

Introduction to CNC for a Total Novice

Part of a series by Graham Bland

Spindle speed and Laser power

Contents

What this covers (supposedly).....	1
Introduction	1
Disclaimer.....	1
Spindle speed and Laser power	2
How Grbl controls spindle speed	2
How it processes a Sxxx command	2
How Grbl controls laser power	4
Why the fixation on values of 255 or 1000.....	4
What do I use	5
A Note on Laser Mode	5

What this covers (supposedly)

A brief description of how Grbl handles spindle speed and laser power and why you can use values other than the outdated 255 or 1000 values for the spindle speed and laser power.

Introduction

I am producing this as a document, mainly because It is an edited series of notes I made, partly because I am tired of watching instructional videos on YouTube which play at Mach 2 and so I am constantly stopping, rewinding and re-watching. *Also I have been told I have a body made for Radio, a voice most suited to the Deaf and I still haven't worked out how to use the video camera on my phone properly.*

Disclaimer

Yeah nowadays it has to be included, and it makes sense as I accept NO liability for anything I may say, write or think.

1. I am a total novice (or was when I started out). **I MAKE NO CLAIM THAT ANYTHING I SAY IS CORRECT!** All this is based on searching the Internet, forum comments.... and my own experiences.
2. This is based solely on the one I have, if there are variations or different versions out there I have not seen them.
3. If you have and are using a Laser module, even the lowest powered is powerful enough to destroy your vision completely, and that of any pets, children, spectators etc. **SO EXCLUDE THEM AND USE THE GOGGLES!**

4. These are power tools, sharp pieces of metal designed to cut things while rotating at high speed and throwing small bits away also at high speed. **Keep everybody's fingers out!** A wood chip flying up your nose is nothing to be sneezed at, never mind in one of your eyes.
5. Take account of the materials you are using and take adequate precautions, dust and ashes can be harmful.
6. I work in mm. Why anyone still uses inches for design work is a mystery. *But then I still think of temperatures in Fahrenheit, anything beyond 100m in miles and property plot sizes in acres!*
7. I have been told by lots of people that my sense of humour is at the least a bit strange. I do not apologise for it.

Any corrections, clarifications or discussion welcome. You can find me on the Facebook SainSmart Genmitsu CNC Routers Group as Graham Bland. <https://www.facebook.com/groups/SainSmart.GenmitsuCNC/>
Public comments are much preferred over Messages.

Spindle speed and Laser power

It doesn't matter what laser you are using or what spindle you use, this is how Grbl works!

When people started developing laser engravers controlled by GCode and bolting lasers onto their CNC machines they needed a way to control the laser power.

They had 2 choices:

1. Develop a series of new GCode commands specifically to handle lasers
2. Re-use existing GCode commands to use Lasers.

Option 2 was chosen as it was the simplest, cheapest and easiest. So Laser power is actually set by the Sxxx command where xxx is the RPM. OK Lasers don't have an RPM, the same command has just been reused to set the laser power.

How Grbl controls spindle speed

Firstly Grbl has no idea how fast the spindle is actually rotating, it has no way of measuring it, so what it does is use what is called PWM (Pulse width modulation) to turn on and off the power to the spindle motor very rapidly. The PWM range on the controller boards is from 0-255, or 256 values (including zero) where 0 is totally off, 255 is totally on and anything in between is some off and some on. So 128 would be approximately 50% on and 50% off, as the normal spindle motor voltage is 24V and it switches very rapidly effectively this makes the motor think it is getting 12V or 50% power.

The PWM value is calculated from some parameters set on the board for maximum spindle speed and minimum spindle speed and takes the Sxxx value set in the GCode commands and calculates the PWM value accordingly. So what it is doing is varying the voltage across the motor and assuming that this translates into a rotation speed of the spindle.

The stored parameters that it uses are \$30 and \$31

- \$30 is the maximum spindle speed
- \$31 is the minimum spindle speed

You can set these values to anything you want.

How it processes a Sxxx command

Firstly xxx is the desired RPM of the spindle.

Firstly if xxx is zero then it sets the spindle to Off or a 0 PWM output, it's the same thing.

Then Grbl checks to see if xxx is above the \$30 value, if it is then it changes the xxx to the \$30 value. If your maximum spindle speed is set to 10,000 and it is processing a S15000 command then it can't reach 15000 so just sets it to full speed.

Then it checks if xxx is below the \$31 value, if it is then it changes the xxx to the \$31 value. If your minimum spindle speed is set to 500 and it is processing a S250 command then it can't reach 250RPM so just sets it to the minimum speed.

Now it calculates the required PWM value (from 0 to 255) to send to the motor controller by $PWM = \$30 / xxx * 255$.

NOTE: S0 will always set the PWM value to 0 regardless of the \$31 minimum spindle speed.

So some examples, all using 24V as the maximum spindle voltage:

\$30=255, \$31=10

S value	Adjusted Value	PWM output	Effective Spindle Voltage
0	0	0	0.00
5	10	1	0.09
10	10	1	0.09
100	100	94	8.85
254	254	253	23.81
1000	255	255	24.00
10000	255	255	24.00

\$30=1000, \$31=0

S value	Adjusted Value	PWM output	Effective Spindle Voltage
0	0	0	0.00
5	5	2	0.19
10	10	3	0.28
100	100	26	2.45
254	254	65	6.12
1000	1000	255	24.00
10000	1000	255	24.00

\$30=9250, \$31=0

S value	Adjusted Value	PWM output	Effective Spindle Voltage
0	0	0	0.00
5	5	1	0.09
10	10	1	0.09
100	100	3	0.28

254	254	7	0.66
1000	1000	28	2.64
10000	9250	255	24.00

As you can see to make the actual spindle speed match the requested Sxxx value as closely as possible then set \$30 to the actual maximum spindle speed!

I have measured the spindle speeds, on my SainSmart 3018 PRO the maximum spindle speed is ~8,250RPM, on the SainSmart 3018-PROVer it is ~9,250RPM.

How Grbl controls laser power

Well your GCode re-uses exactly the same Sxxx commands which are processed in exactly the same way based on the same \$30 and \$31 settings.

But I want to set Laser Power and it doesn't rotate! Well the spindle speed command controls the laser power and it works in exactly the same way, the laser is turned on and off rapidly by a PWM signal, it's just for the laser it is a separate control signal sent to the laser turning it on and off rather than being fed to a voltage controller for the spindle.

Exactly the same tests and calculations are applied by Grbl for a Laser as a spindle.

So you can set the \$30 and \$31 values for maximum laser power in exactly the same way, If these were \$30=9250 and \$31=0 then when using a laser S9250 would be 100% power and S0 would be 0% power with everything else between mapped in the same way.

BUT

When using a laser the power is more important and should be more accurate, when using a bit a rotation speed of 9,000 vs 10,000RPM will make a difference, but not that much. For a laser engraving 90% power is a lot different than 100% power.

Well your software that is generating the Gcode is aware of this!

It will have a value that it uses for maximum laser power, in LaserGrbl this is S-MAX, it doesn't matter what the value is **AS LONG AS IT MATCHES THE \$30 VALUE IN YOUR GRBL SETTINGS!** It can be 255, 1000, 100000 it's not important what the value is **AS LONG AS IT MATCHES THE \$30 VALUE IN YOUR GRBL SETTINGS!**

It will calculate the laser power based on 0 to the S-MAX value (the name is from LaserGrbl, the software you use may use a different name) to set the power.

Why the fixation on values of 255 or 1000

History!!!!

Earlier versions of Grbl were less sophisticated and expected a direct PWM value (0-255) to be supplied for the laser power. So this only applies to Grbl versions later than 0.9....

I am guessing for the 1,000 but again I believe that early hobby engravers and routers were lower powered than today and 1,000 was a convenient number to use rather than measure the actual spindle speed.

Then these values just stuck as they were in common use.

What do I use

Well I have 2 routers, both have lasers and different maximum spindle speeds, 8,250 and 9,250RPM and as I am lazy I have the \$30 for both set at 9,000, it doesn't make a lot of difference when routing, 99% of the time it will be at max speed anyway. But it means I can use the lasers on either without having to change the settings depending on which one I am using.

In all Laser software I have the Max power set to 9,000.

If I upgrade the spindle motor on one (or both) I may have to revisit this but until then it works and I don't have to keep on swapping the values around.

A Note on Laser Mode

Grbl has a laser mode, \$32, set it to 1 when using a laser, set it to 0 when using a spindle motor.

This is very useful and needs to be set correctly **but it ONLY does the following;**

- It turns the laser off when performing G0 (non cutting) movements.
 - These moves are used to position the bit, not to cut. When using a spindle motor then Grbl will normally leave the spindle rotating during these moves because the bit will not be touching anything so it means it doesn't have to wait for it to accelerate up to speed before starting to cut again.
 - Not true for a laser, if it is on it will burn, so it needs turning off when just moving from one place to the next.
- When making a cutting move Grbl uses the maximum acceleration values (\$120-\$122) to work out how fast it can accelerate the spindle or laser up to the requested speed. When using a laser it applies some compensation to reduce the laser power while it is accelerating and so moving more slowly than requested.

That is all it does!

Any comments, proofreading, other ideas or discussions welcomed.

Regards

Graham Bland.