# 5-Axis 3D Printer User Manual

This guide will cover the required steps for setting up the 5-Axis 3D Printer as well as steps for utilizing the program. If you do not need to set the program up on your own computer, you can skip to “Overall Process”.

It is presumed that you downloaded this manual from the open Github repository, so this guide assumes you have successfully downloaded and unzipped the program.

### **Set-up**

1. Before utilizing this program, you will want to install python on your computer.   
   Python installations can be found at <https://www.python.org/>, no particular version is required, using the most up-to-date version should suffice.
2. Run the “Set-up for Program” file found in the main folder. This will automatically install all necessary components needed to run the remaining program.
3. TROUBLESHOOTING:
   1. If your program will not run after these steps, ensure that the following folders are present:
      1. “INPUT STL FILE HERE” in the general files.
      2. “Finished G-Code”, “Finished Models”, “INPUT STL FILE HERE” in SLICER.
      3. “G-Code (Finished)”, “G-Code (Translated)”, “INPUT G-CODE HERE” in G-Code-Converter.
   2. If your stl files are causing errors when running the program, consider reimporting them or port them from a different program. Certain programs will port the stl file differently. We specifically utilized Ultimaker Cura to export any utilized STL files.

### **Overall Process**

Now that you have all of the required files, you may now utilize the entire program.

1. On the file level of “fullTransition.py” and “Run Overall Program”, insert any STL File into the folder “INPUT STL FILE HERE”.   
   The program can handle more than one of these files being in the input folders, but it will only handle one stl file per run.   
   Thus, it’s recommended to utilize only one file when running any segment.
2. Run the batch file labeled “Run Overall Program”, and a command prompt should open. The program will automatically start the “Slicer/Simulation” process.
3. After this process concludes, the program will utilize a utility script to move the finished G-Code over to the G-Code-Converter file, where it will start the “G-Code Translation” process.
4. Once this process has finished, you have now seen a simulation of the G-Code and possess a translated version of the code.

### **Slicer/Simulation Process**

The following processes can be run individually, but are additionally utilized during the “Overall Process”, as referenced above.

1. On the file level of “slice.py” and “Run Slicer Program”, insert any STL File into the folder “INPUT STL FILE HERE”. Similar rules apply as in the Overall Process (see above).
2. Run the program by clicking on the “Run Slicer Program” batch file. Once the program has started, you will see an output depicting the model being translated into G-Code inputs. This is an indication that the STL model was successfully read, and also acts as an indication of the simulation length (a large amount of inputs will take longer to render).
3. You will then be asked if you’d like to simulate the model. This is generally up to user taste, as it’s unnecessary for some models, but can be helpful for others. If you do decide to simulate the model, be aware that this process can take some time, especially for larger models.
4. Once either choice is made and the process has finished, the process with either close or you will be left with a finished model, with the process prompting you to press any key. Doing so will finish the process and close the model.

### **G-Code Translation Process**

The following processes can be run individually, but are additionally utilized during the “Overall Process”, as referenced above.

1. On the file level of “G-converter.py” and “Run G-Code Program”, insert any generalized G-Code into “INPUT G-CODE HERE”.
2. Run the program by clicking the “Run G-Code Program” batch file. This process should continue without input until the translation has finished.

1. You will then receive the prompt "INSERT ADDITIONAL CONFIGURATIONS IN G-CODE? (Y/N)". Typing “N” will continue to step 4, but typing “Y” will let you use additional specifications, noted in 3a-g. You may also type “SKIP” to avoid putting specific values into your G-Code.
   1. Probe Speed

Helps to determine the precise precision of specific printed items.

Increasing this speed increases the probing speed.

Default Value: 6000

* 1. Regular Maximum Speed

The fastest rate at which the motors can travel.

Faster speeds transmit more kinetic energy to the print, so the print quality can be affected. Slower speeds could be necessary depending on the build of the model.

Default Values: 300, 600, 120, 240, 45, 4

* 1. Steps Per Unit

The number of “steps” between each movement.

Calculates a number of different resources, including the filament printing itself. Modifying this will lead to

Default Values: 25, 5, 25, 2.381, 1, 6, 1

* 1. Maximum Instantaneous Speed Changes (mm/min)

When a stepper motor changes too quickly, it can end up skipping steps.

Changing this value will change the stability of the system.

Default Values: 200, 200, 20, 20, 20, 120

* 1. Fan Options

Sets fan power for each fan.

Ranges from 0 to 255

Default Value: 255, 255, 255

* 1. Temperature Limit

The limit for the heater section of the printer.

This may need to be changed depending on the utilized filament.

This also modifies the general range of the heater sections, which should be monitored closely.

Default Value: 262 and 300.

* 1. Idle Timeout

Sets the time before motors are disabled. General value is set low, so adjust to fit your own time (in seconds)  
Default Value: 30

Recommended Value (12 hours): 43200

1. At this point, you will be prompted on if you’d like to package your G-Code or not. Typing “Y” will produce an individual file complete with config folder, whereas typing “N” will end the program. Either way, your G-Code will have been translated and placed into a folder.