



<b>Publication Year</b>	2018
<b>Acceptance in OA</b>	2020-10-07T12:00:58Z
<b>Title</b>	Control software for the Multi-Channel Led starlight simulator
<b>Authors</b>	SALASNICH, Bernardo, CLAUDI, Riccardo, Alei, E., Barbisan, D., BARUFFOLO, Andrea, Cocola, L., Erculiani, M. S., La Rocca, N., Pace, E., Poletto, L., Trivellin, N.
<b>Publisher's version (DOI)</b>	10.1117/12.2311436
<b>Handle</b>	<a href="http://hdl.handle.net/20.500.12386/27647">http://hdl.handle.net/20.500.12386/27647</a>
<b>Serie</b>	PROCEEDINGS OF SPIE
<b>Volume</b>	10707

# PROCEEDINGS OF SPIE

[SPIDigitalLibrary.org/conference-proceedings-of-spie](https://spiedigitallibrary.org/conference-proceedings-of-spie)

## Control software for the Multi-Channel Led starlight simulator

Salasnich, B., Claudi, R. U., Alei, E., Barbisan, D., Baruffolo, A., et al.

B. Salasnich, R. U. Claudi, E. Alei, D. Barbisan, A. Baruffolo, L. Cocola, M. S. Erculiani, N. La Rocca, E. Pace, L. Poletto, N. Trivellin, "Control software for the Multi-Channel Led starlight simulator," Proc. SPIE 10707, Software and Cyberinfrastructure for Astronomy V, 107071I (6 July 2018); doi: 10.1117/12.2311436

**SPIE.**

Event: SPIE Astronomical Telescopes + Instrumentation, 2018, Austin, Texas, United States

# Control Software for the Multi-Channel Led Starlight Simulator

B. Salasnich<sup>\*a</sup>, R.U. Claudi<sup>a</sup>, E. Alei<sup>a</sup>, <sup>e</sup>D. Barbisan, A. Baruffolo<sup>a</sup>, L. Cocola<sup>c</sup>, M.S. Erculiani<sup>c</sup>, N. La Rocca<sup>b</sup>, E. Pace<sup>d</sup>, L. Poletto<sup>c</sup>, N. Trivellin<sup>f</sup>

<sup>a</sup>Osservatorio Astronomico di Padova, INAF, Vicolo Osservatorio 5, 35122, Padova, Italy;

<sup>b</sup>Dipartimento di Biologia, Università degli Studi di Padova, Viale G. Colombo 3, 35131, Padova, Italy; <sup>c</sup>Laboratori LUXOR CNR-INF, Via Trasea 7, 35131, Padova, Italy; <sup>d</sup>Dipartimento di Fisica e Astronomia, Università degli Studi di Firenze, Via G. Sansone 1, 50019 Sesto Fiorentino, Italy;

<sup>e</sup>Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova, Via Gradenigo 6/B, 35131 Padova, Italy; <sup>f</sup>LightCube SRL, Viale della Navigazione Interna 51, 35129 Padova, Italy.

## ABSTRACT

We built a multi-channel led starlight simulator capable to reproduce the radiation of stars of F, G, K and M spectral types in the wavelength range 365-940 nm. This range overlaps the photosynthetic active interval allowing us to use the simulator for biological experiments under radiation and atmospheric conditions close as much as possible to those expected on extrasolar planets.

The simulator is a laboratory tool which is part of the “Atmosphere in a Test Tube” project, aimed to study the photosynthetic efficiency of bacteria under alien environmental conditions and their possible impact on the atmosphere of the host planet.

This paper describes the software developed to control the simulator. We begin by presenting a conceptual overview of the instruments and then illustrating the top-level requirements and the architecture of the control software. Finally, we give a description of the graphical user interface.

**Keywords:** Exoplanets, laboratory tool, control software.

## 1. INTRODUCTION

We developed a multi-channel led illuminator able to radiate in the 365-940 nm range, composed by twenty-five dimmable channels, for a total of 273 air cooled diodes arranged in five concentric rings<sup>1</sup>. Each channel is tunable enough to allow us to reproduce the radiation of stars of F, G, K and M spectral types. This feature makes the illuminator a unique laboratory tool we used for biological experiments under radiation conditions close as much as possible to those expected on extrasolar planets<sup>2</sup>. The offered wavelengths indeed overlap the 400-700 nm range, the so called photosynthetically active radiation (PAR) interval, where on the Earth the solar radiation is used by photosynthetically active organisms to create oxygen<sup>3</sup>. The control software has been designed to allow the user to select a star spectrum from a set, to properly dim each light channel to reproduce, as a whole, the input spectrum and to finally verify through a spectrometer that the produced spectrum can fit the input one.

Our Multi-Channel LED Starlight Simulator is a tool which is part of the “Atmosphere in a Test Tube” project, aimed to study the photosynthetic efficiency of bacteria under alien environmental conditions and their possible impact on the atmosphere of the host planet<sup>4</sup>.

The project is led by INAF-OAPD, the Astronomical Observatory of Padova, in collaboration with the University of Padova (Biology Department and Information Engineering Departments), the CNR-IFN-Luxor laboratory of Padova, the INFN-National Laboratories of Frascati (Roma) and INAF-OAPA, the Astronomical Observatory of Palermo.

This paper describes the software developed to control the starlight illuminator. We begin by presenting a conceptual overview of the instruments and then illustrating the top-level requirements and the architecture of the control software.

\*[bernardo.salasnich@inaf.it](mailto:bernardo.salasnich@inaf.it)

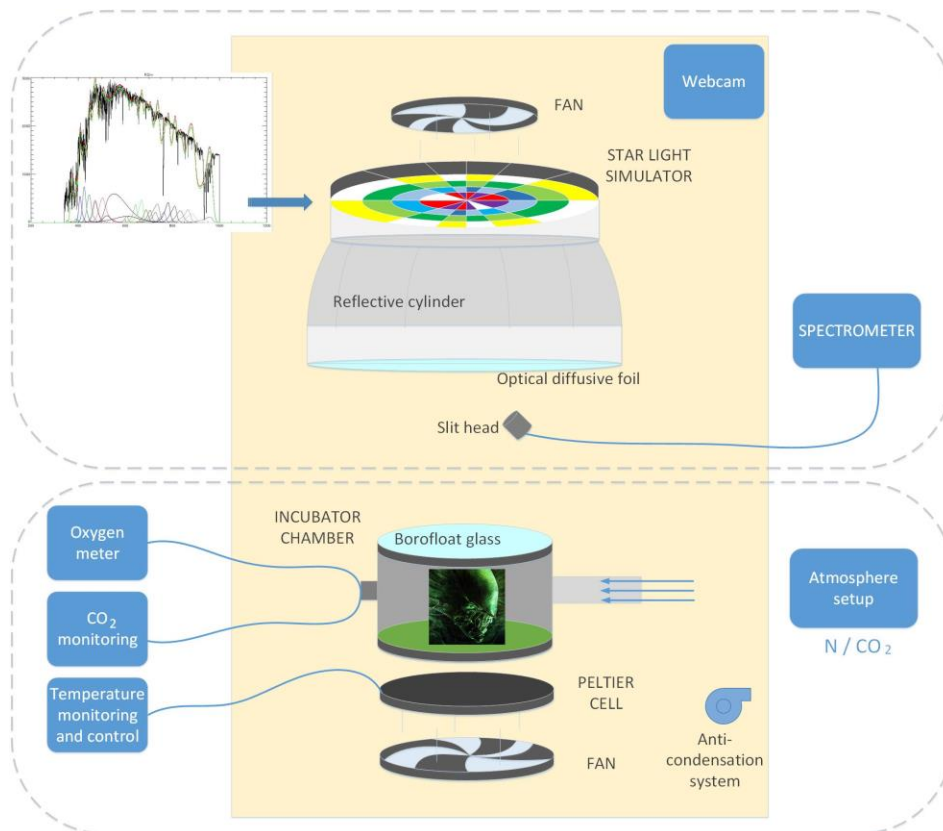
## 2. CONCEPTUAL OVERVIEW

The experimental ensemble is illustrated in Figure 1. It can be conceptually split into two parts: on top we have the starlight simulator, composed by the illuminator and a spectrometer; on the bottom the radiation reaches an incubator chamber, where the biota used for the experiments are hosted. The entire system is isolated in a dark container and is cooled by two fans, it is equipped with an anti-condensation system and can be remotely inspected through a webcam.

The radiative system is composed of five concentric rings on which the mosaic of circuit boards is arranged in a pie-chart shape, on the surface of which the diodes have been welded. Twenty-five led channels have been used. The modularity design of the board permits an easy maintenance in case of damage, allowing to remove only the piece where the problem may appear. The number of led channels has been limited both by the cost of extended wavelength diodes and by the availability on the market. The disposition of diodes on the board has been designed to reduce the not-uniformity of the flux, due to the intrinsic light exit angle of each led. Moreover, a reflective cylinder and an optical diffusive foil have been mounted to increase the uniformity. Since the thermal power of the system is bigger than the radiation power, the diodes are cooled by a fan set on the back of the board.

A spectrometer collects the light through a slit head located at a manually adjustable distance from the diffusive foil. The adopted spectrometer is a COTS component; we selected the FLAME VIS-NIR5<sup>5</sup> by Ocean Optics, its 2040x2040 pixel detector covering the range 190-1100 nm.

The incubator chamber is a steel cylinder in which the light enters through SCHOTT Borofloat borosilicate glass<sup>6</sup>, thermally resistant and not filtering the radiation. The atmosphere in the cell is composed mainly by Nitrogen and carbon dioxide and the mixture is set manually. The chamber is kept at constant temperature by controlling a Peltier cell which also supports the cylinder. Photosynthetically active biota inside the chamber are expected to produce oxygen, therefore the cell is connected to an oxygen sensor while the carbon dioxide concentration is monitored via Tunable Diode Laser Absorption Spectroscopy (TDLAS).

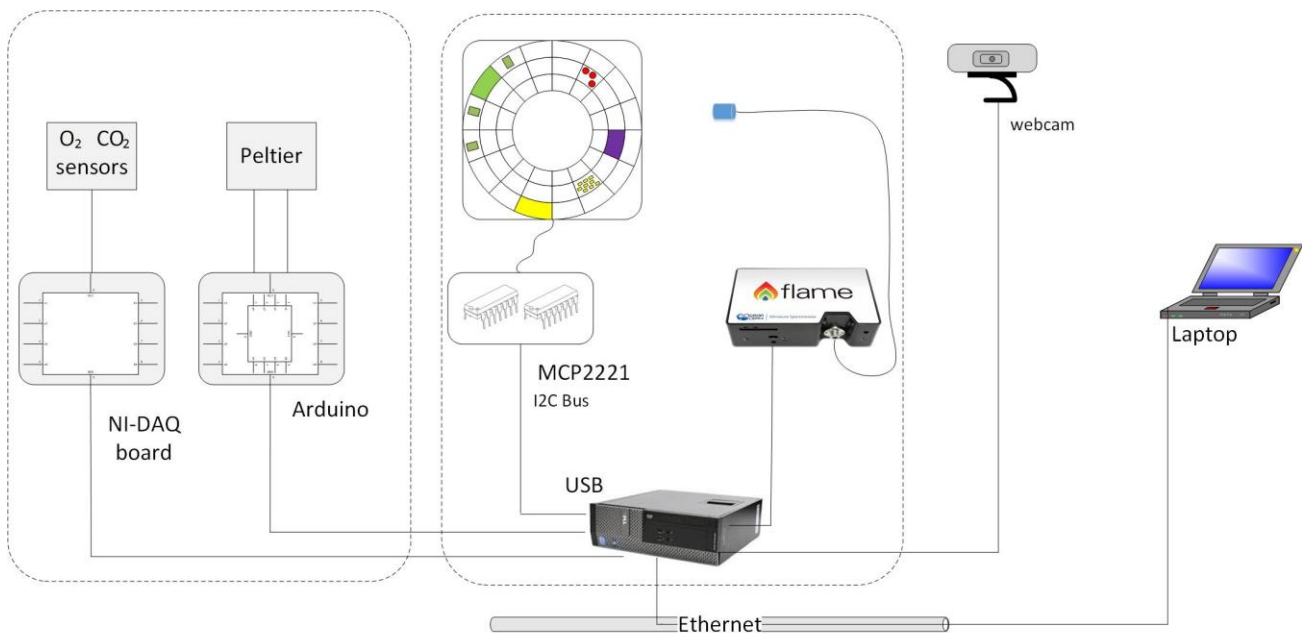


**Figure 1:** Schematic overview of the Atmosphere in the Tube experiment.

Figure 2 shows on the left side the diodes mounted on the board in five concentric rings with the circuits arranged in pie-chart shape. On the right side the same artifact is shown with some diodes switched on.



**Figure 2:** LEDs as mounted on the board (left) and illuminated (right).



**Figure 3:** Hardware architecture

Figure 3 shows the hardware architecture of the system. The control computer, located below the experiment box, communicates with the spectrograph, the illuminator, the Peltier cell, oxygen and carbon dioxide sensors through four serial ports. The Flame is a microcontroller-based miniature fiber optic spectrometer that communicates via the Universal Serial Bus. The diode intensities are regulated through one MCP2221<sup>6</sup> micro-controller, which is connected over I2C bus. The Peltier cell is controlled via Arduino and the oxygen and carbon dioxide sensors are read through a National Instruments DAQ board.

The control software runs under Windows 7SP1 64-bit Operating System on top of a Dell Optiplex 3020M computer, remotely accessible for testing and debugging purpose.

### 3. THE CONTROL SOFTWARE

The starlight simulator (illuminator and spectrometer) and the incubator cell environment (gas sensors and Peltier cell) are controlled by two separated processes, running on the same computer. We have chosen this approach in order to keep the simulator as a separate device. This allows the simulator to be used for other applications than the incubator chamber. In this paper we will focus on the control software of the simulator. The control software of the chamber environment uses MATLAB to access the Arduino and DAQ board, and it will be not described here.

### 3.1 Software requirements

The first two top-level requirements are obvious and simple:

1. the software has to communicate with the MCP2221 chip in order to set the intensity of each channel to the given percentage;
2. the software has to communicate with the spectrometer in order to perform the following operations:
  - a. to setup the integration time of the exposure;
  - b. to start the exposure;
  - c. to get the acquired spectrum;
  - d. to convert the measured photon counts into physical units [ $\text{W}/\text{m}^2/\mu\text{m}$ ].

Moreover, the software has to compute, for each channel, the intensities needed to reproduce any selected input stellar spectrum.

Additional requirements are:

- it has to be possible to perform a self-test of the hardware at start-up. The test consists into switching on the channels at 50% of their power, one by one, taking the spectrum and displaying it to the user.
- the software has to switch off all diodes and acquire a dark frame at the start-up, which is subtracted to each acquired spectrum. It has to be possible to skip this step and load a dark previously acquired and saved on disk.
- it has to be possible to save on disk the acquired spectrum and the used set of led intensities.
- since the power supply cannot manage all diodes when switched on at full intensity, the software has to check that the current absorption does not exceed a given threshold. In that case, the software shall clearly warn the user.
- to provide a control GUI which allows to load the spectrum from file, display it on a plot, set the intensity of each channel, take an exposure, display the acquired spectrum on a plot.
- to compute the integral flux in some ranges of biological interest (PAR, 600-800 nm, 430-450 nm) and convert it in units adopted in biological studies [ $\mu\text{E}/\text{m}^2/\text{s}$ ].

There are not tight performance requirements. Moreover, we wanted a platform-independent software. Both previous requirements together with the fact that both MCP2221 and the spectrograph API are available on Java made us choose Java as the application development language.

### 3.2 Implementation

Figure 4 shows a simplified class diagram of the control application, which has been called MUCHLESS (MULTI-Channel LED Starlight Simulator).

The Main class instantiates one object of class MuchLessManager which has three protected members: the spectrum loaded from file, the one acquired with the spectrograph and the photon counts of the currently used dark.

MuchLessManager class provides five methods for 'simple' operations: load a new spectrum, acquire a new spectrum, acquire a new dark and switch on/off a led channel. Moreover, it provides some methods for 'complex' operations, shown in the figure as "runSequence", composed by sequence of simple operations. For example the startup sequence switches off all LEDs, takes a dark, switches on each channel and takes an exposure for each channel.

MuchLessManager class uses two objects of class Spectrometer and Illuminator. Spectrometer uses one instance of a low level class, Wrapper, in charge to open the communication with the spectrograph, after loading the corresponding API library, and to execute the required method. For the spectrometer such methods are Setup (not displayed in the figure) to set the integration time and Acquire. A similar design is used for the Illuminator class, which uses an instance of the ChipManager class, in charge to open the communication with the MCP2221 chip, load the dynamic library which implements the chip driver and execute the method Write, which setup the intensity on the chip.

MuchLessManager moreover is in charge, through the method EstimateIntensities, to compute the twenty-five intensities needed to reproduce the given stellar spectrum. We adopted a multiple linear regression algorithm to compute the coefficients, using the OLSMultipleLinearRegression class offered by the Apache Math3 package<sup>8</sup>.

Figure 5 shows the spectra produced by each LED channel set at maximum intensity together with the spectrum of a K7 star. On top the best fit obtained with the adopted linear regression routine has been overlaid. Two fits are displayed, the second one obtained using a different programming language (IDL by Exelis<sup>9</sup>) for testing purpose.

MuchLess is a multi-threads application. The user requests are passed to MuchLessManager through the GUI, which inherits from the Java Swing JFrame class and uses Swing components<sup>10</sup>. To avoid freezing the Event Dispatch Thread, used by the graphical interface, we used some worker threads to execute time-consuming operations, like the “runSequence” described above. We adopted the same strategy also for the ‘simple’ operations, therefore each method of the MuchLessManager class is just a wrapper which creates and run a dedicated thread object, like for example the IlluminatorOnThread.

All thread classes are inheriting from the SwingWorker class provided by the Swing library.

Finally, we used the JFreeChart package<sup>11</sup> to plot the spectra.

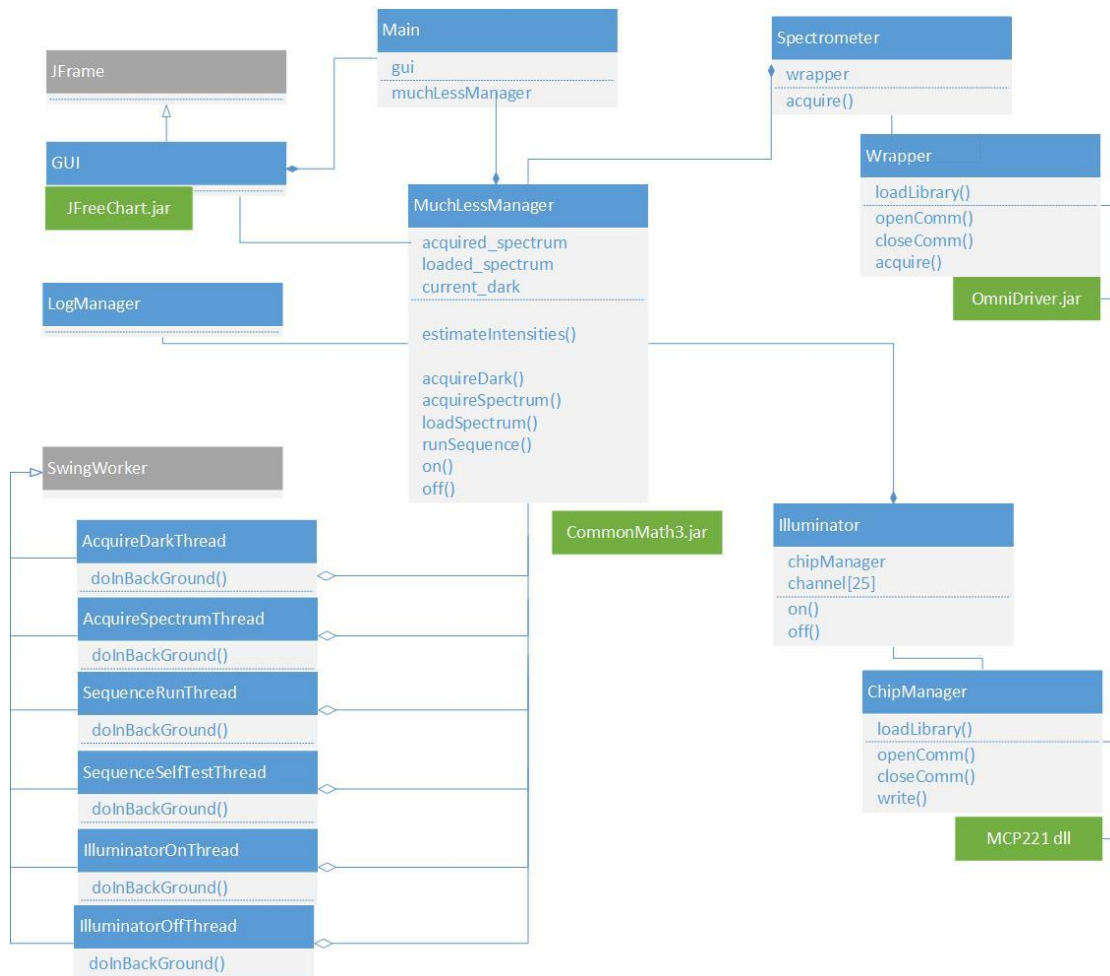
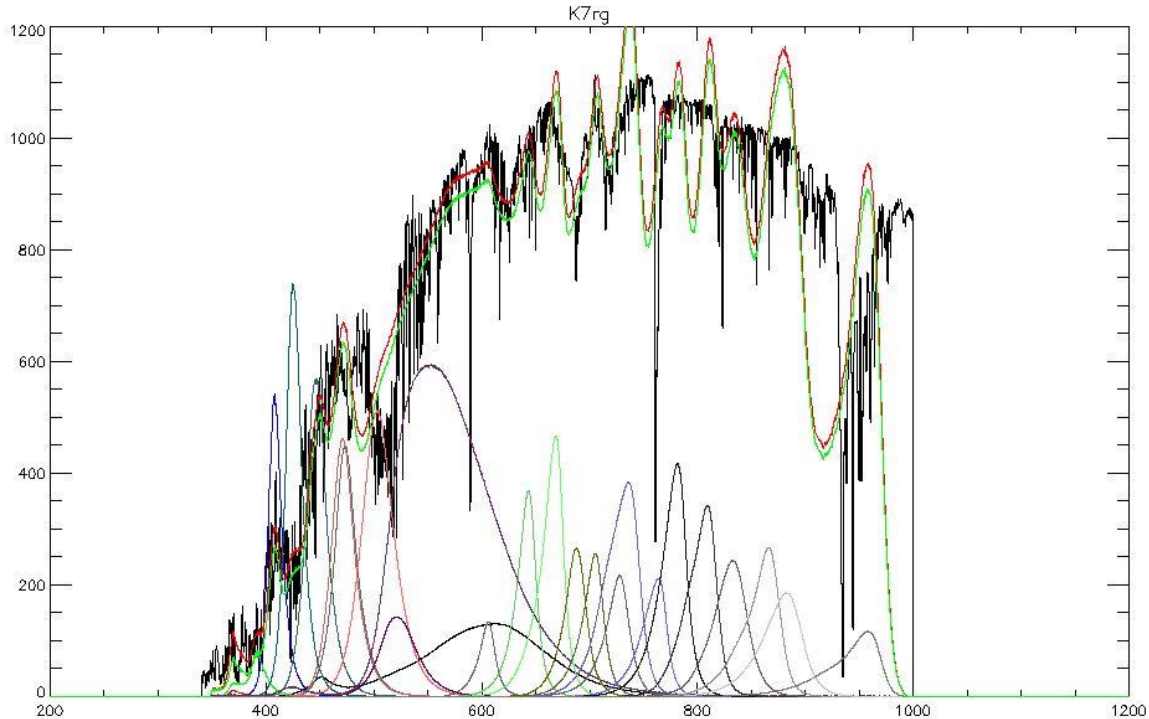


Figure 4 Class diagram of the control software



**Figure 5:** Flux of a star of spectral type K7 together with the spectra of each LED channel. On top the best fit obtained with the linear regression has been overlaid. Wavelengths are given in nm and fluxes in  $\text{W}/\text{m}^2/\mu\text{m}$ .

### 3.3 Graphical interface

Figure 6 shows the graphical interface of the control program. On the left side the user can control the LEDs by switching on and off each channel after selecting the wanted intensity. The first column indicates the intensities in the MCP2221 range 0-1023, the second column shows the corresponding percentage. An icon clearly indicates the state of the channel (gray if switched off, green if on, red if MCP2221 is not reachable).

The total absorbed power in Watt is displayed on top of the channels. The value is highlighted as a warning if it exceeds a given threshold.

On the right side of the panel the spectrum of the star is plotted together with the acquired one. The integral fluxes of the acquired spectrum over some intervals of biological interest are displayed on top, in both  $[\mu\text{E}/\text{m}^2/\text{s}]$  and  $[\text{W}/\text{m}^2]$  units.

In case the spectrograph detects a saturation condition, an icon appears on top of the panel to warn the user to repeat the acquisition after reducing the integration time. Changing of the detector integration time is possible by operating on the dedicated spinner component.

On the upper part of the panel the user can start the execution of the “run” sequence, which sequentially switches on each LED according to the intensities set on the left and acquires the new spectrum.

Moreover, it is possible to activate a “scan” feature useful when the user wants to manually change one or more intensities and immediately see the resulted spectrum. In this case a timer is started and the program acquires a frame per minute (the period is configurable).

The user can save both spectrum and intensities on disk. When loading a spectrum, the dialog in Figure 7 appears, which allows to choose the spectrum and to enable two additional features. The “regression” check-box allows to enable the computation of the intensities to fit the loaded spectrum while the “scale” check-box allows to scale the total flux of the loaded spectrum to a selectable value in  $[\mu\text{E}/\text{m}^2/\text{s}]$  units.

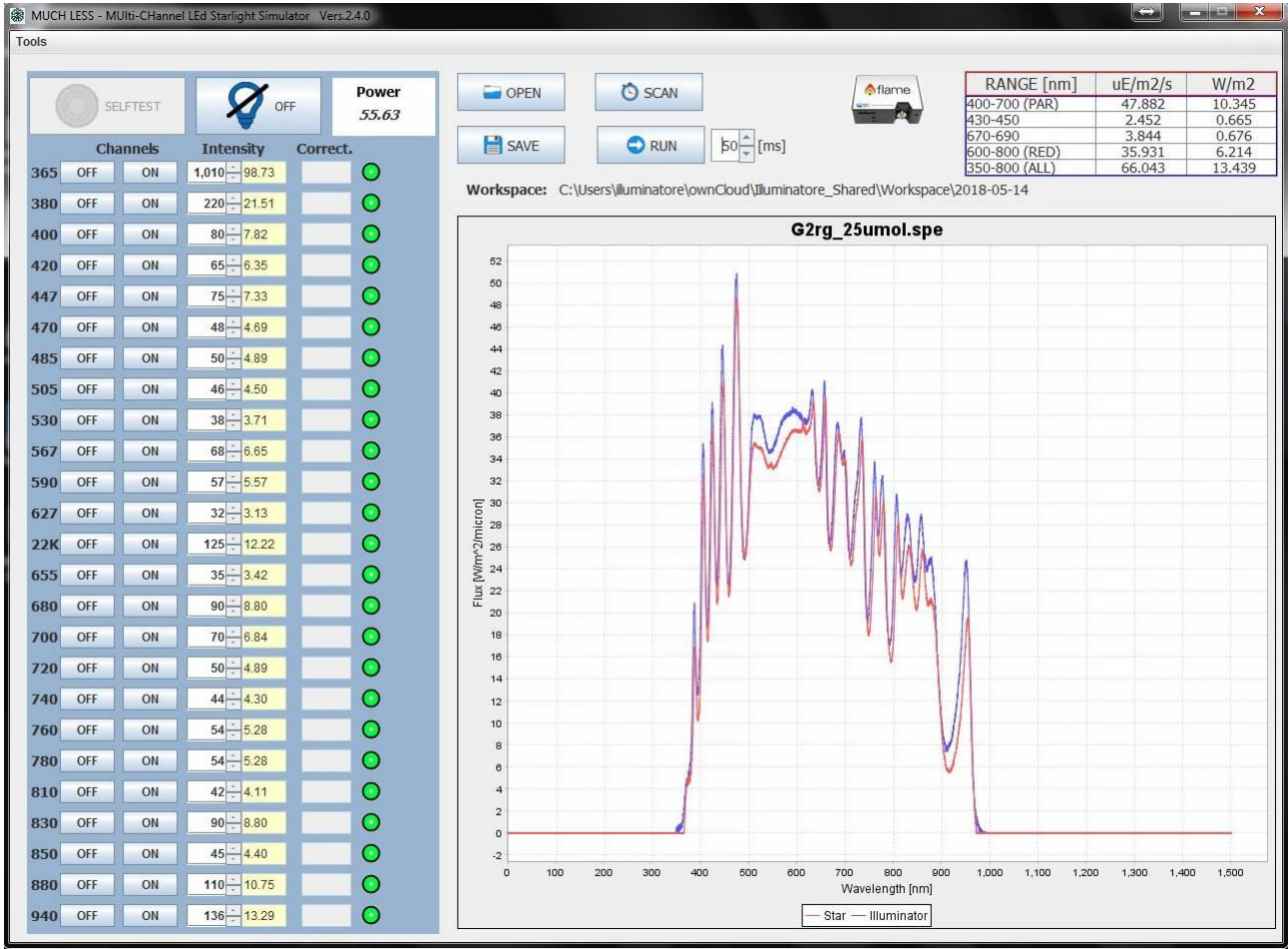


Figure 6: Graphical interface of the control program. See text for the details.

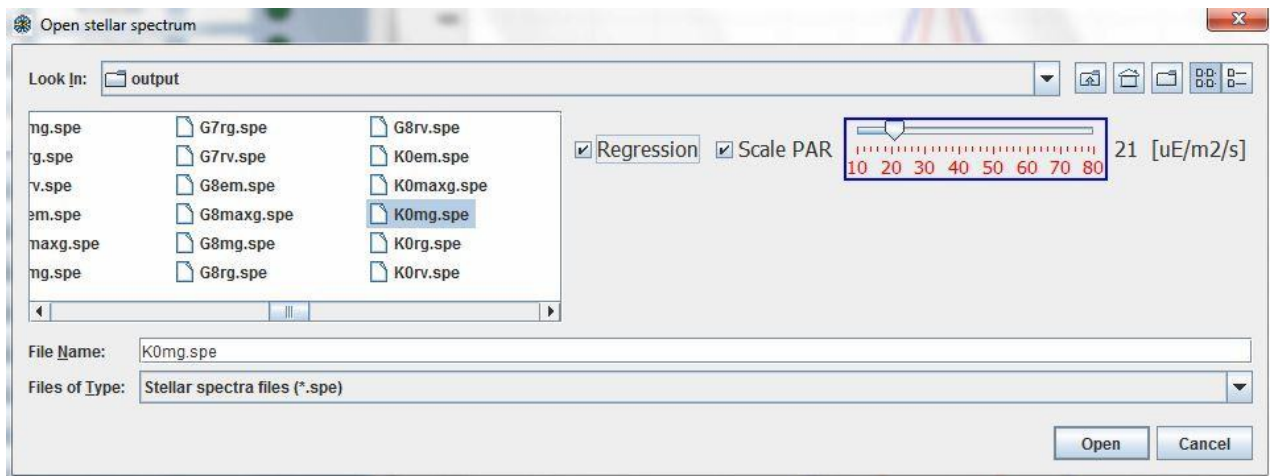


Figure 7: Open spectrum dialog, where the user can also enable the regression and the scaling of the total flux.

## REFERENCES

- [1] Erculiani, M.S. et al, "A tunable integrated system to simulate colder stellar radiation", Proc. SPIE 9626, Optical Systems Design 2015: Optical Design and Engineering VI.
- [2] Claudi, R.U. et al, "Simulating Super Earth atmospheres in the laboratory", International Journal of Astrobiology Vol. 15, Issue 1, 2015
- [3] La Rocca, N. et al, "Microorganisms suitable for studying biomarkers within the atmosphere in a test tube project", 5th Workshop of the Italian Astrobiology Society, Life in a Cosmic Context, 2015
- [4] Trivellin, N. et al, "Adaptive multi-wavelength LED star simulator for space life studies", Light-Emitting Diodes: Materials, Devices, and Applications for Solid State Lighting XX conference, Volume: 9768, 2016
- [5] Ocean Optics FLAME Spectrometer, <https://oceanoptics.com/product/flame-spectrometer/>
- [6] SCHOTT Borofloat Borosilicate Glass, [https://www.schott.com/advanced\\_optics/english/products/optical-materials/thin-glass/borofloat-33/index.html](https://www.schott.com/advanced_optics/english/products/optical-materials/thin-glass/borofloat-33/index.html)
- [7] MCP2221 converter by MICROCHIP, <http://www.microchip.com/wwwproducts/en/MCP2221>
- [8] Apache Commons Math, <http://commons.apache.org/proper/commons-math/>
- [9] IDL by HARRIS Geospatial Solutions, <http://www.harrisgeospatial.com/SoftwareTechnology/IDL.aspx>
- [10] A Swing Architecture Overview <http://www.oracle.com/technetwork/java/architecture-142923.html>
- [11] JFreeChart library, <http://www.jfree.org/jfreechart/>