**TinyG Setup**

This document is meant to supplement the information used at the following websites to download and install TinyG.

* Install hardware according to - <https://github.com/synthetos/TinyG/wiki/TinyG-Start> (you do not need a fan nor a program debugger)
* Connect Tiny G according to - <https://github.com/synthetos/TinyG/wiki/Connecting-TinyG>

Refer to each section when encountered in the tinyG setup documents in the above URLs.

Stepper Power Supply

1. Wire a 24V power supply (White: L, Black: N, Green: Ground)
2. Check output of power supply is 24V.
3. Wire red wire (gauge: )from +V output of power supply to +Vmot terminal of TinyG
4. Wire black wire from -V output of power supply to GND terminal of TinyG

<https://github.com/synthetos/TinyG/wiki/Connecting-TinyG>

Limit Switches, Homing, Travel Limits

1. Switches: SS-10GL or similar (SPDT)
2. Drill holes (#50) and tap 2-56
3. Install switches with 2-56x1/2 SHCS
4. Wire C terminals together on all switches and connect this to GND input
5. Wire the NO terminal from each switch to corresponding spot on TinyG
6. By convention:
   1. Ymin = Front
   2. Ymax = Back,
   3. Xmin = Left,
   4. Xmax = Right
   5. Zmin =Down
   6. Zmax = Up
   7. Note: These are defined as the tool movements. In our case the Y axis is not attached to the tool, but is attached to the base. Thus, the baseplate moving forward is the same as the tool moving back. Because of this, we need to switch the direction of the Y stepper motor, either by wiring it opposite of the X and Z steppers or changing it in software. Likewise, we will use the rear limit switch as the forward limit/home of the tool.
   8. To change the y orientation in software use the polarity command for motor 2 $2po=0
7. Discussion on limits: <https://github.com/synthetos/TinyG/wiki/Homing-and-Limits-Description-and-Operation>
8. The following settings should be set for a homing and limit configuration
   1. $ST Switch Type 0=NO  
      $XSN X Minimum Switch Mode 3=limit-and-homing (See Modes, below)  
      $XSX X Maximum Switch Mode 2=limit-only  
      $YSN Y Minimum Switch Mode 3=limit-and-homing  
      $YSX Y Maximum Switch Mode 2=limit-only  
      $ZSN Z Minimum Switch Mode 0=disabled  
      $ZSX Z Maximum Switch Mode 3=limit-and-homing  
      $ASN A Minimum Switch Mode 0=disabled  
      $ASX A Maximum Switch Mode 0=disabled
9. Check system units are mm (type ‘?’ and enter)
10. Double check and set the distance moved per revolution (1.25mm for a M8x1.25mm leadscrew. Can double check pitch with calipers)
    1. $1tr can be used to check and set the configuration
11. Set axis limits
    1. $ytm=100 (units are mm)
    2. $xtm=190
    3. $ztm=0 (z max)
    4. $ztn=-45

Tuning Motor current and velocity

<https://github.com/synthetos/TinyG/wiki/TinyG-Tuning>

1. Our machine worked with the default G0 speeds except for x, which seems to have more friction in it. The default max traverse speed is 800mm/min which can be found by typing in the $xvm, $yvm, and $zvm commands.
2. Start with the current potentiometer at the middle position - 12:00. Never exceed the potentiometers' range of motion, which is 270 degrees, or from about 8:00 to about 4:00.
3. Set a maximum jerk value ($xjm $yjm $zjm) where you can audibly hear the motor come up to speed. A value of 20,000,000 mm/min^3 was the default for our TinyG. Note: commas are accepted by the configuration routines in text mode, but not in JSON mode.
4. Test traverse rate with a G0, such as G0 X100. The motor should accelerate, cruise at speed, then decelerate to a stop. The motor should not stall or fail to start. Lower the velocity if this is the case.
   1. I lowered $xvm to 600 because it would stall half way through the X- traverse. I did the same for $yvm.
5. If the the motor more or less works but seems to be dropping steps it could be any of the mechanical system (too much friction), the current setting, or the velocity max being too high; or possibly some combination of all three. Experimentation is required. It's best to try to fix them in that order. Start with the mechanical system, then the current, then the setting.
6. We also need to set the absolute maximum feedrates, which default to 800mm/min. I’ll lower them all to 500mm/min, which is lower than the max traverse rate.
   1. This is set with the $xfr, $yfr, and $zfr commands. I set them all to 500.

Coordinate systems

1. <https://github.com/synthetos/TinyG/wiki/Coordinate-Systems>
2. The homing command G28.2 X0 Y0 Z0 runs the machine to the top, left, and front to the limit/homing switches and sets the machine origin here. This sets the absolute coordinate system of the machine (called G53)
3. However, having the origin at an extreme position is not very user friendly, so we will use offset coordinate systems to place the machine in the center of the limits of motion. There are 6 available offset coordinate systems (G54-G59, with ID’s 1-6, so G54 is 1). By convention G54 is left as a 0 offset from G53, so it represents the extreme homed coordinate system
4. I set G55 to the center of the XY motion with the z in the extreme up poisition:
   1. Sending the command G10 L2 sets the coordinate system offset. Parameters that must also be included are the coordinate system ID P1 through P6 (corresponding to the G54-G59 coordinate systems) and the offset values
   2. G10 L2 P1 X0 Y0 will set G54 to match the absolute coordinate system G53
   3. G10 L2 P2 X95 Y50 will set G55 to an offset of 95 in the x-dir and 50 in the y-dir
5. To change to one of the offset coordinate systems, just type it in. So, if you type G55, you’ll now be in the coordinate system centered in the XY motion limits. Once you are in G55, you can type G0 X0 Y0 Z0 to move to that origin.
6. To find out which coordinate system you are in just type ‘?’ for the machine status, which lists the current coordinate system.
7. The nice thing about the G54-G59 offset coordinate systems is that they are saved to TinyG’s memory, so once the homing operation is performed at system turn on, one can immediately change to a more convenient coordinate system.

Coolterm Settings notes

1. Coolterm is a useful serial interface tool for setting up TinyG. You can do all of this inside the python code, but Coolterm is a little less intimidating at first
2. Download it here
   1. <http://freeware.the-meiers.org/>
3. Instructions for setup with TinyG here
   1. <https://github.com/synthetos/TinyG/wiki/Connecting-TinyG>
4. 115,200 baud
5. 8 data bits
6. no parity
7. 1 stop bit
8. XON flow control (or use CTS, but make sure the $ex setting agrees, **I used CTS**)
9. It's also useful to set the following - but not strictly necessary
10. Options/Terminal - Line Mode
11. Options/Enter Key Emulation - CR
12. Hit OK to leave the Options menu