**TinyG Start, Connecting, Drive Testing and Board Diagram**

*Frode Lillerud edited this page on Jan 9 ·* [*108 revisions*](https://github.com/synthetos/TinyG/wiki/TinyG-Start/_history)

*(copied from Synthetos github; https://github.com/synthetos/TinyG/wiki/TinyG-Start)*

If you have a TinyG v8 this is the place to start. The board revision is printed on the silkscreen at the edge of the board. If you do not have a v8 board you may want one of these links instead:

* [TinyG version 7](https://github.com/synthetos/TinyG/wiki/TinyG-Start-v7/)
* [TinyG version 6 or earlier](https://github.com/synthetos/TinyG/wiki/TinyG-Start-v6-and-Earlier)

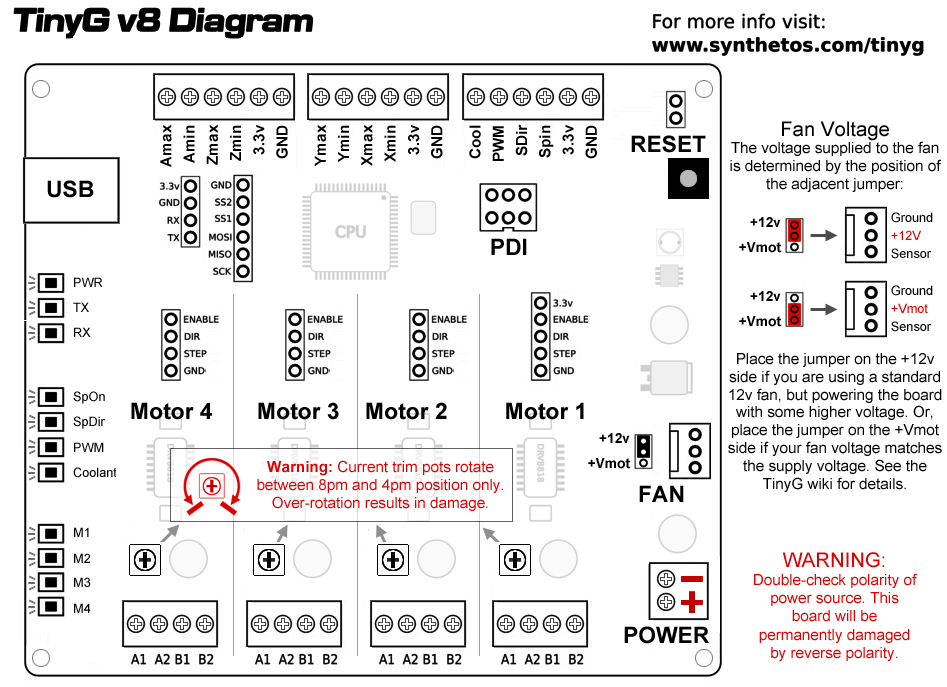
Here's some background if you want to know [about TinyG](https://github.com/synthetos/TinyG/wiki/What-is-TinyG)  
Here's some additional information about [Initial Setup](https://github.com/synthetos/TinyG/wiki/Initial-Setup)  
Here is a list of [differences between the v7 and v8 boards](https://github.com/synthetos/TinyG/wiki/TinyG-Start#changes-in-v8-from-v7)

Here are the steps to get started. We recommend following them in order.

* [What You Need](https://github.com/synthetos/TinyG/wiki/TinyG-Start#what-you-need) (TinyG board, motors, power supply, fan (optional))
* [Connecting TinyG](https://github.com/synthetos/TinyG/wiki/connecting-tinyG)
* [Setting Up Your Machine and Configuring TinyG](https://github.com/synthetos/TinyG/wiki/tinyg-configuration)
  + About your CNC
    - Shapeoko2 section
  + Configuring motors
  + Configuring axes
  + Configuring system and communication parameters

**Read This First**

The getting started page is your first place to go to figure out what you need to get to get your TinyG up and running quickly. However before we dive into hooking up wires, configuring, and running Gcode files the image below is a "diagram" of the important sections / parts of your TinyG board.



To highlight a few things in the above diagram:

* The **MOST IMPORTANT** thing to do is to wire your power input correctly. The input will take up to 30 volts, but most people use 24, 19 or 12 volt power supplies. Double check that the polarity for the GND and Vmot are correct BEFORE plugging in your TinyG board. If you have ANY doubt about the power supply output please check it with a volt meter first.
* Power output for the PC fan is important to make sure you have right! Failure to set your fan jumper may result in providing 24v to a 12v fan and maybe blowing it up. See more about this below.
* All logic input voltages are limited to 3.3v MAX!
* Start with the current setting trim pots in the middle, 6:00, straight-up-and-down position. These are single-turn trim pots that travel about 270 degrees. DO NOT over torque the trim pots!

**What You Need**

Here is what you are going to need in order to use TinyG:

* [**TinyG board**](http://synthetos.myshopify.com/products/tinyg)
* [**Stepper motors**](https://github.com/synthetos/TinyG/wiki/TinyG-Start#stepper-motors)
* [**Power supply**](https://github.com/synthetos/TinyG/wiki/TinyG-Start#power-supply)

Optional

* **Fan** - A 12VDC or 24VDC fan is recommended if the motors are pulling more than 2 amps per winding, especially if the board is in an enclosure.
* **Programmer/Debugger** You can use TinyG's [Bootloader](https://github.com/synthetos/TinyG/wiki/TinyG-Boot-Loader) for firmware updates, so you don;t need a programmer. But if you want to do real-time debugging or serious development we recommend picking up a programmer.

**TinyG Board**

You can get the TinyG controller board fully assembled from the [Synthetos Store](https://synthetos.myshopify.com/products/tinyg). Details of the board are in the diagram at the beginning of this page.

**Stepper Motors**

The next thing you chose (or are chosen by your machine) are the stepper motors. TinyG will work with most NEMA17 and NEMA23 motors, and will work with both bipolar and unipolar stepper motors. This covers most small motors you are likely to encounter. It will not work with motors wired as 5 wire "star configurations", so avoid these.

We have never found a NEMA17 that would not work with TinyG, and almost every NEMA23 we have tried will work if rated up to about 3 amps per winding. The drivers on the TinyG v8 are rated to 2.5A per winding per motor, but will actually do up to 3 amps with proper cooling. Almost every NEMA17 motor we have seen draws less than 2 amps. The exceptions are some of those very long NEMA17s with torque ratings above 90 Oz-in. Most NEMA 23's we are familiar with are between 2 and 3 amps (and should therefore be fan cooled).

We also routinely run NEMA34's, but not in high mechanical load situations. The motor's rated voltage is irrelevant and can be ignored. When running NEMA23's or any motor that draws more than 2 amps we recommend fan cooling. Note that most of the heat comes off the bottom copper, so be sure to provide air circulation for the **bottom of the board** as well as the top.

Some of out favorite sources for stepper motors are:

* [Automation Technology Inc. (Keling)](http://www.automationtechnologiesinc.com/)
* [Alltronics](http://www.alltronics.com/cgi-bin/category/55%20www.alltronics.com/cgi-bin/category/55)
* [MPJA](http://www.mpja.com/Stepper-Motors/products/101/%20www.mpja.com/Stepper-Motors/products/101/)
* [All Electronics](<http://www.allelectronics.com/make-a-store/category/400/Motors/1.html>
* [Sparkfun](https://www.sparkfun.com/categories/178)
* [Phidgets](http://www.phidgets.com/products.php?category=23)
* [Oriental Motor Company](http://www.omc-stepperonline.com/)

**Power Supply**

We highly recommend a 24v-30v power supply. While 12 volt operation is possible and entirely fine, running with a 24v power supply will allow the motors to be more responsive and actually run cooler (ironically).

You might think that 4 motors at 2 amps per winding would require 4 motors \* 2 windings \* 2 amps = 16 amps to drive, but you'd be wrong. 4 to 4.5 amps or above will handle this as not all motors + phases are ever maxed out at the same time. At 24 volts we like to have at least 4.5 amps for NEMA17's and 6 or more is recommended for NEMA23s.

Here are a couple power supplies we like:

* [Meanwell NES-150-24](http://www.mouser.com/ProductDetail/Mean-Well/NES150-24/?qs=sGAEpiMZZMsPs3th5F8koDNPbuqd%252bfezne6r6bnnXjA%3d)
* [Meanwell NES-350-24](http://www.mouser.com/ProductDetail/Mean-Well/NES-350-24/?qs=%2fha2pyFaduhxfhzsenBkIkgMfhBr0hSVdTJWNZMLFL2wp6eI7VH7oQ%3d%3d)

You can usually hunt around and find them for < $50. They also make lower amperage supplies that are cheaper.

PS Caveat: Be sure to switch the AC input from 240v to 120v if you are in the US or a 120v mains country. Most PSs will still work, but will audibly "click" and lose power if you don't set this correctly. Also, we've gotten Meanwell clones ordering from eBay - like Meigwei. Sheesh - they said I was ordering a Meanwell.

**Cooling**

The main heatsinking provided for TinyG is the expanse of 2 oz. copper on the bottom and top of the board. You can see this by inspection. This is usually sufficient for NEMA17 installations and many NEMA23 applications. If you experience thermal shutdown or if you feel the chips are running too hot we recommend fan cooling.

The TI drivers on the TinyG are incredibly robust and will shut down in case of over-current instead of blowing up (unlike some other brands that shall remain nameless). But you don't want to go into thermal shutdown as it will will ruin your job even though the board is still OK. Thermal shutdown is evidenced by anything from a slow on-off cycling of the motor power, getting shorter as the current raises, to a stutter in extreme cases. The chips will be quite hot to the touch.

Fan cooling is the most effective way to cool and far more effective than heatsinking. Just putting little heatsinks on the top doesn't do much good. We used to sell TinyG with these but don't any more.

The TinyGv8's come equipped with a 3-pin fan connector that can be used to power a standard 12vdc PC fan, or a 24vdc fan - depending on a jumper setting and your board voltage (Vmot). Please read the silkscreen designators as below:

* The +12v jumper position connects the fan header to the on-board 12 volt regulator. This should be used if you are running a 12 volt fan and your board power (Vmot) is greater than 14 volts. This is the default jumper position and your v8 should have been shipped with the jumper in this position.
* The +Vmot jumper position connects the fan header directly to the board power (Vmot). In this case your fan voltage **needs to be the same** as your motor voltage, be it 12 volts, 24 volts, or anything else.  
  **Use this side with caution as applying 24 volts to some 12 volt fans will burn them out.**

**Programmer/Debugger**

If you are doing serious development or want to do real-time debugging we recommend getting a programmer debugger. The unit of choice these days is the [ATMEL-ICE-BASIC programmer](http://www.digikey.com/product-detail/en/ATATMEL-ICE-BASIC/ATATMEL-ICE-BASIC-ND/4753381). The BASIC sells for about $50 and will program and debug AVRs (like the Xmega on the TinyG v8) as well as the Atmel ARM chip on the pre-release TinyG v9. Note that as of now AVRdude will not program using the ATMEL-ICE. In addition to multi-chip programming, the Atmel-ICE offers real-time debugging via Atmel Studio or Open OCD.

The older [Atmel AVRISP MKII programmer](http://www.mouser.com/Search/ProductDetail.aspx?qs=sGAEpiMZZMsaJrqdZ%252b6EWyua%252bG%2FwcOQP26MNKN%252bCIDE%3D) will also program the Xmega, and does work with AVRdude. But it doesn't do debugging and won't work with the ARM chips.

Note that older AVR ICSP programmers will not work.

**Connecting TinyG**

At this point you can move on to [Connecting TinyG](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG)

**Changes in v8 from v7**

* LEDs moved to a single edge for better mounting / readout
* USB changed to USB-B (from mini) for better mechanical integrity
* IO connections changed from 0.100 headers to screw terminal blocks
* step/direction/enable signals broken out on headers for external stepper drivers
* logic power supply is now a switcher (as opposed to a linear) on the v7 for better thermal characteristics
* FTDI USB chip changed to FT230X from FT232R
* switch inputs have RC circuit built in for better noise rejection
* RS-485 removed and replaced with SPI connection
* transient voltage suppressors added to Vmotor and USB lines
* mounting hole pattern is slightly different to accommodate about 1/4" growth in the board

**Connecting TinyG**

Alden Hart edited this page on Aug 6 · [94 revisions](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG/_history)

If you are on this page you have already selected your motors and power supply and are ready to hook them up.

Connecting TinyG Steps

* [Connect power](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG#connect-power)
* [Connect USB](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG#establish-usb-connection)
* [Test connection](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG#test-connection-to-tinyg)
* [Wire your motors](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG#wire-your-motors)
* [Set motor current](https://github.com/synthetos/TinyG/wiki/Connecting-TinyG#setting-motor-current)
* [Test drive](https://github.com/synthetos/TinyG/wiki/Test-Drive-TinyG)

## Connect Power

The **MOST IMPORTANT** thing to do is to wire your power input correctly. So check and double check this before actually turning on the power.

**THIS PAGE ASSUMES THAT THE POWER SUPPLY'S BLACK IS GROUND AND RED OR YELLOW IS +24VOLT (HOT)**. Not all power supplies adhere to this convention so you still need to check this carefully. Read below:

1. Check you have the correct power supply. You should have a DC power supply between 12 and 30 volts --- 24 volts is ideal. It should be capable of providing 4 to 15 amps.
2. If you have a power supply where you are responsible for wiring the AC lines (like a Meanwell or some other supply with a terminal strip) - make sure the AC is wired correctly. By convention, the AC neutral is the white wire in the power cord, the hot is the black wire and the ground is green. Make sure the ground is connected. Not only for safety, but also because you may experience spurious resets if the system is not properly grounded.
3. BEFORE connecting it to TinyG, turn on the power supply and make sure you have correct voltage. Make sure it is DC, not AC. Hopefully you have a volt meter or can get your hands on one. If you don't you might consider a trip to the Radar Shed for something like [this one from Sparkfun](https://www.sparkfun.com/products/12966) or [this one from Adafruit](http://www.adafruit.com/products/2034).
4. With the power supply off, wire the negative to the GND terminal of the power input block, and the positive to the +Vmot side. Negative is BLACK by convention and positive is RED or YELLOW, or some other bright color. Check you have the correct voltage and polarity before connecting the power supply to TinyG. We have encountered power supplies that do not obey these conventions, and incorrect voltage or polarity can destroy your board!
5. Turn on the power supply. If the blue light turns on this is a good sign. If not, blow away the smoke and send us an email.

## Establish USB connection

Next establish USB connection with your host computer.

#### Install FTDI Drivers

* If you do not have the FTDI VCP USB drivers for your host computer you will need to install them. It's possible they are already on your system as many applications use them, including the older Arduinos.
* PC Installation - Get the latest Windows driver from the [FDTI VCP Driver Page](http://www.ftdichip.com/Drivers/VCP.htm). You want the VCP driver for your host, not some of the other drivers they offer. Install and reboot your system.
* MAC Installation - As of Mavericks (OSX 10.9.x) OSX will appear to communicate with TinyG without loading the FTDI supplied drivers. However, the native mac drivers do not perform flow control with the FTDI on the v8, so they will not work. You need the VCP 2.3 driver from the [FDTI VCP Driver Page](http://www.ftdichip.com/Drivers/VCP.htm). You must also reboot your system once the installation is finished (they don't tell you this). If installed correctly you should see something like usbserial-DA00Y5MM when you scan for new serial ports.

Note that the Mac native driver will also show the usbserial port, so the only real way to tell if you have the FTDI driver is to look in APPLE\_IN\_THE\_UPPER\_LEFT\_CORNER / About This Mac / System Report / Software / Extensions for FTDIUSBSerialDriver, version 2.3)

#### Set up Coolterm

Once you have the FTDI drivers in place you will want Roger Meier's Coolterm to connect and test your tinyg. Coolterm is a terminal emulator that provides command line access to TinyG and can also stream files to TinyG. We use Coolterm as the preferred way to test the board without introducing variables from advanced UIs and host controllers such as [ChiliPeppr](https://github.com/synthetos/TinyG/wiki/Chilipeppr), which is a better way to actually run jobs once you have the system set up.

* Download and install [Coolterm](http://freeware.the-meiers.org/). For the Mac we have noticed that the latest version (1.4.5) does not do well on large file transmissions. So we still use version 1.4.3. This is available on the Coolterm page under [Previous Releases](http://freeware.the-meiers.org/previous/).
* Go to the Options menu and Re-Scan Serial Ports. You should see something like usbserial-AE01DVWD or COM12. Configure the following settings:
  + 115,200 baud
  + 8 data bits
  + no parity
  + 1 stop bit
  + XON flow control (or use CTS, but make sure the $ex setting agrees)
* It's also useful to set the following - but not strictly necessary
  + Options/Terminal - Line Mode
  + Options/Enter Key Emulation - CR
* Hit OK to leave the Options menu

#### Test Connection to TinyG

* Hit the "Connect" button. Enter a few carriage returns. TinyG should respond with prompts. If not, hit the reset button on the TinyG. You should see some JSON startup messages wrapped in JSON curly braces something like this:
* {"r":{"fb":371.030,"fv":0.950,"hv":7.000,"id":"9H3583-RMP","msg":"Loading configs from EEPROM","f":[1,15,0,8891]}}
* {"r":{"fb":371.030,"fv":0.950,"hv":7.000,"id":"9H3583-RMP","msg":"SYSTEM READY","f":[1,0,0,8820]}}
* tinyg [mm] ok>
* If not, go back and check your driver, your serial settings, your USB cable, and that you have a blue light and not blue smoke. For help from the command line enter 'h' for TinyG help, or '$h' for configuration help
* **If you simply cannot connect try powering down the TinyG and quitting Coolterm (or your terminal program), powering back up and restarting the terminal. There is a known bug in the FTDI drivers that can cause this sometimes.**

**Verify Flow Control** Once you are connected it's a good idea to verify you have the correct flow control settings

* The default flow control for TinyG is CTS, which uses the following settings:
  + Coolterm: CTS checked
  + TinyG $ex=2
* XON/XOFF flow control is also available. Both Coolterm and TinyG must be configured the with these settings:
  + Coolterm: XON checked
  + TinyG: $ex=1

## Wire Your Motors

It's best to wire your motors when they are not yet in your machine. This way you can test drive them without worrying about mechanical issues such as excessive friction or crashing into the side of the machine.

If your motors are already in a machine sometimes it's easier just to use one or more spare motors rather than remove them. It will not damage the TinyG board to drive an axis with no motor attached.

But first, turn off the power to TinyG. Never connect or disconnect anything (except possibly USB) with the power on.

### Motor Connectors (was "Synthetos Connector Kit")

Most TinyG v7s and above have screw terminals and therefore do not need a motor connector kit. If you have a board that has quick release headers on it here are the mating parts:

* Terminal Shell - Molex 09-50-3041 (need 1 per motor)
* Crimp Terminals - Molex 08-50-0134 (need 4 per motor, and usually some extras)

You need 4 terminals for each housing (shell). You can get these from Mouser or DigiKey. TE/Amp also makes some nice parts with the shell and terminals all-in-one:

* TE/Amp 3-644463-4 (need one per motor)

### Find the Coil Pairs

The first step is make sure you know which wires are connected to the same coil.

The following are common motor types:

* Bipolar motors have 4 wires (2 pairs). This wire color code is typical for many bipolar motors:

| **Wire Color** | **Winding** |
| --- | --- |
| Green | Winding A1 |
| Black | Winding A2 |
| Red | Winding B1 |
| Blue | Winding B2 |

* Unipolar motors typically have 6 wires also connected to to 2 windings. The additional wires are the center taps for each winding. These can usually be ignore (not connected). This wire color code is typical for many unipolar motors:

| **Wire Color** | **Winding** |
| --- | --- |
| Green | Winding A1 |
| Yellow | Center tap A |
| Black | Winding A2 |
| Red | Winding B1 |
| White | Center tap B |
| Blue | Winding B2 |

* 8 wire motors: 8 wire motors are usually wired as 2 sets of bipolar windings - essentially these are 2 bipolars wired together. They can be wired by finding which pairs are in parallel with each other and either not using the second winding, or wiring it in parallel for more torque (and more current draw). There is no common color code for 8 wire motors. (Please correct me if I'm wrong).
* 5 wire motors are in a "star" configuration that has a common ground and require a specialized driver. TinyG cannot drive 5 wire steppers.

To determine which wires go to which windings use a volt meter to verify that green and black connect together, and red and blue connect together, and that they don't connect to the other pair. Typical DC resistance across a winding is about 1 to 5 ohms. If you have a Unipolar motor the center taps will show 1/2 the resistance of the full winding.

Here's a shortcut to finding wire pairs for a bipolar (4 wire) motor.

Spin your stepper motor with your fingers. Depending on the size / holding torque this could be easy or pretty hard. All you really want from this is to get a feel how the motor spins without any of the wires connected to each other. Now that you know how hard it is to spin with your fingers, connect 2 wires together. Just pick any two. Try to spin the motor again. If it feels the same then more than likely these are NOT connected to the same coil. Disconnect these wires. Connect one of the other wires to one of the first wire pairs you tried. Try to spin the motors again. This should be much harder. If so, you have found your wire pairs. Tape these 2 together (not wired but just taped to group them). Tape the remaining 2 wires together as well.

Unipolars are a bit more complicated, but not much. To do a series wiring find the outer taps of each coil. These are often color coded by convention (see below). Using a volt meter to find the resistance across the outer pair. The resistance between the center tap an an outer tap will be 1/2 the resistance between the outer taps.

Some useful information on wiring steppers can be found here: <http://reprap.org/wiki/Stepper_motor>

### Connector Pinouts

Back to connecting your TinyG. Each of the 4 motor outputs have a four pin terminal block wired as:

* A1
* A2
* B1
* B2

Attach one pair to A1/A2 and the other pair to B1/B2. If your motor spins the wrong way, you can swap A1 with A2 and vice versa. This can also be done in software by using the polarity setting, e.g. $1po=1 to invert.

## Setting Motor Current

**WARNING: Do not turn the current trimpots too hard. They will break. They adjust from the 8pm position to 4pm, only.**

* Motor current for each axis is adjusted with the trimpot nearest that axis. Clockwise increases current, counter-clockwise decreases current.
* You want the motor current set slightly above the range you need for your application, but not much higher. Overdriving the motors draws more current and risks overheating or thermal shutdown.

A motor that is receiving too much current may also run "rough". If you experience this try backing off the current and see if the motor runs more smoothly.

Overcurrent also causes your motors to heat up a lot more. Stepper motors draw the most current when they are not moving so this is when they will get the hottest. Motors that run too hot run the risk of being ruined by demagnetizing.

* Follow these steps to set the motor current:
  + Set current to zero by gently turning the trimpot all the way counter-clockwise
  + Send a relatively slow move to that axis, something like g1 f400 x50
  + Adjust the current up (clockwise) until the motor moves or starts humming (meaning it's stalled)
  + Try the move again at this new setting. If it's still stalled increase the current some more and try again, or drop to a slower feed rate (F value).
  + Once the motor is running turn the current down until it stops. Mark this as the low current spot.
  + Run the motor again and adjust current up until it runs rough or goes into thermal shutdown (see below).
  + Now back off until the cycling stops and the motor runs smoothly, and mark this as your upper limit.
  + Repeat the above with a high speed move, like a g0 x100. This should also work without stalling. If it stalls you may need to adjust the maximum velocity or jerk settings - you may need to tune those. See TinyG Tuning.
  + When you actually run jobs you want to back off the current as much as you can while still running reliably to somewhere between your upper and lower limit. This will minimize board and motor heating.

If the motor starts to cycle on and off it indicates thermal shutdown is occurring. Cycling will occur under thermal shutdown, and only gets more severe as the current goes up - where it appears the motor is stuttering. Thermal shutdown is, of course, to be avoided, but we've never seen it actually damage the drivers or motors and we've seen some pretty abusive cases (actually, we caused them)!

* Now is also a good time to check out [Power Management](https://github.com/synthetos/TinyG/wiki/TinyG-Configuration-for-Firmware-Version-0.97#1pm---power-management-mode). By default power management is set to $\_pm=2: "Motor powered during a machining cycle" (i.e. when any axis is moving).

## Cooling

Most NEMA17 applications we have seen do not require any additional cooling. Passive cooling as fine as long as you allow for convection. The chip transfers most of the heat to the 2 oz. copper on the bottom of the board. Make sure you have adequate airflow (convection) across the bottom and the top of the board. Vertical orientations are nice for this.

If you are using those large NEMA17s (e.g. 125 oz.in) or NEMA23's you may need fan cooling. Again - provide adequate airflow across both sides of the board.

Heatsinks are not necessary and should be avoided unless you really need them. If you do decide you need heatsinks attach them to the bare copper on the bottom - not the tops of the chips. Use the exposed pads and be careful not to exceed the pads or you risk shorting

# Test Drive TinyG

peterarien edited this page on Dec 10, 2014 · [7 revisions](https://github.com/synthetos/TinyG/wiki/Test-Drive-TinyG/_history)

This page discusses how to test TinyG to make sure everything is working properly. (Obviously this page is still in work)

You can use the built-in tests to check that TinyG is working properly. Type $test for a list of tests.

$test=1 is the basic test that will show if the motors and output bits are running. This test is best performed with motors that are not installed in a machine. Other tests are for installed machines.

From here you may want to look at

* [TinyG Configuration](https://github.com/synthetos/TinyG/wiki/TinyG-Configuration)
* [TinyG Tuning](https://github.com/synthetos/TinyG/wiki/TinyG-Tuning)
* [Sending Gcode files to TinyG](https://github.com/synthetos/TinyG/wiki/TinyG-Sending-Files)

## Self Tests

The following self tests can be run from the command line. Format is $test=1 for text mode, {test:1} for JSON mode

$test=1 smoke test (best performed on a machine with a large table or motors not connected to a machine)

$test=2 homing test (you must trip homing switches)

$test=3 square test (a series of squares)

$test=4 arc test (some large circles)

$test=5 dwell test (moves spaced by 1 second dwells)

$test=6 feedhold test (enter ! and ~ to hold and restart, respectively)

$test=7 M codes test (M codes intermingled with moves)

$test=8 JSON test (motion test run using JSON commands)

$test=9 inverse time test

$test=10 rotary motion test

$test=11 small moves test

$test=12 slow moves test

$test=13 coordinate system offset test (G92, G54-G59)