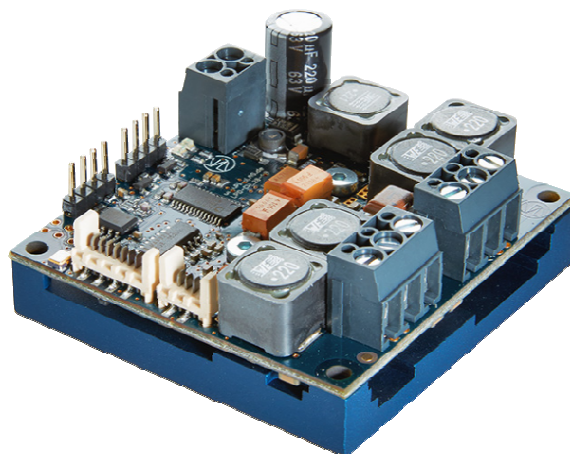




# User Manual

PL-TEC 2-1024



PicoLAS GmbH  
Burgstr. 2  
52146 Würselen  
Germany

Phone: +49 (0) 2405-64594-60  
Fax: +49 (0) 2405-64594-61  
E-mail: [info@picolas.de](mailto:info@picolas.de)  
Web: [www.picolas.de](http://www.picolas.de)

## Table of Contents

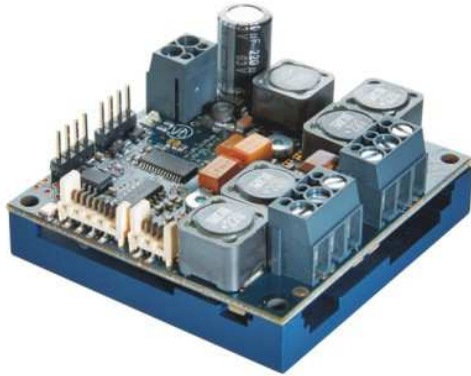
Product Overview .....	3
Description of Connections .....	4
How to get started with a PLB-21 .....	8
How to get started without a PLB-21 .....	8
Dos and Don'ts .....	9
Absolute maximum Ratings .....	9
Functional Description .....	10
Power Supply.....	11
Cooling .....	11
Overtemperature Shutdown.....	11
Mechanical Dimensions .....	12
Power on Self Test.....	13
Controlling the Driver .....	13
Controlling the Driver using a PLB-21 .....	14
Controlling the Driver via PC.....	17
Register Description .....	31



## PL-TEC 2-1024

Rev. 1905

### Fully digital controlled Driver for TEC Modules



- Bipolar  $\pm 10$  A or dual  $\pm 5$  A output
- Temperature stability up to 0.001 K
- Adjustable TEC current and voltage limiting to protect TEC module
- Direct TEC temperature control
- Low ripple and low noise design
- Multiple monitoring outputs

### Product Description

The PL-TEC 2-1024 is a very small, safe, and most accurate temperature controller for Peltier thermoelectric cooler (TEC) modules.

An ultra low-drift chopper amplifier maintains  $\pm 0.001$  K temperature stability. Output current, rather than voltage, is directly controlled to eliminate current surges. Individual heating and cooling current limits and voltage limits provide the highest level of TEC protection.

The PL-TEC 2-1024 operates from a single supply and provides a single bipolar  $\pm 10$  A output or two independent  $\pm 5$  A outputs (user selectable with a switch). The bipolarity of the TEC is realized through the outputs of four synchronous buck regulators. Therefore, it is possible to keep the temperature constant even by low load currents without "dead zones" or other nonlinearities.

An analog control signal precisely sets the TEC temperature. The actual TEC temperature and current can be monitored over the control interface. In addition, separate over- and under-temperature information can be monitored over the digital interface. An on-chip voltage reference provides bias for a thermistor bridge.

Through the new efficient design, just the baseplate of the PL-TEC 2-1024 needs to be cooled. Additional forced air cooling is not necessary.

### Technical Data\*

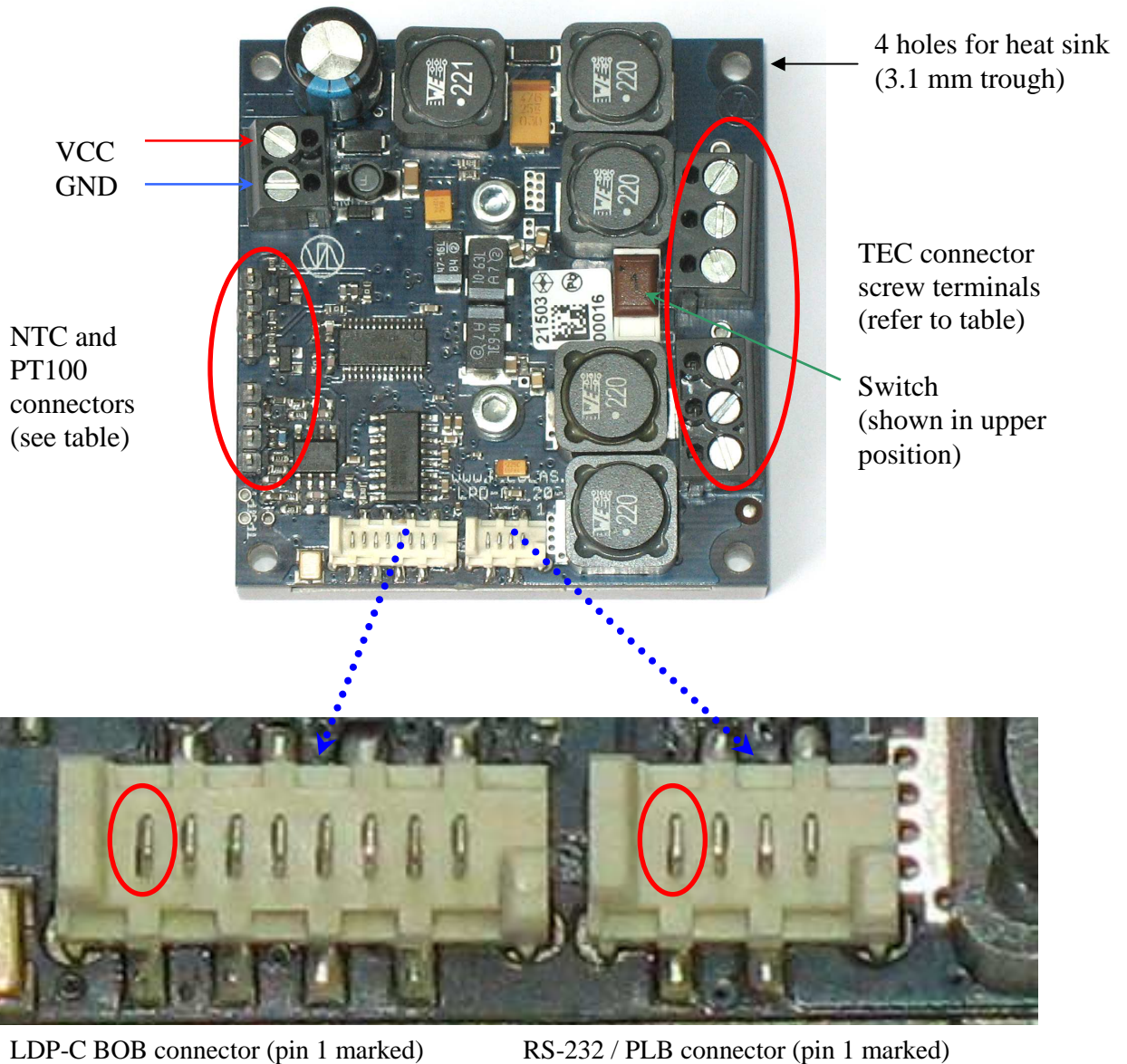
<u>Single output mode</u>	
Output current	Bipolar 10 A
Output current limiting	Heating 0 .. 10 A
Range	Cooling 0 .. 10 A
	User settable via interface
<u>Dual output mode</u>	
Output current	Bipolar 5 A, each
Output current limiting	Heating/cooling each
Range	channel independent:
	0 .. 5 A
Max. output voltage	-24 .. +24 V
Output voltage limiting	Heating/cooling independent
Range	0 .. 24 V user settable
<u>Both operation modes:</u>	
Supply voltage	+12 .. +50 V, min. 2 V
	above TEC voltage
Temperature stability	0.001 K
Thermistor inputs	A) NTC 10k/NTC100k
	B) PT100
	C) NTC 10k/NTC100k
	D) PT100
Temperature set point	0 .. 5V (each channel)
	or via digital interface
Monitor outputs	Via digital interface
Power dissipation	< 45 W
Dimensions in mm	60,9 x 57,8 x 29
Weight	194 g
Operating temperature	0 to + 55 °C

\* Technical data is subject to change without further notice.

Optional Accessories: PLB-21

## Description of Connections

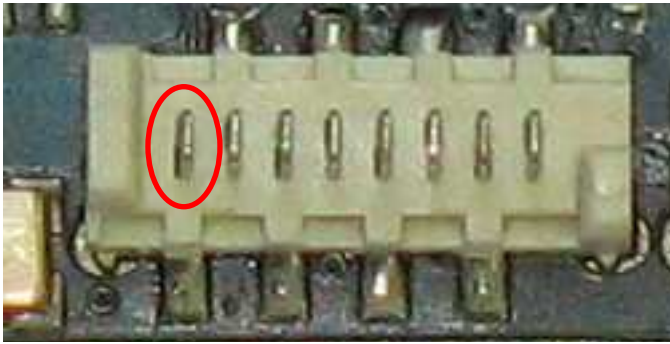
The following drawing shows all connections which are available to the user.



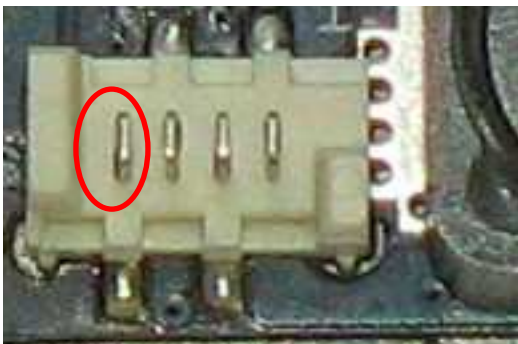
VCC	Supply voltage 12 .. 48 V, 50 V max.
GND	Supply ground
LD+	Positive laser diode output (anode). Use both connectors parallel for high currents.
LD-	Negative laser diode output (cathode). Do not connect to ground! Use both connectors parallel for high currents.
LDP-C BOB Conn	Mini MOLEX connector for analog/TTL control of the driver (see below)
RS-232/PLB21-Conn	Mini MOLEX connector for RS-232 communication or PLB-21 (see below)

For a more convenient use of the driver (e.g. in laboratory use) we recommend the optional available product accessory LDP-C BOB. Please see LDP-C BOB manual for further details.

Pin assignments and description of the MOLEX pin headers. The pictures are in the same view than the overview picture.



Pin Number	Description	Direction	Comments
1 (marked in red)	reserved		Do not connect!
2-3	reserved		Do not connect!
4	ENABLE	Input	0V = disabled, 3.3 V = driver enabled Please note: Some digital commands are not available when this pin is pulled high
5	TEC_OK	Output	0V = Driver not ok or power up self test not passed yet; 3.3V = Driver ok / test passed
6	GND	Output	
7	reserved		Do not connect
8 (right)	GND	Output	Since HW rev. 2.1: 5 V / 100mA output



Pin Number	Description	Direction	Comments
1 (marked in red)	+12 V	Output	Only for the use with the PLB-21, prevent any overload or short! (Max. 100 mA)
2	TXD	I/O	RS-232 - Standard serial interface
3	RXD	I/O	RS-232 - Standard serial interface
4 (right)	GND	Output	

Pin assignments and description of the thermistor pin headers. The pictures are in the same view than the overview picture.

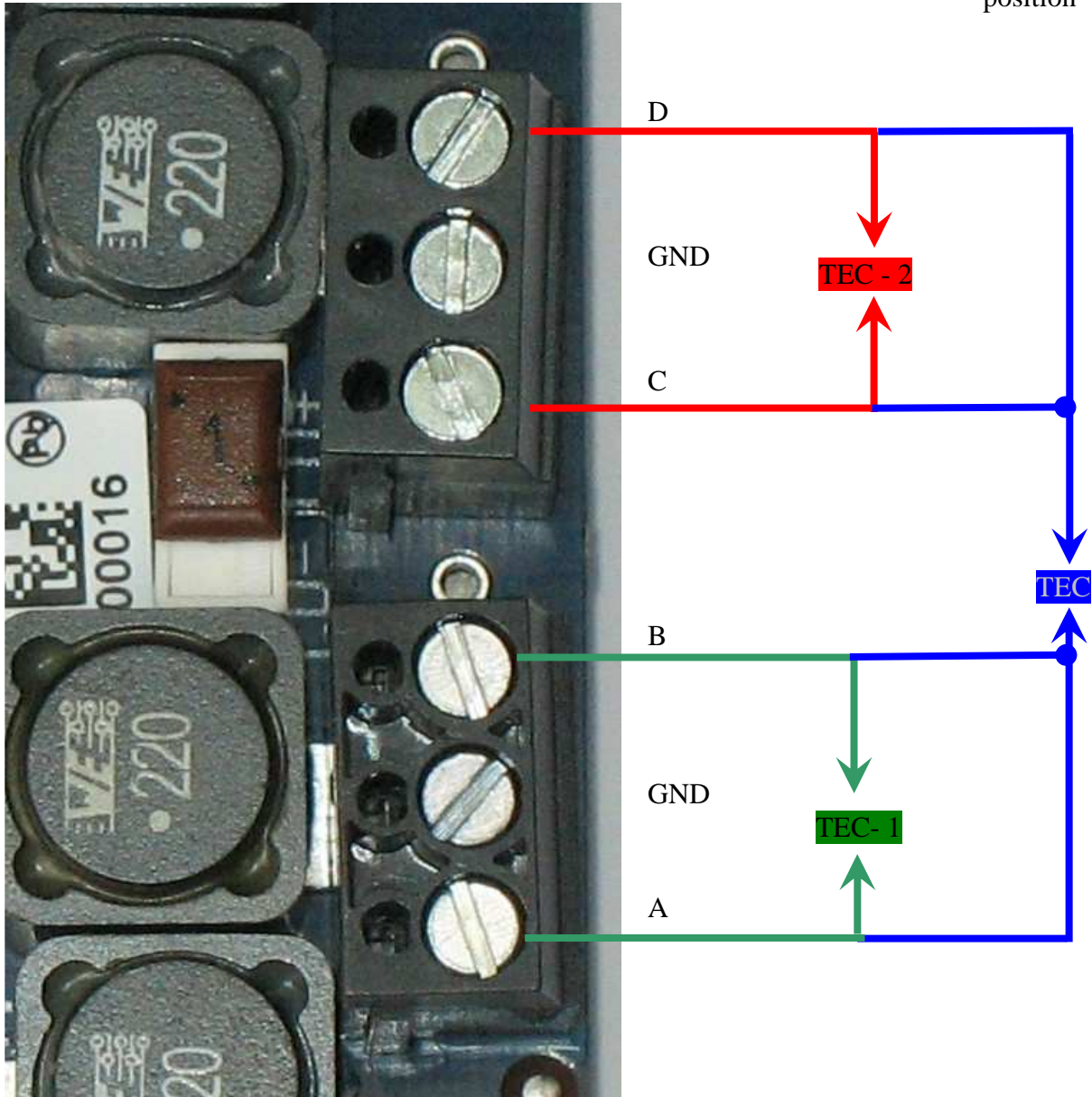
**Warning:** For improved EMI noise immunity the leads of the thermistors must be kept floating. The GND pins of the PT100 can be used for shielding. Shielding must be kept single ended, too.

**Remark:** The targeted value for the NTC is 10k (factory set). If you need 1k NTCs or a low temperature, switch from 10k to 1k via software command.

	Single channel operation Switch in lower position	Dual channel operation Switch in shown position	
Channel 1 NTC 10 k / 1 k	NTC Ch- 1	NTC Ch- 1	
	NTC Ch- 1 Return	NTC Ch- 1 Return	
Channel 2 NTC 10k / 1k	No function, do not connect	NTC Ch- 2 Return	
	No function	NTC Ch- 2	
Channel 1 PT 100	PT100-Ch1	PT100-Ch 1	
	PT100-Ch 1 Return (GND)	PT100-Ch 1 Return (GND)	
Channel 2 PT 100	No function (GND)	PT100-Ch 2 Return (GND)	
	No function	PT100-Ch 2	

Pin assignments and description of the TEC screw terminals. The pictures are in the same view than the overview picture.

Picture of the pcb in same position as in main overview	Output name	Dual Channel operation Switch in shown position	Single channel operation Switch in lower position
---	-------------	--	--



**Remark:** The polarity of the TEC cannot be changed by software command. Therefore, if the TEC heats or cools uncontrollable, the polarity has to be changed manually.

**Remark:** If the switch is altered during power-on the TEC will shutdown itself until the power is toggled.

**Warning:** For single channel operation the user must connect output A with B and C with D as shown.

## How to get started with a PLB-21

Step	What to do	Note
1	Unpack your device.	
2	Connect the PLB-21.	Use PLB-21 pin header and special cable
3	Apply the supply voltage.	Apply any voltage between 12 V and 48 V at “VCC” with respect to GND.
4	Wait until “pulser-ok” is high.	Pin 5 with respect to pin 6 must turn from 0 V to 3.3 V
5	Apply the temperature setpoint.	Use the PLB-21 to adjust the setpoint temperature by turning the dial.
6	Set enable pin “high”.	Apply 3.3 V to pin 4 of the BOB connector. The current will start up now.
7	Check	If the driver is not properly cooled, it will shut down.

## How to get started without a PLB-21

**Remark:** The temperature setpoints can not be applied analog. However, the usage of the driver without any digital interface (RS-232 via PC or PLB-21) is possible when all necessary settings have been done previously.



## Enabling / Disabling the Driver

The output of PL-TEC 2-1024 can either be enabled via an external signal (BOB connector) or a bit in the LSTAT register. This behaviour can be altered via the config menu in the PLB-21 menu or via the serial interface. When the PL-TEC 2-1024 is configured for internal enable, the ENABLE pin of the BOB connector is ignored.

## Dos and Don'ts

Never make a short at the output. This will not do any harm to the laser driver but will yield in an incorrect current measurement.

Never disconnect the Peltier module while the output is enabled.

Never switch between single and dual channel operation while the PL-TEC 2-1024 is powered up.

Mount the driver on an appropriate heat sink. The driver will shut down under overtemperature circumstances. Depending on the desired current a slight airflow across the black coils on top is necessary. Please keep the temperature of these devices below 85 °C

Never connect either of the TEC Outputs to GND / VCC as the polarity of the output clamps changes when the TEC switches from heating to cooling or from cooling to heating.

## Absolute maximum Ratings

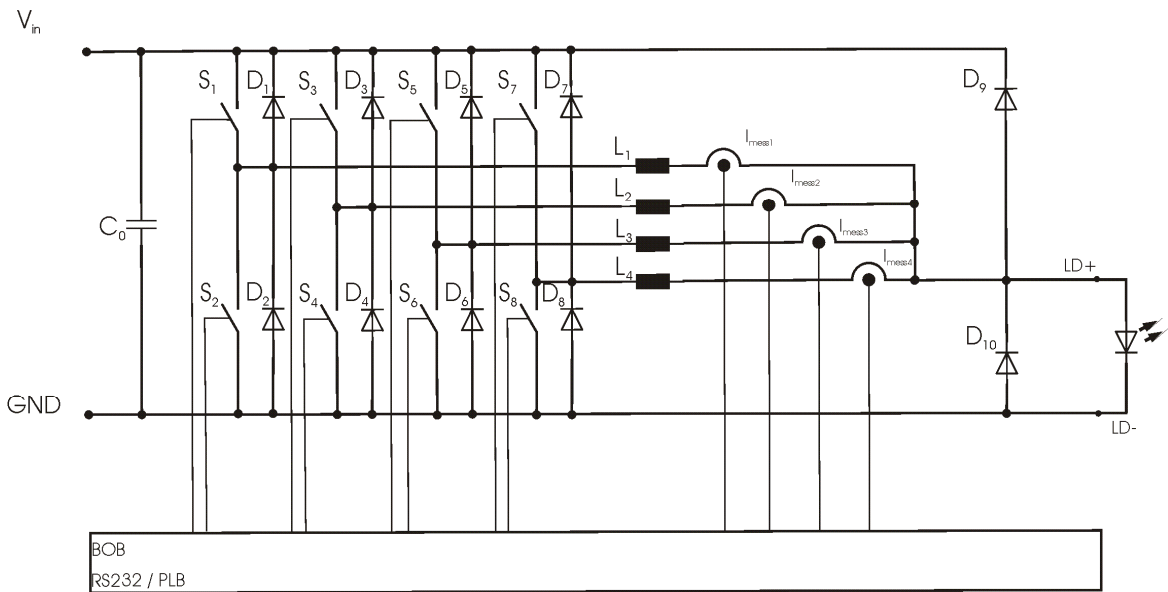
- Supply voltage range: +12 V .. +50 V max. during operation. Destroying limit: 56 V
- Maximum input current: 20 A
- Maximum TEC output current in single channel mode: 10 A
- Maximum TEC output current in dual channel mode: 5 A
- BOB connector input and output voltages: 0 V to 3.3 V (terminals are 5 V proof)
- BOB connector output currents: 1 mA
- Auxiliary 5 V supply voltage output current: 30 mA (pin 8 of BOB connector planned for Rev 2.1)
- Auxiliary 12 V supply voltage output current: 100 mA (pin 1 of PLB-21 connector)
- Operating temperature: 0 - 55 °C

## Functional Description

The PL-TEC operating principle is based on the LDP-CW 20-50. The PL-TEC operates with four parallel buck converters (S1, S2, D1, D2, L1; S3, S4, D3, D4, L2; S5, S6, D5, D6, L3; S7, S8, D7, D8, L4). Every single converter has an independent control loop with a current sensor (I<sub>meas1</sub>, I<sub>meas2</sub>, I<sub>meas3</sub> and I<sub>meas4</sub>). The setpoint current that is pretended by the user is evenly spread over all four converters.

Several security features protect the laser diode and PL-TEC from damage. D8 protects the laser diode from reverse currents while D7 protects the driver in case of a load failure. In case of a failure, the control unit disables the PL-TEC. A soft start mechanism slowly raises the current after enabling the PL-TEC.

In contrary to the LDP-CW 20-50 all 4 channels are linked to separate screw terminals. Furthermore they are terminated with an RC filter to reduce the EMI noise on the TEC.



Operation principle of LDP-CW 20 50 driver

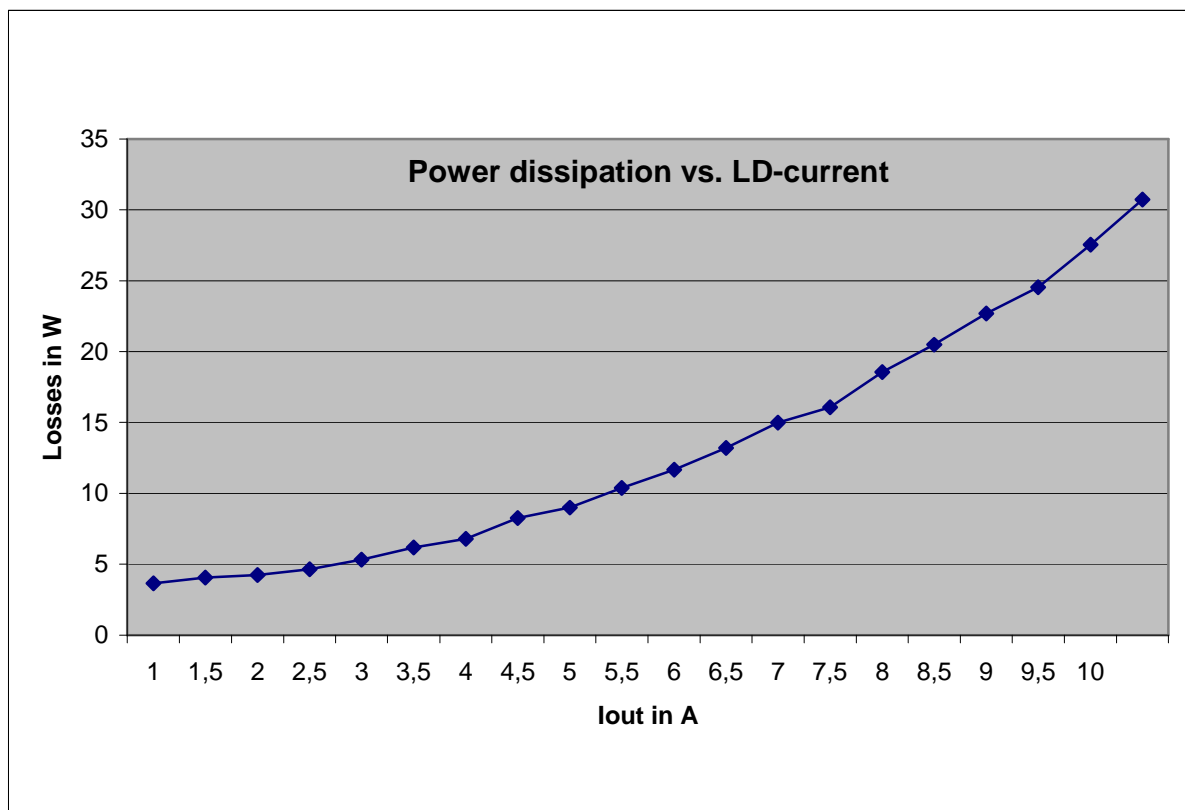
Element	Function
S1, S2, S3, S4, S5, S6, S7, S8, D1, D2, D3, D4, D5, D6, D7, D8, L1, L2, L3, L4	Buck converter
C0	Input buffer capacitor
D7, D8	Free wheeling and driver protection diodes

## Power Supply

The power supply must be able to cover the output power plus the internal power losses (refer to next section). Example: You have a 48 V power supply. The maximum TEC power will be 240 W at 6 A. The total power consumption will be:  $240 \text{ W} + 12 \text{ W} = 252 \text{ W}$ . Divided by 48 V this yields 5.25 A (better 6 A) of current consumption at the DC terminals.

## Cooling

The driver produces up to 31 W of losses. Thus the base plate has to be mounted on a heat sink to ensure proper operation and prevent an overtemperature shutdown. If working with high currents above 5 A, it is recommended to cool the power inductors as well. This can be achieved easily by placing the driver with its heat sink in the air flow of a fan.



Power dissipation over output current

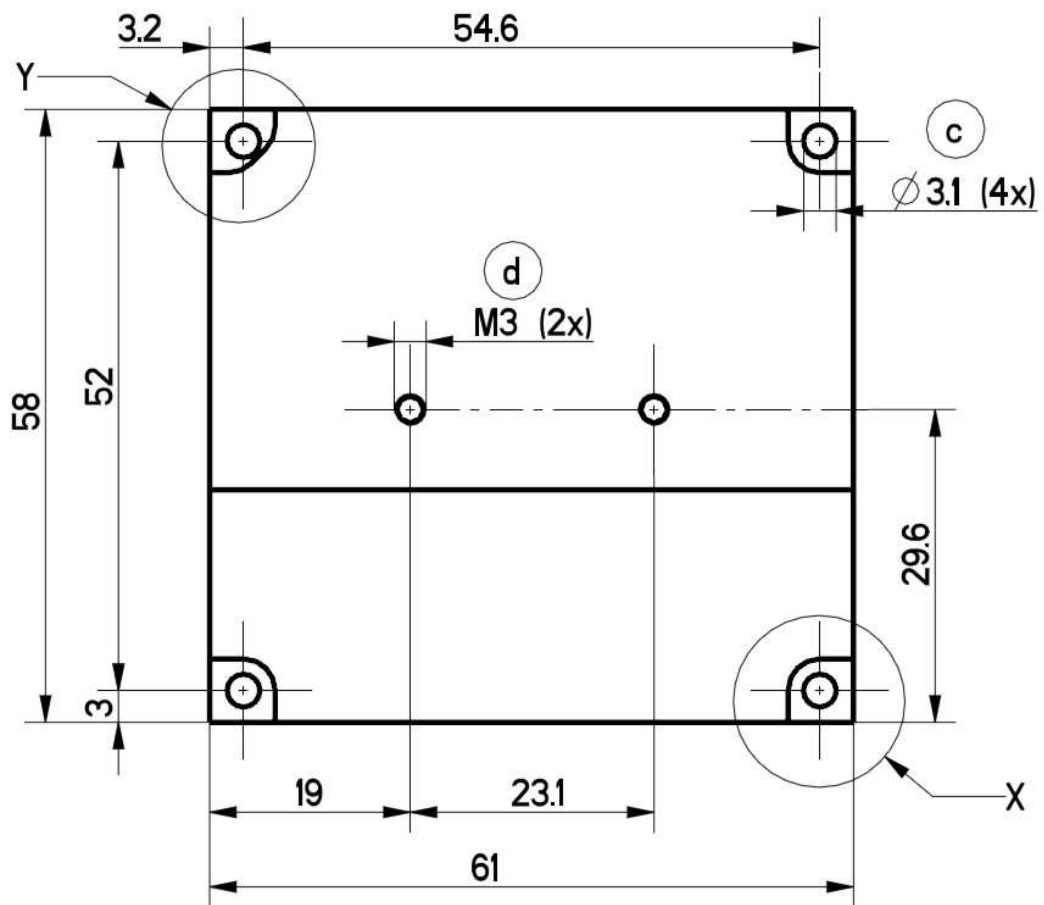
## Overtemperature Shutdown

To protect itself, the PL-TEC automatically shuts down itself if its temperature rises above 80 °C. This condition is latched and the appropriate bit in the ERROR register is set. To re-enable the PL-TEC the ENABLE pin must be toggled.

## Mechanical Dimensions

Over all height: 30 mm

All dimensions in mm



## Power on Self Test

Each time the driver is powered up it performs a test of its internal safety features. The driver cannot be enabled until a self test is performed successfully. The PULSER\_OK signal will be pulled high when the test was successful.

The test will take less than 5 seconds, but can take up to 15 seconds due to internal time-outs if any failure is detected.

## Controlling the Driver

The PL-TEC-2-1024 is designed to operate with a connected digital control. However, it can be operated stand alone when all necessary settings are configured previously as it remembers all settings from the last time it was powered on. Furthermore, it can be configured to load default values on power-on. In this case it loads pre configured settings each time the power is toggled.

Connecting a digital control to the driver does not alter the internal settings.

### **No digital control**

If no digital control is attached, only the BOB connector could be used to enable / disable the driver. To do so the input pin 4 at the BOB connector (“ENABLE”) must be set HIGH. If an error occurs (e.g. overtemperature), the TEC will be disabled and the pin 5 of the BOB connector (“TEC\_OK”) is pulled low. The “ENABLE” pin has to be toggled to enable the driver again.

### **PLB-21**

If a PLB-21 is attached to the TEC, it can be used to control its behavior. The PLB-21 may ask for a driver to download. This must be confirmed with “yes” in order for the PLB-21 to work properly. This must always be done when the PLB-21 was connected to any other PicoLAS product. After the download all operating parameters can be accessed using the PLB-21. For a detailed description see chapter PLB-21 below.

### **PC**

If the TEC is connected to a PC, all operating parameters can be accessed via a serial RS-232 terminal program or the PicoLAS protocol. For a detailed description of the serial text protocol and the PicoLAS protocol see below.

## Controlling the Driver using a PLB-21

When the PLB-21 is connected the first time to the driver the user is asked to download a new driver. This must be confirmed with “yes” for working the PLB-21 properly.

### Menu Structure

The following diagram shows the structure of the PLB-21 menu which affects the driver. All entries are described in detail. All other menu entries are described in the PLB-21 manual. For detailed instructions see the PLB-21 manual.

Note that if the TEC is used in dual channel mode there *Config* and *Controller* entries for both channel 0 and channel 1.

Menu root

- Setpoint
  - o 1: stp.  $F_1$
  - o 1: act.
  - o 2: stp.  $F_1$
  - o 2: act.
  
- Config Ch0 / 1
  - o Input
  - o R (Ohm)
  - o B
  - o T ( $^{\circ}$ K)
  
- Controller Ch0 / 1
  - o I<sub>max</sub> (A)
  - o P
  - o I
  - o D
  
- Config
  - o Enable Ext
  - o Enable
  
- Defaults
  - o Def. pwron
  - o Load defaults
  - o Save defaults

## Setpoint

In this menu point the setpoint temperature for channel 0 and 1 can be modified. Furthermore the actual temperatures can be read.

### 1: stp. F1

This value defines the setpoint temperature of TEC channel 1 in dual channel mode or the setpoint temperature in single channel mode.

### 1: act.

This read only value shows the actual measured temperature which is used by the control loop of channel 1.

### 2: stp. F1

This value defines the setpoint temperature of TEC channel 2 in dual channel mode. When the single channel mode is used this menu point will not be shown.

### 2: act.

This read only value shows the actual measured temperature which is used by the control loop of channel 2. When the single channel mode is used this menu point will not be shown.

## Config Ch0/I

Each TEC channel can be configured independently from each other. Note that the configuration of a single input (R/B/T) applies to both channels if both channels have the same input selected.

If the TEC is configured in single channel mode only “Config Ch0” is shown.

### Input

Possible values are NTC1, NTC2, PTC1 or PTC2 which represents the four different input channels.

### R

This value defines the base resistance of the connected temperature sensor measured in Ohms.

### B

This value defines the B value of the connected temperature sensor. It is only used in case of an NTC.

### T

This value defines the base temperature of the connected temperature sensor measured in °C. It is only used in case of a NTC.

## Controller CH0/I

Each TEC control loop has its own P, I and D values as well as a maximum output current.

If the TEC is configured in single channel mode only “Controller Ch0” is shown.

### Imax (A)

This value defines the maximum output current of the TEC channel measured in amperes.

### P

This value defines the parameter oh the P-controller.

### I

This value defines the parameter oh the I-controller.

### D

This value defines the parameter oh the D-controller.

## Config

The driver can be configured to use a software enable signal instead of the hardware pin of the BOB connector. This can be configured here.

### **Enable Ext.**

When this item is set to “on” the driver uses the hardware enable pin of the BOB connector. Otherwise it uses the menu point below.

### **Enable**

When “Enable Ext” is set to “off” the PL-TEC 2-1024 becomes enabled when this item is set to “on”. Otherwise this shows the state of the ENABLE pin of the BOB connector.

## Defaults

The driver can load a default setting each time it powers up or the user commands it to do so. This is done within this submenu.

### **Def. pwron**

When enabled, the driver loads the saved settings each time it powers up.

### **Load**

When activated via turning the Jogdial or the ENTER key all internal registers are changed to the previously saved values. The output stage has to be re-enabled via the L\_ON bit or the FIRE Key afterwards.

### **Save**

When activated via turning the Jogdial or the ENTER key all internal registers are stored into an internal EEPROM for later usage.

## If an Error occurs

If an error occurs during operation the pulse output is switched off, the “TEC\_OK” signal on the BOB connector is pulled low and a message is displayed on the PLB-21. If no other action is described on the display, a toggle of the ENABLE pin resets the error condition and re-enables the driver.



# Controlling the Driver via PC

## Introduction

When the TEC is connected to a PC, it allows communications over a serial text interface as well as the PicoLAS protocol. While the text interface is designed for communication with a terminal program, the PicoLAS protocol is designed as a system interact protocol.

The switching between the two protocols occurs automatically as soon as the driver receives a certain sequence. The corresponding commands are:

- **PING** for the PicoLAS protocol
- **“init”** followed by <Enter> for the text interface

## Description of the RS-232 Interface

The PL-TEC 2-1024 implements a standard RS-232 interface. It can be connected to a PC using a tree-wire connection. The connection settings are:

Baud rate	115200
Data bits	8
Stop bits	1
Parity	even

## The Serial Text Interface

The following section describes the structure and commands of the text interface.

### Structure

Every command that is sent to the TEC must be completed with a CR (Enter). It consists of a command word followed by a parameter. If the command was successfully executed a “00” is sent, otherwise a “01”. If there is an error pending, the response will be “10”, otherwise “11”. If the command requires an answer parameter, this parameter is sent before the confirmation is given.

Example:

The user would like to read out the actual setpoint temperature of channel 0:

User input:               gsoll 0<Enter>  
Output of the TEC:    12.2<CR><LF>  
                          00<CR><LF>

Example 2:

The user would like to set a new setpoint temperature to channel 0:

User input:               scurrent 0 25.7<Enter>  
Output of the LDP-CW:    25.7<CR><LF>  
                          00<CR><LF>

Input is done in ASCII code and is case sensitive. Every terminal can be used that supports this standard.

## Commands for the TEC 2-1024

The following table contains a command reference for the PL-TEC 2-1024

Command	Parameter	Answer	Description
ssoll	<channel> <value>	--	Sets the setpoint temperature th the given value. A dot is used as decimal point. No more then two decimal places are used! (12.225 is the same as 12.22) Note: In single channel mode the first parameter <channel> must not be set!
gsoll	<channel>	temperature in °C	Outputs the actual setpoint temperature Note: In single channel mode the first parameter <channel> must not be set!
gsollmin	--	temperature in °C	Outputs the minimum output current
gsollmax	--	temperature in °C	Outputs the maximum output current
gist	<channel>	temperature in °C	Outputs the actual measured temperature of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
con	<channel>	--	Activates the control loop of the given channel Note: In single channel mode the first parameter <channel> must not be set!
coff	<channel>	--	Deactivates the control loop of the given channel Note: In single channel mode the first parameter <channel> must not be set!
gerr	--	32 bit number	Outputs the ERROR register
gerrtxt	--	error text	Description of every pending error
gstat	--	--	Returns the value of the internal status register
gserial	--	serial number	Returns the device serial number
gname	--	device name	Returns the devices internal name
ghwver	--	hardware version	Prints out the hardware version
gswver	--	software version	Prints out the software version
ps	--	current settings	Prints out an overview of all settings
loaddefault	--	--	Loads previously saved settings
savedefault	--	--	Saves the current settings as defaults
enable_ext	--	--	Switches the driver to external enable
enable_int	--	--	Switches the driver to internal enable
enable	--	--	Enables the driver (when internal enable is used)
disable	--	--	Disables the driver (when internal enable is used)
gtemp	--	--	Returns the actual measured PCB temperature in °C
gtempphys	--	--	Returns the re-enable temperature after an overtemperature shutdown in °C

Command	Parameter	Answer	Description
gtempoff	--	--	Returns the shutdown temperature in °C
sp	<channel> <P>	--	Sets the parameter of the P-controller of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
gp	<channel>	P value	Returns the parameter of the P-controller of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
gpmin	--	minimum P value	Returns the minimum parameter of the P-controller
gpmax	--	maximum P value	Returns the maximum parameter of the P-controller
si	<channel> <I>	--	Sets the parameter of the I-controller of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
gi	<channel>	I value	Returns the parameter of the I-controller of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
gimin	--	minimum I value	Returns the minimum parameter of the I-controller
gimax	--	maximum I value	Returns the maximum parameter of the I-controller
sd	<channel> <D>	--	Sets the parameter of the D-controller of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
gd	<channel>	D value	Returns the parameter of the D-controller of the given channel. Note: In single channel mode the first parameter <channel> must not be set!
gdmin	--	minimum D value	Returns the minimum parameter of the D-controller
gdmax	--	maximum D value	Returns the maximum parameter of the D-controller
grntc	< input >	resistance in Ohm	Returns the configured resistance of the given input channel (NTC)
srntc	<input> <R>	--	Sets the resistance of the selected input to the given value
grntcmin	--	minimum resistance in Ohm	Returns the minimum resistance value of the connected NTC

Command	Parameter	Answer	Description
grntcmax	--	maximum resistance in Ohm	Returns the maximum resistance value of the connected NTC
gbntc	< input >	B value	Returns the configured B value of the given input channel
sbntc	<input> <B>	--	Sets the B value of the selected input to the given value
gbntcmin	--	minimum B value	Returns the minimum B value of the connected NTC
gbntcmax	--	maximum B value	Returns the minimum B value of the connected NTC
gtntc	< input >	temperature in °K	Returns the configured norm temperature of the given input channel
stntc	<input> <T>	--	Sets the norm temperature of the selected input to the given value
gtntcmin	--	minimum temperature in °K	Returns the minimum norm temperature of the connected NTC
gtntcmax	--	maximum temperature in °K	Returns the maximum norm temperature of the connected NTC
grptc	< input >	resistance in Ohm	Returns the configured resistance of the given input channel (PTC)
srptc	<input> <R>	--	Sets the resistance of the selected input to the given value (PTC)
grptcmin	--	resistance in Ohm	Returns the minimum resistance value of the connected PTC
grptcmax	--	resistance in Ohm	Returns the maximum resistance value of the connected PTC
gmaxcur	<channel>	current in A	Returns the configured maximum output current of the given controller. Note: In single channel mode the first parameter <channel> must not be set!
smaxcur	<channel> <current>	--	Sets the maximum current of the selected channel to the given value. Note: In single channel mode the first parameter <channel> must not be set!
gmaxcurmin	--	maximum current in A	Returns the minimum output current of the TEC
gmaxcurmax	--	minimum current in A	Returns the maximum output current of the TEC
ginput	<channel>	selected input	Returns the current input number of the given channel. 0: NTC 1 1: NTC 2 3: PTC 1 4: PTC 2 Note: In single channel mode the first parameter <channel> must not be set!

Command	Parameter	Answer	Description
sinput	<channel> <input>	--	Sets the input of the selected channel to the given value: 0: NTC 1 1: NTC 2 3: PTC 1 4: PTC 2 The ENABLE pin must be pulled low when using this command. Otherwise the new input will not be used. Note: In single channel mode the first parameter <channel> must not be set!
gregler	<channel>	see text	This command prints out the internal controller values. The format is: <setpoint> tab <actual value> tab <err> tab <err sum> tab <previous err> tab <duty cycle in A>

### If an Error occurs

If an error occurs during operation, the pulse output is switched off and the return value of a command is no longer “0” or “1” but “10” or “11”. Errors have to be acknowledged with a toggle of the ENABLE signal, otherwise switching on again of TEC output is not possible. For more details see the description of the ERROR register.

To retrieve the error, use the **gerror** command for the content of the ERROR register or the **geretxt** command for a human readable form.

## The PicoLAS Protocol

The following section describes the structure and possible commands of the PicoLAS protocol.

### Structure

Each transmission consists of 12 bytes – called a frame as follows – which must be sent consecutively. Otherwise the system times out and the transmission must start again from the beginning.

A frame has a fixed structure. The first two bytes describe the command, the following eight bytes the parameters, followed by one reserved byte and one checksum byte. The checksum is calculated out of the first 11 bytes which are linked by a bitwise XOR.

Thus a frame has the following structure:

Byte	Meaning
1	Bit 8-15 of the command
2	Bit 0-7 of the command
3	Bit 56-63 of the parameter
4	Bit 48-55 of the parameter
5	Bit 40-47 of the parameter
6	Bit 32-39 of the parameter
7	Bit 24-31 of the parameter
8	Bit 16-23 of the parameter
9	Bit 8-15 of the parameter
10	Bit 0-7 of the parameter
11	Reserved, always 0x00
12	Checksum

A properly received frame must be acknowledged by the recipient with an answer, which is also a frame. If the acknowledgement does not occur the command has not been processed and the sending procedure should be repeated.

If the recipient recognizes the command as valid, but not the parameters, then it will answer with an ILGLPARAM (0xFF12) as command.

In the case that the recipient receives an invalid command it will answer with UNCOM (0xFF13).

If a faulty checksum is recognized, then the answer is RXERROR (0xFF10). If this error occurs often, the connection should be checked.

Using the REPEAT (0xFF11) command the recipient can instruct the sender to send the most recent frame again.

## General Commands

The following list contains an overview of the general commands which are supported by every product from PicoLAS which makes use of this protocol. The explanation of the individual commands is given further below.

Command Name	Sent Frame		Answer Frame	
	Command	Parameter	Command	Parameter
<b>PING</b>	0xFE01	0	0xFF01	0
<b>IDENT</b>	0xFE02	0	0xFF02	ID
<b>GETHARDVER</b>	0xFE06	0	0xFF06	Version
<b>GETSOFTVER</b>	0xFE07	0	0xFF07	Version
<b>GETSERIAL</b>	0xFE08	0 ... 20	0xFF08	Refer to description
<b>GETIDSTRING</b>	0xFE09	0 ... 20	0xFF09	Refer to description

### **PING**

This command is used to determine the presence of a connected driver and to initialize its interface. It does not change any registers. The command parameter is always 0, the answer parameter too.

### **IDENT**

This command is used to determine the device ID of an attached recipient. It has no effect on the condition of the recipient. The parameter is always 0. The answer contains the ID.

### **GETHARDVER**

Instructs the driver to send back the version number of its hardware. The parameter is always 0. The answer contains the hardware version number. The format of the answer is: 0x000000<major><minor><revision>. In other words: one byte for each of the three elements of the version number.

As example, version 1.2.3 has the parameter 0x000000010203.

### **GETSOFTVER**

Instructs the driver to send back the version number of its firmware. The parameter is always 0.

The answer contains the software version of the recipient. The format of the answer is: 0x000000<major><minor><revision>. In other words: one byte for each of the three elements of the version number.

As example, version 2.3.4 has the parameter 0x000000020304.

### **GETSERIAL**

Instructs the driver to send back its serial number. If 0 is sent as parameter, the answer contains the number of (ASCII) digits of the serial number; otherwise the respective position of the serial number is sent in ASCII format.

### **GETIDSTRING**

Instructs the driver to send back its name. If 0 is sent as parameter, the answer contains the number of digits of the string, otherwise the respective position of the serial number is sent in ASCII format.



## Commands for the Driver

The following table contains a list of the commands which the TEC supports in addition to the generally applicable commands. An explanation of the individual commands and its parameters follows afterwards.

Command	Sent Frame		Received Frame	
	Command	Parameter	Command	Parameter
<b>GETTEMP</b>	0x0001	--	0x0113	16 bit signed integer
<b>GETTEMPOFF</b>	0x0002	--	0x0113	16 bit signed integer
<b>GETTEMPHYS</b>	0x0004	--	0x0113	16 bit signed integer
<b>GETSOLL</b>	0x0010	0	0x0101	32 bit value
<b>GETSOLLMIN</b>	0x0011	0	0x0101	32 bit value
<b>GETSOLLMAX</b>	0x0012	0	0x0101	32 bit value
<b>SETSOLL</b>	0x0013	Refer to description	0x0101	32 bit value
<b>GETLSTAT</b>	0x0020	0	0x0103	32 bit value
<b>GETTEMP</b>	0x001A	refer to description	0x0102	32 bit value
<b>GETERROR</b>	0x0021	0	0x0114	32 bit value
<b>GETREGS</b>	0x0022	0	0x0105	64 bit value
<b>SETLSTAT</b>	0x0023	Refer to description	0x0103	32 bit value
<b>CLEARERROR</b>	0x0024	0	0x0104	0
<b>SAVEDEFAULTS</b>	0x0027	0	0x0112	0
<b>LOADDEFAULTS</b>	0x0028	0	0x0112	0
<b>GETKPMIN</b>	0x0040	0	0x010A	32 bit value
<b>GETKPMAX</b>	0x0041	0	0x010A	32 bit value
<b>GETKP</b>	0x0042	Refer to description	0x010A	32 bit value
<b>SETKP</b>	0x0043	Refer to description	0x010A	32 bit value
<b>GETKIMIN</b>	0x0044	0	0x010B	32 bit value
<b>GETKIMAX</b>	0x0045	0	0x010B	32 bit value
<b>GETKI</b>	0x0046	Refer to description	0x010B	32 bit value
<b>SETKI</b>	0x0047	Refer to description	0x010B	32 bit value
<b>GETKDMIN</b>	0x0048	0	0x010C	32 bit value
<b>GETKDMAX</b>	0x0049	0	0x010C	32 bit value
<b>GETKD</b>	0x004A	Refer to description	0x010C	32 bit value
<b>SETKD</b>	0x004B	Refer to description	0x010C	32 bit value
<b>GETRNTCMIN</b>	0x0050	0	0x010D	32 bit value
<b>GETRNTCMAX</b>	0x0051	0	0x010D	32 bit value
<b>GETRNTC</b>	0x0052	Refer to description	0x010D	32 bit value
<b>SETRNTC</b>	0x0053	Refer to description	0x010D	32 bit value
<b>GETBNTCMIN</b>	0x0054	0	0x010E	32 bit value

Command	Sent Frame		Received Frame	
	Command	Parameter	Command	Parameter
<b>GETBNTCMAX</b>	0x0055	0	0x010E	32 bit value
<b>GETBNTC</b>	0x0056	Refer to description	0x010E	32 bit value
<b>SETBNTC</b>	0x0057	Refer to description	0x010E	32 bit value
<b>GETTNTCMIN</b>	0x0058	0	0x010F	32 bit value
<b>GETTNTCMAX</b>	0x0059	0	0x010F	32 bit value
<b>GETTNTC</b>	0x005A	Refer to description	0x010F	32 bit value
<b>SETTNTC</b>	0x005B	Refer to description	0x010F	32 bit value
<b>GETRPTCMIN</b>	0x005C	0	0x0110	32 bit value
<b>GETRPTCMAX</b>	0x005D	0	0x0110	32 bit value
<b>GETRPTC</b>	0x005E	Refer to description	0x0110	32 bit value
<b>SETRPTC</b>	0x005F	Refer to description	0x0110	32 bit value
<b>GETIMAX</b>	0x0060	Refer to description	0x0111	32 bit value
<b>GETIMAXMAX</b>	0x0061	0	0x0111	32 bit value
<b>GETIMAXMIN</b>	0x0062	0	0x0111	32 bit value
<b>SETIMAX</b>	0x0063	Refer to description	0x0111	32 bit value
<b>GETREGLERPARAM</b>	0x0070	Refer to description	0x0115	32 bit value

## Description of the individual Commands

### **GETTEMP**

The return value contains the actual measured PCB temperature in steps of 0.1 °C.

### **GETTEMPOFF**

The return value contains the overtemperature shutdown value in steps of 0.1 °C.

### **GETTEMPHYS**

The return value contains the temperature at which the driver will be re-enabled after an overtemperature shutdown. The value is measured in steps of 0.1 °C.

### **GETSOLL**

This command returns the setpoint temperature of the given channel [0|1]. The lower 32 bit of the return value represents the setpoint in a 32 bit integer format in steps of 0.01 °C.

### **GETSOLLMIN**

The return value contains the minimum setpoint temperature in steps of 0.01 °C.

### **GETSOLLMAX**

The return value contains the maximum setpoint temperature in steps of 0.01 °C.

### **SETSOLL**

This command sets the current setpoint to the given value in steps of 0.01°C. The bits 56 ... 63 select the channel, the bits 0 ... 31 represents the setpoint value in a 32 bit signed integer format.

The setpoint value must be within the borders defined by the minimum and maximum temperature values given by the *GETSOLLMIN* and *GETSOLLMAX* command.

The return value contains the actual setpoint value (See *GETSOLL* command).

### **GETTEMP**

This command returns the actual measured temperature of the given channel. It is measured in steps of 0.001°C

### **GETLSTAT**

This command returns the value of the LSTAT register. For a complete description of this register see below.

### **GETERROR**

This command returns the value of the ERROR register. For a complete description of this register see below.

### **GETREGS**

This command returns the value of the LSTAT and ERROR register combined in a single 64 bit value. The lower 32 bit represents the LSTAT register, the upper 32 bit the ERROR register.

### **SETLSTAT**

This command sets the LSTAT register to the given value. The return value contains the new register value.

### **CLEARERROR**

This command clears a part of the internal ERROR register. For a detailed description of the ERROR register see below.

### **SAVEDEFAULTS**

This command saves all settings to an internal EEPROM.

### **LOADDEFAULTS**

This command loads previously saved settings into the driver.

### **GETKPMIN**

The return value contains the minimum value of the P-controller.

### **GETKPMAX**

The return value contains the maximum value of the P-controller.

**GETKP**

This command returns the value of the P-controller of the given channel [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format.

**SETKP**

This command sets the value of the P-controller to the given data. The bits 56 ... 63 select the channel, the bits 0 ... 31 represents the setpoint value in a 32 bit signed integer format. The setpoint value must be within the borders defined by the minimum and maximum temperature values given by the *GETKPMIN* and *GETKPMAX* command. The return value contains the actual setpoint value (See *GETKP* command).

**GETKIMIN**

The return value contains the minimum value of the I-controller.

**GETKIMAX**

The return value contains the maximum value of the I-controller.

**GETKI**

This command returns the value of the I-controller of the given channel [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format.

**SETKI**

This command sets the value of the I-controller to the given data. The bits 56 ... 63 select the channel, the bits 0 ... 31 represents the setpoint value in a 32 bit signed integer format. The setpoint value must be within the borders defined by the minimum and maximum temperature values given by the *GETKIMIN* and *GETKIMAX* command. The return value contains the actual setpoint value (See *GETKI* command).

**GETKDMIN**

The return value contains the minimum value of the D-controller.

**GETKDMAX**

The return value contains the maximum value of the D-controller.

**GETKD**

This command returns the value of the D-controller of the given channel [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format.

**SETKD**

This command sets the value of the D-controller to the given data. The bits 56 ... 63 select the channel, the bits 0 ... 31 represents the setpoint value in a 32 bit signed integer format. The setpoint value must be within the borders defined by the minimum and maximum temperature values given by the *GETKDMIN* and *GETKDMAX* command. The return value contains the actual setpoint value (See *SETKD* command).

**GETRNTCMIN**

The return value contains the minimum possible resistance value of the NTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format.

**GETRNTCMAX**

The return value contains the maximum possible resistance value of the NTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format.

**GETRNTC**

This command returns the configured NTC resistance of the given NTC input [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format.

**SETRNTC**

This command sets the NTC resistance to the given data. The bits 56 ... 63 select the input NTC channel, the bits 0 ... 31 represents the resistance value in a 32 bit signed integer format.

The setpoint value must be within the borders defined by the minimum and maximum values given by the *GETRNTCMIN* and *GETRNTCMAX* command.

The return value contains the actual resistance (See *SETRNTC* command).

**GETBNTCMIN**

The return value contains the minimum possible B-value of the NTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format.

### **GETBNTCMAX**

The return value contains the maximum possible B-value of the NTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format.

### **GETBNTC**

This command returns the configured B-value of the given NTC input [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format.

### **SETBNTC**

This command sets the NTC B-value to the given data. The bits 56 ... 63 select the input NTC channel, the bits 0 ... 31 represents the B-value in a 32 bit signed integer format. The setpoint value must be within the borders defined by the minimum and maximum values given by the *GETBNTCMIN* and *GETBNTCMAX* command. The return value contains the actual resistance (See *GETBNTC* command).

### **GETTNTCMIN**

The return value contains the minimum possible norm temperature of the NTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format in steps of 0.1 °K.

### **GETTNTCMAX**

The return value contains the maximum possible norm temperature of the NTC channels. The lower 32bit of the return value represents the data in a 32 bit integer format in steps of 0.1 °K.

### **GETTNTC**

This command returns the configured norm temperature of the given NTC input [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format in steps of 0.1 °K.

### **SETTNTC**

This command sets the NTC norm temperature to the given data. The bits 56 ... 63 select the input NTC channel, the bits 0 ... 31 represents the norm temperature in a 32 bit signed integer format in steps of 0.1 °K. The setpoint value must be within the borders defined by the minimum and maximum values given by the *GETBNTCMIN* and *GETBNTCMAX* command. The return value contains the actual resistance (see *GETBNTC* command).

### **GETRPTCMIN**

The return value contains the minimum possible resistance value of the PTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format.

### **GETRPTCMAX**

The return value contains the maximum possible resistance value of the PTC channels. The lower 32 bit of the return value represents the data in a 32 bit integer format.

### **GETRPTC**

This command returns the configured resistance of the given PTC input [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format.

### **SETRPTC**

This command sets the PTC resistance to the given data. The bits 56 ... 63 select the input NTC channel, the bits 0 ... 31 represents the resistance value in a 32 bit signed integer format. The setpoint value must be within the borders defined by the minimum and maximum values given by the *GETRPTCMIN* and *GETRPTCMAX* command. The return value contains the actual resistance (See *GETRPTC* command).

### **GETIMAXMIN**

The return value contains the minimum possible output current of the TEC. The lower 32 bit of the return value represents the data in a 32 bit integer format in steps of 0.01A.

### **GETIMAXMAX**

The return value contains the maximum possible output current of the TEC. The lower 32 bit of the return value represents the data in a 32 bit integer format in steps of 0.01A.

### **GETIMAX**

This command returns the configured maximum output current of the TEC of the given channel [0|1]. The lower 32 bit of the return value represents the data in a 32 bit integer format in steps of 0.01A.

### **SETIMAX**

This command sets the PTC output current of the TEC to the given data. The bits 56 ... 63 select the TEC channel, the bits 0 ... 31 represents the resistance value in a 32 bit signed integer format in steps of 0.01 A.

The setpoint value must be within the borders defined by the minimum and maximum values given by the *GETIMAXMIN* and *GETIMAXMAX* command.

The return value contains the actual maximum current (See *GETIMAX* command).

### **GETREGLERPARAM**

This command returns the actual parameters used by the PIC controller. The given parameter selects the data which is to be returned. The first byte (most significant byte) selects the channel while the second byte selects the value. Possible values are:

<b>Parameter</b>	<b>Return value</b>
0	reserved
1	reserved
2	reserved
3	calculated error (used by P-controller)
4	summed error (used by I-controller)
5	previous error (used by D-controller)
6	calculated TEC duty cycle

## Register Description

### Description of the STAT Register

The following list contains a description of the individual STAT bits. These can be read with GETSTAT and written with SETSTAT. With SETLSTAT a complete 32 bit word must always be written. Thus, to change individual bits the register must first be read out with GETSTAT, then the desired bits changed and then with SETSTAT passed again to the TEC.

Bit	Name	Read/Write	Meaning
0	CH0_TEC_ON	Read/write	Switches the control loop of channel 0 on or off. Note that the ENABLE pin of the BOB connector must also be set high in order to enable the TEC.
1-2	CH0_INPUT	Read/Write	These bits select the input for channel 0. Possible values are: 0 : NTC input 0 1 : NTC input 1 2 : PTC input 0 3 : PTC input 1
3	CH1_TEC_ON	Read/Write	Switches the control loop of channel 1 on or off. Note that the ENABLE pin of the BOB connector must also be set high in order to enable the TEC.
4-5	CH1_INPUT	Read/Write	These bits select the input for channel 1. Possible values are: 0 : NTC input 0 1 : NTC input 1 2 : PTC input 0 3 : PTC input 1
6	ENABLE_OK	Read	This bit indicates the state of the ENABLE pin of the BOB
7	TEC_OK	Read	This bit indicates the state of the TEC_OK pin of the BOB
8	reserved	Read	reserved
9	DEFAULT_ON_PWRON	Read/Write	If this bit is set to “1”, the TEC will load the previously saved settings every time the power is toggled.
10	SWITCH	Read/Write	This bit indicates the state of the mode switch. A value of “1” means that the TEC is in single channel mode.
11	ENABLE_EXT	Read/Write	Determines weather the driver is enabled via the ENABLE pin of the BOB connector (“1”) or via the ENABLE bit in this register(“0”)
11-31	Reserved	Read	Reserved

## Description of the ERROR Register

The following list contains a description of the individual bits of the ERROR register. A “1” as a bit leads to a deactivation of the drivers output.

The bits 0 and 1 are cleared every time the ENABLE pin is set low or when the CLEARERROR command is used.

The bits 4 and 6 are set during the power on self test. They can not be cleared by the ENABLE pin or the CLEARERROR command. If one of these bits is set, the supply voltage should be switch off and on again. If the error persists, the PL-TEC 2-1024 needs to be repaired.

Bit	Name	R/W	Meaning
0	DRV_OVERTEMP	Read	Indicates an overtemperature shutdown of the output stage.
1	DRV_FAIL	Read	Indicates a shutdown of the LDP-CW due to a load failure or an overcurrent
2	VCC_FAIL	Read	Indicates that the supply voltage is not within safe operation range.
3	TEK_SWITCHERR	Read	Indicates that the channel mode switch was switched.
4	CRC_DEVDRV_FAIL	Read	Indicates that the internally stored PLB-21 driver is invalid. The driver can still be used, but it is impossible to download the driver into the PLB-21
5	CRC_DEFAULT_FAIL	Read	Indicates that the internal default registers are corrupt. The default values should be re- saved.
6	CRC_CONFIG_FAIL	Read	Indicates that the internal configuration register is corrupt. If this error persists, the LDP-CW needs to be repaired.
7	Reserved	Read	Reserved
8	TEC_ADC_FAIL	Read	This bit indicates an internal error. If this bit persists, TEC needs to be repaired.
9	FAILED_TO_LOAD_DEFAULTS	Read	Indicates that the last attempt to load the default values failed.
10	TEMP_OVERSTEPPED	Read	Indicates that the maximum operating temperature was overstepped.
11	TEMP_HYSTERESIS	Read	Indicates that the driver is cooling down after an overtemperature shutdown. The driver cannot be re-enabled until this bit flips back to “0”.
12	TEMP_WARNING	Read	Indicates that the operation temperature is near the shutdown temperature.
13	ENABLE_DURING_POWERON	Read	Indicates then the ENABLE pin of the BOB connector was high during power-on. Only valid if ENABLE_EXT the LSTAT register is set to “1”
14	ENABLE_DURING_ENCHANGE	Read	Indicates that the ENBLE pin of the BOB connector was high while setting ENABLE_EXT to “1” in the ENABLE register.
15	Reserved	Read	Reserved
16-31	Reserved	Read	Reserved