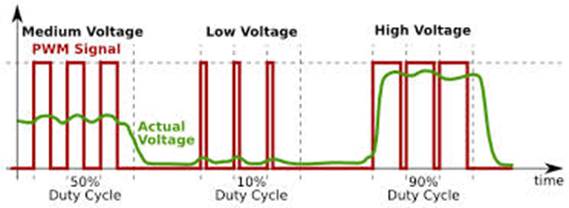
**Laser Cutter Firing Mechanism (Laser Power)**

*PWM/Laser Firing Profile*

Laser Intensity is determined by pulse width modulation (PWM) of signals sent from the Arduino controller to the laser. The Arduino can send two voltage intensities to the laser cutter through the RAMBo microcontroller: a high and a low voltage. To modulate laser intensity with higher precision than on or off, the Arduino can send modulated square waves with controlled relative proportions of high vs. low signal that are integrated into an analog signal of approx. constant voltage by a capacitor (See PWM figure below).



<http://www.wayneandlayne.com/projects/video-game-shield/design/>

Essentially, the laser accepts signals at a frequency determined by the internal capacitor. During each cycle, there are subcycles during which the signal can be a high or low voltage. The new firmware by Tim attempts to maximize control over laser intensity signaling by computing the smallest subcycle possible as determined from the physical capacitor and the Arduino processing power. Our laser cutter was similar to Tim’s laser cutter, so we estimated the capacitor frequency to be 25kHz. The Arduino has 16 MHz computing speed, so the number of subcycles possible are 16MHz / 25kHz = 640. By dividing the power range (0 to 100%) by 640, we calculate the approximate resolution of the power setting to be about 0.15625. Tim suggested that the resolution is probably about 0.2.

Laser settings can be changed in the Laser.cpp file (Ex. Capacitor frequency and Arduino Computing speed)

*Beam Types*

There are three potential beam types in the current marlin configuration: Continuous, Pulsed, and Raster.

* 1. The continuous mode is the traditional laser cutting mode; i.e. during a movement, the laser is continuously open.
  2. The pulsed mode only fires the laser a set number of pulses per millimeter of movement,
     1. At low enough speeds (still relatively faster than previous ablation speeds), there is very nice ablation of the surface.
     2. This mode was determined to be extremely useful for my project.
  3. The raster code is not fully functional yet, and I don’t anticipate using that setting for my project, so I won’t discuss it here.

**Signal firing characterization by oscilloscope observation**

*Background*

We know nothing about the actual signals being conveyed from the RAMBo microcontroller to the laser cutter. Thus, I used an oscilloscope to generally observe output signals from the microcontroller in response to various basic GCODE commands.

* 1. M3 [timed laser fire command] does not pause the command execution queue
     1. Movement commands will be executed immediately causing possible movement while laser is still firing

Solution: Insert a G4 [pause] command after every M3 command to force a pause in the GCODE for the M3 laser fire.

NOTE: G4 [pause] command cannot pause for less than 16 ms in actuality (even if less time is specified).

* 1. If laser intensity is set, G0 movements become laser on movements (I have no idea why)

Solution: Manually set power to 0 by command “M3 S0 L10000”.

**PWM Intensity Resolution**

*Background*

It is also important to understand the resolution of the laser power intensity setting. From a signal standpoint, this means the smallest difference in input power settings that is distinguishable. To test this, I simply took measurements of the baseline PWM firing signal after changing the laser power. The microcontroller is constantly sending the intensity signal, so no code was needed to make measurements.

As a note, intensity can be effectively changed by issuing an M3 command like so:

"M3 Sx"

Where x denotes the Pulse Width Modulation ratio (PWM) in %.

System has integer value resolution of PWM settings with 0.1% accuracy

S1 is the lowest observable signal on the oscilloscope.

S1 input  1.1% output

S2 input  2.0% output

S1.5 input  1.1% output

S12 input  12% output

Further testing may yield different results, but tentatively, integer values of power input should safely give accuracy of +/- 0.1% power.

**Power Resolution**

*Background*

Laser signal PWM input does not necessarily correlate perfectly to max power. Here I measured the actual power output at different % power input settings to empirically determine actual power output. Unfortunately with a high powered laser, I cannot test the power at the focal region without potentially destroying the power meter. So, I placed the power meter on the laser cutter bed as far from the focal zone as possible to take power measurements.

Machine generated alternative text:
0.00E + 00 
5b0E102 
1b0E101 
1.50E101 
2b0E101 
2.50E101 
3b0E101 
3.50E101 
0 
PWM Inputvs.Actual Power(Watts) 
100 
120 