User Manual

BFS-VDIG 03

-preliminary-

PicoLAS

PicoLAS GmbH
Burgstr. 2
52146 Würselen
Germany

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

Rev. 2201

Table of Contents

How to get started

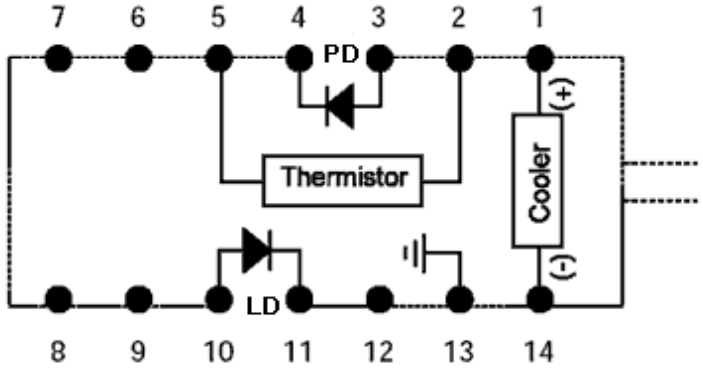
The basic settings (temperature, BIAS current etc.) can be adjusted using the RS-232 or the PLB-21. You need an additional arbitrary waveform generator (AWG) which has to be connected to the Input MMCX-Terminal.
There is no AWG on board of the seed driver!

The PCB is very thin. Mechanical force has to be avoided to prevent damage. Mechanical damages are not covered by warranty.

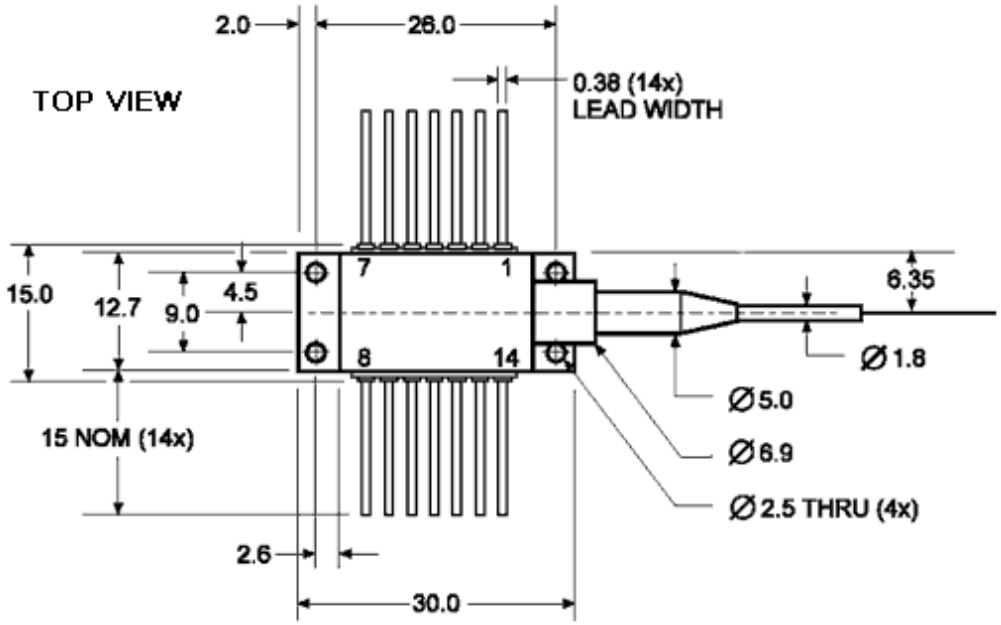
Step	What to do	Check
1	Assemble the driver onto an appropriate heat sink.	Refer to section “Cooling”.
2	Solder the Laser diode into the driver	Refer to section “Required laser diode”.
3	Connect GND and +5 V to the power connector while the power supply is disabled.	Refer to section “Power Supply” for connector layout.
4	Connect pulse generator on input connector (no pulse before power supply is enabled).	Refer to section “Input” for allowed signal parameters.
5	Connect PC via D-SUB9 to micro match cable or connect to a PC.	Refer to cable drawings.
6	Turn the power supply on.	The green LED 2 should turn on after a few seconds. It indicates that the self-test is successfully passed.
7	Communication with the driver must be possible at this point.	Check the +5 V TEC supply for this step with PLB-21 or the response to “ps” with the terminal interface.
8	Configure the TEC parameters as required	Refer to section “TEC controller”
9	Set the pulse data	Refer to section “Generating a pulse shape”
10	Enable external trigger	
11	Check the optical pulse shape	

Required Laser Diode Pinout

The BFS-VDIG 03 is designed for the use with laser diodes in a butterfly package according to the following pin layout:



Pin	Connection
1	TE Cooler (+)
2	Thermistor
3	PD Anode (-)
4	PD Cathode (+)
5	Thermistor
10	LD Anode (+)
11	LD Cathode (-)
14	TE Cooler (-)



The driver is connecting pin 9 & 10 and pin 11 & 12 together.
Pitch of the legs: 2.5 mm.
Distance of the legs to bottom of the chassis: 4.5 mm.

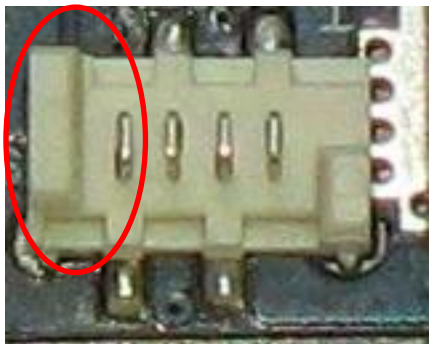
Please shorten the legs to avoid short circuits inside the driver

The maximum internal stray impedance must not exceed ~ 10 nH to achieve best performance

Required electrical Characteristics of the Laser Diode

Parameter	Comment	Min.	Typ.	Max.
Laser diode forward voltage		0 V	1.8 V	2.2V
Back facet monitor reverse voltage			5 V	15 V
Back facet monitor responsivity	5 V reverse voltage	0.5 µW/mA		20 µW/mA (10 µW/mA typ)
Back facet monitor dark current	5 V reverse (bias) voltage			40 nA
Thermistor resistance	At 25 ºC	9.5 kΩ		10.5 kΩ
TEC current	Peak current (short term)			2.0 A
TEC current	Continuous current	0 A	1.0 A	1.5 A
TEC voltage	Depending on type of laser diode			3 V

Description of the RS-232 Connector



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	

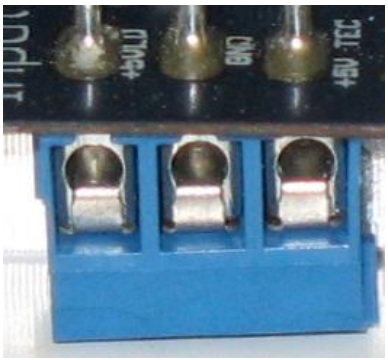
The RS-232 interface can be used with a PC and a terminal program.
Connector type Molex 90327-0304

Description of the Trigger Input

The Trigger is terminated with 50 Ohm and requires a 2.0 V .. 3.3 V rising edge.
The connector is of type MMCX, female [MOLEX - 73415-1001]

Description of the Power Connector

The power connector provides tree screw terminals for supplying the 5 V DC supply voltage.



1 2 3

Pin	Allowed range	Best performance	Destroying limit
1 (µC & AUX)	4.75 V .. 5.25 V	+5.0 V	+6 V
2 (GND)	GND	GND	---
3 (TEC)	4.75 V .. 5.25 V	+5.0 V	+6 V

Power Supply

Any non stabilized power supply capable of driving 5 V DC with no less than 20W output power can be used to operate the driver.

TEC Controller

The driver is equipped with a thermal electric controller (TEC for short). It allows the usage of the integrated thermistor / cooler combination in the connected laser diode and generates a constant temperature for the laser crystal.
It is configurable via RS-232 and allows control over the following parameters:

- Enable / Disable
- Temperature
- Maximum TEC current
- Control loop response time (PID parameters)

The according commands are listed later in this document.

Generating a Pulse Shape

The driver uses an internal data array of 150 data words to generate a current pulse with up to 300ns length. These data values must be set via the RS-232 interface using the **spulscur** command. In order to generate a pulse shorter than 300ns, the data values of the positions not used must be set to 0.

The following example will generate a rising current pulse of 100ns from 0mA to 100 mA:

spulscur 0 0
spulscur 1 2
spulscur 2 4
spulscur 3 6
…
…
…
spulscur 48 98
spulscur 49 100
spulscur 50 0
spulscur 51 0
…
…
…
spulscur 149 0

The pulse output is triggered by a rising edge on the trigger input connector. The signal used must be able to drive at least 2.0V into 50 Ohms.
A signal amplitude of more than 3.3V is not recommended.

Be aware that in the current hardware version the driver will always generate pulses if a trigger is applied even if there are no valid data programmed into it. If the data points are modified, the trigger signal must be disabled.

For testing purposes, the commands **spulscurx** and **spulsrisex** can be used:

The **spulscurx** command takes a current as parameter and will fill the data registers with this value. This will generate a near rectangular pulse of 300ns length.
Example: **spulscurx 100**

The **spulsrisex** command takes two currents as parameters. It will fill the data registers with a linear interpolation between the two values and therefore generate a rising or falling current pulse from current 1 to current 2 and a length of 300ns.
Example: **spulsrisex 0 1000** or **spulsrisex 500 200**

Controlling the Driver via a PC

Introduction

When the driver 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 (binary command, see below)
- "**init**" followed by <Enter> for the text interface

Description of the RS-232 Interface

The driver implements a standard RS-232 interface. It can be connected to a PC using a three-wire connection. For USB connection you need an USB RS-232 adaptor (USB serial adaptor).
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 driver must be completed with a **CR** (Enter). It consists of a command word followed by one or more parameters. 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 1:
The user would like to read out the actual temperature setpoint:
User input: gtsoll<Enter>
Output of the driver: 250<*CR*><*LF*>
00<*CR*><*LF*>

Example 2:
The user would like to set a new setpoint current:
User input: stsoll 270<Enter>
Output of the driver: 270<*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 BFS-VDIG 03

The following table contains a command reference for the BFS-VDIG 03.

Command	Parameter	Description
ghwver	-	returns the hardware version number
gswver	-	returns the software version number
gserial	-	returns the serial number
gname	-	returns the device name
ps	-	prints out all settings
gerrtxt	-	returns the error register in text form
gerr	-	returns the error register as a 32 bit number
glstat	-	returns the laser status register
slstat	32 bit number	sets the laser status register to the given value
gitec	-	returns the actual TEC current in [A]
gttec	-	returns the actual TEC temperature in [°C]
gtist	-	returns the actual laser diode temperature in [°C]
gtsollmin	-	returns the minimal TEC temperature setpoint in [°C]
gtsollmax	-	returns the maximal TEC temperature setpoint in [°C]
gtsoll	-	returns the actual TEC temperature setpoint in [°C]
stsoll	-	sets the TEC temperature setpoint to the given value in [°C]
gkpmin	-	returns the minimal strength of the proportional part of the TEC controller
gkpmax	-	returns the maximal strength of the proportional part of the TEC controller
gkp	-	returns the actual strength of the proportional part of the TEC controller
skp	value	sets the strength of the proportional part of the TEC controller to the given value
gkimin	-	returns the minimal strength of the integral part of the TEC controller
gkimax	-	returns the maximal strength of the integral part of the TEC controller
gki	-	returns the actual strength of the integral part of the TEC controller
ski	value	sets the strength of the integral part of the TEC controller to the given value
gkdmin	-	returns the minimal strength of the differential part of the TEC controller
gkdmax	-	returns the maximal strength of the differential part of the TEC controller
gkd	-	returns the actual strength of the differential part of the TEC
skd	value	sets the strength of the differential part of the TEC controller to the given value

Command	Parameter	Description
gimaxmin	-	returns the minimum current limiter value in [A]
gimaxmax	-	returns the maximum current limiter value in [A]
gimax	-	returns the actual current limiter value in [A]
simax	current in [A]	sets the TEC current limiter to the given value, e.g. *0.5*
tenable	-	enables the TEC controller
tdisable	-	disables the TEC controller
gpulscurmin	-	returns the minimum valid pulse current value
gpulscurmax	-	returns the maximum valid pulse current value
gpulscur	pos	returns the pulse current value at the given position
spulscur	<pos> <mA>	sets the pulse current at the given position to the given value
spulscurx	<mA>	sets all pulse current data points to the given value
gpulsposmax	--	returns the maximum valid pulse data position value
gpulsdata	--	returns all pulse current data values
spulscurx	Current in mA	creates a static pulse shape with the given current
spulsrisex	<c1 in mA> <c2 in mA>	creates a linear pulse shape ranging from current c1 to current c2
gbias	*bias in mA*	*get the current bias in [mA]*
sbias	*bias in 0.1mA*	*set the bias current in mA/10, e.g. 100 → 10mA*
gbiasmin		*get the minimum bias current*
gbiasmax		*get the maximum bias current*
gvol	*voltage in V*	*get the current voltage for the laser diode*
svol	*voltage in 0.1V*	*set the voltage for the laser diode, e.g. 120 → 12V*
gvolmin		*get the minimum voltage*
gvolmax		*get the maximum voltage*

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 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**
PING	0xFE01	0	0xFF01	0
IDENT	0xFE02	0	0xFF02	ID
GETHARDVER	0xFE06	0	0xFF06	Version
GETSOFTVER	0xFE07	0	0xFF07	Version
GETSERIAL	0xFE08	0 … 255	0xFF08	Refer to description
GETIDSTRING	0xFE09	0 … 255	0xFF09	Refer to description

In addition to these commands there are some answers, which can be given by every command:

Answer	Answer Frame	
	Command	**Parameter**
RXERROR	0xFF10	0
REPEAT	0xFF11	0
ILGLPARAM	0xFF12	0
UNCOM	0xFF13	0

RXERROR
If a frame is repeated four times and still broken this answer will be sent.

REPEAT
The last frame was received in a broken state. The transmission must be repeated. This can be up to four times before a RXERROR will be send.

ILGLPARAM
The parameter of the last frame had an incorrect value.

UNCOM
The command of the last frame is unknown by the device.

Description of the LSTAT Register

The following list contains a description of the individual LSTAT bits. These can be read with the GETLSTAT and written with SETLSTAT command.

Bit	Name	Read/Write	Meaning
0	PULSER_OK	ro	When “1” there is no error pending.
1	DEF_PWRON	r/w	When “1”, the driver loads its default values on start up.
2	SAVE_DEF	w	When set to “1” the driver load its default values. Always read “0”.
3	LOAD_DEF	w	When set to “1” the driver save the current settings as new default values. Always read “0”.
4-31	Reserved	ro	Reserved

Description of the ERROR Register

The following list contains a description of the individual bits of the ERROR register. It can be read with the GETERROR command.

Bit	Name	Read/Write	Meaning
0	CFG_CHKSUM_FAIL	ro	A CRC error was detected in the internal configuration values. Please contact your distributor.
1	PLB_CHKSUM_FAIL	ro	A CRC error was detected in the PLB driver. The driver cannot be used. This does not affect the device but the PLB.
2	DEF_CHKSUM_FAIL	ro	A CRC error was detected in the default values. A re-save of the values should correct this.
3	VCC_LD_FAIL	ro	The measured voltage on the +5 V LD input is not within valid range.
4	VCC_TEC_FAIL	ro	The measured voltage on the +5 V TEC input is not within valid range.
5-31	Reserved	ro	