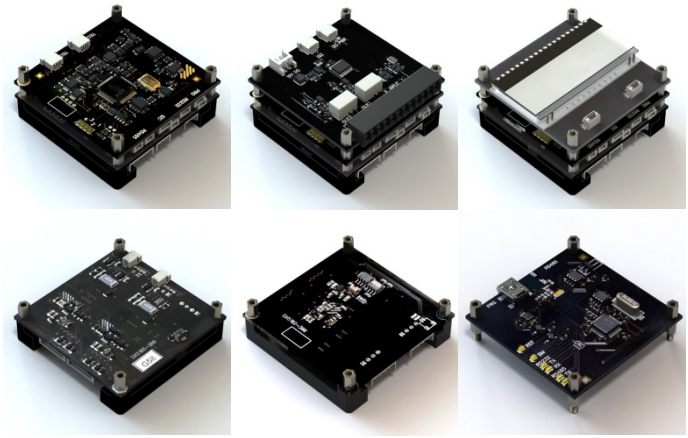
PL Engineering Ltd.

TEC Controller DX5100
Technical Manual

SYSTEM OF COMMANDS



Moscow, 2017

Version 3.13

Edition July 2017

CONTENTS

1. COMMUNICATION PROTOCOL

1.1. Interfaces RS-485 and RS-232

In the controller two physical interfaces RS-485 and RS-232 are realized.

Both interfaces are serial and use the following adjustments:

Possible rates of exchange (baud)	9600 19200(default) 38400 57600 115200
databits	8
parity	no
Stop bits	1
flow control	no

The program exchange protocol is «WAKE».

One of the interfaces is used for management of control of the TEC controller (sending of commands and reception of responses). I.e. the protocol WAKE can function by either of the interfaces. For definiteness, we shall name the interface on which the WAKE - **as the command** interface.

The protocol WAKE suggests a half-duplex exchange. It does not give enable the device to output information on its own initiative via the command interface. For this purpose (outputting of the telemetry on the initiative of the device) the other interface (where WAKE does not work) is used.

Thus, each of the interfaces realized in the Controller can be command.

The non-command interface is used for telemetry output. Telemetry is outputted in a symbolic form. Hence for display and storage of the

information transmitted by the device it is possible to use any terminal program properly adjusted.

The interface RS-485 supports the network topology that allows connecting up to 32 devices to one bus RS-485. The physical environment of signaling in the bus RS-485 is a twisted pair. The physical realization of the protocol RS-485 suggests a half-duplex mode of bus exchange.

The device is always a slave. It sends a response to every frame received.

The package WAKE received by slaves is further named - «command».

The response frame is named «informational».

1.2. Protocol WAKE

The protocol WAKE is a logic level of the control interface with the help of the asynchronous serial channel. The physical level of the interface is not defined by the protocol, e.g. either RS-232 or RS-485 can be used. The report allows exchanging data frames, length up to 255 bytes, with addressed devices, which can be up to 127. In the Controller the total length of a frame is limited by 64 bytes.

The base of the protocol WAKE is the protocol SLIP (UNIX™ Serial Link Interface Protocol).

Data transmission is carried out in a binary form, i.e. all possible byte values are used (00h…FFh).

For transfer of service information two codes are reserved: FEND = C0h (Frame End) and FESC = DBh (Frame Escape).

The control code FEND serves for a designation of the sending start, and the code FESC serves for the transmission of ESC-sequences.

If in a data stream there bytes whose values coincide with control codes, these bytes are replaced by ESC-sequences. This mechanism is named byte stuffing.

The code FEND is replace by the sequence <FESC>, <TFEND>, and the code FESC – by the sequence <FESC>, <TFESC>, where TFEND = DCh (Transposed FEND), TFESC = DDh (Transposed

FESC). The codes TFEND and TFESC are control only in the ESC-sequences, as at data transmission they need not be replaced.

Table 1. Protocol WAKE control codes

Designation	Explanation	HEX-value
FEND	Frame End	C0h
FESC	Frame Escape	DBh
TFEND	Transposed Frame End	DCh
TFESC	Transposed Frame Escape	DDh

Table 2. Substitution of databytes by ESC-sequences

Databyte	Sequence transmitted
C0h	DBh, DCh
DBh	DBh, DDh

The structure of the packet WAKE is the following: it always begins with the control code FEND (C0h). Then an optional address byte follows after which there is a command byte. The byte of data amount and actually databytes go farther. The optional byte of the control sum terminates the packet. CRC-8.

Table 3. Packet WAKE structure

FEND	ADDR	CMD	N	Data1	…	DataN	CRC

FEND: The control code FEND (C0h) is an attribute of the beginning of a packet. Due to the stuffing, this code does not exist anywhere else in a stream, which unequivocally allows determining the beginning of a packet.

<ADDR>: The address bytes are used for addressing separate devices. To be able to determine unequivocally whether the second and the third bytes in a packet are an address or a command, there

are some restrictions. For addressing byte 7 is used, and the MSB transmitted together with the address should be always set =1.

	D7	D6	D5	D4	D3	D2	D1	D0
ADDR =	1	A6	A5	A4	A3	A2	A1	A0

Sometimes there is a necessity to send a command or data simultaneously to all the devices. For this purpose the broadcast is carried out by transfer of the zero address (taking into account MSB=1 the transmitted byte is equal 80h).

The transfer of the zero address in a packet is completely similar to transfer of a package without an address. Therefore for the protocol realization it is possible to exclude the zero address from a packet automatically.

Taking into account the word length and one address reserved for a broadcast, the maximum number of addressed one-type devices equals 127.

If it is a necessary to send the value of address 40h or 5Bh (transmitted bytes in this case will be equal C0h or DBh), the stuffing is done, i.e. the transmission of the ESC-sequence (see table 2).

The devices with such addresses demand one byte longer packet. It can be considerable when short packets are used. In such cases it is necessary to avoid assignment to devices of the named addresses.

CMD: The command byte should always have a zero MSB.

	D7	D6	D5	D4	D3	D2	D1	D0
CMD =	0	C6	C5	C4	C3	C2	C1	C0

The command code occupies 7 bits, which allows transmitting up to 128 various commands.

The codes of commands are chosen in an arbitrary way depending on requirements.

As the command code has always a zero MSB, this code never coincides with control codes. When sending a command the stuffing it is never done.

N: The byte of data amount has a value equal to the number of transmitted databytes.

The code of amount of data occupies 8 bits, so one packet can contain up to 255 databytes. The value N does not take into account packet service bytes FEND, ADDR, CMD, N and CRC. As a result of the stuffing the packet actual length can increase. The value N **does not take into account** this fact and expresses the number of useful databytes (i.e. the value N is always such as though there were no stuffing).

If the transmitted command has no parameters, N = 00h and databytes are omitted.

If it is necessary to transfer the value N equal C0h or DBh the stuffing is done, i.e. the transfer of the ESC-sequence (see Table 2). However at such big values N the packet length is so great, that its lengthening by a byte is practically imperceptible.

Data1…DataN: Databytes whose number is determined by the value N. At N = 00h there are no databytes. The databytes can have any value except FEND (C0h) and FESC (DBh). If it is necessary to transfer one of these values the stuffing is done, i.e. the transfer of the ESC-sequence (see Table 2), consisting of the control code FESC and code TFEND (TFESC).

The command parameters are sent in the data field of the command frame of the Controller.

The information frame data field contains a response answer generated by the device.

In the Controller the databytes represent the ASCII symbols. Thus all the field of the data can be considered as a symbolic line.

CRC: Byte of the control sum CRC-8. It can be absent in some options of the protocol. The control sum CRC-8 is calculated before the stuffing for the entire packet, beginning with the byte FEND and finishing with the last databyte. If a packet transmits an address, when calculating the control sum, its true value is used, i.e. MSB=1 is not taken into account.

For the calculation of the control sum the polynomial is used. CRC = X8 + X5 + X4 + 1.

Before the calculation the number DEh initializes the value CRC**.** By transfer of the value of the control sum byte, C0h and DBh are replaced with ESC-sequences (see Table 2).

1.3. Structure of Buffer of Data Transfer/Reception

The structure of the data field of the frame (Data1 … DataN) of the realized protocol for all the commands has the same features.

Each command (command frame) causes a response (information frame).

In each **command frame** there are two (first) bytes that stand for the following.

The first byte is the identifier of the device type and is an expansion of the addressing field. The device on the bus perceives the command frame as directed to it, if the address byte and identifier coincide with those of the device.

By broadcasting transfer the identifier, as well as the address, is accepted equal to zero. The described device has an identifier of the type 2.

The second byte is reserved for further applications and is not used in the Controller.

In each **information frame** there are two (last) bytes that stand for the device status. The status bytes should be interpreted as a set of bits, each of which signals about an event. The correspondence of status bits to the events will be given below.

As an example of parameters, the data field can contain the following types of data.

Data Types	Bits	Bytes	Value Range
unsigned char	8	1	0 — 255
unsigned int	16	2	0 — 65535

Data Types	Bits	Bytes	Value Range
unsigned long	32	4	0 — 4294967295
float	32	4	±1.175494E-38 — ±3.402823E+38

In this system of commands no sign integers are used, though they may be used in a general case.

Besides the types given in the Table the data field can contain a symbol line finished by zero (0x00).

Two kinds of data fields are supported: symbolic and binary (corresponding modes of WAKE).

1.3.1. *WAKE Binary Mode*

The integer types of data are given by the sequence of bytes of the number shown in the Table below. The upper bytes are given first.

The number 7459 is stored as a hexadecimal value of 0x1D23. In memory, this value appears as follows:

		Address+0	Address+1	Address+2	Address+3
Contents	int	0x1D	0x23	-	-
Contents	long	0x00	0x00	0x1D	0x23

A floating-point number is expressed as the product of two parts: the mantissa and a power of two. For example:

$$\pm \text{mantissa} \times 2^{\text{exponent}}$$

The mantissa represents the actual binary digits of the floating-point number.

The power of two is represented by the exponent. The stored form of the exponent is an 8-bit value from 0 to 255. The actual value of the exponent is calculated by subtracting 127 from the stored value (0 to 255) giving a range of –127 to +128.

The mantissa is a 24-bit value (representing about seven decimal digits) whose most significant bit (MSB) is always 1 and is, therefore, not stored. There is also a sign bit that indicates whether the floating-point number is positive or negative.

Floating-point numbers are stored on byte boundaries in the following format:

	Address+0	Address+1	Address+2	Address+3
Contents	SEEE EEEE	EMMM MMMM	MMMM MMMM	MMMM MMMM

where

S	represents the sign bit where 1 is negative and 0 is positive.
E	is the exponent with an offset of 127.
M	is the 24-bit mantissa (stored in 23 bits).

Zero is a special value denoted with an exponent field of 0 and a mantissa of 0.

Using the above format, the floating-point number -12.5 is stored as a hexadecimal value of 0xC1480000. In memory, this value appears as follows:

	Address+0	Address+1	Address+2	Address+3
Contents	0xC1	0x48	0x00	0x00

1.3.2. *WAKE Symbol mode*

In the symbol mode the field **Data1…DataN** is considered a list of parameters given as symbols and separated by the space from each other. The first two bytes of the command frame and the last two bytes of the information frame are not separated.

The integer parameters can be both in the decimal and hexadecimal format. In the hexadecimal form each nibble (a tetrad) has the following form:

'0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F'.

A couple of adjacent symbols corresponds to a byte written in a hexadecimal form.

Thus, from the above-said it follows that for the successful connection with the Device a proper command interface, a mode (symbol or binary) and an exchange rate should be chosen.

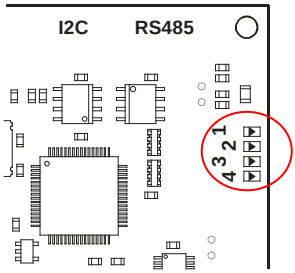
1.3.3. ***Connection***

The device can be adjusted to different rates of communication. Each of the interfaces available can be command. The WAKE data format may be either symbol or binary.

During 1 sec after the Device switched on, the LEDs indicate the rate of exchange via the interfaces and the mode WAKE as shown below.

There are four LEDs on the Digital Control Board (LED 1…LED 4).

Just after switching ON the Device, during 1 sec LEDs show the exchange rate per interface and the mode WAKE corresponding to the Table below.



WAKE	BIN					SYM				
Rate (baud)	9600	1920	3840	5760	1152	9600	1920	3840	5760	1152
LED 1	■		■		■	■		■		■
LED 2		■	■				■	■		

□ Light is OFF

Also after the switching on, the Device outputs an information line of the format given in the Table below. The line is outputted into the interface RS232 as well as into the interface RS485 (unless it is command).

The output is carried out with the rate to which the Device is tuned.

	TEC controller	NetAdr=	hh	DevId=0200 WAKE-RS	485	-SY
			1		2	3
1	Network device address-*01(default)…7F*					
2	Command interface *RS232 (default)* *RS485*					
3	Data format WAKE *BIN (default)* *SYM*					

This information can be used in case there are difficulties while connecting with the Device.

If connecting to the interface that at present is not the command one, it is impossible to operate the device at once. To give the connected interface the status command for the current session it is necessary to send the sequence of symbols «$&%» to an appropriate COM-port.

The reception of this sequence via not the command interface switches the device into the mode of commands reception via this interface. The specified sequence of symbols can be given in any terminal program. Also it is possible to output it by the command 02h - «Transfer an echo».

02	**__@$&%__**

	1	2
1	Command code hex	
2	Symbol line switch the reception of commands to the interface on which this command is accepted. The response to this command is not sent.	

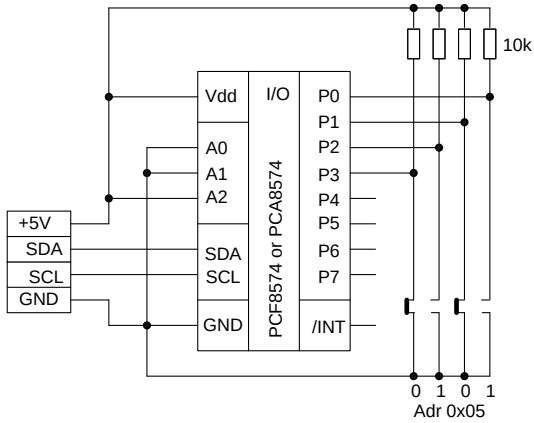
The protocol WAKE will further work via the interface by which the command has arrived. However, after turning off and repeated switching on of the device, the command interface will be again the one established by the command 4Bh.

In version 3 of the firmware it is possible to set the controller address not programming by a command but from the outside device.

The presence of the device number 05 (for the command 0x44) on the bus I2c allows setting the address of the controller on the bus RS-485. This device should be the PCA8574 or PCF8574 - Remote 8-bit I/O expander for I2C-bus. The address inputs of this device should be A0=0, A1=0, A2=1. In this case four less significant bits of data input are interpreted as an address of the controller in the network RS-485. The address should be generated by the appropriate signals on pins of the microcircuit. The availability of the microcircuit is checked just after switching on of the controller.

P3	bit D3 of controller address in the network RS-485
P2	bit D2 of controller address in the network RS-485
P1	bit D1 of controller address in the network RS-485
P0	bit D0 of controller address in the network RS-485

If the address is set this way, it has a priority over the address set by command 07.



More significant bits of this microcircuit can be used, for example, for LED control. Then it is always necessary to set less significant bits in a high level.

2. SYSTEM OF COMMANDS

1.4. Commands Performance

A command is performed under the conditions:

- Coincidence of the device address transmitted in the field ADDR with the address of the device receiving the packet or ADDR=0x00 (broadcasting transmission). If the interface RS232 is used as the command one, the value of the field ADDR may not coincide with the Device net address.
- Coincidence of the device identifier transmitted in the data block, with the identifier of the device receiving the packet;
- The command format is not broken and the values of parameters do not leave beyond allowable limits.

For the description of commands the command code and its parameters are specified. The following designations are used:

- number of ADC or number of TEC channel (of PID regulator) or number of DAC.

ADC numbers and appropriate measured parameters are in the table:

0	Supply voltage
1	TEC1 voltage
2	TEC2 voltage
3	TEC1 current
4	TEC2 current
5	TEC1 temperature
6	TEC2 temperature

TEC channel number value and DAC number can be «0» or «1».

The following abbreviations are used for the description of parameters depending on their types:

uc	unsigned char	integer decimal number
ud	unsigned int	integer decimal number
ul	unsigned long	integer decimal number
e	float	floating point number
f	float	floating point number
s	string	line of symbols
h	hex	hexadecimal number (0…9, A…F)

For the symbol mode these abbreviations determine a type and form of parameters. For the binary mode they only do a type of parameters. The figure after "e" and "f" indicates a number of digits after the point (comma).

1.5. Device Status

The LSB status byte (the last byte) value:

0x01	error EEPROM
0x02	unknown command
0x04	no ready data for telemetry (response)
0x08	TEC voltage at Z-metering does not drop for too long
0x10	error in parameters or command format
0x20	reception RS-232 buffer overfilling
0x40	reception RS-485 buffer overfilling
0x80	voltage supply error

Attention! *The error of supply voltage (0x08) is accompanied by switching-off of voltage converters. The error remains even after voltage returning to allowable limits 12±10%*

The MSB status byte value:

0x01	TEC1 temperature is beyond the limitations
0x02	TEC2 temperature is beyond the limitations
0x04	TEC1 temperature is within the setting
0x08	TEC2 temperature is within the setting
0x10	Command performance is interrupted

1.6. Table of Commands

Mnemomics	Code	Parameters	Description	Inf Frame
		System commands		
CMD_ECHO	0x02	s	// send echo	s
CMD_INFO	0x03		// send device identifier	hhhh
CMD_GetVer	0x04		// send device firmware version	s
CMD_GetInfo	0x05		// send device information	s
CMD_SetInfo	0x06	s	// write device information	
CMD_SetAdr	0x07	uc	// set network address	uc
CMD_StTel	0x40	uc hh hh	// set telemetry status	hh hh
CMD_get_PRM	0x41		// send parameters stored in FLASH (backup)	
CMD_set_PRM	0x42		// receive parameters stored in FLASH (recovery from backup)	
CMD_I2C	0x44	uc hh hhhh hh	// operations with bus I2C	hh hh hh hh hh hh hh hh
CMD_Prog_T	0x45	uc uc uc f ud hh uc	// Managing of programs of change in time	uc uc uc f2 ud hh uc
CMD_get_Tel	0x46		// send telemetry line	
CMD_Krt_OK	0x49	№ uc uc f	// criterion of signal of settling	

Mnemomics	Code	Parameters	Description	Inf Frame
CMD_St_HW	0x4a		// status of devices in system	hh hh hh
CMD_Infs_Wk	0x4b	uc uc uc	// set status to interface	
CMD_Dig_Out	0x4d	uc	// enable of digital output	uc
CMD_Dig_In	0x4e	uc hh	// control of digital intput	uc hh
		Commands of Work with ADC		
CMD_ClbrADC	0x10	№ uc	// hardware calibration of ADC	hh hh
CMD_ClbrK_ADC	0x11	№ f	// calibration of ADC (calculation of calibrating ADC coefficients)	
CMD_Wr_K_ADC	0x12	№ e	// writing calibrating ADC coefficients	
CMD_Kfiltr	0x13	№ uc	// writing ADC filter coefficient	
CMD_AskKADC	0x14	№	// sending ADC conversion coefficient and filter coefficient	hh e6 uc hh
CMD_AskOfst	0x15	№	// sending ADC register of offset	hh hh hh hh
CMD_StartADC	0x16	№	// starting measurement in ADC channel	hh hhhhhhh h e6 e6
CMD_Only_1	0x17	№ 1/0	// measurements of one ADC channel (fast measurement)	hh
CMD_Sever	0x18	hh	// mask on ADC channels	hh
CMD_PGA	0x19	№ uc	// PGA of ADC thermistor channel	
CMD_Polinom	0x1a	№ uc	// writing thermistor polynomial order	
	0x1a	№ uc f	// writing thermistor polynomial coefficients	
CMD_ask_Pol	0x1b	№ uc	// sending thermistor polynomial coefficient	hh uc uc e6
			// Save the Current Settings of Thermistor Input	
CMD_saveTerm	0x1c	uc	// Save the Current Settings of Thermistor Input	
CMD_loadTerm	0x1d	uc uc	// Restore Thermistor Input Settings	hh
CMD_get_T	0x1e		// Save Table of Settings (backup)	

Mnemomics	Code	Parameters	Description	Inf Frame
BL				
CMD_set_TBL	0x1f		// Restore Table of Settings (Backup)	
		Commands of work with DAC		
CMD_set_DAC	0x21	№ f	// setting DACs in Volts	hh ud
CMD_seth_DAC	0x22	№ ud	// setting DACs directly (no control of limitations)	hh ud
CMD_Wr_K_DAC	0x23	№ f f	// writing DAC calibrating coefficients	
CMD_AskKDAC	0x24	№	// sending convertion coefficient and DAC max values	hh e6 e6 f2
CMD_DAC_max	0x25	№ f	// writing max voltage	
CMD_U_Treg	0x26	№ f	// voltage of T-regulation	hh f2
		Commands of work with PID controller		
CMD_Pol_TEC	0x30	№ uc	// setting of TEC polarity	
CMD_set_PID	0x31	№ f f f	// writing parameters of PID controller	
CMD_ask_PID	0x32	№	// sending parameters of PID controller	hh f6 f6 f6
CMD_setCurrT	0x33	№ uc	// current thermistor 0-10uA 1-93uA	hh uc
CMD_askT_PID	0x34	№ [f]	// sending/set setpoints of PID controller	hh f2 f2 uc uc
CMD_strt_PID	0x35	№ uc f	// starting controll	
CMD_tun_PID	0x36	№ hh	// output parameters of PID controller (bits)	hh hh
CMD_Zmetr	0x37	№ uc uc	// starting Z-meter	s
CMD_Zprmtr	0x38		// storage of Z-metering parameters (as reference)	
CMD_Z_I	0x39		// sending Z-meter current	e6
CMD_Z_I	0x39	f	// storage of Z-meter current	
CMD_Boot	0x3b	№ uc f ud	// start of regulation after restarting	hh hh f2 ud

Mnemomics	Code	Parameters	Description	Inf Frame
CMD_set_LimT	0x3c	№ f f uc	// writing limiting temperatures	
CMD_get_LimT	0x3d	№	// sending limiting temperatures	hh f2 f2 uc
CMD_ResZmtr	0x3e		// sending Z-metering results	hh f2 e2 f2
CMD_TecZmtr	0x3f	№	// sending Z-metering parameters	hh f2 e2 f2
CMD_PID_tun	0x51	№	// auto tuning PID	s
CMD_REST	0x53		// reset controller	
CMD_EKR	0x54	[#screen]	// control of indication board	

1.7. Commands Descriptions

1.7.1. *Send Echo*

In response a frame with data field the same as the field (Data1 … DataN) of the received frame is sent.

	02	s
	1	2
1	Command code hex	
2	Symbolic line	

In response a frame with data field the same as the field (Data1 … DataN) of the received frame is sent.

By sending this command, one can interrupt long procedures (for example, those of Z-meter).

1.7.2. *Send Device Identifier*

	03
	1

1	Command code hex

In response network address and device identifiers are sent.

Example:

0102
TEC controller (02) with address 01 on bus RS-485 (01)

1.7.3. *Send Device Firmware Version*

04
1

1	Command code hex

In response the device name and firmware version are sent.

Example:

DX5100.022
Device DX5100 with firmware version - 022

1.7.4. *Send Information of Device*

05
1

1	Command code hex

In response serial number and date of issue of device are sent.

Example:

	#C09-P16-P17-I06 10.05.2009
	Serial number of boards and date of issue of device

1.7.5. *Write Information into Device*

	06	**s**
	1	2
1	Command code hex	
2	Serial number of boards and date of issue of device (max 32 symbols)	

1.7.6. *Set Network Device Address*

	07	**uc**
	1	2
1	Command code hex	
2	Network device address (1…127)	

Command 07 can be transferred at broadcasting addressing.

1.7.7. *Set Telemetry Status*

	40	**uc**	**hh**	**hh**
	1	2	3	4
1	Command code hex			
2	Telemetry output period			
3	Telemetry status MSB			
4	Telemetry status LSB			

Command 40 can be transferred at broadcasting addressing. When a broadcasting command is received, the telemetry output stops.

The parameter period of output is set in 0.01 s. To set the period equal 1 s one should set d equal 100 (uc=100). The maximal value uc=255. The period of telemetry output also depends on required parameters being ready. Thus, if switching on the output of all parameters (LSB is 7F), the minimal period of telemetry will be about 0.5 s. This time corresponds to the time of conversion of all the ADC channels. To increase the telemetry rate it is necessary to include only the needed parameters in the telemetry block and to use the commands reducing quantity of processable ADC channels (described further).

The functions of all status LSB and MSB are given in the table below.

Each bit allows/forbids (1/0) either a function, or inclusion of a parameter in the telemetry block.

LSB			
BIT	Mask	Function or parameter in telemetry block	Units
0	01	Supply voltage measured (f2)	V
1	02	TEC1 voltage measured (f2)	V
2	04	TEC2 voltage measured (f2)	V
3	08	TEC1 current measured (f1)	A
4	10	TEC2 current measured (f1)	A
5	20	TEC1 temperature measured (f3)	K
6	40	TEC2 temperature measured (f3)	K
7	80	Reserved	
MSB			
BIT	Mask	Function or parameter in telemetry block	Units
0	01	TEC1 channel status (hh) (for bit values - see command 4A)	
1	02	TEC2 channel status (hh) (for bit values - see command 4A)	
2	04	Device status (hhhh – two bytes at the end of	

		the information frame) bits values Device status see section 2.2 The bytes of the Device status are only added to the telemetry of the non-command interface, as in the telemetry by the command 46 the Device status is present at the end of the information frame	
3	08	Enable anything besides the telemetry be outputted into a non-command interface	
4	10	TEC1 temperature setpoint (f2)	K
5	20	TEC2 temperature setpoint (f2)	K
6	40	Permission of telemetry output into the command interface. It should be noted that the given mode is not supposed by the protocol WAKE, i.e. it is non-standard since it results in periodic output of the frames WAKE on the bus by the slave-device without reception of a command frame.!!! See the command 46h.	
7	80	Permission of telemetry output into a non-command interface	

The first parameter in the telemetry is the time in the units 0.01 sec (ul). The time is continuously calculated and reset after sending the command 40.

The telemetry line is finished by the symbol ";".

If the command is performed successfully, the values of two status bytes are outputted (hh hh).

Attention! If the telemetry line length exceeds 62 symbols, that may tell upon the time-dependent functions of the system. For example, the PID period “beating” can be observed.

At the telemetry status 0xB7, 0x66 the line will contain values of two channels voltages, two channels temperatures, PID two channels statuses, device status, setting values of PID two channels. The line length will not exceed 62 symbols.

To estimate the telemetry line length the following Table can be used (taking into account that the parameters are separated by spaces):

Parameter	Symbols number	Example
Time	7	1364400
Supply voltage measured (f2)	5	12.02
TEC1 voltage measured (f2)	5	-4.12
TEC2 voltage measured (f2)	5	-1.23
TEC1 current measured (f2)	4	0.53
TEC2 current measured (f2)	4	2.54
TEC1 temperature measured (f2)	6	299.53
TEC2 temperature measured (f2)	6	310.12
TEC1 channel status (hh)	2	10
TEC2 channel status (hh)	2	00
Device status (hhhh – two bytes corresponding to those at the end of the information frame)	4	0000
TEC1 temperature setpoint (f2)	6	300.00
TEC2 temperature setpoint (f2)	6	310.00

1.7.8. ***Send Parameters Stored in FLASH (Backup)***

41	
1	
1	Command code hex

After sending this command the device outputs contents of structure with parameters kept in non-volatile memory to a command or non-command interface. These parameters are such as calibrating coefficients, network addresses, etc.

Each byte is transferred as two hexadecimal figures. The first 4 figures (2 bytes) are the size of structure outputted, in bytes. This command is used for backup of stored parameters. Output of

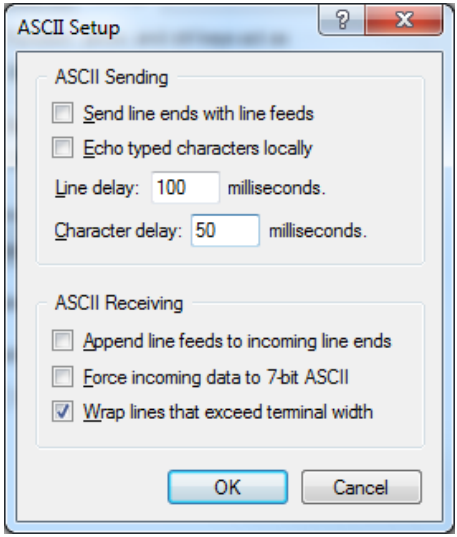
parameters will begin into both the interfaces after reception of any symbol by any interface.

1.7.9. *Write Parameters Stored in FLASH*

42	
1	
1	Command code hex

After sending this command, the device expects for data to come and interprets them as given by the command 41. The data received are stored in the structure.

For the correct execution of this command in a terminal program to set the delay for characters 50mS.



For the storage of the entered data in the non-volatile memory it is necessary to send a command of setting any parameter. For example, 0x07 - "Set network address".

1.7.10. *I2C Bus Operations*

	44	**uc**	**hh**	**hhhh**	**hh**
	1	2	3	4	5
1	Command code hex				
2	0-writing 1-reading				
3	№ of device on bus I2C 0- EEPROM 24c256 1-PCF8574 with data bus LCD 2-PCF8574 with control bus LCD 3-RTC DS1307 4- PCF8574 control digital output relay and digital input 5- PCF8574 LEDs and a digital input RS485 address				
4	Address in a device on the bus I2C				
5	If reading, bytes number (no more than 8) If writing, byte for writing				

By this command it is possible to read or write data in devices connected to the bus I2C.

The bytes read (hh hh…) are transferred into the information frame.

1.7.11. *Management of Program in Time*

The Controller has a function “Program” - changing of operation mode in time. This function is carried out by the programs stored in the non-volatile memory.

- Programs number up to 16 (0-15)
- Number of lines in program up to 50 (0-49)

  Each line in program contains the following:

- Temperature (setpoint) (K) for modes 2 and 3 or Voltage (V) for mode 4 (see below)

- Time after which one has to leave this line of program (number of seconds - max 65535)
- Most significant nibble – mode of current line (interpretation of the field of the first parameter of the program line as temperature or voltage). Possible values:
  - 0 Interdiction of regulation
  - 2 T-regulation
  - 3 Temperature maintenance (PID)
  - 4 Constant voltage
  - 5 Setting the voltage of T-regulatory
  - 6 Adjust the settings
  - 7 Exit from the program without shutting down the last mode
- Least significant nibble - number of program to go (0-15) on the expiration of the period Time
- Number of line to go

Each program (memory space for programming) has a status:

- 255 - there is no program
- 0 - there is a program and it is the program beginning
- 1-244 - there is a program and it is not the program beginning

Stop the regulatory process and to stop work on the program, you can specify in the address line of transition value 0xFF (255).

If the text of the program line number where you want to go - 0xFF, respectively, will be switched off (regulation will be stopped)

It is possible to proceed to any line of any program - programs can be cascaded and cycled (repeat periodically). Cascading provides the duration of the process 50 steps and more.

The cascading means that after one program performed another program starts.

It is possible to stop the regulation process. For this purpose the set number of the program to proceed with should exceed 15.

By this command it is possible to write or read a line of a program, to set or read the status of a program.

					Program line contents			
	45	**uc**	**uc**	**uc**	**f**	**ud**	**hh**	**uc**
	1	2	3	4	5	6	7	8
1	Command code hex							
2	MODE 0-record of program line 1-reading of program line 2-setting of program status 3-reading of program status							
3	Number of program to deal with							
4	Number of program line(0-49) if MODE=0 or MODE=1 Or status (0-255) if MODE=2 or MODE=3							
5	Setpoint - temperature to maintain (K) for modes 2 and 3 Voltage for mode 4							
6	Time (s) to maintain the setting value (setpoint)							
7	Most significant nibble – mode of current line (interpretation of the field fl (parameter 5) as temperature of voltage) 0 Interdiction of regulation 2 T-regulation 3 Temperature maintenance (PID) 4 Constant voltage Least significant nibble - number of program to go (0-15) after expiration of Time							
8	Number of line to go (0-49)							

1.7.12. ***Send Telemetry Line***

	46
	1
1	Command code hex

In response to this command, the device sends an information frame with a telemetry line. The parameters values are given according to status (see the command 40h – set the status of telemetry).

1.7.13. *Criterion of Signal of Settling*

	49	**uc**	**uc**	**uc**	**f**
	1	2	3	4	5
1	Command code hex				
2	TEC channel number				
3	2…255 - number of periods of PID Controller after which if temperature is within the limits, the appropriate status signal is on				
4	2…255 - number of periods of PID Controller after which if temperature has gone beyond the limits, the appropriate status signal is off				
5	Allowable deviation				

The parameter 4 should not exceed parameter 3. Both parameters should not be smaller than «2». When the board DX5107 is used, the relay switches according to parameters of the command 0x49 (see also the command 0x4D).

1.7.14. *Status of Devices in System*

	4A
	1
1	Command code hex

The field «PARAMETERS» of the information frame:

	hh	**hh**	**hh**
	1	2	3
1	Presence of devices on bus I2C 0x01-EEPROM 24c256 0x02-PCF8574 with data bus LCD		

	0x04-PCF8574 with control bus LCD 0x08-RTC DS1307
2	TEC1 channel status 0x01- regulation is on 0x02- TEC1 temperature within the setting 0x04- 1-heating 0-cooling 0x08- operation by program of changing in time 0x10- converters channel is present 3 MSB *0x20…0x80- regulation mode* 0- no regulation 1- by program 2- T-regulation 3- to the setting temperature 4- constant voltage
3	TEC2 channel status 0x01- regulation is on 0x02- TEC2 temperature within the setting 0x04- 1-heating 0-cooling 0x08- operation by program of changing in time 0x10- converters channel is present 3 MSB *0x20…0x80- regulation mode* 0- no regulation 1- by program 2- T-regulation 3- to the setting temperature 4-constant voltage

1.7.15. *Setting Command Status to Interface*

	4B	**uc**	**uc**	**uc**
	1	2	3	4
1	Command code hex			
2	0 - interface RS232 will be command 1 - interface RS485 will be command			
3	0 – binary mode WAKE 1 – symbol mode WAKE			
4	Rate of exchange			

	0-9600 1-19200 2-38400 3-57600 4-115200

Command 4B can be transferred at broadcasting addressing. When a broadcasting command is received, the interface RS485 will be command.

After sending this command a specified interface becomes command after restarting the device.

1.7.16. *Digital Output Enable*

	4D	**uc**
	1	2
1	Command code hex	
2	0- disable, 1- enable of digital output	

By this command it is possible to enable/disable corresponding relay switch of the circuit of the digital input-output of DX5107 when a setpoint temperature is achieved (according to parameters of the command 0x49).

The command can be sent without parameters.

The field «PARAMETERS» of the information frame:

	hh
	1
1	0- disable, 1- enable of digital output

1.7.17. *Control of Digital Input*

	4E	**uc**	**hh**

	1 2 3
1	Command code hex
2	TEC channel number
3	D0…D3 - number of program, which run on falling D4…D7 - number of program, which run on rising

By this command it is possible to set numbers of programs, transition to which is carried out by signals of corresponding inputs of the board of the digital input-output DX5107.

The command can be sent without parameter 3.

The field «PARAMETERS» of the information frame:

	hh hh 1 2
1	TEC channel number
2	D0…D3 - number of program, which run on falling D4…D7 - number of program, which run on rising

1.8. Commands of Work with ADC

In the description of commands of work with ADC when referring to ADC the number of a channel is given. In the data field of the information frame the number of ADC input is displayed. Therefore one ADC input can be used for measurement of two channels since there are external multiplexers in the device.

Measuring channel	Parameter	ADC input
0	Supply voltage	00
1	TEC1 voltage	01
2	TEC2 voltage	01
3	TEC1 current	02
4	TEC2 current	02

Measuring channel	Parameter	ADC input
5	TEC1 temperature	03
6	TEC2 temperature	04

1.8.1. *ADC Hardware Calibration*

The command serves for hardware calibration of ADC.

	10	**uc**	**uc**
	1	2	3
1	Command code hex		
2	ADC channel number		
3	Calibration type 1- Self Calibration for Offset and Gain 4- System Calibration for Offset only 3- Self Calibration for Gain only		

The command starts calibration of ADC channel. At parameter d=4 there is a system offset calibration. When sending this command it is necessary to provide a zero level on the appropriate input of measurement.

At parameter d=3 there is a self-Calibration for Gain by the reference voltage. This calibration is accompanied by filling of the appropriate registers of ADC microcircuit. The command is terminated by record of registers of ADC microcircuit in the non-volatile memory for a further application at switching onto the ADC channel calibrated.

1.8.2. *ADC Calibration*

	11	**uc**	**f**
	1	2	3
1	Command code hex		
2	ADC channel number		
3	Value of a calibrated level of physical value		

The command can be applied to the ADC channel for which all hardware calibrations (see the command 10h) are done. During the calibration the calibrating coefficient of linear function of conversion is calculated and stored in the non-volatile memory. For the calibration of temperature measurement channel the physical value level should be specified resistance of the calibrating resistor.

1.8.3. *Writing Calibration ADC Coefficients*

	12	**uc**	**f**
	1	2	3
1	Command code hex		
2	ADC channel number		
3	Calibration coefficient value		

The command serves for storing the calibration ADC coefficient into the non-volatile memory.

1.8.4. *Writing ADC Filter Coefficient*

	13	**uc**	**uc**
	1	2	3
1	Command code hex		
2	ADC channel number		
3	Filter coefficient value		

The command serves for recording the coefficient of the digital filter that influences the ADC readouts into the non-volatile memory.

If the filter coefficient is K, the filter time constant is tau=K*dT, where dT is time of ADC quantization. Therefore at K=1 the ADC indications are not filtered.

When the PID controller is in work, one should not apply the digital filter to ADC signals of temperature measurement.

1.8.5. *Sending ADC coefficients*

	14	**uc**
	1	2
1	Command code hex	
2	ADC channel number	

The field «PARAMETERS» of the information frame:

	hh	**e6**	**uc**	**hh**
	1	2	3	4
1	Number of ADC input			
2	Calibration coefficient value			
3	Filter coefficient value			
4	determines amplification coefficient PGA= 2^{hh}			
	0-1	2-4	4-16	6-64
	1-2	3-8	5-32	7-128

1.8.6. *Sending Registers of ADC Channel Offset*

	15	**uc**
	1	2
1	Command code hex	
2	ADC channel number	

The field «PARAMETERS» of the information frame:

	hh	**hh**	**hh**	**hh**
	1	2	3	4
1	Number of ADC input			
2…4	ADC channel offset register values obtained as a result of offset system calibration			

1.8.7. *Starting Measurement in ADC Channel*

	16	**uc**
	1	2
1	Command code hex	
2	ADC channel number	

The field «PARAMETERS» of the information frame:

	hh	**hhhhhhhh**	**e6**	**e6**
	1	2	3	4
1	Number of ADC input			
2	Measurement of ADC channel in hexadecimal units			
3	Measurement of ADC channel in physical units taking into account calibration coefficients (for the channels 5 and 6 – value of thermistor resistance)			
4	Measurement of ADC channel in physical units taking into account calibration coefficients (for the channels 5 and 6 – temperature value)			

1.8.8. *ADC One Channel Measurement*

To increase the speed of digitalizing there is a command 17h that transfers ADC into a mode in which the channels are not switched, and the one chosen is measured. This mode can be used for measuring the object dynamic characteristics.

	17	**uc**	**uc**

	1	2	3
1	Command code hex		
2	ADC channel number		
3	1- switching on of the mode of one channel measurement 0- switching off of the mode of one channel measurement		

The field «PARAMETERS» of the information frame:

	hh
	1
1	Byte with bit set in a position corresponding to a chosen channel, if the channel for quick measurement is chosen, otherwise - 0x00 **ADC channel** 01h supply voltage 02h TEC1 voltage 04h TEC2 voltage 08h TEC1 current 10h TEC2 current 20h TEC1 temperature 40h TEC2 temperature

1.8.9. *Mask on ADC Channels*

If some ADC channels are not needed, they can be excluded from the process of measurement, after sending the command 18h.

	18	**hh**
	1	2
1	Command code hex	
2	Mask	

The field «PARAMETERS» of the information frame:

	hh
	1
1	Mask

The mask is interpreted as a byte submitted in the hexadecimal form. Every bit of this byte either includes (bit "1"), or excludes (bit "0") a corresponding ADC channel in/from the process of measurement.

Mask	ADC channel
01h	supply voltage
02h	TEC1 voltage
04h	TEC2 voltage
08h	TEC1 current
10h	TEC2 current
20h	TEC1 temperature
40h	TEC2 temperature

1.8.10. *PGA of ADC thermistor channel*

	19	uc	uc
	1	2	3
1	Command code hex		
2	ADC channel number		
3	determines Programmable Gain Amplifier = 2^{hh} 0-1 2-4 4-16 6-64 1-2 3-8 5-32 7-128		

1.8.11. *Thermistor coefficients*

The command is used for setting the function of conversion of the thermistor resistance into temperature T=f(R).

There may be set two types of such functions – one given by polynomial and one given by the Steinhart and Hart equation.

Thermistor polynomial coefficients

When using a polynomial function, it is supposed to apply a fifth-degree polynomial.

$T=A_0 + A_1*R + A_2*R^2 + A_3*R^3 + A_4*R^4 + A_5*R^5$

If the polynomial degree is lower than five, unnecessary coefficients should be set zero.

Steinhart and Hart equation

The thermistor coefficients A, B, and C linearize the thermistor temperature resistance curve and are related using the Steinhart and Hart equation as follows:

$$\frac{1}{T} = A + B[\ln R] + C[\ln R]^3$$

Where:

T = Temperature (K)

R = Thermistor resistance (ohms)

A, B, C = Thermistor sensor coefficients

These coefficients should be set to values specified by the thermistor manufacturer.

This command is interpreted differently depending on parameters number.

Writing type of thermistor conversion function

	1A	**uc**	**uc**
	1	2	3
1	Command code hex		
2	ADC channel number		
3	Type of conversion function 3 - Steinhart and Hart equation 5 - polynomial function		

Writing thermistor coefficients

	1A	**uc**	**uc**	**f**
	1	2	3	4
1	Command code hex			
2	ADC channel number			
3	thermistor coefficient order (n =0…5)			
4	thermistor coefficients			

n	Steinhart and Hart equation	Polynomial
0	A	A_0
1	B	A_1
2	C	A_2
3		A_3
4		A_4
5		A_5

Sending thermistor coefficient

	1B	**uc**	**uc**
	1	2	3
1	Command code hex		
2	ADC channel number		
3	thermistor coefficient order (n =0…5)		

The field «PARAMETERS» of the information frame:

	hh	**uc**	**uc**	**e6**
	1	2	3	4
1	ADC channel number (5, 6)			

2	Type of conversion function 3 - Steinhart and Hart equation 5 - polynomial function
3	thermistor coefficient order (n =0…5)
4	thermistor coefficients

n	*Steinhart and Hart equation*	*polynomial*
0	A	A_0
1	B	A_1
2	C	A_2
3	-	A_3
4	-	A_4
5	-	A_5

1.8.12. *Save the Current Settings of Thermistor Input*

	1C	**uc**
	1	2
1	Command code hex	
2	TEC channel number	

The current calibration settings of thermistor measurements input are stored in a line, according to the parameter of the set current and ADC gain.

A table line stores: the conversion factor, the values of calibration offset registers and sign validity data.

1.8.13. *Restore Thermistor Input Settings*

	1D	**uc**	**uc**
	1	2	3
1	Command code hex		

2	TEC channel number
3	Parameter corresponding to the set characteristics of the calibration (to the maximum possible thermistor resistance)

It is checked whether the data stored in the table are correct (data saved by the command 1C). If the data are correct, the calibration parameters, the values of the measuring current and channel ADC gain are filled according to them.

The field «PARAMETERS» of the information frame:

	hh
	1
1	The command execution result: 0x01- parameters are restored 0x00- requested parameters aren't stored by the command1C.

1.8.14. *Save Table of Settings (Backup)*

	1E
	1
1	Command code hex

After sending this command, the controller outputs the contents of the table with the parameters stored in the nonvolatile memory into the command and non-command interfaces.

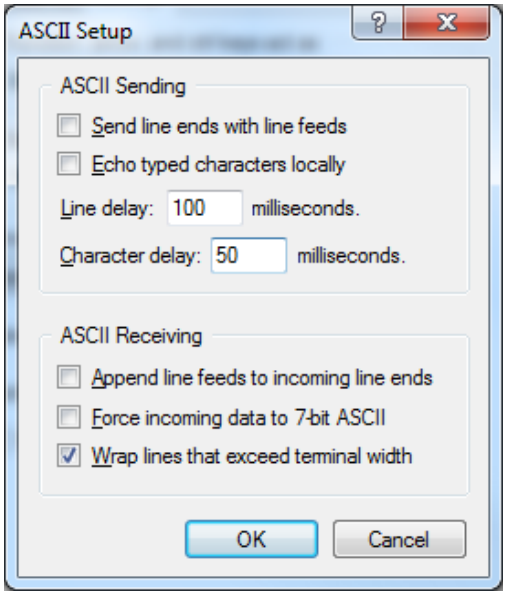
Each byte is transmitted as two characters of the hexadecimal digits. This command is used for backup of parameters stored. The output of parameters will begin into both interfaces after reception of any character by any interface. The data corresponding to one line are stored in 8 bytes.

1.8.15. *Restore Table of Settings (Backup)*

	1F
	1
1	Command code hex

After receiving this command, the device expects data to come by the command or non-command interface and interprets them as stored by the command 1E. The data received are stored in the non-volatile memory.

For the correct execution of this command, set the delay for characters 50 ms when the file is outputted.



1.9. Commands of Work with DAC

In the description of commands of work with DAC, when addressing to DAC the TEC channel number is indicated. In data field of the information frame a number of output of the DAC processor is given.

1.9.1. *Setting DACs in Volts*

	21	**uc**	**f**
	1	2	3
1	Command code hex		
2	TEC channel number		
3	Voltage in Volts		

When the command is executed, the set voltage does not exceed maximal for a chosen channel.

The field «PARAMETERS» of the information frame:

	hh	**ud**
	1	2
1	DAC channel	
2	Voltage in DAC units	

1.9.2. *Setting DACs Directly*

	22	**uc**	**ud**
	1	2	3
1	Command code hex		
2	TEC channel number		
3	Voltage in DAC units		

Attention! *Check for excess of maximal voltage is not done!*

The command is used for calibration. It should be used at voltage up to 8 V.

The field «PARAMETERS» of the information frame:

	hh	ud
	1	2
1	DAC channel	
2	Voltage in DAC units	

1.9.3. *Writing DAC Calibration Coefficients*

	23	uc	f	f
	1	2	3	4
1	Command code hex			
2	TEC channel number			
3	Value of offset of linear function of conversion			
4	Value of proportionality factor of linear function of conversion			

The coefficients determine the function by which a value loaded to DAC is obtained, depending on voltage needed.

1.9.4. *Sending Coefficient and DAC Maximal Values*

	24	uc
	1	2
1	Command code hex	
2	TEC channel number	

The field «PARAMETERS» of the information frame:

	hh	e6	e6	f2
	1	2	3	4
1	DAC channel			
2	Value of offset of linear function of conversion			
3	Value of proportionality factor of linear function of conversion			
4	TEC maximal voltage			

1.9.5. *Writing Maximal Allowable Voltage*

	25	uc	f
	1	2	3
1	Command code hex		
2	TEC channel number		
3	TEC maximal voltage (V)		

1.9.6. *Voltage of T-reg*

	26	uc	f
	1	2	3
1	Command code hex		
2	TEC channel number		
3	voltage of T-regulation (V)		

The field «PARAMETERS» of the information frame:

	hh	f2
	1	2
1	TEC channel number	
2	Value of voltage of T-regulation (V)	

The command can be sent without parameters.

1.10. Commands of Work with PID

1.10.1. *Setting of TEC Polarity*

Management of TEC voltage polarity is carried out by the command 30h.

	30	**uc**	**uc**
	1	2	3
1	Command code hex		
2	TEC channel number		
3	0- TEC is off 1- TEC is heating 2- TEC is cooling		

The command execution is accompanied by the message of the form "bridge AA-BB" sent into a non-command interface, where AA is TEC channel number, BB is the third parameter of the command.

1.10.2. *Writing Parameters of PID Controller*

	31	**uc**	**f**	**f**	**f**
	1	2	3	4	5
1	Command code hex				
2	TEC channel number				
3	Value of proportional coefficient of PID *controller*				
4	Value of integral coefficient of PID *controller*				
5	Value of differential coefficient of PID *controller*				

1.10.3. *Sending Parameters of PID Controller*

	32	**uc**	**f**	**f**	**f**
	1	2	3	4	5
1	Command code hex				
2	TEC channel number				

The field «PARAMETERS» of the information frame:

	hh	**f6**	**f6**	**f6**
	1	2	3	4
1	TEC channel number			
2	Value of proportional coefficient of PID *controller*			
3	Value of integral coefficient of PID *controller*			
4	Value of differential coefficient of PID *controller*			

1.10.4. *Set thermistor current*

	33	**uc**	**uc**
	1	2	3
1	Command code hex		
2	TEC channel number		
3	0- current ~1uA 1- current ~83uA		

The command 33 can be sent without parameter 3.

The field «PARAMETERS» of the information frame:

	hh	**uc**
	1	2
1	TEC channel number	
2	0- current ~1uA	

	1- current ~83uA

1.10.5. *Sending/Set Setpoints of PID Controller*

	34	**uc**	**[f2]**
	1	2	3
1	Command code hex		
2	TEC channel number		
3	If it is present - Value of setpoint temperature of PID (setting value without restarting the PID).		

The field «PARAMETERS» of the information frame:

	hh	**f2**	**f2**	**uc**	**uc**
	1	2	3	4	5
1	TEC channel number				
2	Value of setpoint temperature of PID controller (setting value)				
3	Value of deviation at which the signal of required temperature settling is generated in the status of PID controller				
4	Criterion of signal of setting value achieved (see command 49h)				
5	Criterion of signal of setting value gone (see command 49h)				

1.10.6. *Starting Controller*

	35	**uc**	**uc**	**f**
	1	2	3	4
1	Command code hex			
2	TEC channel number			
3,4	3		4	
	0	Regulation stop	Not present	
	1	Regulation according to program	0…15 program to go to	

2	T-regulation	Temperature to be maintained
3	Temperature maintenance – PID starting – setting regulation	Temperature to be maintained (setting value)
4	Constant voltage	Voltage to be maintained

The command 35 can be sent at broadcasting addressing.

The command execution is accompanied by the message of the form "bridge AA-BB" sent into a non-command interface (if enabled, see the command 40), where AA is TEC channel number, BB is the third parameter of the command.

1.10.7. *Parameters of Output of PID Controller*

	36	**uc**	**hh**
	1	2	3
1	Command code hex		
2	TEC channel number		
3	Status byte - bits turn on "1" or off "0" certain functions		
	0x01	controls voltage output. I it is «0», voltage is off and it is only polarity that switches	
	0x02	reserved	
	0x04	controls a character of regulation. Setting this bit transfers the process into T-regulation (relay)	

The field «PARAMETERS» of the information frame:

	hh	**uc**
	1	2
1	TEC channel number	
2	Status byte	
	0x01	controls voltage output. I it is «0», voltage is off and it is only polarity that switches
	0x02	reserved

0x04	character of regulation. "1"- T-regulation (relay)
0x08	1-thermitor is present 0-thermistor is not present

1.10.8. *Starting Z-meter*

	37	**uc**	**uc**	**uc**
	1	2	3	4
1	Command code hex			
2	TEC channel number			
3	Z-meter measurement time (s) 20…255			
4	if "1" – only resistant measurement			

The field «PARAMETERS» of the information frame:

	" Z-metr	uc	started. Wait!!! "
		1	
1	TEC channel number		

Time indicated in parameters of the command 37h, is not the time of diagnostics. The time obviously should be quite more than the trebled time constant of the object. The diagnostics consists of the several intermediate stages, each of which takes time exceeding that specified in parameters of the command 37h. The time of diagnostics is estimated to be 5 times longer than the specified time.

During the diagnostics, as intermediate stages are over, the debugging information is outputted in the non-command interface.

After Z-metering is over, resistance (R), figure-of-merit Z, time constant (tau) are outputted into non-command interface (if there was no mistake «TEC voltage has not fallen for too long in Z-metering»).

Attention! *The procedure of Z-metering is a long process during which commands are not received and information frames are not created. LED indication is ceased (its beginning again is a signal of Z-metering being over).*

During the diagnostics of any channel, regulation on this channel stops.

It is possible to interrupt the procedure by sending the command “Send echo”.

1.10.9. *Storage of Z-Metering Parameters*

	38
	1
1	Command code hex

By this command the found parameters are stored as reference ones reference for the given object.

1.10.10. *Z-Meter Current*

	39	**f**
	1	2
1	Command code hex	
2	Value of calibration resistance switch on to the channel PID1	

By this command the value of calculated current is stored in the non-volatile memory and used for Z-metering calculations.

The field «PARAMETERS» of the information frame:

	f6
	1

1	Electric current value of Z-meter

The command can be sent without parameters.

1.10.11. *Switching On Regulation after Restarting*

	3B	**uc**	**uc**	**f**	**ud**
	1	2	3	4	5

1	Command code hex		
2	TEC channel number		
3 4		3	4
	0	Regulation stop	Not present
	1	Regulation according to program	0…15 program to go to
	2	T-regulation	Temperature to be maintained
	3	Temperature maintenance – PID starting – setting regulation	Temperature to be maintained (setting value)
	4	Constant voltage	Voltage to be maintained
5	time (s) after which to proceed to *the program (only for Regulation according to program)*		

The field «PARAMETERS» of the information frame:

	hh	hh	f2	**ud**
	1	2	3	4

1	TEC channel number		
2 3		2	3
	0	Regulation stop	Not present
	1	Regulation according to program	0…15 program to go to
	2	T-regulation	Temperature to be maintained
	3	Temperature maintenance – PID starting – setting regulation	Temperature to be maintained (setting value)
	4	Constant voltage	Voltage to be maintained

4	time (s) after which to proceed to the program *(only for Regulation according to program) or 0*

The command can only be sent with the parameter 2.

1.10.12. *Writing Limiting Temperatures*

	3C	**uc**	**f**	**f**	**uc**
	1	2	3	4	5
1	Command code hex				
2	TEC channel number				
3	Minimal allowable temperature				
4	Maximal allowable temperature				
5	If during this time temperature happens to be beyond the set limits - the error signal is outputted - see the device status				

1.10.13. *Sending Limiting Temperatures*

	3D	**uc**
	1	2
1	Command code hex	
2	TEC channel number	

The field «PARAMETERS» of the information frame:

	hh	f2	f2	**uc**
	1	2	3	4
1	TEC channel number			
2	Minimal allowable temperature			
3	Maximal allowable temperature			
4	If during this time temperature happens to be beyond the set limits - the error			

	signal is outputted - see the device status

1.10.14. *Sending Z-metering Results*

	3E
	1
1	Command code hex

The field «PARAMETERS» of the information frame:

	hh	f2	e2	**f2**
	1	2	3	4
1	TEC channel number whose parameters are given further			
2	Resistance (R) (Ohm)			
3	Figure of merit Z (1000/K)			
4	Time constant tau (s)			

1.10.15. *Sending Z-metering Parameters*

	3F	**cu**
	1	2
1	Command code hex	
2	TEC channel number	

The field «PARAMETERS» of the information frame:

	hh	f2	e2	**f2**
	1	2	3	4
1	TEC channel number whose parameters are given further			
2	Resistance (R) (Ohm)			

3	Figure of merit Z (1000/K)
4	Time constant tau (s)

1.10.16. *Autotuning PID*

	51	**uc**
	1	2
1	Command code hex	
2	TEC channel number	

The field «PARAMETERS» of the information frame:

	“Tuning PID	uc	started. Wait!!! ”
		1	
1	TEC channel number		

The autotuning function searches the values of proportional, integral and differential coefficients of the PID algorithm.

During the process, as the intermediate stages being over, the debugging information is outputted into a non-command interface.

After the end of autotuning the coefficients of a proper PID channel are set.

The coefficients are obtained for the cooling mode in which all the ADC channels are allowed to be measured – the longest sampling period of PID.

Attention! *The procedure of adjustment may be long process and during it no commands are received and no information frames are formed. The LED indication stops (the beginning of LED indication can be a signal of the autotuning end). During autotuning of a channel, regulation of the other*

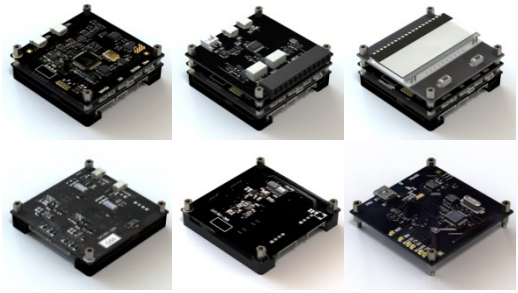
channel stops.

1.10.17. *Reset of controller*

	53
	1
1	Command code hex

1.10.18. *Management of indication board*

	54 [#screen]
	1 2
1	Command code hex
2	Most significant nibble – unit temperature display 1 - degree Celsius, 0- Kelvin Least significant nibble - number Start screen 0- "E1" ... 3 - "E4"



PL Engineering Ltd.

46 Warshawskoe shosee
Moscow 115230 Russia
e-mail: info@promln.com
phone: +7-499-678-3231
fax: +7-499-678-3258
website: www.promln.ru

Overseas Sales representative

TEC Microsystems GmbH
Schwarzschildstrasse 8
Berlin 12489, Germany
phone: +49-(0)30-6789-3314
fax: +49-(0)30-6789-3315
e-mail: info@tec-microsystems.com
website:www.tec-microsystems.de