

TECHNICAL NOTE

PYTHON EXAMPLE FOR INTEGRA

1. GETTING STARTED

The SP_Integra_Python.zip folder contains an example of Python code for serial communication with standard INTEGRA devices from Gentec-EO. Open the unzipped folder with an integrated development environment (IDE) to view and run the example. Three Python modules must be installed to run this code: [matplotlib](#), [time](#), and [pyserial](#).

RUN "INTEGRA.PY"

The example contains four scripts: *integra.py*, *port.py*, *scope.py*, and *status.py*. The *integra.py* file contains the main function, which allows the user to communicate with the program. Once the script is executed, a list of serial ports will be displayed on the terminal for the user to choose the desired one.

SELECT A SERIAL PORT

The first action required from the user is to choose the serial port with which they wish to communicate. If no serial port is connected, an error message is displayed and the program terminates. Otherwise, the user is prompted for the port name (COM#).

Opening the serial port is carried out by the "port_open" function of the PortSerial class of the *port.py* file, while the selection is carried out by the "port_select" function in the same location. Since the program is suitable for INTEGRA devices, the transmission rate is set to 115,200.

2. GENERAL OPERATION

Once the COM port is selected, a menu is displayed on the terminal with the following five options:

```
-----  
Menu:  
  
1. Display detector/Monitor info  
2. Change the wavelength  
3. Change the range  
4. Display measurement  
5. Quit  
  
Your choice: 
```

Each time an option is selected and the corresponding action is performed, the menu reappears until the user decides to exit. Below, how each option works and the required actions are explained.

OPTION 1: DISPLAY DETECTOR/MONITOR INFORMATION

This option lists the information of the connected Gentec-EO device using the *STS command, which is sent and read by the "get_status" and "print_status" functions of the INTEGRA class of the *integra.py* file. The information displayed includes, among other things, the device name, wavelength, and current range.

TECHNICAL NOTE

OPTION 2: CHANGE THE WAVELENGTH

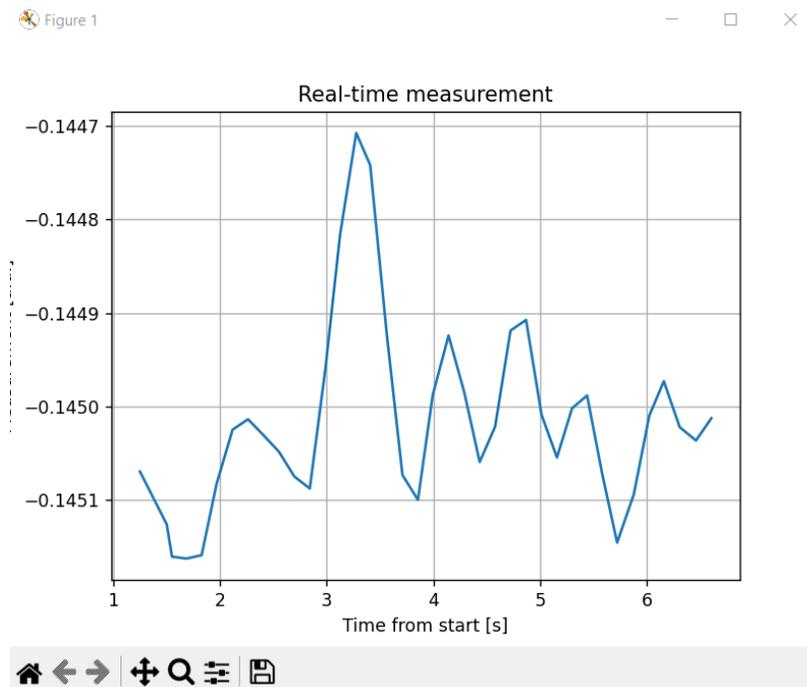
When option 2 is selected, the user is prompted to enter the desired wavelength. The range (minimum and maximum value) is displayed when option #1 is selected. However, if the number entered exceeds this range, the value is automatically adjusted to the nearest limit. Note that the number must be less than 99999 and that decimal numbers are not allowed, as are negative numbers. It is the "set_pwc" function (INTEGRA class) which is called to make the change.

OPTION 3: CHANGE THE RANGE

Option 3 is very similar to the previous one. The main difference is that the "show_validscale" function (INTEGRA class) is called to display the possible ranges. Next, the user is prompted to enter a range index from the choices displayed (number in brackets). As before, the value is adjusted to the nearest limit if the value entered is not within the allowed range. The number entered must be less than 99, and the same restrictions as in section 2.3 apply. After entering the number, the "set_range" function (INTEGRA class) is called.

OPTION 4: DISPLAY MEASUREMENT

When option 4 is selected, a graph appears with real-time measurements (see figure below). It is the Scope class that constructs and updates this graph. The *CAU command allows you to get the data continuously and *CSU stops the sending. Once the window is closed, the menu reappears.



3. GOING FURTHER

Now that you have a basic example, you can build your own Python code to control your INTEGRA device. The complete list of serial commands is detailed in the INTEGRA user manual. You can download the latest version of the user manual on the Gentec-EO website: <https://www.gentec-eo.com/products/integra>