8 Hummingbird Control API (Serial Interface)

The Hummingbird ultrafast laser oscillator provides two independent serial communication interfaces for remote control and monitoring:

- **USB** (enumerates as a virtual COM port)
- **RS485** (half-duplex, differential signaling for long-distance or noise-sensitive environments)

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

8.1 State Machine

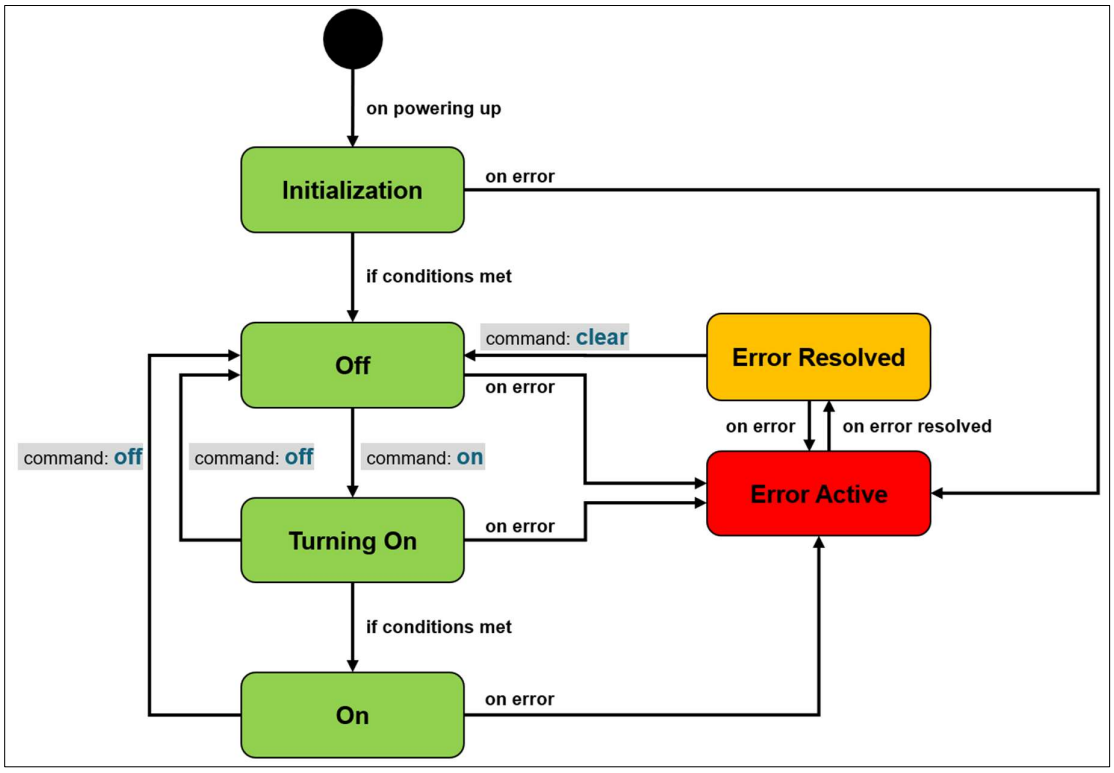

Figure 17: State machine of the HUMMINGBIRD-1030 laser oscillator.

8.2 Communication Basics

- **Baud Rate**: 115200 (RS485 default; USB is virtual COM and ignores baud settings)
- **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`

8.3 Available Commands

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

Command	Description	Response Format
`on`	Turns the system **on** (only valid in **Off** state).	`OK`
`off`	Turns the system **off** (only valid in **On** state).	`OK`
`status?`	Returns current system status and parameters in JSON dictionary format.	`OK {<JSON>}`
`clear`	Clears a **resolved** error and transitions the system back to normal operation.	`OK`
`on?`	Queries if the system is currently in the **On** state.	`OK true` or `OK false`
`off?`	Queries if the system is currently in the **Off** state.	`OK true` or `OK false`

Table 12: Available Commands

8.4 System States

The Hummingbird system operates in the following states:

State	Description	Valid Commands
Initialization	System is waiting until the supply voltage is within specification, the safety interlock is engaged, and the temperature is within the permissible range.	`status?`
Off	System is off. Waiting for user command.	`on`, `status?`, `off?`
Turning On	Transitioning to **On**, performing internal checks and stabilization.	`status?`, `on?`, `off?`
On	System is fully operational.	`off`, `status?`, `on?`
Error Active	A fault is currently active (hardware or safety condition not met).	`status?`
Error Resolved	Fault condition has cleared but requires user acknowledgment via `clear` before normal operation can resume.	`status?`, `clear`

Table 13: System States

8.5 The `status?` Response

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

Field	Type	Description
`state`	string	Current system state (e.g., `"on"`, `"off"`, `"turning_on"`, `"error_active"`, `"error_resolved"`, `"initializing"`).
`error`	null or array	`null` if no error; otherwise `[code, category, description]`. Please refer to Table 9 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 Table 9 for a complete list of the available codes and categories.
`supply_voltage_measured`	float (V)	Measured supply voltage.
`laserdiode_temperature_measured`	float (°C)	Measured laser diode temperature.
`pcb_temperature_measured`	float (°C)	Measured PCB temperature.
`laserdiode_voltage_measured`	float (V)	Measured laser diode voltage.
`laserdiode_current_measured`	float (A)	Measured laser diode current.
`tec_output`	float (%)	-100% to 100%; A negative `tec_output` value indicates heating, whereas a positive value indicates cooling.
`frequency_measured`	float (MHz)	Measured repetition rate of laser system; not available on all systems.
`interlock_closed`	boolean	`true` if interlock circuit is closed; `false` if open.

Table 14: The status? Response

Future firmware releases may introduce additional fields.

Example:

```
OK {"state": "off", "error": null, "warnings": [[3, "TEC Temperature", "TEC Temperature is 27.08C (lower limit at 34.00C)"]], "supply_voltage_measured": 19.96, "laserdiode_temperature_measured": 27.08, "pcb_temperature_measured": 27.89, "laserdiode_voltage_measured": 0.00, "laserdiode_current_measured": -0.00, "tec_output": -37.68, "frequency_measured": 0.0}
```

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.

8.6 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.