temperatures from the last measurement, a telegram with FC355 and two user data bytes

- *char* '?'
- *uint8\_t* fiber number

has to be sent to the controller. This request triggers notice 1964/7X if no measured data is available.

**Note:** A query is only answered if the controller start-up has been completed.

### 3.2.4 Maximum Zone Temperatures

---

#### Function Code 356

**Usage** Telegrams with function code 356 transmit maximum zone temperatures. Each telegram contains the maximum zone temperatures of up to 50 zones. For higher numbers of zones per fiber, up to 19 further telegrams with increasing block numbers are transmitted. The structure of the user data section of these telegrams is

- `uint8_t` fiber number;
- `uint8_t` block number (1 to 20);
- 1 to 50 x `float` maximum zone temperatures in order of zone numbers. The maximum zone temperatures are given in °C. For hidden zones and zones completely located behind a fiber break, the returned value is -1000.0. For zones containing a fiber break, the maximum zone temperature is calculated only from locations before the fiber break position.

**Query** To enquire the maximum zone temperatures from the last measurement, a telegram with FC356 and two user data bytes

- `char` '?'
- `uint8_t` fiber number

has to be sent to the controller. This request triggers notice 1964/7X if no measured data is available.

**Note:** A query is only answered if the controller start-up has been completed.

### 3.2.5 Minimum Zone Temperatures

---

#### Function Code 361

**Usage** Telegrams with function code 361 transmit minimum zone temperatures. Each telegram contains the minimum zone temperatures of up to 50 zones. For higher numbers of zones per fiber, up to 19 further telegrams with increasing block numbers are transmitted. The structure of the user data section of these telegrams is

- `uint8_t` fiber number
- `uint8_t` block number (1 to 20);
- 1 to 50 x `float` minimum zone temperatures in order of zone numbers. The minimum zone temperatures are given in °C. For hidden zones and zones completely located behind a fiber break, the returned value is -1000.0. For zones containing a fiber break, the minimum zone temperature is calculated only from locations before the fiber break position.

**Query** To enquire the minimum zone temperatures from the last measurement, a telegram with FC361 and two user data bytes

- `char` '?';
- `uint8_t` fiber number

has to be sent to the controller. This request triggers notice 1964/7X if no measured data is available.

**Note:** A query is only answered if the controller start-up has been completed.

### 3.2.6 Acknowledge and Reset

---

#### OTS3

#### Function Code 395

**Usage** Function code 395 triggers an Acknowledge or a Reset operation of the controller. The controller also transmits function code 395 after execution of Acknowledge and Reset operations.

**Report** The OTS3 reports an Acknowledge or Reset operation in a telegram with function code 395 and two data bytes

- *char* operation, with value 'R' for Reset or 'A' for Acknowledge;
- *uint8\_t* source triggering the operation:
  - 1: key switch of the OTS3
  - 2: input via relay
  - 3: command from FibroManager
  - 4: command from a third-party system (e.g. function code 395 or SCADA protocol<sup>3)</sup>)
  - other values: not used.

<sup>3)</sup> SCADA protocols like MODBUS, DNP3 or IEC 60870-5-104 are only available if the respective licenses have been purchased and installed.

**Initiate** The Acknowledge or Reset operations are initiated by a telegram with function code 395 and one user data byte:

- *char* operation, with the value 'R' for Reset or 'A' for Acknowledge

The controller executes the requested operation and replies with a FC395 telegram as defined above.

**Note:** If the Acknowledge or Reset operations change the state of any output, the system transmits a device status telegram (see 3.4.4).

Function code 395 replaces function code 1515 of previous versions of the protocol.

## 3.3 Data Transmission

---

**Note:** The optional data transmission to a third-party system is only available if the corresponding license has been purchased and installed.

Temperature and backscatter data is too large for the 214 byte user data section of a single telegram.

Therefore, it is split into

- a start telegram with FC374,
- the required number of data telegrams with function code 371 and
- an end telegram with function code 372.

Data is compressed for transmission according to the Internet standards RFC 1950 [5] and RFC 1951 [6] to minimize transmission time.

Data is transmitted automatically after each measurement as defined in the OTS3 configuration (see the Software Reference Manual FibroManager for details). The transmission of the last measured temperature profile can be additionally requested using FC374.

### 3.3.1 Start Telegram

---

#### Function Code 374

**Usage** Function code 374 indicates the start of a data transmission. The telegram consists of

- the general header,
- the specific header and
- the first part of the compressed data stream.

**General Header (34 user data bytes)** The general header starts at offset 0 in the user data area of the start telegram.

- *uint16\_t* type of the transmitted data:
  - 0 (0000hex): temperature profile in °C;
  - 1 (0001hex): backscatter profile, Stokes signal;
  - 2 to 65535 (0002hex – FFFFhex): not used.
- 32 x *uint8\_t* not used.

**Specific Header (33 user data bytes)** The specific header starts at offset 34 (22hex) in the user data area of the start telegram.

- *uint8\_t* fiber number;
- *uint32\_t* number of data points;
- float spatial resolution in mm;
- 22 x *char* date and time of the measurement (see 2.2);
- 2 x *uint8\_t* not used.

**Data (up to 147 user data bytes)** The first chunk of data starts at offset 67 (43hex) in the user data area of the start telegram. If the total number of compressed bytes is smaller than 147 bytes, the length of the start telegram is shortened accordingly.

**Note:** Temperature data behind a fiber break location are transmitted as -1000.0.

**Query** The transmission of the temperature profile from the last measurement can be requested by a telegram with FC374 and two user data bytes:

- *char* '?';
- *uint8\_t* fiber number.

It is not possible to request the transmission of backscatter profiles. The request triggers notice 1964/7X if no temperature data is available.

### 3.3.2 Data Telegram

---

#### Function Code 371

**Usage** If the length of the compressed data stream exceeds the available numbers of bytes in the start and end telegrams, data telegrams are transmitted between start and end telegram.

The user data section of the data telegrams contains:

- *uint16\_t* sequential number of telegram: The sequential number of the data and end telegrams starts with 0 for the first data telegram in each sequence. If this number overflows at 65535 (FFFFhex), it rolls over to 0.
- 212 x *uint8\_t* compressed data.

### 3.3.3 End Telegram

---

#### Function Code 372

**Usage** The end telegram indicates the end of a data transmission sequence and contains the last part of the compressed data stream. The user data section of this telegram contains:

- *uint16\_t* sequential telegram number (see 3.3.2);
- 0 to 212 x *uint8\_t* compressed data.

## 3.4 Miscellaneous

### 3.4.1 Software Version

---

#### Function Code 1005

**Usage** The software version of the controller can be requested using a FC1005 telegram with one user data byte char '?'. The request triggers a telegram with FC1005 and 6 user data bytes:

- *float* software version xx.xxyyy, e.g. the value '40.00104' represents version 4000, revision 104;
- *int16\_t* release code

### 3.4.2 Attendance Check

---

#### Function Code 1099

**Usage** Function code 1099 transmits the system status and status changes. The system transmits a telegram with function code 1099 before starting each measurement or measurement sequence.

**Query** The status information can be requested using a FC1099 telegram with one user data byte

- *char* '?'.

The request triggers a FC1099 telegram containing three user data bytes:

- 1 x *uint8\_t* status bit mask.

The status bits represent the following information:

Bit	Description	Information
0	measurement status	This bit is set if the OTS is measuring.
1	full alarm processing	This bit is set if valid alarm zones are defined for all fibers in the measurement sequence including spare fibers.
2	measurement cycle separator	A telegram with this bit set is transmitted by the controller before starting any measurement in a sequence. The fiber number in this telegram represents the fiber to be measured next.
3	measurement sequence separator	A telegram with this bit set is transmitted by the controller at the beginning of each measurement sequence. The fiber number (Byte 3) is

<b>OTS3</b>		set to -1 (FFhex) in the measurement sequence separator telegram.	
	4	always set 1	Note: This bit has been used in previous versions as internal status flag.
	5	No fiber break	This bit is set if none of the fiber involved in the active measurement sequence including spares exhibits a fiber break.
	6	single-fiber measurement	This bit is set if just one fiber is to be measured and no sequence of fibers.
	7	end of measurement	The controller transmits a telegram with this bit set after the end of each measurement or when a measurement has been cancelled. In both cases, the fiber number represents the last measured fiber.

Tab. 6 Status bits

- 1 x *uint8\_t* general operating mode bit mask:
  - Bit 0: always 0;
  - Bit 1: always 1;
  - Bit 2 to 7: not used.

**OTS3**

In previous versions of the protocol, bits 0 and 1 have been used to indicate the recipient of the data. The values given above are compatible with 'transmit data to third-party systems' (LON 0) of previous protocol versions.

- 1 x *uint8\_t* fiber number 0 to 47 or -1 (FFhex) if the telegram is not assigned to a specific fiber (see above).

**Notes** When transmitting OTS data over an Ethernet connection Attendance Check requests should be sent periodically during periods of network inactivity to 'keep alive' ('ping') the connection. See section 2.1.2 for further details.

**OTS3**

The 'detailed system status' function of previous protocol versions is no further supported.

### 3.4.3 Controller Address

---

**Function Code 1800**

**Usage** An empty FC1800 telegram sent to the controller triggers a reply with a FC1800 telegram containing one user data byte uint8 controller LON address.

### 3.4.4 Device Status

---

**Function Code 382**

**OTS3**

**Usage** Function code 382 is used to report the status of the OTS controller's inputs, outputs, explosion protection and test key switch.

**Note:** This function code replaces function code 381 of previous protocol versions.

The system can be configured to transmit a FC382 telegram whenever the state of an input or output changes. See the Software Reference Manual FibroManager for details.

The FC382 telegram contains 32 user data bytes:

- 14 x *uint8\_t* output status as a set of bits, starting with the state of the first output (output 1) in bit 0 of the first element. The index *i*, counting from 0 to 13, inclusively, and bit *b*, counting from 0 to 7, inclusively, containing the logical state of output *n* are calculated as follows:

$$i_o = \left\lceil \frac{n_o - 1}{8} \right\rceil \quad b_o = (n_o - 1) - 8i_o$$

The usage of the output bits depends on the number of extra output interfaces of the OTS controller. Bits corresponding to non-present outputs are always 0.

- 5 x *uint8\_t* input status as a set of bits, starting with the state of the first input (input 1) in bit 0 of the first element. The index *i*, counting from 0 to 4, inclusively, and bit *b*, counting from 0 to 7, inclusively, containing the logical state of input *n* are calculated as follows:

$$i_i = \left\lceil \frac{n_i - 1}{8} \right\rceil \quad b_i = (n_i - 1) - 8i_i$$

The usage of the input bits depends on the number of extra input interfaces of the OTS controller. Bits corresponding to non-present inputs are always 0.

- *uint8\_t* logical state of the system fault and common alarm outputs and the optional explosion protection circuit:
  - Bit 0: logical state of the system fault output, i.e. 0 if the output is closed (no system fault);
  - Bit 1: logical state of the common alarm output, i.e. 0 if the output is open (no alarm);
  - Bit 2: indicates that the explosion protection circuit is installed. It is 0 for systems without explosion protection option.
  - Bit 3: indicates that the key switch is in the Test Mode position;
  - Bit 4 to 7: not used.
- *float* internal device temperature in °C.
- *float* internal device humidity in % rel. humidity. If the device is not equipped with an internal humidity sensor, this element contains the special value Not a Number (NaN).
- *float* power supply DC input voltage in V.

**Query** The status information is requested using a FC382 containing one user data byte

- *char* '?'.

The request triggers a FC382 telegram from the controller as defined above.

**Note:** Function code 383 replaces function code 381 of previous protocol versions.

### 3.4.5 Event Memory

---

OTS3
------

#### Function Code 383

**Usage** Function code 383 queries the event memory of the OTS controller.

**Query** The status information is requested using a FC383 telegram and one user data byte

- *char* '?'.

The request triggers a series of FC383 telegrams from the controller with 9 or 13 user data bytes, each containing one of the events from the memory:

- *time\_t* date and time of the event;
- *int8\_t* fiber number or -1 for a system event;
- *uint16\_t* event code (see 3.5.2);
- 2 x *char* event extension. Is filled with ASCII NULL characters (00hex) for events without extension (see 3.5.2);
- *float* fiber break position for event 1904 Fiber Break (see 3.5.2).

**Notes:** The report function of FibroManager can create a detailed status report. Function code 383 replaces function code 380 of previous protocol versions.

### 3.4.6 Date and Time

---

OTS3
------

#### Function Code 391

**Usage** Function code 391 queries the date and time from the OTS controller. The date and time of a controller is requested using a FC331 telegram with one user data byte

- *char* '?'.

The request triggers a FC391 telegram from the controller containing 23 user data bytes:

- 22 x *char* date and time. The format is given in section 2.2;
- *bool* NTP status. If set, the OTS controller's clock is synchronized with the configured NTP server.

**Note:** Function code 391 replaces function code 390 of previous protocol versions.

## 3.5 Errors and Notices

### 3.5.1 General Telegram Structure

An unified telegram user data structure is used for the transmission of error and notice messages. The telegram may contain

- the number of the fiber triggering the condition,
- an extension (2 char values) and/or
- one or more additional data elements.

Data sizes of fiber number and extension are fixed thus allowing a definite assignment of data. The fiber number F is transmitted as 1 byte (*uint8\_t*), the extension E as two bytes (2 x *char*), and the data D as a multiple of 4 bytes (e.g. *uint32\_t* or *float*).

Accordingly, the relation between telegram user data length and the information in the user data is as follows:

User Data Length	Data (bytes)	Fiber Number	Extension	Additional Data
0	--	--	FF	--
1	F	F	FF	--
2	EE	--	EE	--
3	FEE	F	EE	--
4	DDDD	--	--	DDDD
5	FDDDD	F	--	DDDD
6	EEDDDD	--	EE	DDDD
7	FEEDDDD	F	EE	DDDD

Currently, additional data are only transmitted for error code 1904 (fiber break, 1 float fiber break position in m). The presence of the fiber number and/or extension code in different errors and notice messages is shown in section 3.5.2.

### 3.5.2 Error and Notice Codes

Error and notice codes are listed in the following two tables. All error messages also trigger a General Error message with FC1900 and no additional data. The controller can be set not to transmit notices. See the Software Reference Manual FibroManager for details.

The entry in the column 'Data' indicates the extension to be sent. The column 'Fiber' indicates if this condition is fiber-specific.

#### OTS3

The error messages are compatible with previous protocol versions. Nevertheless, several error conditions were subsumed under new error codes to simplify their use. Some error codes from previous protocol versions have been omitted since they are not relevant for OTS3.

**Error messages**

Code	Data	Fiber	Description
1900			General error. A general error telegram is triggered by any other error, and it is sent immediately after the triggering error telegram.
1902			Measurement canceled. The measurement has been canceled by a user command or by an internal process.
1903		yes	Measurement error.
1904		yes	Fiber break. The additional data element contains the position of the fiber break in m.
1955			Measurement not active or disturbed.
1961	4O	yes	No sensor fiber connector plugged. The measurement is going to be canceled.
1962	5A		NV RAM battery empty.
1962	5E		Additional IO cards were incorrectly installed.
1970			Hardware error. The controller hardware must be repaired. Use FibroManager to create a Device Status Report and contact the LIOS support.
1971			Configuration error. No measurement is taken. Check the information reported for further details.
1972	AA		The internal humidity of the device is above the error level.
1972	AB		The power supply input voltage is below the lower or above the upper error level, respectively.
1972	AC		The internal device temperature is above the error level. The measurement will be stopped.
1972	AD		The internal device temperature is below the error level.

**Notices**

Code	Data	Fiber	Description
1925			Wrong recipient.
1928			CRC8 error in received data.
1952			Reboot after power-down.
1964	7N		Too many open TCP/IP connections. The attempt to establish a new connection is rejected, i. e. the connection will be closed (see 2.3.4).
1967	AN	yes	Communication protocol error.
1967	AQ		Unknown function code.
1967	AS		The requested data is not available. This condition can appear e.g. when temperature data is requested without any finished measurement.
1973			Internal warning.
1974	D0	yes	More than 106 alarm triggering locations have been detected (see section 3.2.1).
1974	D1		Storing event memory data on the CF card failed, some event memory entries were lost.
1974	D2		Event memory damaged, some entries were lost.

1974	D4		No CF card installed, or the CF card is damaged.
1974	DZ	yes	Internal measurement warning.
1975	AA		The internal humidity of the device is above the warning level.
1975	AB		The power supply input voltage is below the lower or above the upper warning level, respectively.
1975	AC		The internal device temperature is above the warning level.
1975	AD		The internal device temperature is below the warning level.
1976			Communication warning, check events.
1977	B1		A reset command has been received.
1977	B1		An acknowledge command has been received.
1978	C1		Test mode started.
1978	C1		Test mode ended, starting normal operation.

## 4 References

---

- [1] IEEE Standard 754-1985: IEEE Standard for Binary Floating Point Arithmetic.
- [2] The Open Group Base Specifications Issue 7: IEEE Std 1003.1™-2008 (<http://www.opengroup.org/onlinepubs/9699919799/>).
- [3] International Standard ISO/IEC 9899:1999: Programming languages — C.
- [4] INFORMATION SCIENCES INSTITUTE, UNIVERSITY OF SOUTHERN CALIFORNIA: Transmission Control Protocol (IETF RFC 793), September 1981.
- [5] P. DEUTSCH and J-L. GAILLY: ZLIB Compressed Data Format Specification version 3.3 (IETF RFC 1950), May 1996.
- [6] P. DEUTSCH: DEFLATE Compressed Data Format Specification version 1.3 (IETF RFC 1951), May 1996.

## A. Appendices

### A.1 Discontinued Function Codes

---

The following function codes from previous versions of the Communication Protocol and Public Codes are not available with OTS3:

- FC 354: alarms. Replacement: FC 379 (see 3.2.2).
- FC 360: sub-zones. The sub-zone feature is obsolete for OTS3 because of the availability of 1000 zones per fiber.
- FC 380: event memory. Replacement: FC 383 (see 3.4.5).
- FC 381: device status. Replacement: FC 382 (see 3.4.4).
- FC 390: date and time. Replacement: FC 391 (see 3.4.6).

- FC 1515: reset outputs. Replacement: FC 395 (see 3.2.6).

## A.2 CRC8 Algorithm

This implementation example of the CRC8 algorithm is suitable for all standard C compilers.

```

/** \brief Calculate a CRC8 hash value
 *
 * \param len length of data field
 * \param data data bytes
 * \return CRC8 value
 *
 * Calculate and return the CRC8 value of len bytes of data starting at
 * address data.
 */
uint8_t
calculate_crc8(int len, uint8_t * data)
{
    /* look-up table for CRC codes */
    static const uint8_t crc_table [256] =
    { 0, 94, 188, 226, 97, 63, 221, 131,
      194, 156, 126, 32, 163, 253, 31, 65,
      157, 195, 33, 127, 252, 162, 64, 30,
      95, 1, 227, 189, 62, 96, 130, 220,
      35, 125, 159, 193, 66, 28, 254, 160,
      225, 191, 93, 3, 128, 222, 60, 98,
      190, 224, 2, 92, 223, 129, 99, 61,
      124, 34, 192, 158, 29, 67, 161, 255,
      70, 24, 250, 164, 39, 121, 155, 197,
      132, 218, 56, 102, 229, 187, 89, 7,
      219, 133, 103, 57, 186, 228, 6, 88,
      25, 71, 165, 251, 120, 38, 196, 154,
      101, 59, 217, 135, 4, 90, 184, 230,
      167, 249, 27, 69, 198, 152, 122, 36,
      248, 166, 68, 26, 153, 199, 37, 123,
      58, 100, 134, 216, 91, 5, 231, 185,
      140, 210, 48, 110, 237, 179, 81, 15,
      78, 16, 242, 172, 47, 113, 147, 205,
      17, 79, 173, 243, 112, 46, 204, 146,
      211, 141, 111, 49, 178, 236, 14, 80,
      175, 241, 19, 77, 206, 144, 114, 44,
      109, 51, 209, 143, 12, 82, 176, 238,
      50, 108, 142, 208, 83, 13, 239, 177,
      240, 174, 76, 18, 145, 207, 45, 115,
      202, 148, 118, 40, 171, 245, 23, 73,
      8, 86, 180, 234, 105, 55, 213, 139,
      87, 9, 235, 181, 54, 104, 138, 212,
      149, 203, 41, 119, 244, 170, 72, 22,
      233, 183, 85, 11, 136, 214, 52, 106,
      43, 117, 151, 201, 74, 20, 246, 168,
      116, 42, 200, 150, 21, 75, 169, 247,
      182, 232, 10, 84, 215, 137, 107, 53 };
    uint8_t CRC8 = 255;
    for (; len > 0; len--)
        CRC8 = crc_table[CRC8 ^ (*data++ & 0xff)];
    return CRC8;
}

```

## A.3 TPC access code example

---

**Note:** This TCP access code example contains the minimum commands required to access an OTS controller over a TCP connection [2]. This example does not include any error handling. For details about the system calls used in this example see reference [2].

```
int handle;
struct sockaddr in address;
unsigned char buffer [220];
int running;

handle = socket (AF_INET, SOCK_STREAM, 0);
// fill address with the IP address and port of the OTS controller
Connect (handle, (struct sockaddr *) &address, sizeof (address));

running = 1;
// main loop: evaluate received telegrams
while (running) {
    fd set read rdy;
    // sleep on the handle, waiting for data
    FD_ZERO (&read rdy);
    FD_SET (handle, &read rdy);
    if (select (handle + 1, &read rdy, NULL, NULL, NULL) > 0) {
        // read telegram header
        recv (handle, buffer, 6, 0);
        // read telegram body (if any)
        if (buffer[5] > 0)
            recv (handle, buffer + 6, buffer[5], 0);
        // check CRC
        // evaluate data
    }
}

// shut down connection and close
shutdown (handle, SHUT_RDWR);
close (handle);
```



Siemens Switzerland Ltd  
Infrastructure & Cities Sector  
Building Technologies Division  
International Headquarters  
CPS Fire Safety  
Gubelstrasse 22  
CH-6301 Zug  
Tel. +41 41 724 24 24  
[www.siemens.com/buildingtechnologies](http://www.siemens.com/buildingtechnologies)

---

Document no. **A6V10335141\_b\_en\_--**  
Edition 01.2015

Manual FibroLaser  
Section 2