LIDAR CLI User Manual - 1.0.0

David Roman
droman@ifae.es

IFAE

November 20, 2019

Contents

1 Running licli **1**
- 1.1 First steps . . . 1
- 1.2 Examples . . . 1
- 1.3 Start up . . . 1
- 1.4 Data acquisition . . . 1
- 1.5 Shutdown . . . 1
- 1.6 Warm up . . . 2
- 1.7 Connect to internal network . . . 2

2 Commands reference **2**
- 2.1 LLC . . . 2
  - 2.1.1 Arms . . . 2
  - 2.1.2 DAC . . . 2
  - 2.1.3 Drivers . . . 2
  - 2.1.4 Hotwind . . . 2
  - 2.1.5 Laser . . . 3
  - 2.1.6 Relay . . . 3
  - 2.1.7 Sensors . . . 3
- 2.2 Motors . . . 4
  - 2.2.1 Doors . . . 4
  - 2.2.2 Petals . . . 4
  - 2.2.3 Telescope . . . 4
- 2.3 Monitoring . . . 4
  - 2.3.1 Sensors . . . 4
  - 2.3.2 Motors . . . 4
- 2.4 Alarms . . . 4
- 2.5 Config . . . 5
- 2.6 Trace . . . 5
- 2.7 Operation . . . 5
  - 2.7.1 Acquisition . . . 5
  - 2.7.2 Telescope . . . 5
  - 2.7.3 Low level . . . 6

3 Configuration **6**

4 Troubleshooting **6**
- 4.1 The Token's Signature resulted invalid... . . . 6
- 4.2 This method is not safe/read-only . . . 6
- 4.3 Where is X,Y,Z file/directory . . . 6

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 jwt 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

Open doors: **./licli motors doors --open**
Get converted sensors values: **./licli llc sensors**
Make 4 shots with disc level 2: **./licli operation acq --shots=4 --disc=2**. Output data will be in *~/.local/share/lidar/0.1*.

1.3 Start up

To start up the system in the standard way execute the "startup" command of 2.7.

If for some reason this fails we can do the same manually with a couple of commands:

- **./licli motors doors --open**
- **./licli motors telescope --sz=600**
- **./licli operation ll --micro-init**
- **./licli motors petals --open**
- **./licli llc laser --init**
- **./licli llc laser --power=5**

1.4 Data acquisition

When an acquisition is made the server stores the data in the licel format and returns an ID, this file can be afterwards downloaded using that identifier. For example to make an acquisition with discriminator at 4 and 100 shots:
./licli operation acq --shots=100 --disc=4 (this will return the file ID, 1234 for example)
./licli operation acq --download=1234 --disc=4

1.5 Shutdown

To shutdown the system in the standard way execute the "shutdown" command (section 2.7 briefly describes what it does).

1.6 Warm up

To shutdown the system in the standard way execute the "warmup" command (section 2.7 briefly describes what it does).

1.7 Connect to internal network

There are two switches. One of them is used for all the devices that are accessible from outside the container, the other contains all devices connected to the internal network. To access this network connect a laptop with an ethernet cable with these settings: mask 255.255.255.0, ip 192.168.127.230 or below. Inside this local network the server is at 192.168.127.253.

2 Commands reference

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

Unless other commands, operation commands (2.7) are complex commands. Those complex commands can be macros (ie: a couple of commands executed sequentially) or control functions which makes use of the other functions.

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 the last read of laser temperature.
- **read-temp**: Read laser temperature. If the laser is not powered it returns the last value, it doesn't return an error. This command should be used with caution. When a lot of read commands are sent at the same time it may return random numbers and the control on server side can be buggy.
- **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. When the temperature reaches the target it will be stopped.
- **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 from low level control board. There are other sensors that this command doesn't read, for example the temperature of the head of the laser.
If no option is specified it will print values in human readable form.

- **raw** (debug): Get raw sensor values

2.2 Motors

2.2.1 Doors

- **close**: Close doors.
- **open**: Open doors
- **stop**: Stop doors.
- **status**: Print current status of the doors (OPEN, CLOSE or INTERSTATE if it's something in the middle)

2.2.2 Petals

- **close**: Close petals.
- **open**: Open petals
- **stop**: Stop petals.
- **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

-- **startup**: Prepare the laser to shoot. Open the doors, power on all subsystems and move telescope to a good position (ie:above tilt).

-- **shutdown**: Power off all subsystems, close petals, move the telescope to parking position and close the doors.

-- **warmup**: Start warm up. Enable hotwind and wait until it gets to the target temperature.

2.7.1 Acquisition

When taking an acquisition by default the data is stored in the licel format in the server HDD. The server by default gives us a file ID of the file of the acquisition. After that if we want we can use that ID to download the file. Files are not download automatically.

-- **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

-- **analog-data**: Save a file with all analog values using space character as separator.

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.

4 Troubleshooting

In case of unknown problems, increase verbosity level to gather more information. To do this define *LIDAR_VERBOSITY* with an integer in the range of [1, 4].

4.1 The Token's Signature resulted invalid...

This happens when the cached token is not correct. To fix it simply remove *~/.cache/lidar/0.1/token* and try again. This way a new token will be generated.

4.2 This method is not safe/read-only

This means that the keylock is on which means that only those commands that don't produce side effects are allowed (ie: read-only commands) for example to read the status, etc. To fix it someone have to go physically and switch off the keylock.
Note that enabling the keylock doesn't kill current running commands

4.3 Where is X,Y,Z file/directory

This program follow the XDG Base Directory specification. The full specification may be found on `https://standards.freedesktop.org/basedir-spec/basedir-spec-latest.html`

In short: config files are in *~/.config/lidar*. Data is in *~/.local/share/lidar*. Cache content is in *~/.cache/lidar*.