

CNC_3018_Tests

August 29, 2017

1 CNC Laser Power Settings Test

1.0.1 Reference:

-<http://www.cnccookbook.com/CCNCNCCodeG20G21MetricImperialUnitConversion.htm>
- <http://marlinfw.org/meta/gcode/> - <https://github.com/grbl/grbl/wiki> -
<http://www.linuxcnc.org/docs/2.5/html/gcode/other-code.html>

2 Code:

```
In [1]: %load_ext autoreload
        %autoreload 1
```

```
In [2]: %import GCode
        %import GRBL
```

```
In [3]: cnc = GRBL.GRBL(port="/dev/cnc_3018")
```

```
In [4]: cnc.laser_mode
```

```
Out[4]: 1.0
```

```
In [5]: def init(M3 = True, feed = 200, laser = 25):
        program = GCode.GCode()
        program.G21() # Metric Units
        program.G91() # Absolute positioning.
        program.G1(F=feed) #
        program.M3(S=laser) # Laser settings.
        return program
```

```
In [6]: def end():
        program = GCode.GCode()
        program.M5() # Laser settings.
        return program
```

```
In [7]: def square(size=20):
        program = GCode.GCode()
        program.G1(X=size)
```

```

        program.G1(Y=size)
        program.G1(X=-size)
        program.G1(Y=-size)
        return program

In [8]: # Set minimal power setting to focus and position laser
        cnc.cmd("M3 S1")

Out[8]: ['ok', 'ok']

In [11]: cnc.cmd("GO X0") # Laser off

Out[11]: ['ok', 'ok']

In [10]: cnc.cmd("G1 X0") # Laser On

Out[10]: ['ok', 'ok']

In [ ]: # JogY to position table.
        cnc.cmd("GO Y+80")

In [ ]: # Write function to JogX.
        def jogx(x=10):
            program = GCode.GCode()
            program.G0(X=x)
            return program

In [ ]: cnc.run(jogx(-20))

In [ ]: for laser in [1, 10, 50, 100, 150, 255, 1024]:
        print("\t"*3+"Lasers Set To: {}".format(laser))
        program = init(M3=True, laser=laser) + square(size=10) + end()
        cnc.run(program)
        cnc.run(jogx(20))

```

3 Experimental Setup.

Assembled Chinese CNC 3018. GRBL Version

```

In [12]: cnc.cmd("$ $")

Out[12]: ['ok',
          '$0=10',
          '$1=25',
          '$2=0',
          '$3=5',
          '$4=0',
          '$5=0',
          '$6=0',

```

```
'$10=3',
'$11=0.010',
'$12=0.002',
'$13=0',
'$20=0',
'$21=1',
'$22=0',
'$23=0',
'$24=25.000',
'$25=500.000',
'$26=250',
'$27=1.000',
'$30=1000',
'$31=0',
'$32=1',
'$100=800.000',
'$101=800.000',
'$102=800.000',
'$110=800.000',
'$111=800.000',
'$112=500.000',
'$120=10.000',
'$121=10.000',
'$122=10.000',
'$130=200.000',
'$131=200.000',
'$132=200.000',
'ok']
```

```
In [13]: cnc.cmd("$#")
```

```
Out[13]: ['ok',
 '[G54:0.000,0.000,0.000]',
 '[G55:0.000,0.000,0.000]',
 '[G56:0.000,0.000,0.000]',
 '[G57:0.000,0.000,0.000]',
 '[G58:0.000,0.000,0.000]',
 '[G59:0.000,0.000,0.000]',
 '[G28:0.000,0.000,0.000]',
 '[G30:0.000,0.000,0.000]',
 '[G92:0.000,0.000,0.000]',
 '[TL0:0.000]',
 '[PRB:0.000,0.000,0.000:0]',
 'ok']
```

```
In [14]: cnc.cmd("$I")
```

```
Out[14]: ['ok', '[VER:1.1f.20170801:]', '[OPT:V,15,128]', 'ok']
```

```
In [15]: cnc.cmd("$N")
```

```
Out[15]: ['ok', '$NO=', '$N1=', 'ok']
```

4 Results

- 1 Can not be seen.
- 10 Can not be seen.
- 50 Cut through 1 piece of paper & marked one under it.
- 100 cut through 2 pieces and light etch on clipboard.
- 150 Cut through 2 pieces and dark etch on clipboard.
- 255 & 1024 look identical. Etched 'well' into clipboard.

5 Test Conclusion.

- Need to wrap Keyboard Kill with a CNC Kill.
- Need to test 10-50 in smaller increments.

```
In [ ]:
```