Flashing the Arduino

1. Download the Grbl source code.
   * https://github.com/grbl/grbl/archive/master.zip
   * Once downloaded, unzip it and you'll have a folder called grbl-master or something similar.
   * (On Arduino IDE < 1.0.6 or < 1.5.6) Re-name the folder to Grbl. You have to do this, because the Arduino IDE does not like the - hyphen in their library names.
2. Make sure are using the most recent version of the Arduino IDE (last tested on v1.05).
   * NOTE: For pre-v1.05 Arduino IDE users, you may need to manually add Grbl's source code into your Arduino libraries, so that it will appear in the Import Library... menu. Google the internet for how to do this, and once installed, skip to step 4.
3. Load Grbl into the Arduino IDE as a Library.
   * Launch the Arduino IDE.
   * Click the Sketch drop-down menu, navigate to Import Library..., and click Add Library....
   * Select the Grbl folder when asked to select a zip or folder for the library you'd like to add.
   * It may take a few seconds for the Arduino IDE to import it.
4. Create a Grbl sketch in the Arduino IDE.
   * Start a new sketch, if one isn't already up on the screen.
   * Click the Sketch drop-down menu, navigate to Import Library..., and click the new menu option Grbl at the bottom of the list.
   * A long list of #include<>'s used by Grbl will be added to the sketch. Ensure (edit if necessary) that "system.h" is at or near the top of the list of includes. At this point, *don't add anything else! No void, no loop, nothing!*
5. Compile and upload Grbl to your Arduino.
   * Connect your Arduino Uno to your computer.
   * Make sure your board is set to the Arduino Uno in the Tool->Board menu and the serial port is selected correctly in Tool->Serial Port.
   * Click the Upload, and Grbl should compile and flash to your Arduino! (Flashing with a programmer also works by using the Upload Using Programmer menu command.

Configuring grbl for your printer

First, connect to Grbl using the serial terminal of your choice. Set the baud rate to **115200** as 8-N-1 (8-bits, no parity, and 1-stop bit.)

Once connected you should get the Grbl-prompt, which looks like this:

Grbl 0.9g ['$' for help]

**Type $** and press enter to have Grbl print a help message. You should not see any local echo of the $ and enter. Grbl should respond with:

$$ (view Grbl settings)

$# (view # parameters)

$G (view parser state)

$I (view build info)

$N (view startup blocks)

$x=value (save Grbl setting)

$Nx=line (save startup block)

$C (check gcode mode)

$X (kill alarm lock)

$H (run homing cycle)

~ (cycle start)

! (feed hold)

? (current status)

ctrl-x (reset Grbl)

We need to set the steps per millimeter variable equal to the pitch of your lead screw. **Type $$** to list variables

$100=314.961 (x, step/mm)

$101=314.961 (y, step/mm)

$102=314.961 (z, step/mm)

$110=635.000 (x max rate, mm/min)

$111=635.000 (y max rate, mm/min)

$112=635.000 (z max rate, mm/min)

$120=50.000 (x accel, mm/sec^2)

$121=50.000 (y accel, mm/sec^2)

$122=50.000 (z accel, mm/sec^2)

Setting variables for your system

**Type $100**=(pitch of lead screw) to set the x motors variable

**Type $101**=(pitch of lead screw) to set the y motors variable

**Type $120**=5 setting x motors acceleration to 5mm/sec^2 allowing the motor to accelerate during the move, rather than starting at the maximum velocity

**Type $121**=5 setting y motors acceleration to 5mm/sec^2

Next you will need to adjust the maximum velocity of the motors, depending on the pitch of the lead screw you will need a faster or slower maximum velocity in order to prevent miss-stepping and losing track of the positioning. Our system works fine at 250mm/min, adjust accordingly

**Type $110**=(x motor velocity)

**Type $111**=(y motor velocity)

AFMS Implementation

The Adafruit motor shield uses 4 h-bridge drivers, capable of driving 2 stepper motors. It is a cheaper alternative to the CNC Motor Shield that is currently compatible with the GRBL firmware.

In order to interface with the motors there are already libraries supported by Adafruit to drive the motors. The AFMS Library and the AccelStepper library are provided by Adafruit for use with their shield. The AFMS library provides general support for driving the motors with little velocity control. With the AccelStepper library the motors can be driven at slightly higher velocities as well as control for variable acceleration.

Due to limitations of the shield, it is unable to drive the motors at high enough velocities and accelerations required to be implemented into the GRBL firmware. Additionally the CNC shield and AFMS use different pins to send signals to the motors. To implement the AFMS into GRBL would require massive changes in the source code and all of the functions required for stepper control would need to reimplemented for the AFMS.