LIDAR CLI User Manual

David Roman
droman@ifae.es

IFAE

October 23, 2019

Contents

1 Running licli **1**

1.1 First steps 1

1.2 Types of commands 1

2 Commands **1**

2.1 LLC 1

2.1.1 Arms 1

2.1.2 DAC 1

2.1.3 Drivers 1

2.1.4 Hotwind 2

2.1.5 Laser 2

2.1.6 Relay 2

2.1.7 Sensors 2

2.2 Motors 3

2.2.1 Doors 3

2.2.2 Petals 3

2.2.3 Telescope 3

2.3 Monitoring 3

2.3.1 Sensors 3

2.3.2 Motors 3

2.4 Operation 4

2.4.1 Acquisition 4

2.4.2 Telescope 4

2.4.3 Low level 4

1 Running licli

1.1 First steps

To be able to run licli we have 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 same computer we can *export LIDAR_ADDR="127.0.0.1"*

1.2 Types of commands

Operation commands can be complex commands (ie: commands that do more than one thing) or checked commands (we check if the result is the expected one). Other commands will simply do the thing without taking care of anything else. In that sense, operation commands are higher level commands and other commands are the low-level commands.

2 Commands

Commands marked as **debug** are commands used for debugging purposes, that is, in the general use case won't be used.

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.

2.4 Operation

2.4.1 Acquisition

2.4.2 Telescope

2.4.3 Low level