Instructions

Username: microscope Password: microscope

Database password: microscope

For Si/SiO2

Without polarizer, filter, etc

Microscope brightness: preset

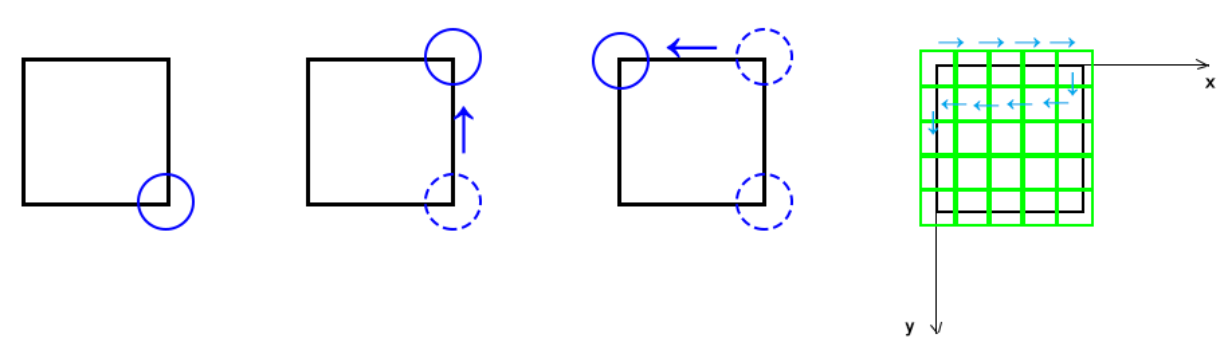
Shutter speed 1/200

ISO 100

White balance: custom

Picture quality: S2 (1920x1080)

1. Open EOS Utility software and run the Python program from Spyder by pressing F5 (or use *scan.bat* on desktop if you just want one scan).
2. Use the knob to locate the lower right corner in the field of view. Focus with the joystick. (See below for the controller setting)
3. Use the joystick to move the lens to the upper right corner of the area and focus. (This relative movement is recorded)
4. Use the joystick to move the lens to the upper left corner of the area and focus. (This relative movement is recorded). The program will take this corner as the origin and calculate how many photos need to be taken. The scanning starts.
5. The image files will be saved to desktop\Pictures\[date], and named as [date]\_[time]\_[total size]\_row\_col



Remark

1. Program structure (main.py)

Cell 1: import modules, set parameters and initialize devices

Cell 2: define functions

Cell 3: scan

Cell 4: image processing

If you want to test or control certain devices, just run Cell 1 and 2, then use the console on the lower right. For example, to enable joystick control, call *joyControl()* .

1. When you run the program in Spyder, relevant variables will be recorded and can be inspected in Variable Explorer in the upper right section.
2. Please try to fix the sample to the glass slide and the stage to prevent drifting. Also please keep the sample to be as level as possible, since a large change in z is likely to cause defocus.
3. Be careful not to hit the limit of the moving stage when using joystick to control, which will make step count inaccurate. The starting position of the stages has been set up to facilitate moving up and left with respect to the chip.
4. Semi-auto mode (under development)

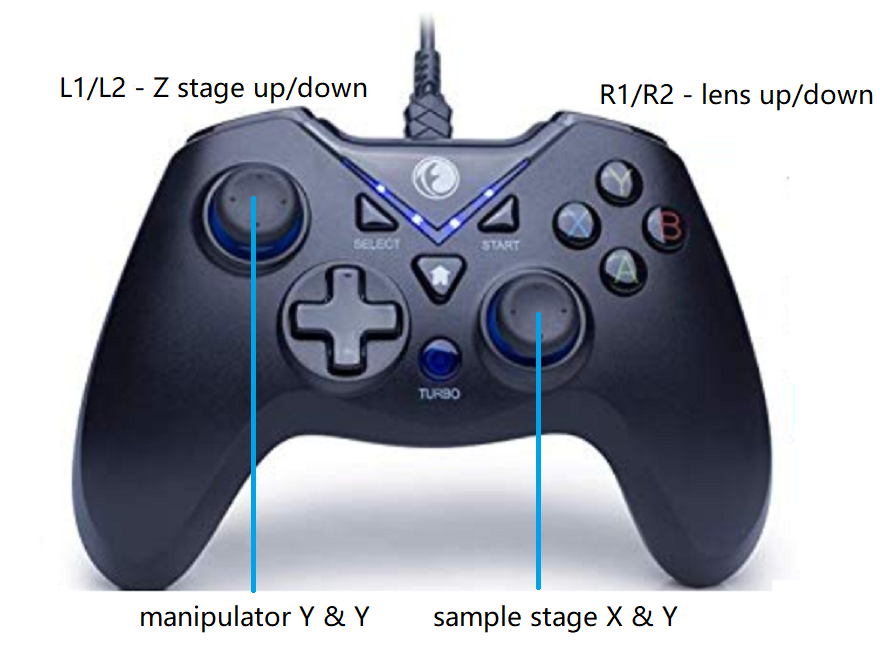
Use *scanArea(size in mm)*, which does the translational motion for us. We wait until it reaches the target position specified in the argument and then we focus it. We may also adjust the position when we focus.

1. It is recommended to calibrate the linear stage after the load changes. This can be done by using the function *calibrate()*, which will give the number of steps for forward and backward motion. Then go to *calibrationResult.xlsx* and fill this in to calculate pixel per step. Write this parameter into the variable *pxPerStep* in *main.py*

B – small movement

A – medium movement

X - large movement



START – proceed