Laser_Tests_02

August 29, 2017

1 CNC Laser Power Settings Test 2

1.1 Objective:

- Make more functions for testing.
- Figure out constant vs dynamic power results.
- Test between 10 & 25

2 Code:

```
In [1]: import GCode
        import GRBL
```

```
In [2]: cnc = GRBL.GRBL(port="/dev/cnc_3018")

        print("Laser Mode: {}".format(cnc.laser_mode))
```

```
Laser Mode: 1.0
```

```
In [3]: from enum import IntEnum
        class Tool(IntEnum):
            SPINDLE = 0
            LASER = 1
```

```
In [4]: from enum import IntEnum
        class LaserPower(IntEnum):
            CONSTANT = 0
            DYNAMIC = 1
```

```
In [5]: LaserPower.CONSTANT
```

```
Out[5]: <LaserPower.CONSTANT: 0>
```

```
In [6]: def init(power = LaserPower(0), feed = 200, laser = 25):
            program = GCode.GCode()
            program.G21() # Metric Units
            program.G91() # Absolute positioning.
```

```
            program.G1(F=feed) #
            if power==LaserPower.CONSTANT:
                program.M3(S=laser) # Laser settings
            else:
                program.M4(S=laser) # Laser settings
            return program

In [7]: def end():
            program = GCode.GCode()
            program.M5() # Laser settings.
            return program

In [8]: 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
```

2.1 Test Setup

Position the paper & other things.

```
In [49]: cnc.cmd("M5") # Laser off

Out[49]: ['ok', 'ok']

In [64]: # Set minimal power setting to focus and position laser
         cnc.cmd("M3 S1")
         cnc.cmd("G1 X0") # Laser On

Out[64]: ['ok', 'ok']

In [51]: def jogx(x=10):
             program = GCode.GCode()
             program.G0(X=x)
             cnc.run(program)
         def jogy(y=10):
             program = GCode.GCode()
             program.G0(Y=y)
             cnc.run(program)
         def jogz(z=10):
             program = GCode.GCode()
             program.G0(Z=z)
             cnc.run(program)

In [13]: def kill_cnc():
             # TODO:
             pass
```

```
In [14]: workspace = 220 # mm

In [15]: square_size = 10
         spacing = 5

In [16]: test_space = square_size + spacing

In [17]: test_space

Out[17]: 15

In [22]: tests = workspace / test_space
         tests

Out[22]: 14.666666666666666

In [25]: import numpy as np

In [26]: tests = np.floor(tests)
         tests

Out[26]: 14.0
```

14 tests.

```
In [61]: laser_pwms = np.linspace(10, 50, 14, dtype=np.uint8)
         laser_pwms

Out[61]: array([10, 13, 16, 19, 22, 25, 28, 31, 34, 37, 40, 43, 46, 50], dtype=uint8)

In [62]: laser_powers = [LaserPower.CONSTANT, LaserPower.DYNAMIC]

In [63]: jogx(-1*test_space)

In [60]: jogy(10)

In [59]: def init(power = LaserPower(0), feed = 200, laser = 25):
             program = GCode.GCode()
             program.G21() # Metric Units
             program.G91() # Absolute positioning.
             program.G1(F=feed) #
             if power==LaserPower.CONSTANT:
                 program.M3(S=laser) # Laser settings
             else:
                 program.M4(S=laser) # Laser settings
             return program

         for laser_power in laser_powers:
             for laser_pwm in laser_pwms:
                 print("Lasers Set To: {} {}".format(laser_power, laser_pwm))
                 program = init(power=laser_power, laser=laser_pwm) + square(size=square_size) +
```

```python
            print(".", end="")
            cnc.run(program)
            jogx(square_size)
            jogx(spacing)
            print("")
        jogx((-1*test_space)*tests)
        jogy(spacing)
```

```
Lasers Set To: 0 10
.
Lasers Set To: 0 13
.
Lasers Set To: 0 16
.
Lasers Set To: 0 19
.
Lasers Set To: 0 22
.
Lasers Set To: 0 25
.
Lasers Set To: 0 28
.
Lasers Set To: 0 31
.
Lasers Set To: 0 34
.
Lasers Set To: 0 37
.
Lasers Set To: 0 40
.
Lasers Set To: 0 43
.
Lasers Set To: 0 46
.
Lasers Set To: 0 50
.
Lasers Set To: 1 10
.
Lasers Set To: 1 13
.


---------------------------------------------------------------------------

KeyboardInterrupt                         Traceback (most recent call last)

<ipython-input-59-334601ea897c> in <module>()
 16         print(".", end="")
```

```
        17          cnc.run(program)
---> 18          jogx(square_size)
        19          jogx(spacing)
        20          print("")


    <ipython-input-51-b27fb8b7e610> in jogx(x)
        2      program = GCode.GCode()
        3      program.G0(X=x)
----> 4      cnc.run(program)
        5 def jogy(y=10):
        6      program = GCode.GCode()


    ~/python_cnc3018/python_rs274/GRBL/__init__.py in run(self, program, compact)
    117
    118              while len(results) == 0:
--> 119                  sleep(0.5)
    120                  results = self.read(multiline=True, timeout=0.1)
    121


    KeyboardInterrupt:
```

3 Experimental Setup.

Assembled Chinese CNC 3018. GRBL Version

```
In [66]: cnc.cmd("$$")

Out[66]: ['ok', 'error:8']

In [67]: cnc.cmd("$#")

Out[67]: ['ok', 'error:8']

In [68]: cnc.cmd("$I")

Out[68]: ['ok', 'error:8']

In [69]: cnc.cmd("$N")

Out[69]: ['ok', 'error:8']
```

4 Results

4.1 M3

- Laser needs to 'warm up'(?) on constant power mode

5

- Starts to smoke @ 22 on graphite.

- Starts showing up @ 25, 40 nearly completed a cut, but laser wasn't on whole cycle. 43+ all cut.

- 40 barely marked second paper.

- 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.

4.2 M4

- Very itermittent.
- 50 didn't cut all the way through. Almost no marking on second paper.

5 Test Conclusion.

- Need to wrap Keyboard Kill with a CNC Kill.
- Need more tests.

In [ ]: