Optocon ASCII Communication Protocol for Fotemp of the Series Trafo, Compact and OEM

Overview

This manual is valid for all Fotemp devices of the second generation. It describes the interface communication protocol using RS232, RS485 or USB (FTDI Max230) interface.

System description

All Optocon devices have an internal RS232 Interface or RS485 bus, USB is only available as converter providing a virtual serial port. For every temperature's value has to be asked separately by the end user software, parameters are set equally. There is no timed auto-measuring/sending mode implemented in the device.

Serial port settings

The settings for the serial port are: 57600 bps connection with 8 data bits, 1 stop bit, no parity and no flow control. Please make sure to use the correct settings.

Command syntax

Fotemp devices use an ASCII protocol. A Fotemp accepts commands and requests. Commands start with a colon ":", requests with a question mark "?", followed by function number and up to two parameters finalized by a carriage return (<CR>) . In-between the separator • means <space> (ASCII 0x20). The basic structure of a command is: ?XX•CH•PM1•PM2•PM3<CR>

Requests are answered with hash mark followed by function number, channel number (when channel specific function) and the desired value, finalized with carriage return an Line feed <CR><LF> : #XX•CH•PM1<CR><LF>

Thereafter follows a positive ACK: *FF<CR><LF>

In case of an error there only is a negative ACK send from the device: *FF<CR><LF>

Commands only get an ACK – positive or negative.

Protocol Extension for Module Devices

FOTEMPMK-19'' devices contains several modules. Each of the modules has its own measuring system and a RS485 interface. In the FOTEMPMK-19'' the modules are connected via RS485 bus. Each module has its own device address depending of the slot where module is pluged in. For this reason the communication protocol is extended by address information. The address information is in the front of each telegram. The address part begins with the character 'A' followed by two hexadecimal digits representing the address number in hexadecimal format: AXX•<Telegram>
Then the telegram part described in chapter Command syntax follows separated by space character.

Example for reading temperature of channel 2 from module with device address 5:
Send telegram: A05 ?01 02<CR>
Response: A05 #01 01 235<CR><LF>*00<CR><LF>
The received temperature is valid and has a value of 23.5 °C.

Commands

This chapter describes all commands a requests known by Optocon Fotemp devices of the 2nd generation. Some commands depend on the devices' hardware configuration.

Temperature Readings

01 –Averaged Temperature of a Channel

Command 01 read only

Parameter 1: Number of channel to be read, channels are numbered from 1 to the channel count of your device.

Request example: `?01•2<CR>` asks for the averaged temperature of channel 2. For averaging settings refer to command 53 below.

Answer example: `#01•1•-135<CR><LF>*00<CR><LF>` The first parameter indicates the temperature state, where 1 means „new temperature" and 0 means the temperature has already been read. In the second parameter the temperature is encoded as 1/10 degree Celsius: in this case -13.5°C. Negative temperature values begins with a minus (ASCII 0x2D). If a channel shows „9999" it will have one of the 3 reasons:

1. No senor connected
2. sensor is defect
3. channel is switched off

02 –Averaged Temperatures of all Channels

Command 02 read only.

Request example: `?02<CR>` asks for the averaged temperature of all channels of the device.

Answer example: `#02•234•-114•----•2345<CR><LF>*00<CR><LF>` Every parameter represents the temperature of the channel in the order of channel numbers (the example shows a 4 channel device). Temperatures below 0°C begins with a minus character (ASCII 0x2D). In the example the third channel showing „---". That means a disconnected, switched off or defective sensor.

03 - Current Temperature of a Channel

Command 03 read only

Parameter 1: Number of channel to be read, channels are numbered from 1 to the channel count of your device.

Request example: `?03•1<CR>` asks for the temperature of channel 1.

Answer example: `#03•1•234<CR><LF>` The first parameter indicates the temperature state, where 1 means „new temperature" and 0 means the temperature has already been read. The second parameter represents the temperature encoded as 1/10 degrees Celsius. Negative values begins with a minus character (ASCII 0x2D). A channel value of 9999 could be cause by the following reasons:

1. No senor connected
2. sensor is defect
3. channel is switched off

04 – Current Temperature of all Channels

Command 04 read only

Request example: `?04<CR>` asks for the temperature of all channels of the device.

Answer example: `#04•234•-114•---•2345<CR><LF>*00<CR><LF>` Every parameter represents the temperature of the channel in the order of channel numbers. Temperatures below 0°C begins with a minus character (ASCII 0x2D). In the example the third channel is showing „---“. That could be cause by the following reasons:

1. No senor connected
2. sensor is defect
3. channel is switched off

05 –Current Temperature of a Channel with Datetime

Command 05 read only

This command only makes sense on devices with integrated RTC, which is currently just Fotemp Trafo. Other devices will reply with `*FF<CR><LF>` .

Parameter 1: Number of the channel to be read, channels are numbered from 1 up to the channel count of your device.

Request example: `?05•6<CR>` asks for the temperature of channel 6 and its time of measurement with a precision of 1 second.

Answer example: `#05•1•456•14110412132456<CR><LF>*00<CR><LF>` The first parameter indicates the temperature state, where 1 means „new temperature“ and 0 means the temperature has already been read.
In the second parameter the temperature is encoded as 1/10 degrees Celsius: in this case -13.5°C. Negative temperature values begins with a minus character (ASCII 0x2D). A channel showing „9999“ indicates a disconnected, switched of or defective sensor.
The last parameter encodes the date and time of measurement. Two signs are one value of the date & time, as shown in the example:

Within -40°C and +85°C the RTC has an accuracy of ±6 ppm (±18 seconds per month). It’s calendar is aware of leap years.

06 –Minimum and Maximum Temperature of one Channel

Command 06 read only

This command reads the minimum and maximum value of temperature since the device has rebooted. With the command 13 the time interval can reset (See chapter 13 – reset extreme values).

Parameter 1: Number of channel to be read, channels are numbered from 1 to the channel count of your device.

Request example: `?06•2<CR>` asks the minimum and maximum temperature of channel 2.

Answer example: `#06•-135 1952<CR><LF>*00<CR><LF>` The first parameter indicates the minimum temperature of -13.5 °C the second parameter a maximum temperature of 195.2 °C. The temperature values are encoded as 1/10 degrees Celsius. Negative temperature values are prefixed with a minus (ASCII 0x2D).

This feature is available since firmware release 2.118

07 –Temperature Error

Command 07 read only

Parameter 1: Number of channel to be read, channels are numbered from 1 to the channel count of your device. This parameter is optional. The command without this parameter will receive the error codes of all channels

Request example: `?07•2<CR>` asks the temperature error code of channel 2.

Answer example: `#07•2•4CR><LF>*00<CR><LF>` The first parameter indicates number of the channel where the error is received from. The second parameter indicates the error code.

This feature is available since firmware release 2.118

13 – Reset Extreme Values

Command 13 write only

Parameter 1: Number of the channel from which to be reset extreme values.

Command example: `:13 02<CR>` Reset minimum and maximum temperature value of channel 2. The values will be set to the actual measured temperature value.

Device Information

0F – Channel Count of the Device

Command 0F read only

Request example: `?0F<CR>` asks for the channel count of the device.

Answer example: `#0F•8<CR><LF>*00<CR><LF>` The only parameter indicates the channel count: 2 in the example. For actual Fotemps values between 1 and 8 are valid replies.

10 – Currently Active Channels

Command 10 read and write.

READ:

Request example: `?10<CR>` asks the currently active channels of the device.

Answer example: `#10•0B<CR><LF>*00<CR><LF>` In the received parameter the active channels are encoded bitwise, the value is presented hexadecimal. In case of the example 0x0B is in binary 0000 1011. The LSB represents channel 1, the MSB channel 8. If the bit is set (value is 1) than it means the channel is enabled. If not the channel is disabled. In the example the channels 1, 2 and 4 are switched on, whereas channels 3, 5, 6, 7 and 8 are switched off.

WRITE:

Parameter 1: In parameter 1 the switched on and off channels are encoded. The bits of the ASCII-encoded hexadecimal byte are the channels, with bit0 representing channel 1 to bit7 for channel 8.

Command example: `:10•1E<CR>` will switch on channels 2, 3, 4 and 5. Channels 1, 6, 7 and 8 are switched off. The binary conversion of 0x1E is 0001 1110.

40 – Model Name

Command: 40 read only

Request example: `?40<CR>` asks for the model name of the device.

Answer example: `#40•43•4F•4D•50•32<CR><LF>*00<CR><LF>` Every parameter encodes a byte value of the model name. Each value is a hexadecimal ASCII-code. In case of the example it reads „COMP2”.

41 – Device Serial Number

Command: 41, read only.

Request example: `?41<CR>` asks for the serial number of the device.

Answer example: `#41•30•30•31•30•30•32•31<CR><LF>*00<CR><LF>` Every parameter encodes a byte value of the serial number where each value is a hexadecimal ASCII-code. In the example the serial number 0010021 is read.

42 – Firmware Version

Command: 42 read only.

Request example: `?42<CR>` asks for the firmware version on the device.

Answer example: `#42•32•2E•31•31•38<CR><LF>*00<CR><LF>` Every parameter encodes a byte value of the firmware version, where each value is a hexadecimal ASCII-code. In the example the firmware release „2.104" is read.

Parameters

53 – Temperature Averaging

Command: 53 Read and write

READ:

Parameter 1: Channel number for which the averaging count for the moving averaging is asked. For compatibility reasons the parameter is optional. Up to firmware version 2.102 averaging can only be set for all channels at once and can therefor only be read at once. Leaving the channel blank in firmware versions 2.103 and newer will give the averaging count of the currently measuring channel.

Request example: `?53•3<CR>` asks the number of temperatures taken for averaging on channel 3

Answer example: `#53•3•4<CR><LF>*00<CR><LF>` The first parameter is the channel number, the second indicates the number of temperature values used for calculating the moving average temperature, 4 is the factory default.

WRITE:

Parameter 1: Channel number of which the averaging count for the moving average will be set. For compatibility reasons the parameter is optional. Up to firmware version 2.102 averaging can only be set for all channels at once. Leaving the channel blank in firmware versions 2.103 and newer will set the averaging for all channels at once. Averaging counts from 2 up to 20 are valid.

Parameter 2: Count of temperature values to calculate the moving average.

Command example: `:53•3•5<CR>` will set the averaging count of channel 3 to 5 temperature values taken for calculating the moving average.

75 – Temperature Offset

Command 75 read and write.

READ:

Parameter 1: Channel number from which to read the temperature offset.

Request example: `?75•4<CR>` asks temperature offset of channel number 4.

Answer example: `#75•001E<CR><LF>*00<CR><LF>` the parameter represents the 2 byte signed temperature offset value normalized in tenth degree Kelvin. The value is displayed as 4-digit hexadecimal number. The example 0x001E decodes to 30_{10}, which is in tenth of Kelvin: +3.0 K. Giving an example for Offsets `#75•FFE6<CR><LF>*00<CR><LF>` below zero would decode to -26, meaning -2.6 K.

WRITE:

Parameter 1: Channel number on which to set the temperature offset.

Parameter 2: is the hexadecimal interpretation of temperature offset value normalized in tenth degree Kelvin. This value will be added to the current temperature offset of the channel.

Command example: `:75•4•000B<CR>` will add another 1.1 K to the temperature offset of channel number 4. To give a more descriptive example, suppose the temperature offset is set to 3.0 K. Sending 0x15 (2.1 K) to channel 4 lets the temperature offset add up to 5.1 K. To achieve an offset of 0 K one needs to add -5.1 K: as signed 2-byte value in hex it is 0xFFCD.

81 – Analog Output Boundaries

Command 81 read and write.

With this command the analog output boundaries of each channel can be read and set. Between the low and the high boundary in degrees Celsius the analog current or voltage (depending on your devices hardware equipment) is linear to the temperature. Temperatures below keep the analog output at 4 mA / 0 V, above the output is kept at 24 mA / 10 Volts. Disconnected, switched off or defective sensors set the output to 24 mA / 10 Volts.

Parameter	Output Current	Output Voltage
Minimum Temperature	4 mA	0 V
Maximum Temperature	24 mA	10 V

This command only makes sense on Fotemp devices equipped with analog outputs, but regardless of the devices configuration the command can be executed.

READ:

Parameter 1: is the channel number for which the boundaries are to be set. Channels are numbered from 1 to the channel count of your device. This parameter is optional, leaving it blank will result in as many answers as channels in your device, with the boundaries of each channel.

Request example: `?81•3<CR>` asks the analog output boundaries of channel number 3.

Answer example: `#81•3•FF9C•012C<CR><LF>*00<CR><LF>` The first parameter is the channel number for which the boundaries are shown. Parameters 2 and 3 are the low and the high analog output boundary. They are hexadecimal encoded in tenth of degree Celsius. In case of the example the low boundary is -10°C and the high is 300 °C. Factory preset is 0°C to 300°C.

WRITE:

Parameter 1: channel number for which the boundaries are set. This Parameter is optional, leaving out a channel number results in setting the analog output boundaries for all channels in the device.

Parameter 2: lower analog output boundary in tenth of degree Celsius.

Parameter 3: higher analog output boundary in tenth of degree Celsius.

Command example: `:81•3•FC18•0064<CR>` will set the lower analog output boundary to -100°C and the higher to +100°C for channel number 3.

82 –Temperature Limits for Relay Output

Command 82 read and write.

With this command the temperature limits for relay output of each channel can be read and set.

There are changings in the behavior since firmware release 2.117

Behavior until Firmware release 2.116:

The relay switched on if the upper temperature limit will be exceed. The relay switched off when temperature underrun the low limit. In this release only one threshold with a hysteresis can for the output relay be configured.

Behavior since Firmware since 2.117:

The switching of the relays depends of relay's configuration described in chapter. In this release it is possible to signal overrun and underrun temperature by the relays separately.

This command only makes sense on Fotemp devices equipped with relay outputs. On devices without relay outputs it results in an negative ACK: `*FF<CR><LF>`.

Parameter 1: channel number of which the relay output boundaries are asked from. This parameter is optional. Left without there will be send as many replies as channels in the device, each with the boundaries for one channel.

READ:

Request example: `?82•1<CR>` asks the relay output boundaries for channel number 1.

Answer example: `#82•1•00C8•00FF<CR><LF>*00<CR><LF>` The first parameter indicates channel number 1 for which the boundaries are shown. Parameters 2 and 3 are the switch-off and the switch-on relay output boundary. They are decimal encoded in tenth of degree Celsius. In the example a rising temperature will switch on the relay at 25.5°C. A falling temperature switches off the relay at 20.0°C.

WRITE:

Parameter 1: channel number for which the boundaries are set.

Parameter 2: switch-off relay output boundary in tenth of degrees Celsius.

Parameter 3: switch-on relay output boundary in tenth of degrees Celsius.

Command example: `:82•1•00C6•00CA<CR>` will set the switch-on temperature to 20.2°C and the switch-off temperature to 19.8°C for the relay of channel number 1. It creates a hysteresis of 0.4°C for the relay switching.

84 Relay Switch Configuration

Command 84 read and write.

With this command the relay's behavior depending of the configured output boundaries can be configured. This feature is available since firmware release 2.118. Each of the used relays has its own configuration. The parameter is a decimal number representing 3 flags:

Bit	Meaning	Details
0	Activate upper limit monitoring	If this bit is set and the Invert Bit is set to zero the relay will switched ON if actual temperature exceeds the upper limit. In the other case the relay will switched off.
1	Activate lower limit monitoring	If this bit is set and the Invert Bit is set to zero the relay will switched ON if actual temperature underruns the lower limit. In the other case the relay will switched off.
2	Invert output signal	If this bit is set the relay output will be inverted.

For the relays a static switching hysteresis of 1 K is set.

For details see manual Relays_FW2_2.118_En_v01.pdf.

This command only makes sense on Fotemp devices equipped with relay outputs. On devices without relay outputs it results in an negative ACK: `*FF<CR><LF>`.

Parameter 1: channel number of which the relay's properties are asked from. This parameter is optional. Left without there will be send as many replies as channels in the device, each with the relay's properties for one channel.

READ:

Request example: `?84 1<CR><LF>` asks the relay's properties for channel number 1.

Answer example: `#82•1•03<CR><LF>*00<CR><LF>` The first parameter indicates channel number 1 for which the relay configuration are shown. Parameter 2 is a hexadecimal number represents the configuration flags of the relay configuration. The number 3 means that the relay switched on if the measured temperature higher than the upper limit and lower than the lower limit.

WRITE:

Parameter 1: channel number for which the boundaries are set.

Parameter 2: switch-off relay output boundary in tenth of degree Celsius.

Parameter 3: switch-on relay output boundary in tenth of degree Celsius.

Command example: `:82•1•00C6•00CA<CR>` will set the switch-on temperature to 20.2°C and the switch-off temperature to 19.8°C for the relay of channel number 1. It creates a hysteresis of 0.4°C for the relay switching.

Real Time Clock

The integrated real time clock is an optional feature of Fotemp Trafo devices.

90 – real time clock date and time

Command 90 read and write.

Where available this command will read or set the current date and time of the integrated real time clock. Fotemps without real time clock reply with `*FF<CR><LF>`.

READ:

Request example: `?90<CR>` asks the current date and time of the integrate real time clock.

Answer example: `#90•14•11•05•13•12•25•37<CR><LF>*00<CR><LF>` Each parameter encodes a value of the current date and time: Thursday 13th of November at twelve o'clock, twenty five minutes and 37 seconds.

14	11	05	12	13	24	56
Yeah	Month	Day of Week	Day of Month	Hour	Minute	Second
From 00=2000 to 83=2083	01=January 12=December	1=Sunday 2=Monday … 7=Saturday	1…30/31 29th February exists in a leap year	24 hour format 00…23	00…59	00…59

Within -40°C and +85°C the RTC has an accuracy of ±6 ppm (±18 seconds per month). It's calendar is aware of leap years.

WRITE:

Parameters: are in the following order: year month, day of week, day of month, hour, minute and seconds. Please refer o the table above.

Command example: `:90•15•01•07•29•15•45•11<CR>`will set the real time clock to Saturday, 29th of January at 15:45 o'clock an 11 seconds.

Data Logging on SD Card

Some devices are supplied with a SD card which is used for data logging. The SD card has no file system. The data will be written in raw format. The size of one section have to set to 512 byte. The first section contains the status data:

Addr (Hex)	Size (Bytes)	Data Type	Description
0	4	ASCII	ID String („OPTO“)

4	4	UINT32	Address of first data section
8	14	ASCII	time stamp of first data section
16	4	UINT32	Address of last data section
1A	14	ASCII	time stamp of last data section
28	4	UINT32	count of data sets
2C	1	UINT32	channel count
2D	467		Unused

All other sections are used as data sections. A data section has the capacity for 8 data records (for 8 channels).A data record contains the following data:

Data Element	Description
Channel Nummber	1 ... 8
Valid Flag	0: Unvalid 1: Valid
Temperature Value	ASCII DECIMAL
Time Stamp	ASCII DECIMAL

Independent of the channel count for each data save process a new section will written. The write cycle interval can be configured by setting command 93 (see chapter 93 – Real Time Clock Timer Interval) The time interval is controlled by RTC. This means the RTC equipment is required for data logging on SD card.

B0 – Read dataset from SD Card

Command B0 read only

This command reads a temperature value with timestamp from SD card from the data section on which the read section pointer is placed of the channel on which the channel pointer is placed. After this the channel pointer will be incremented by one ore if the last channel is reached the channel pointer will be set to channel 1 and the section pointer will be set to the next data section.

Request example: `?B0<CR>` asks for the next temperature data saved on SD card.

Answer example: `#B0•2•1 1075 17040502145751<CR><LF>*00<CR><LF>` The received parameters corresponds to table data element (see above in section Data Logging on SD Card)

The parameters are all encoded as decimal ASCII digits.

B1 – Get Dataset Count SD Card

Command B1 read only

With this command the count of saved datasets can be read. A dataset consists of the temperature and timestamp of measurement of all channels of the device. Each data set is placed in one of the sections of SD card.

Reading the data happens channel-wise, so the count of datasets multiplied with the channel count of the device is the number of B0-reads to read all saved datasets.

Request example: `?B1<CR>` asks for the count of datasets saved on SD card.

Answer example: `#B1•3<CR><LF>*00<CR><LF>` states 3 datasets on internal flash. If it is an 4-channel Fotemp 12 temperatures with their time of measurement and channel number are saved.

B2 – Delete Datasets

Command B2 write only

This command deletes a given count of datasets on SD card of the Fotemp. Like only the eldest data on SD card can be read, only the eldest datasets can be deleted too. A dataset consists of the temperature and time of measurement of all channels of the device.

Command example: `:B2•2<CR>` will delete the two eldest datasets (2*channel count of temperatures) on the internal flash memory.

BE – Reset Read Counter

Command B3 write only

This command resets the read counter used by command B0. B0 only allows to read 254 datasets, then some data has to be deleted to read the next datasets. In case of an error the read counter can be reset to read the datasets again. Already deleted datasets can not be regained!

Command example: `:B3•<CR>` will reset the read counter to reread the datasets.

BF – Erase all Data Sets

Command BF write only

With this command the SD card can be reset and cleared. All data is deleted.

Command example: `:BF•<CR>` will delete all datasets on the internal flash memory. Please note the trailing <space>, it is necessary!

B3 –Set Timer Interval for Data Logging

Command 93 read and write.

With this commando the write cycle of logging data can be set.

READ:

Parameter 1: sets the write cycle for data logging in seconds.

Parameter 2: sets a multiplier for a 2nd dependent timer function (special use on customer request), set to 1, when not in use.

Request example: `?B3<CR>` asks for the write cycle for data logging

Answer example: `#B3•60•3<CR><LF>*00<CR><LF>` states a write cycle of 60 seconds and a multiplier of 3 for a 2nd function.

WRITE:

Parameter 1: sets the write cycle for data logging in seconds.

Parameter 2: multiplier of the cycle time can used by a 2nd function.

Command example: `:B3•140•2<CR>` will set the write cycle to 140 seconds and the multiplier is set to 2, meaning the 2nd function is issued every 280 seconds.

B4 – Read Data State

Command B4 read only

This commands is for reading information about the written log data on SD card.

Request example: `?B4<CR>` asks for the state of logging data.

Answer example: `#B4•166171•263982•97811•4•3<CR><LF>*00<CR><LF>`

The following values are received:

Parameter	Description	Value
Start Sector	The section on the SD card with the first written data record	166171

End Sector	The section on the SD card with the last written data record	263982
Data Count	The count of written data sections	97811
Read Sector Offset	Sector Offset of the section where the data record is read by the command B0 Read Section = Start Section + RS Offset -1	4
Read Channel Offset	Channel Offset where the data record is read by the command B0	3

B5 – Read Data Set from SD Card

Command B5 read only

This command reads a specified data record from SD card.

Parameter 1: Section number where data to read from

Parameter 2: Channel Number of data record to read.

Request example: `?B5•166100•3<CR>` Read data set from section 166100 of channel 3.

Answer example: `#B5•3•1•428•17030614031347CR><LF>*00<CR><LF>`

The following values are received:

Parameter	Description	Value
Channel Number	The channel number from which data read	3
Valid Flag	A value of 1 means temperature value is valid. A value of 0 menas the temperature value is invalid.	1
Temperature Value	The temperature value is 42.8 °C	428
Time Stamp	The timestamp represents the time event when the temperature value has been measured.	17030614031347

This timestamp „17030614031347“ can be interpreted as follow 14.03.2017 03:13:47

Format of Time Stamp

Size	Comment
2	Year
2	Month
2	Day of Week
2	Day of Month
2	Hour
2	Minute
2	Second

BA - Read SD card properties

Command BA read only

This command reads some properties from the detected SD card.

Request example: `?BA<CR>` asks for SD card properties.

Answer example: `#BA•1•2•512•30253056<CR><LF>*00<CR><LF>`

The following values are received:

Parameter	Description	Value
Flags	Status Flag	1
SD Version	SD card Specification (is 1 or 2)	2
Block Length	Length of one data block	512
Block Count	Count of data blocks	30253056

The memory capacity is the result of multiplying block length with block count:
512 x 30123056 = 15.423.004.672 = 16 GB

The Meaning of the flag parameter is described in the following table:

Bit	Meaning if it set
0	Initialization OK
1	Write Error occurred
2	Read Error occurred