CLI - Command-Line Interface

The cli offers a easy way to take sensor measurements and readings. Multiple sensors can be connected to the host running the CLI. To allow a level of automation and integration into CI pipelines, several different configurations can be created to perform different measurements.

Note

For the following cli examples, a working sensor with `UnifiedMagBoardFirmware` is required.

Installation

Note

As the package has not been published on PyPi yet, it CANNOT be install using pip. So for the CLI, the same installation procedure is used as for the main `MagneticReadoutProcessing` package.

```
$ git clone https://github.com/LFB-MRI/MagneticReadoutProcessing
./MagneticReadoutProcessing
$ cd ./MagneticReadoutProcessing
$ pip3 install -r requirements.txt


# RUNNING THE CLI USING VENV (if library is not installed on system)
$ ./venv/bin/python3.9 ./cli/cli.py --help
```

Note

In order to allow direct running of the examples below, the `venv` is used. So all cli commands are changed to: `./venv/bin/python3.9 ./cli/cli.py` instead of `python3 ./cli/cli.py`.

Usage

```
$ ./venv/bin/python3.9 ./cli/cli.py --help
Usage: cli.py [OPTIONS] COMMAND [ARGS]...
Options:
--install-completion  Install completion for the current shell.
--show-completion     Show completion for the current shell, to copy it or
                      customize the installation.
--help                Show this message and exit.
Commands:
config
measure
sensor
```

Configuration

The following example shows how to create a configuration and start a measurement run.

The first step is to test the connected sensors. For this task a new configuration is needed. The ``configname``parameter is equal to the filename of the generated configfile.

The `setupsensor` parameter launches the sensor configuration wizard. Here all connected sensors are listed. If an other sensor hardware is used (with different vid/pid), its possible to specify the port manually

```
$ ./venv/bin/python3.9 ./cli/cli.py config --help
Usage: cli.py config [OPTIONS] COMMAND [ARGS]...
Options:
--help  Show this message and exit.
Commands:
list
reset
setup
setupsensor


# cli.py config setupsensor <configname>
$ p./venv/bin/python3.9 ./cli/cli.py config setupsensor testcfg
0 > Unified Sensor 386731533439 - /dev/cu.usbmodem3867315334391
Please select one of the found sensors [0]:
sensor connected: True 1243455
SENSOR SETUP COMPLETE: cli/configs/testcfg_config.json

# cli.py config setupsensor <configname> <device_path>
$ ./venv/bin/python3.9 ./cli/cli.py config setupsensor testcfg /dev/tty
sensor connected: True 54224326
SENSOR SETUP COMPLETE: cli/configs/testcfg_config.json
```

Sensor information and readout

After the sensor setup is finished for this sensor. Its possible to query the sensor manually.

```
$ ./venv/bin/python3.9 ./cli/cli.py sensor --help
Options:
--help  Show this message and exit.
Commands:
info
query


# cli.py sensor info <configname>
$ ./venv/bin/python3.9 ./cli/cli.py sensor info testcfg
SENSOR INFORMATION
NAME:
ID: 525771256544952
CONNECTED SENSORS: 2
CAPABILITIES: ['static', 'axis_b']


# cli.py sensor query <configname>
$ ./venv/bin/python3.9 ./cli/cli.py sensor query testcfg
QUERY RESULT FOR SENSOR_ID:525771256544952 SENSOR_NUMBER:0
> B:47.66
QUERY RESULT FOR SENSOR_ID:525771256544952 SENSOR_NUMBER:1
> B:44.63
```

Automatic measurement configuration

After a manual readout-test, the configfile can be modified to allow automatic measurements with specified settings. The config also contains information about type of reading, number datapoints and averaging. To set these the `config` option offers a setup wizard.

```
# cli.py config setup <configname>
$ ./venv/bin/python3.9 ./cli/cli.py config setup testcfg
CONFIGURE testcfg
READING-NAME: [testreading]: testreading
OUTPUT-FOLDER [/cli/reading]: /tmp/reading_folder_path
NUMBER DATAPOINTS: [1]: 10
NUMBER AVERAGE READINGS PER DATAPOINT: [1]: 100
MEASUREMENT SETUP COMPLETE: cli/configs/testcfg_config.json
```

Note

To setup another configuration just change the `<configname>` paramter in each command. To edit a configuration, re-run the commands.

Note

To delete a configuration delete the `<configname>_config.json` file in the `cli/config/` directory.

Run automatic measurement

After this step it is possible to execute a measurement using all saved configuration files. First its possible to list all found configuration files inside od the `cli/configs/` folder.

```
$ ./venv/bin/python3.9 ./cli/cli.py config list
FOUND CONFIGURATIONS IN. cli/configs/
0> testcfg
1> calibration
```

To start a measurement run the `measure run``option is used. Its possible to run all or a specified configuration by using the ``<configname>` parameter.

The system performs a pre-check of the sensor and configuration to avoid any misconfiguration errors before a long measurement run.

```
$ ./venv/bin/python3.9 ./cli/cli.py measure --help
Usage: cli.py measure [OPTIONS] COMMAND [ARGS]...
Options:
--help  Show this message and exit.
Commands:
run

# RUN ALL FOUND CONFIGURATIONS
$ ./venv/bin/python3.9 ./cli/cli.py measure run
STARTING MEASUREMENT RUN WITH FOLLOWING CONFIGS: ['testcfg', 'calibrationreading']
# RUN SPECIFIED CONFIGURATION
# ./cli.py measure run <configname>
$ ./venv/bin/python3.9 ./cli/cli.py measure run testcfg
STARTING MEASUREMENT RUN WITH FOLLOWING CONFIGS: ['testcfg']


PRERUN CHECK FOR testcfg [cli/configs/testcfg_config.json]
> config-test: OK
> sensor-connection-test: OK
START MEASUREMENT CYCLE
perform_measurement for testcfg
sampling 10 datapoints with 100 average readings
SID:0 DP:0 B:47.35999999999999 TEMP:23.55443
SID:0 DP:1 B:47.35999999999999 TEMP:23.55443
....
dump_to_file testreading_ID:525771256544952_SID:0_MAG:N45_CUBIC_12x12x12.mag.json
```