LIDAR CLI User Manual

David Roman
droman@ifae.es

IFAE

October 24, 2019

Contents

1 Running licli

1.1 First steps

To be able to run licli the first step is to specify the IP address of the server. This is done by defining LIDAR_ADDR environment variable. For example if the server is running in the local host we can *export LIDAR_ADDR="127.0.0.1"*

If it's the first time we run licli it would ask us to introduce a password. This password is defined in the configuration file of the lidar server software. In case that it's not the first time and wt secret has changed we can have some errors like **"The Token's Signature resulted invalid when verified using the Algorithm"**. If it's the case remove *~/.cache/lidar/0.1/token* and try again.

1.2 Examples

2 Command reference

Commands marked as **debug** are commands used for debugging purposes.

Unless other commands, operation commands are a composition of different commands which are executed sequentially. Sometimes we call it macros.

2.1 LLC

This command contains all the commands related with the low level control board.

2.1.1 Arms

Command to control and monitor the laser arm. Only one action will be executed at a time. If more than one action is specified, only one will be executed, other will be ignored.

- check-node (**debug**): Check communication with node N (N must be either "1" or "2")
- emergency-stop (**debug**): Execute emergency stop
- get-pos: Get current position
- go: Go to current position. Argument must follow the format X:Y (eg: 1000:10000)
- init: Initialize arm
- set-speed (**debug**): Set speed. Argument must follow the format Axis:speed.

2.1.2 DAC

Control dac voltages.

- set-voltage: Set dac voltage. Argument must follow the format dac:voltage (0:100).

2.1.3 Drivers

Get information about the drivers

- get-status: Get a list of drivers status

2.1.4 Hotwind

Controls the hotwind. Only one action will be executed at a time. If more than one action is specified, only one will be executed, other will be ignored.

- error (**debug**): Set error ¿?
- lock (**debug**): Lock
- unlock (**debug**): Unlock

2.1.5 Laser

Commands to control the laser. Only one action will be executed at a time. If more than one action is specified, only one will be executed, other will be ignored.

- check (**debug**): Check laser communication
- fire: Fire the laser
- get-temp: Get laser temperature.
- init: Initialize laser
- pause: Pause the laser
- power: Set power in %
- stop: Stop the laser

2.1.6 Relay

Commands to switch on/off relays. Only one action will be executed at a time. If more than one action is specified, only one will be executed, other will be ignored.

- get-status: Get status
- hotwind-off: Disable hotwind
- hotwind-on: Enable hotwind
- laser-on: Enable laser
- laser-off: Disable laser
- licel-on: Enable laser
- licel-off: Disable laser
- get-status: Get relay status
- set-status: Set relay status. Argument must follow the format idx:status where status is true or false (eg: 0:false)

2.1.7 Sensors

Commands to get sensor information. Only one action will be executed at a time. If more than one action is specified, only one will be executed, other will be ignored.

- converted: Get sensors with a human-readable form
- raw (**debug**): Get raw sensor values

2.2 Motors

2.2.1 Doors

- close: With an argument which must be ”0” or ”1”. ”1” to start closing the doors, ”0” to stop the movement.
- open: With an argument which must be ”0” or ”1”. ”1” to start opening the doors, ”0” to stop the movement.
- status: Print current status of the doors (OPEN, CLOSE or INTERSTATE if it’s something in the middle)

2.2.2 Petals

- close: With an argument which must be ”0” or ”1”. ”1” to start closing the petals, ”0” to stop the movement.
- open: With an argument which must be ”0” or ”1”. ”1” to start opening the petals, ”0” to stop the movement.
- status: Print current status of the petals (CLOSED or UNKNOWN if there is something else)

2.2.3 Telescope

- ga: Get current azimuth position
- gz: Get current zenith position
- home: Go home. If tries to go to the position registered before opening the doors, if it’s not possible it defaults to the hardcoded values, which may not be the correct ones. For this reason it’s important to start the server when the doors are closed.
- sa: Go to the given azimuth encoder position (an unsigned integer)
- sz: Go to the given zenith encoder position (an unsigned integer)

2.3 Monitoring

Monitoring commands

2.3.1 Sensors

Monitoring of sensors.

- humidity: Show readings of humidity values
- env-temperature: Show readings of environment temperatures
- last-value (can be used in combination): Show the last value reading

2.3.2 Motors

Monitoring of the motor monitoring board. Currently it only prints the encoders position.

2.4 Alarms

Waits for alarm messages. Alarms are triggered whenever a sensor exceeds the safe limit. Message are sent in the form of strings informing of the problem.

2.5 Config

Returns current configuration of the server.

2.6 Trace

Command for debugging purposes

2.7 Operation

2.7.1 Acquisition

- disc: Discriminator level. Always required.
- download: Download licel file. Argument is the ID (unsigned integer) of the licel file we want to download.
- shots: Acquire a given number of shots. Argument is an unsigned integer greater than 1.
- start: Start acquisition manually
- stop: Stop acquisition

2.7.2 Telescope

Telescope motors operations. When going to parking position it will first try to go to the last know good value (ie: the position we know that doors can be closed), if it's not possible it will go to the hardcoded values.

- get-azimuth: Get azimuth position
- go-azimuth-parking: Go to azimuth parking position
- go-parking: Go to parking position
- go-zenith-parking: Go zenith parking position
- start-test: Execute telescope tests (move azmiuth and zenith to their maximum, minimum and parking position).
- stop: Stop any telescope movement
- to-max-azimuth: Go to maximum azimuth
- to-max-zenith: Go to maximum zenith
- to-min-azimuth: Go to minimum azimuth

2.7.3 Low level

Low level operations

- arm-align: Move arm to alignment position (first it must be initialized)
- arm-init: Initialize arm
- laser-init: Initialize laser (first it must be powered-on)
- laser-power-off: Power off laser
- laser-power-on Power on laser

- micro-init: Micro initialization sequence. Prepare laser, arm and rump up all DACs
- micro-shutdown: Micro shutdown sequence. Power off laser hotwind and ramp down DACs
- ramp-down: Ramp down all DACs
- ramp-single: Modify voltage of a single DAC. Argument must follow the format idx:voltage.
- ramp-up: Ramp up all DACs

3 Configuration

Configuration files can be found on *~/.config/lidar/0.1/client/*. Currently there is only one file, micro_init_sequence.properties which can be used to defines arm alignment position, pmt voltages and temperature threshold at which the hotwind should be powered on.

4 Troubleshooting