1. **Physical Setup**

    a. Attach channels to MCS
    b. Attach actuators to channels
        i. First channel is X (long actuator mounted to optics block), 2nd channel is Y, 3rd channel is Z (updown/most important actuator)
    c. Plug in ethernet from MCS to PC
    d. Plug MCS into outlet (make sure power is off)
        i. Then power on
    e. For calibration place a gold bar with tip at known X-Y position.
        i. Connect arduino to PC with a USB-C to USB-A wire
        ii. Attach the arduino's pin 7 to the gold bar - no need for a ground wire.

2. **Digital Setup**

    First time?:

    There are multiple ways to control actuators, via code and via a GUI from Smaract. The MCS is connected with ethernet so knowing the network port is important.

    1. Open up the MCSNetworkInterfaceConfig.
    2. Should be an available interface with (192.168.1.2). Select that number.
    3. Get the current configuration. It should match this:
        IP: 192.168.001.200
        GW: 192.168.001.001
        SN: 255.255.000.000
        Port: 05000
    4. If it doesn't match. You can reset to default configuration or set a new configuration to match. If you are really adamant about working on an ip other than the default. You will NEED to change the parameters in the python file "ActScript.py". Just make sure to change the ip and port address in lines 16 & 17.
    5. Check to make sure this is working with the provided GUI
        a. Open MCSConfiguration
            i. If this doesn't open you have an incomplete software package. The software provided FROM smaract is incomplete. I provided a completed working file that includes that python files for operation
        b. Go to "select MCS" and have input: network:192.168.1.200:05000
        c. Connect

    If connecting in MCSConfiguration does not work I recommend restarting your computer. Make sure the physical setup was done in correct order (turn on power last). You can also try to disable the windows firewall (may not be possible in your circumstances). The MCS Configuration-Tool Manual.pdf can also be used for reference if you have any problems/questions.

If not using the default multimeter, then there is a required code change in the Multimeter.py file. It's currently configured for an Arduino R4 Uno. Any arduino should work to replace, the function could also be used for a digital multimeter continuity function. It's just one function that must return True when continuity is reached.

After ensuring that the ip configuration is up and running (a USB connection will require some changes to the python file) then the code can begin.

If using the provided multimeter then no code changes necessary.

There are two primary functions. An alignment and calibration function. The calibration function is for the gold bar method, and will require user operation.

All code must run on Python 32 bit, 64 bit will <u>not</u> work. Pandas version is 2.0.3. Newer versions are not 32 bit compatible.

Using Arduino UNO R4 For multimeter:

Required items:

- Arduino Uno R4
- Male USB-a to Male USB-c
- Minimum of 1 pinned wires

The arduino code name "ArdSketch.cpp" should be uploaded to the Arduino via the publicly available Arduino IDE software. For testing purposes, you can view the "serial monitor" in the arduino IDE. However, this should be closed when running the python code.

3. **Gold Bar Calibration User Guide**
    a. Z-calibration
        i. Attach multimeter hardware. If using an arduino, boot up the arduino IDE and run cpp code.
        ii. Run calibrate_aperture() from ActScript.py
        iii. The initial X-Y calibration position is very important. Make sure the aperture slit is positioned above the gold bar, but the knife edge is NOT positioned over the gold bar.
        iv. Continue through program
        v. The output will be in xyz_positions.csv. The function also returns the XYZ values.
    b. Tilt test
        i. Attach multimeter hardware. If using an arduino, boot up the arduino IDE and run cpp code.
        ii. Run tilt_calibrate() from ActScript.py

iii. The initial X-Y calibration position is very important. Make sure the aperture slit is positioned above the gold bar, but the knife edge is NOT positioned over the gold bar. The aperture will move along the X-axis (the long actuator direction), and will repeatedly find the position.
iv. Continue through program
v. The output will be in x_z_positions.csv. The function also returns the XZ values.

c. Align aperture
i. This will move the aperture to the desired position 10 microns above the wafer. No need for multimeter hardware. You need to have previously run the Z-calibration and have an xyz_positions.csv file.
ii. Run align_aperture()
iii. Continue through program