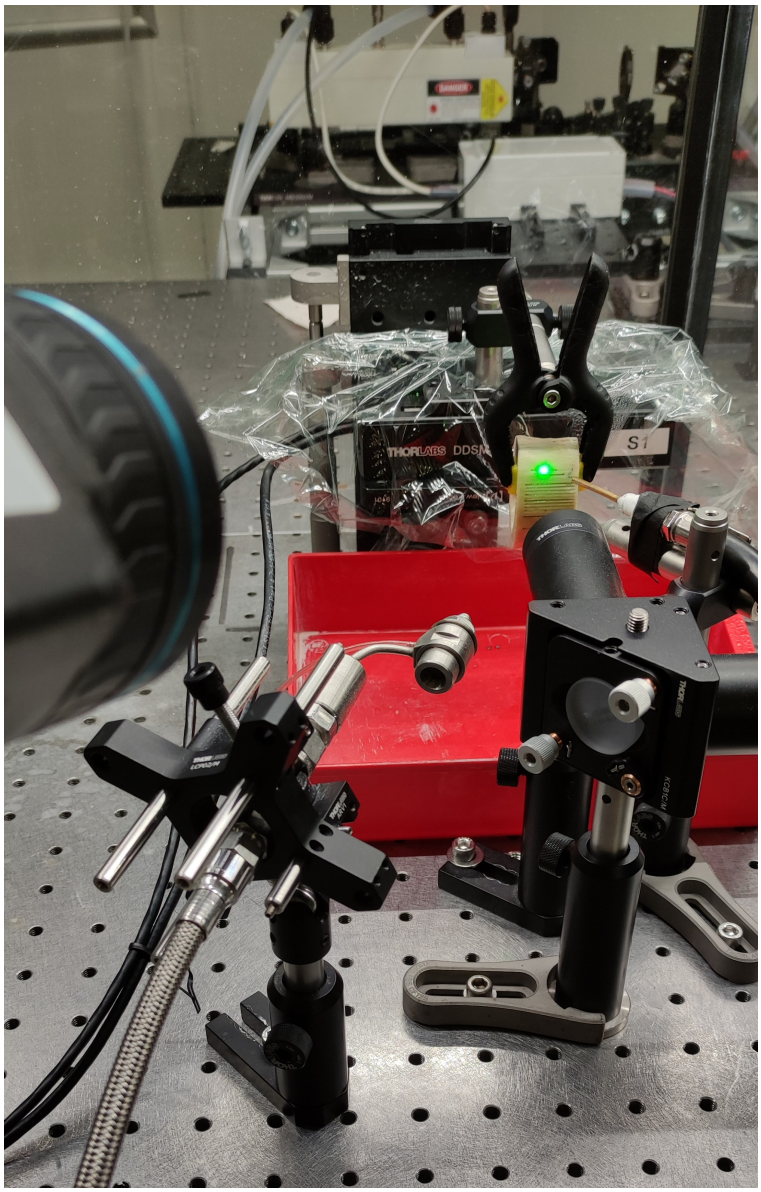
MANUAL for the Integrated setup

Dr. Sandra Drusová (Sandora)

December 9, 2021



Contents

1 Introduction **2**

2 Program logic and synchronization of devices **3**

3 Quick start **3**

4 Testing and troubleshooting **4**

4.2 Irrigation 5

4.3 Air controller 5

4.4 Thermal camera 5

4.5 Translation stage 6

1 Introduction

Several electronic devices were synchronized and integrated together to help optimizing the laser ablation process in the BLOG lab. This guide describes how to control, test and troubleshoot these devices.

The involved devices are:

Laptop	Dell Latitude 3400
Irrigation system	TESCOM ER5000
Air flow controller	Eleveflow OB1
Thermal camera	FLIR A655
Shutter	Thorlabs SH1
Brushed motor controller (cube)	Thorlabs KBD101
Solenoid controller (cube)	Thorlabs KSC101
Translation stage	Thorlabs DDSM50/M

How to connect the devices:

Since the laptop has only 3x USB-A ports, you will need a docking station or any extender with 4+ USB-A ports.

Irrigation system	blue USB-A to the USB extender
Air flow controller	black USB-A to the USB extender
Thermal camera	Ethernet cable to the laptop
Shutter	solenoid controller via PIN cable
Translation stage	brushed motor controller via PIN cable
Cube controllers	Thorlabs USB-A to the USB extender

How to control the devices:

Irrigation system	*TEST_Irrigation_Air.py, RUN_only_Irrigation.py*	*RUN_all.py*
Air flow controller	*TEST_Irrigation_Air.py, RUN_only_Air.py*	
Thermal camera	ResearchIR, *RUN_only_Camera.py*	
Shutter	manually via controller cube, Kinesis, *RUN_only_Shutter.py*	
Translation stage	manually via controller cube, Kinesis, *RUN_only_Stage.py*	

2 Program logic and synchronization of devices

Translation stage is controlled by Kinesis software with a loop sequence *ThorlabsStageLoop.sequence* or a Python program *RUN_only_Stage.py* (will be integrated into the main program in the future if needed). The stage is moving back-and-forth within a user-defined range with a given speed.

The ablation laser starts while the shutter is still closed blocking the beam. The main program *RUN_all.py* starts, the irrigation starts, water jet is cooling down the sample. Within roughly 2 seconds the shutter opens, the ablation starts and the timer is on. When the user-defined time is up, the shutter closes. All other devices are automatically closed.

The irrigation is always on during the ablation. The water jet is deflected by air jet based on temperature of the sample. Temperature of the sample is registered by the thermal camera. If the maximum temperature from the entire camera image exceeds the threshold temperature, the air jet is switched off to let the water cool down the sample. Otherwise the air jet is on and deflects the water jet.

There is an option to apply adaptive temperature thresholding: The program looks at maximum temperature from n samples in the past measurements. It makes and average and scales it by a scaling factor. This value becomes a new threshold temperature. The camera registers only surface temperature. Temperature at the surface might decrease as the cut gets deeper, that's why the adaptive threshold was implemented.

3 Quick start

Loop sequence for the stage *ThorlabsStageLoop.sequence* is located in C:/Users/OceanSpectro/Desktop.
All necessary Python files are located in the folder
C:/Users/OceanSpectro/Desktop/Sandra/Code/integratedsetup.
The code to control the devices is written in Python3.
All manuals are located in C:/Users/OceanSpectro/Desktop/Sandra/Manuals.

1. Set-up the experiment, warm-up the laser

2. Connect all devices

3. Translation stage: Enable the movement using a button on the controller. Home the stage using the controller or in Kinesis. Move the stage manually with the joystick and choose the range of distances where you want it to move during ablation.
4. Open Sequence *ThorlabsStageLoop.sequence* in Kinesis program and edit user parameters (position1, position2, max_velocity, acceleration)
5. Run the sequence
6. (optional) Test the irrigation, air controller, shutter, thermal camera. Info in section 4.
7. Make sure that the shutter blocks the beam, manually with the controller.
8. Start the laser
9. Start the pressurized air from the lab. It blows the water and debris away from the sample.
10. Open *RUN_all.py* in Spyder 4 (Anaconda3) program and edit user parameters
11. Run *RUN_all.py* using green triangle button or F5 key. The program should stop after a pre-defined period of time and close connections to all devices. Irrigation and air stops, shutter is closed.
12. In case you want to interrupt the execution of *RUN_all.py*, click in the command window in Spyder and press CTRL+C.

4 Testing and troubleshooting

In case of any issues, contact @Sandora for IT support. Before you do that, please try turning the devices on and off again. Python programs can be interrupted with CTRL+C. Consult the manuals (manufacturers and Lina's manual) for the range of acceptable settings for different devices.

4.1 Shutter

Testing:

- Enable the shutter on the cube controller. Try opening and closing the shutter using the wheel on the cube controller.
- Run *RUN_only_Shutter.py* in Spyder, edit user parameters. Check if the shutter opens and closes in a pre-defined sequence.

Troubleshooting:

- If the shutter doesn't react to the wheel on the controller, it was not enabled.
- If Python throws an error, the shutter was not connected well. Try reconnecting the device or change USB port.

4.2 Irrigation

Testing:

- Run *TEST_Irrigation_Air.py* in Spyder. The program will guide you.
- Run *RUN_only_Irrigation.py* in Spyder, edit user parameters. Check if the water jet starts and stops in a pre-defined sequence.

Troubleshooting:

- If you get SettingsNotAcceptedError, restart the program
- If Python throws any random error, try reconnecting the irrigation system or connecting it to a different USB port
- If the water jet doesn't look nice, open the nozzle, take the inner part out and let it dry.

4.3 Air controller

Testing:

- Run *TEST_Irrigation_Air.py* in Spyder. The program will guide you.
- Run *RUN_only_Air.py* in Spyder, edit user parameters. Check if the air flows and stops in a pre-defined sequence.

Troubleshooting:

- If Python throws an error, try connecting the device to a different USB port.

4.4 Thermal camera

Testing:

- Run *RUN_only_Camera.py* in Spyder. Put your hand in front of the camera and observe the command line output to see if the temperature changes.

Troubleshooting:

- Unplug the camera. Wait until it makes the clicking "predator noise" twice. It's most likely camera calibrating itself. Restart the .py code.
- Make sure that the camera is connected directly to the PC, not via the USB extender!

4.5 Translation stage

Testing:

- Enable the stage on the cube controller. Home it. Try moving it using the wheel on the cube controller.
- Move the stage using Kinesis
- Run *RUN_only_Stage.py* in Spyder, edit user parameters. Check if the stage loops back-and-forth.

Troubleshooting:

- If the stage doesn't react to the wheel on the controller, it was not enabled.
- The stage stops reacting if it gets wet from the inside. Stop the program and dry it well before starting new experiments.
- If you forcefully moved the stage, home it and restart the program.