

# **FiSens FiSpec Programmers Manual**

Python code example

## **1) Required libraries**

All libraries used by the developers are stored inside the 'requirements.txt' file which will be provided together with the python code.

The most important libraries for running the python code example are

- tkinter: generating the graphic user interface (GUI)
- matplotlib: creating graphs from data in list format
- pySerial: establishing a serial connection to a given COM-Port
- threading: running multiple tasks simultaneously

The user needs to install named libraries using the following commands in Windows command prompt.

```
python -m pip install tkinter
python -m pip install matplotlib
python -m pip install pySerial
```

### ***Side note – virtual environment:***

If different versions of the named libraries are used for other projects on target device the user can do so by creating virtual environments for those projects. After that it will be possible to run code with specific, project-depending requirements.

A virtual environment can be created and controlled with the command prompt by using the following commands:

- 1) Navigate to the directory in which 'main.py' and 'FiSpec\_GUI.py' are located by using

```
cd [path to directory]
```

- 2) Create a virtual environment by using

```
python -m venv venv
```

which will create a virtual environment called 'venv' in the given directory. An additional folder called 'venv' should now be visible inside the directory.

- 3) Activate the virtual environment by running

```
venv\scripts\activate.bat
```

in case of success the activation is shown by '(venv)' at the beginning of a new line in windows command prompt.

- 4) Install the necessary libraries by running the command lines from above or install all at once by running

```
python -m pip install -r requirements.txt
```

- 5) If you want to add new libraries to your project you can do so by installing them and afterwards save them in requirements.txt by running

```
python -m pip freeze > requirements.txt
```

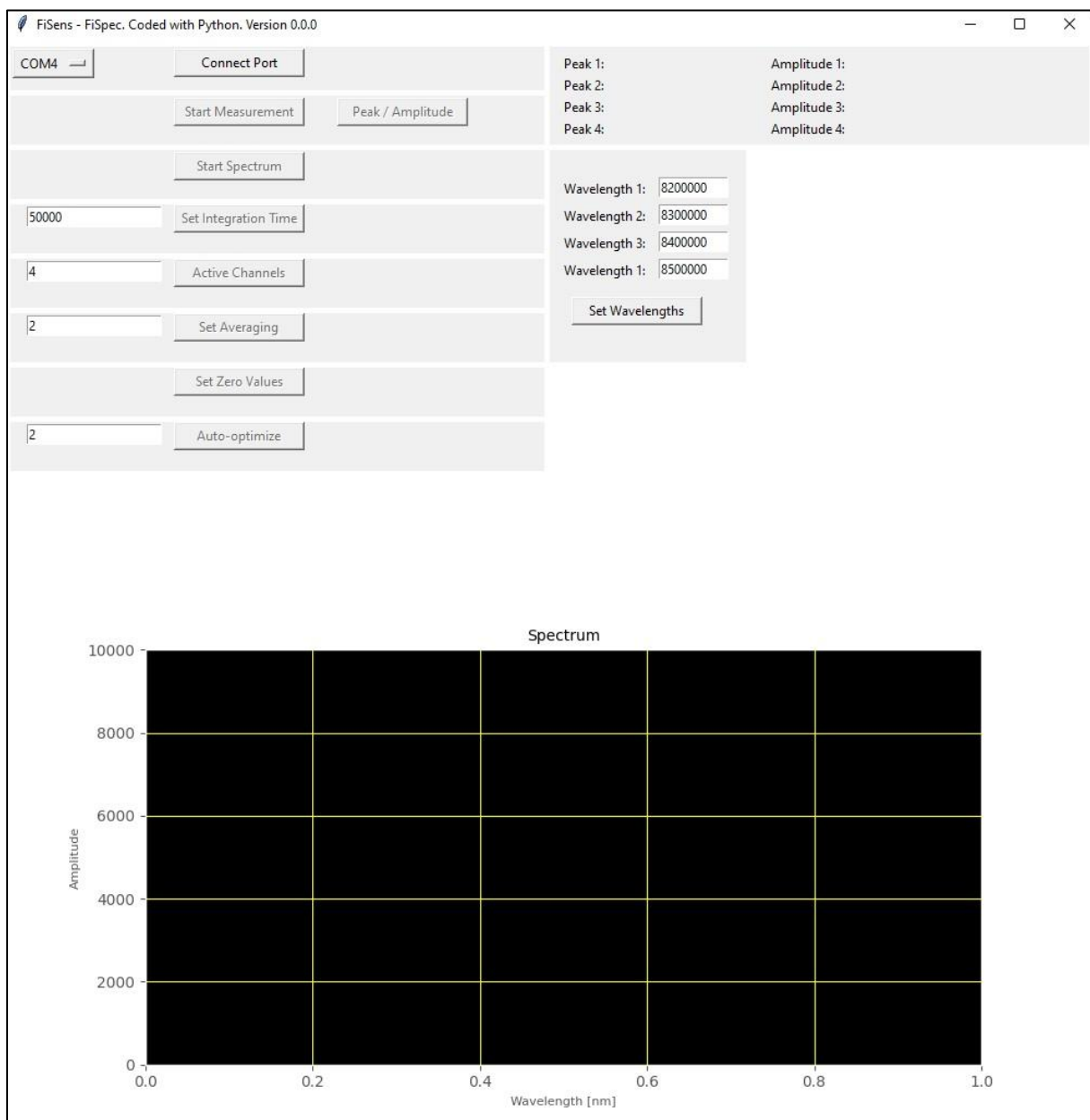
If you're using source control (for example GitHub) it will now be easier for other project participants to reconstruct the code to a running version by passing the 'requirements.txt' file together with the code.

## 2) The Code main.py & FiSpec\_GUI.py

The python code example consists of two files.

- main.py – sending commands, receiving measurement data, mathematical operations
- FiSpec\_GUI.py – Design of example GUI

To run the code first execute FiSpec\_GUI.py followed by main.py. If successful, the graphic user interface as shown below will pop up. Modification regarding the GUI can be achieved by editing FiSpecGUI.py.



After starting the program, the user has to pick a COM-Port by selection in the optionsmenu. If the “Connect Port”-button is pressed and if a FiSens device can be found under the given COM-Port the program connects to the device via the pySerial package. Upon start an object of the Serial class will be created. By initializing the attributes of this object the serial port can be configured, for example setting values for timeouts, stopbits, baudrate etc. If the program is successfully connected to a device the other buttons of the graphic user interface become enabled.

By using the Buttons and entry fields the user can try and test some fundamental functions from the FiSpec program. The table below shows all implemented commands in the python example code.

Command	Description	Button in GUI
Ke,x,y,z>	Peak detection details	Connect Port
LED,1>	Switches internal light source	Connect Port
WLL>	Get wavelength of pixel list	Connect Port
OBB,x>	Switch on board calculation	Peak / Amplitude – Temperature/Strain
a>	Globally start measurement	
iz,x>	Set integration time	Set Integration Time
KA,x>	Transmit channel number	Active Channels
m,x>	Set averaging	Set Averaging
OBn,x	Set zero values	Set Zero Values
AO,x>	Auto-optimization	Auto-Optimize
P>	Get measurement data	Start/Stop Measurement
s>	Get Spectrum	Start/Stop Spectrum