

User Manual

PEREGRINE API

SCPI communication guide for the PEREGRINE ultrafast laser system via RS485 and Ethernet.



Dear Valued Customer,

Thank you for choosing the PEREGRINE laser system from LaserAtWork GmbH and for placing your trust in our products. We are pleased to support you with this API communication manual.

This document provides the information required to remotely control, monitor, and integrate the PEREGRINE laser system using its SCPI-based communication interface. It describes the supported command set, command and response formats, interface-specific considerations for RS485 and Ethernet communication, and general principles for reliable system integration.

Please read this manual carefully before developing or commissioning a remote-control application. Correct implementation of the commands and communication procedures described in this document is essential for reliable operation and effective monitoring of the laser system.

Keep this document available as a reference during software development, system integration, and service activities. If you have questions, require technical support, or need additional documentation, please contact LaserAtWork GmbH or your authorized representative. Our team will be pleased to assist you throughout the lifetime of your product.

Best regards,
The LaserAtWork Team

Contact

LaserAtWork GmbH
Buendenstrasse 6a
AT-6973 Hoechst
Austria

Mobile: +43-660-655-5875
eMail: info@laseratwork.com
Web: <https://www.laseratwork.com>

Table of Contents

1	PEREGRINE CONTROL API (SERIAL INTERFACE)	4
1.1	STATE MACHINE	4
1.2	COMMUNICATION BASICS	5
1.3	AVAILABLE COMMANDS	5
1.4	THE REPRATE? RESPONSE	6
1.5	THE STATUS? RESPONSE	7
1.6	SYSTEM STATES	8
1.7	ERROR HANDLING	9
1.8	WARNING AND ERROR CODES (COMING SOON)	9

Table of Figures

FIGURE 1:	STATE MACHINE OF THE PEREGRINE LASER SYSTEM.	4
-----------	--	---

Table of Tables

TABLE 1:	AVAILABLE COMMANDS	6
TABLE 2:	THE REPRATE? RESPONSE	6
TABLE 3:	THE STATUS? RESPONSE	7
TABLE 4:	SYSTEM STATES	8
TABLE 5:	WARNING AND ERROR CODE	9

1 PEREGRINE Control API (Serial Interface)

The PEREGRINE ultrafast laser system provides two independent serial communication interfaces for remote control and monitoring:

- **RS485** (half-duplex, differential signaling for long-distance or noise-sensitive environments)
- **Ethernet** (optional)

Both interfaces support the same command set, based on the **SCPI** (*Standard Commands for Programmable Instruments*) standard. SCPI is a widely adopted protocol in instrumentation and control, offering human-readable ASCII commands and structured responses.

Additionally, both Ethernet and RS485 interfaces can be used simultaneously; however, commands are processed sequentially in the order received.

1.1 State Machine

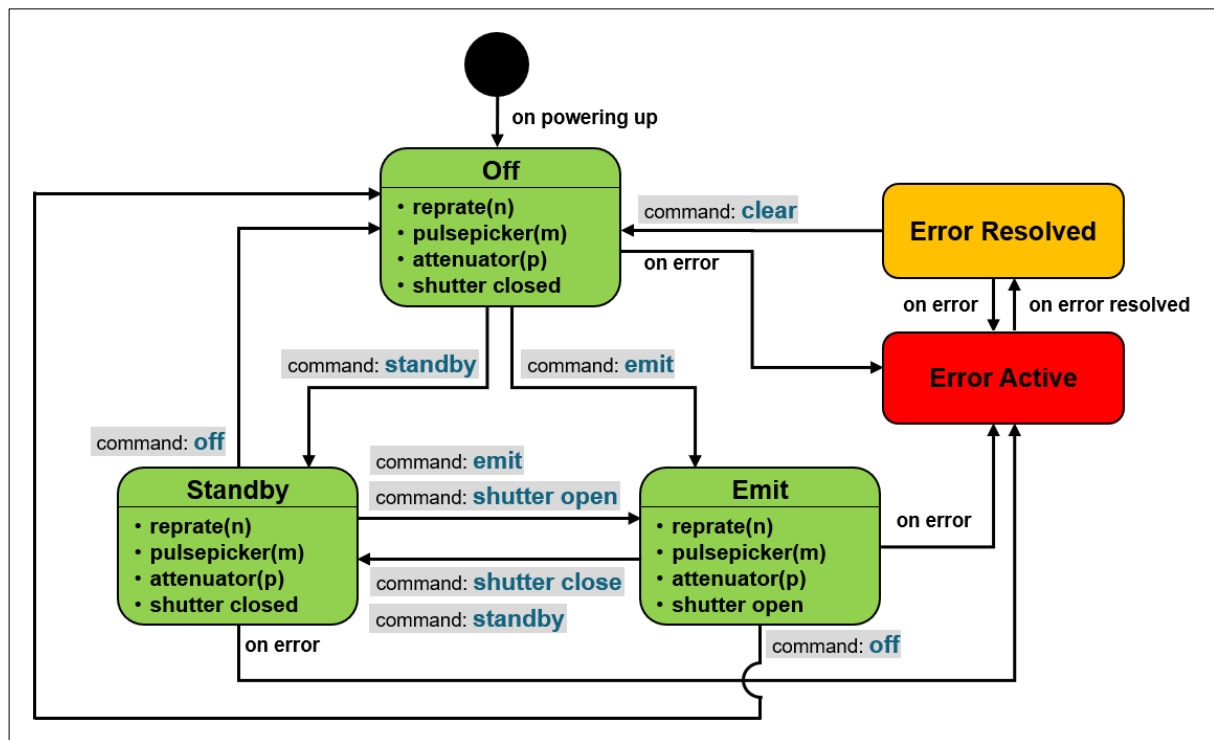


Figure 1: State machine of the PEREGRINE laser system.

1.2 Communication Basics

- **Baud Rate:** 115200 (RS485 default)
- **Data Format:** 8 data bits, no parity, 1 stop bit (8N1)
- **Encoding:** ASCII text
- **Termination:** Each command must be terminated with a newline character (`\n`) or carriage return + newline (`\r\n`).
- **Responses:** The system responds with either `OK` or `FAIL`, optionally followed by data.

The general response format is:

`OK [data]`

`FAIL [description]`

If an unrecognized command is sent, the system responds with:

`FAIL unknown command`

1.3 Available Commands

All commands are **case-insensitive** but are conventionally written in lowercase.

Command	Parameter	Description	Response Format
<code>off</code>	<code>none</code>	Turns the system off (<i>available in both standby and emit states</i>).	<code>OK</code>
<code>standby</code>	<code>none</code>	Turns the system on, <u>BUT</u> keeps the shutter closed. Therefore, no laser light will be emitted (<i>available in both off and emit states</i>).	<code>OK</code>
<code>emit</code>	<code>none</code>	Turns the system on (<i>if not already on</i>) <u>AND</u> opens the fast shutter via the shutter open command (<i>available in both off and standby states</i>).	<code>OK</code>
<code>reprate n</code>	<code>n</code> : integer value <code>n</code> ∈ {1,..., 10}	Use the reprate command followed by a space and an integer <code>n</code> from 1 to 10, where <code>n</code> selects the <code>n</code> -th repetition rate from the internal lookup table.	<code>OK</code>
<code>pulsepicker m</code>	<code>m</code> : integer value <code>m</code> ≥ 1	Use the pulsepicker command followed by a space and an integer <code>m</code> ≥ 1, where <code>m</code> specifies that every <code>m</code> -th pulse is selected from the chosen repetition rate.	<code>OK</code>
<code>attenuator p</code>	<code>p</code> : floating-point value in percent <code>p</code> ∈ [0,..., 100]	Use the attenuator command followed by a space and a floating-point percentage value <code>p</code> between 0 and 100, where <code>p</code> = 0 corresponds to no output power and <code>p</code> = 100 corresponds to full output power.	<code>OK</code>
<code>shutter open</code>	<code>none</code>	Opens the fast shutter; the output power is thereby defined by the attenuator p command	<code>OK</code>
<code>shutter close</code>	<code>none</code>	Closes the fast shutter	<code>OK</code>

clear	none	Clears a resolved error and transitions the system back to normal operation.	OK
status?	none	Returns current system status and parameters in JSON dictionary format.	OK {<JSON>}
off?	none	Queries if the system is currently in the off state.	OK true or OK false
standby?	none	Queries if the system is currently in the standby state.	OK true or OK false
emit?	none	Queries if the system is currently in the emit state.	OK true or OK false
reprate?	none	Queries both the actual and the available repetition rates of the system.	OK {<dict>}
pulsepicker?	none	Queries the pulsepicker value m that has been set by the command pulsepicker m	OK m
attenuator?	none	Queries the attenuator value p that has been set by the command attenuator p	OK p
shutter?	none	Queries if the shutter of the system is open or closed	OK open or OK closed

Table 1: Available commands

1.4 The **reprate?** Response

The command **reprate?** returns the current repetition rate of the laser system, followed by all available repetition rate options presented as an indexed list in dictionary-style format.

Index	Dictionary Key	Description	Example Key Value
0	current_reprate	Current repetition rate	2 MHz
1	reprate1	First available repetition rate option	500 kHz
2	reprate2	Second available repetition rate option	1 MHz
3	reprate3	Third available repetition rate option	10 MHz
...	...	...	...
10	reprate10	10 th available repetition rate option	50 MHz

Table 2: The **reprate?** response

1.5 The status? Response

The status? query provides detailed system information as a JSON object. Fields include:

Field	Type	Description
state	string	Current system state ("off", "standby", "emit", "error_active", "error_resolved")
state_ready	boolean	Indicates when system has reached the state selected by the customer; true if state has successfully passed all internal checks and has reached equilibrium, false otherwise.
error	null or array	null if no error; otherwise [code, category, description]. Please refer to Error! Reference source not found. for a complete list of the available codes and categories.
warnings	array	Array of warnings, each warning is encoded via [code, category, description]. Please refer to Error! Reference source not found. for a complete list of the available codes and categories.
supply_voltage_measured	float (V)	Measured supply voltage.
ambient_temperature_measured	float (°C)	Measured ambient temperature.
frequency_measured	float (MHz)	Measured repetition rate of laser system.
interlock_closed	boolean	true if interlock circuit is closed; false if open.

Table 3: The status? response

Example:

```
OK {"state": "off", "state_ready": false, "error": null, "warnings": [[3, "TEC Temperature", "TEC Temperature is 27.08C (lower limit at 34.00C)"]], "supply_voltage_measured": 19.96, "ambient_temperature_measured": 27.08, "frequency_measured": 10.0, "interlock_closed": true}
```

Certain operations, such as turning the system on, involve transitions that take several seconds. During these periods, repeated status? queries can be used to monitor progress.



Future firmware releases may introduce additional fields.

1.6 System States

The PEREGRINE system operates in the following states:

State	Description	Valid Commands
Off	System is off. Once the <code>state_ready</code> flag is true, the system is thermally stabilized.	<code>standby</code> , <code>emit</code> , <code>replate n</code> , <code>pulsepicker m</code> , <code>attenuator p</code> , <code>status?</code> , <code>off?</code> , <code>replate?</code> , <code>pulsepicker?</code> , <code>attenuator?</code> , <code>shutter?</code>
Standby	System is fully operational, but shutter remains closed once the <code>state_ready</code> flag is true. NOTE: No laser light is emitted in the standby state.	<code>off</code> , <code>emit</code> , <code>shutter open</code> , <code>replate n</code> , <code>pulsepicker m</code> , <code>attenuator p</code> , <code>status?</code> , <code>standby?</code> , <code>replate?</code> , <code>pulsepicker?</code> , <code>attenuator?</code> , <code>shutter?</code>
Emit	once the <code>state_ready</code> flag is true, the system is fully operational and the shutter will open automatically. NOTE: laser light will be emitted in the Emit state as soon as the <code>state_ready</code> flag is true!	<code>off</code> , <code>standby</code> , <code>shutter close</code> , <code>replate n</code> , <code>pulsepicker m</code> , <code>attenuator p</code> , <code>status?</code> , <code>emit?</code> , <code>replate?</code> , <code>pulsepicker?</code> , <code>attenuator?</code> , <code>shutter?</code>
Error Active	A fault is currently active (hardware or safety condition not met).	<code>status?</code>
Error Resolved	Fault condition has cleared but requires user acknowledgment via <code>clear</code> before normal operation can resume.	<code>clear</code> , <code>status?</code>

Table 4: System States

1.7 Error Handling

- **Active errors** (`error_active` state) indicate a persistent fault; these must be resolved before normal operation can resume.
- **Resolved errors** (`error_resolved` state) indicate that the fault condition is cleared, but the system requires acknowledgment via the `clear` command.
- Sending `clear` in the `error_resolved` state returns the system to its normal operational state.

1.8 Warning and Error Codes (coming soon)

Code	Category	Type	Description

Table 5: Warning and Error Code



Future firmware releases may introduce additional categories.