Data collection

1) Create empty spectrum files (with regular txt extension) for every type of reading you want to make in /Desktop/Urania-main directory.

1.1) Specify the metadata in the first line of each file in the following format:

```
{"class":"metadata","valve_number":2,"is_a_spectrum":"True","initial_value":1,"amount_of_scans":50,"step":1}
```

1.2) valve_number takes integers from 1 to 16 and specifies the position of multi-inlet valve for the scan.

1.3) initial_value is the first M that you want to scan. For example, if you want to scan from molar mass 1, set initial value to 1.

1.4) amount_of_scans is the amount of scans you want to make. Step is the step between scans, for example, if step is 1, scans would be made for molar masses 1, 2, 3 and so on.

1.5) Class:metadata and is_a_spectrum:true are technical entries that should not be modified

please mind the underscores!

2) Create task list with TaskList.txt name (can be changed in MainConfig) in /Desktop/Urania-main directory.

2.1) Specify the first line as following:

```
{"class": "regular_log", "last_executed": 1}
```

First line is a memory cell that stores the number of last executed regular task.

2.2) Specify the tasks as following:

```
{"class": "task", "name": "pipe1", "type":"emergency", ”how_much_executions”:5, "valve_position":1, "filename":"Pipe1scan", "scans": 10}
```

2.3) Specify the unique name of the task. It is done for user convenience

2.4) specify type of task as “emergency”, “scheduled” or “regular”
2.5) specify valve_position as integer from 1 to 16

2.6) specify the name of spectrum text file that you have created before

2.7) specify the “scans”. It is the amount of scans that are going to be made

2.8) if task type is “emergency”, specify the amount of executions in “how_much_executions” entry as integer or as “inf” for infinite amount of executions. If not specified, emergency task is executed only once and erased from task list

2.9) if task type is “scheduled”, specify the frequency of execution as “freq”:3600. Freq is the amount of seconds between each execution. For example, if you want to execute task once every hour, set freq as 3600. Specify “last_execution”:0. It is a memory cell that stores the last moment of time task was executed

3) To start data collection, execute StartSampling.py script

3.1) Navigate to Urania directory:

```
cd Desktop/Urania-Main
```

3.2) Execute script with sudo privileges:

```
sudo python3 StartSampling.py
```

3.3) Enter admin password

Data visualisation

1) To open data visualisation webapp, start streamlit localhost server

1.1) Navigate to Urania directory:

cd Desktop/Urania-Main

1.2) Execute script that starts localhost server:

streamlit run main.py

1.3) Navigate to localhost:8501 address through web browser (usually done automatically)

2) To open data visualisation page, select the name of desired page on left sidebar

2.1) Data visualisation works either in “last” or “search” mode of operation. It can be changed in Settings menu. If the mode of operation is “last”, program will automatically display the most recent scans. If the mode of operation is “search”, you should select the date and time through prompted widget. Program will display data for this moment of time

2.2) Select how much spectrums you want to display on screen by typing an integer into the field

2.3) There are two types of graphs displayed below: spectrum for given moment of time and concentration vs time graphs for given molar masses

2.4) Each type of graph can be displayed with values in pascals (raw output of RGA) or in PPM (automatically converted). Each type of graph can be displayed either on linear or logarithmic scale.

2.5) For concentration vs time graph, you can input molar masses (or “ox” or oxygen) splitted by comma into the field

2.6) To display table with values, press “display table with values” button. It can be copied and pasted to office programm such as LibreOffice or Excel

2.7) To compare results with tolerance values (file with tolerances can be selected in Settings), press “Find Abnormalities” button

Remote Access

1) System can be accessed remotely through RustDesk program

1.1) Install RustDesk on your system. On Debian systems it can be done by typing “sudo apt install rustdesk”

1.2) Enter 9-digit ID and password of system (not provided in this manual for security reasons)

1.3) Set up 2FA through any 2FA application (I recommend Aegis, as it is free and open source and 100% offline). Credentials are not provided in this manual

1.4) Click “Remote connect” to control the system or “file transfer” to use system as network attached storage.

Manual control of VSC and Multi Inlet Valve

1) VSC and multi inlet valve can be controlled using VSC_Manual_Operation.py

1.1) To run this program, navigate to Urania directory by typing:

cd Desktop/Urania-main/

1.2) When run, program automatically displays readings from all VSC sensors

1.3) Type one of the prompted commands to do the desired action (change mode of operation or setpoint of MFC or pressure controller; or change position of multi inlet valve)