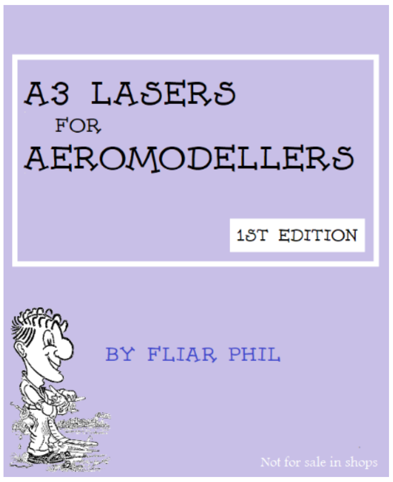
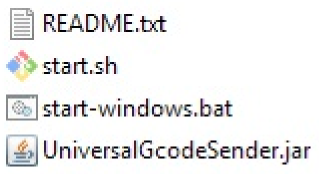
## *grasping the nettle*

Once the A3 Laser is assembled and configured, well "happy days", but before that there's a mountain to climb installing the software. A complete account of this might be called A3 Laser for Dummies but that's not going to happen. On the other hand, it can't be ignored and partly what prompted these articles, so let's grasp the nettle.



if only it were real

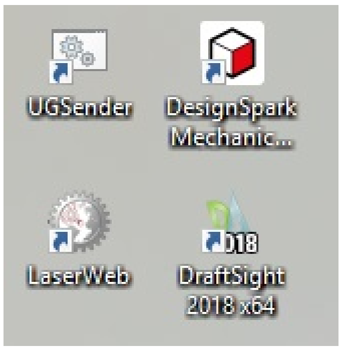
Universal Gcode Sender or UGS is probably the easiest to install, so begin there. Search online for *download UGS* or something along those lines and click to download ver 1.0.9 it comes zipped but Windows Explorer say will unpack it into a temporary folder.



README.txt
start.sh
start-windows.bat
UniversalGcodeSender.jar

the UGS download unpacked

Move these four files from this temporary folder to one you've created and called, say UGS. Once there you I suggest you create a shortcut to *start-windows.bat* rename to something more like *UGSender* and move it onto your desktop.



how part of your desktop might look

UGS comes into it's own in configuring the microprocessor board [Arduino NANO] which runs GRBL on the control board [EleksMaker Mana SE]. This is how you're able to input that the stepper motors take 80 steps/mm, place the origin where you want it [bottom left, say] and quite a few more GRBL settings – more on this below.

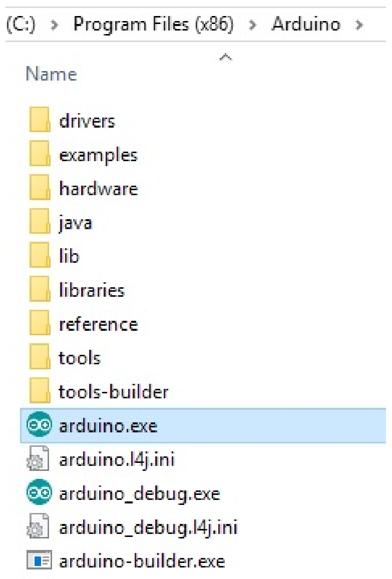
The microprocessor that comes with the A3 Laser kit is a clone of the Arduino NANO and isn't programmed with GRBL. My choice was to exchange the clone for a genuine Arduino NANO rather than overwrite the software it came with, but this isn't strictly necessary.

a genuine Arduino NANO

The role of this board is to talk to the stepper motor drivers [one for each axis] and to control the laser. It gets it's power via the USB cable and takes instruction from LaserWeb4, step by step, in G-code, eg `G1 X15.00 Y15.00 S25.50 F200` as we saw above. It runs open source CNC software called GRBL which needs to be uploaded to it, just the once, in a process taking under a minute. If you lived round the corner from me I'd happily upload GRBL for you, but I've checked and you don't -*phew!* Instead you'll need to install *Arduino IDE* called an integrated development environment which can upload the latest version of GRBL to your Arduino NANO. Did I say there was a mountain to climb?

## *uploading GRBL*

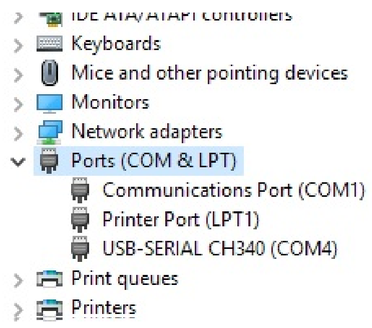
You'll need to search for the latest version of GRBL [ver 1.1f] or go direct to the resource folder for these articles. It needs to be copied to the libraries folder in the Arduino folder:


(C:) › Program Files (x86) › Arduino ›

Name

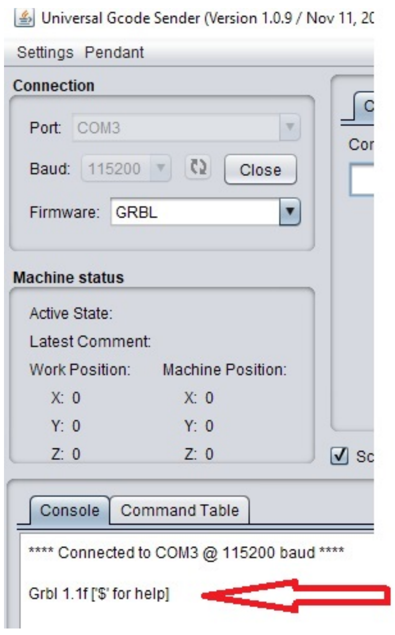
drivers
examples
hardware
java
lib
libraries
reference
tools
tools-builder
arduino.exe
arduino.l4j.ini
arduino_debug.exe
arduino_debug.l4j.ini
arduino-builder.exe

Now you should open Arduino, connect to the NANO with a USB lead and via the Tools tab select the right board [Arduino Nano] and the port.
Next via the File tab select Examples and look down the list for the GRBL 1.1f file, click and select grblUpload. Click on the forward arrow Upload icon and cross your fingers. Arduino will first Verify and then Upload . . it takes a while but you know you've won when you see the message Upload done.
It was during this process that I nearly gave up [and since repeated a number of time as a check while writing this piece]. So what could possibly go wrong: copying GRBL 1.1f to the wrong folder, having more than one Arduino folder or installation . . not being sure which com port the USB connects to – tip: run Device Manager [you can search for it in Settings] and opening up Ports and in this example COM4


IDE ATA/ATAPI controllers
Keyboards
Mice and other pointing devices
Monitors
Network adapters
Ports (COM & LPT)
Communications Port (COM1)
Printer Port (LPT1)
USB-SERIAL CH340 (COM4)
Print queues
Printers

what Device Manager reveals

How do you know if everything really has gone as it should? Open up UGSender [Universal Gcode sender] select the right port [baud should be 115200] and Open. In the Console you need to see the Arduino board answering with Grbl 1.1f
If you took the Arduino NANO out of the control board while you uploaded GRBL, you can return it now.


Universal Gcode Sender (Version 1.0.9 / Nov 11, 20
Settings Pendant
Connection
Port: COM3
Baud: 115200 Close
Firmware: GRBL
Machine status
Active State:
Latest Comment:
Work Position: Machine Position:
X: 0 X: 0
Y: 0 Y: 0
Z: 0 Z: 0
Console Command Table

```
**** Connected to COM3 @ 115200 baud ****
Grbl 1.1f ['$' for help]
```

## *taking stock*

If things have gone well, you've installed UGSender, installed Arduino, copied the latest GRBL [ver 1.1f] into the Arduino Libraries folder and uploaded GRBL to the Arduino NANO microprocessor board on the EleksMaker control board. As a check you've opened the USB connection between UGSender and the Arduino NANO which has confirmed with Grbl 1.1f

## *point of safety*

Just as it is good practice, for example, to have the Tx turned on with throttle set to minimum *before* powering up the Rx in a model you should connect UGSender to the Arduino NANO before switching on or connecting power to the EleksMaker control board. If you forget this then, while opening the port, <u>the laser will turn on at full power for a second or so</u> when you're not expecting it.

## *nudging*

Now would be a good time to get the feel of the machine:

disconnect the cable from the control unit to the laser for now
with power off, push the carriage that holds the laser into the middle of the workspace [it should offer little resistance]
turn on the power and the stepper motors should hold the three carriages in place
click on the Machine Control tab in UGSender, set the step size to 5mm and try nudging with x+ or x-, y+ or y-
you could Reset Zero then nudge here and there before clicking on Return to Zero

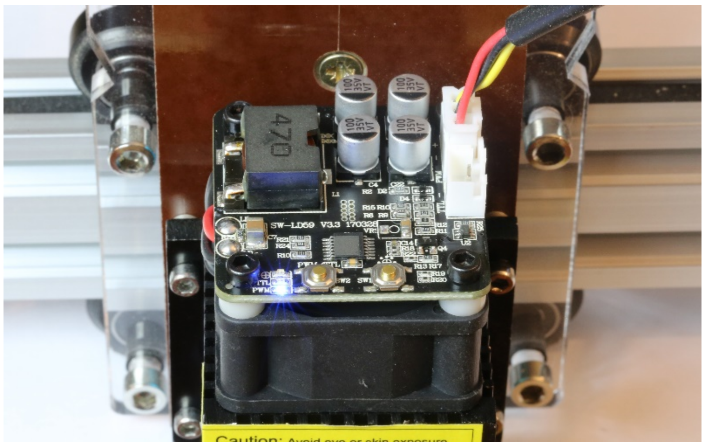
Almost certainly you'll notice a couple of things: while you input a nudge of 10mm say, the carriage moves but not by that amount; then there's the matter of left and right. The Arduino NANO hold a number of parameters which need to be set to match the machine you're asking it to drive. It needs to know how many steps of the stepper motor will result in it moving 1mm. To few steps and the carriage won't move as far as it should, too many and it will overshoot. Taking into account the number of teeth on the stepper motor pulley [20t], the pitch of the toothed belt [2mm] and so on produces the value 80 steps/mm. Select the commands tab and enter the command $$ [followed by return]. You'll see all the settings listed - look for $100 and $101 – the x and y steps/mm. These may contain 250 [forget the decimal part for now] but should contain 80. In the command window enter $100=80 [return] then get it to show al the settings by entering $$ [return]. This time you should see the new value of 80 next to $100. Next set $101 to 80. You could go back to nudging and check that when you input 50mm say the carriage actually moves 50mm.

Next problem to tackle is does x- result in going to the left, or perversely move to the right. Setting $3 controls direction. If the x direction is wrong try setting this to 1 with $3=1. If the y direction is wrong then you'll need to add 2 [so 0 would go to 2 and 1 would go to 3] every chance here of getting yourself thoroughly confused but the *right* answer will be 0, 1,2 or 3.
You'll find the full list if you search online for Grbl Settings. Two more you need to change are $1 should be 255 if you want the stepper motors to hold position and $32 should be set to 1 or Laser mode otherwise GRBL thinks you have a router or somethings and will pause from time to time while the "motor" spins up to speed.
in brief: $1=255, $3= a number from 0 to 3 [what works for you], $32=1, $100=80 and $101=80

Making changes to Grbl settings covers the real reason for installing UGSender and you may not need it again. If you feel "that was a lot of work for very little" you could always select File Mode, import a Gcode file and run it, though I suggest you disconnect the laser for the present.

## *connecting the laser*

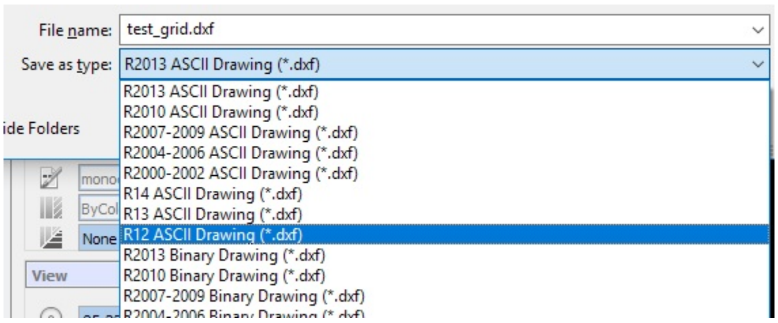
You'll have read earlier that some lasers can operate continuously [or TTL mode] or they can operate in pulse mode [PWM].



The left of the two switches toggles between the two modes TTL [blue LED glows] and PWM [green LED glows]  and needs to be set to PWM with the blue LED glowing. The right hand switch toggles a built in blue LED which shines in place of the laser itself. If you're nudging the laser to the start of where you want to be cutting then you can turn it on and a blue dot will appear on the balsa directly underneath the laser. Meanwhile another red LED will flash, reminding you to turn it off again.
Please note the advice given earlier, to have the control board switched off while your establishing a connection via USB, or the laser will unexpectedly turn on at full power for a second or so.

## *LaserWeb4*

This is the software to be used on a day to day basis and one that will become very familiar. It accepts drawing [dxf files], generates Gcode and talks to the Arduino Nano on the control board.


File name: test_grid.dxf
Save as type: R2013 ASCII Drawing (*.dxf)

R2013 ASCII Drawing (*.dxf)
R2010 ASCII Drawing (*.dxf)
R2007-2009 ASCII Drawing (*.dxf)
R2004-2006 ASCII Drawing (*.dxf)
R2000-2002 ASCII Drawing (*.dxf)
R14 ASCII Drawing (*.dxf)
R13 ASCII Drawing (*.dxf)
R12 ASCII Drawing (*.dxf)
R2013 Binary Drawing (*.dxf)
R2010 Binary Drawing (*.dxf)
R2007-2009 Binary Drawing (*.dxf)

be sure to save in R12 ASCII Drawing format

Here you can specify how bright the laser should be and how fast to cut. It will let you see where your drawing is with respect to the machine origin and allow you to nudge. If you have a file to follow on from cutting to label the parts then it will run that too. On the downside it doesn't come with a whole lot of instructions, or any at all. [and I find it can be reluctant to load and run]

go to: [https://github.com/LaserWeb/LaserWeb4-Binaries/releases](https://github.com/LaserWeb/LaserWeb4-Binaries/releases)
From this page click on the second option for x64.exe to download it. Once downloaded, click on the downloaded file and it should install

## *a couple more points*

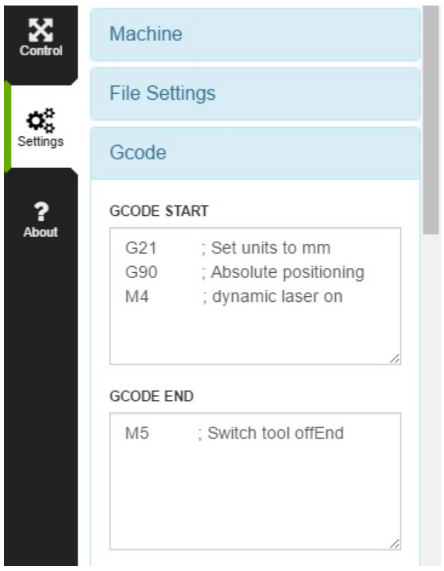
LaserWeb4 generates G code from your dxf with the laser power and cutting speed you decide, but it will also has one more useful function: it will top and tail your code with any commands you'd like to include by default.

Open the Settings tab and enter in GCODE START any codes or commands you'd like to see run before the main G code:

```
G21     ; Set units to mm
G90     ; Absolute positioning
M4      ; dynamic laser on
```

This last one arranges for the laser power to be reduced when the cut rate is low – for example while negotiating a corner

The stepper motor drivers have an adjustment to limit the maximum current available . . about the last thing you'd want is for the carriage to jam and the controller to overheat and burn out.

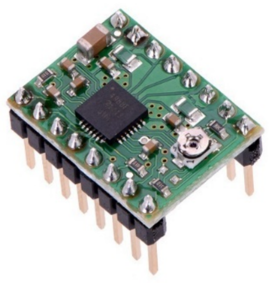


GCODE START

```
G21     ; Set units to mm
G90     ; Absolute positioning
M4      ; dynamic laser on
```

GCODE END

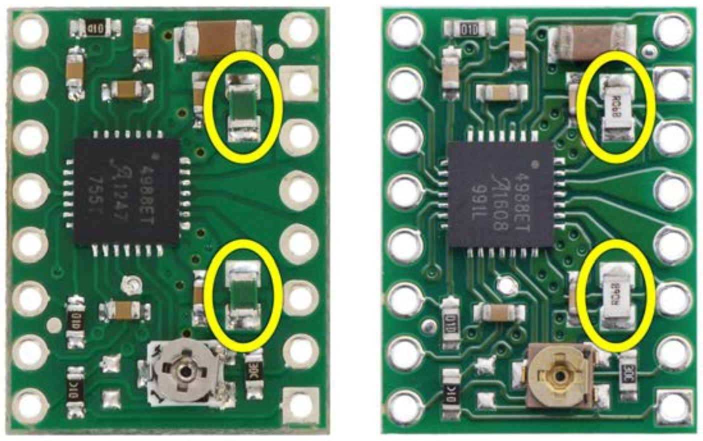
```
M5      ; Switch tool offEnd
```



A genuine Pololu A4988

The Eleks Mana board carries a couple of these, but in Chinese clone form [remember that one has to drive *two* stepper motors in the Y-axis]. Search online and you'll find detailed instructions for setting the maximum current or current limit. The snag is that before you can calculate the voltage generated by this current [and set the potentiometer] you really need to know the value of the two large resistors.

see: https://www.pololu.com/product/1182



$R_{CS}$ = 50 mΩ    $R_{CS}$ = 68 mΩ

Unfortunately, it isn't clear to me what value resistor has been chosen for the Eleks Mana drivers and so I reluctant to guess here. Try using them as factory set, testing occasionally to see that they're not getting too hot. So long as the carriages move smoothly you're probably safe to leave well alone. My own test has been to have the laser make two passes and check that is following precisely the same path [therefore no steps missed].
See the point made above in *GRBL settings* about *$1 should be 255 if you want the stepper motors to hold position.*

## *finally*

In deciding what setting to use I think of Laser Power / Cut Rate as Q a single number:

from the spreadsheet you might see power 100% and cut rate 220 mm/min, so Q is 100/220 or 0.4545
but you could turn the power down provided you cut more slowly: power 50% and a cut rate of 110 mm/min because 50/110 is *also* 0.4545 ie the same Q

Behind the spreadsheet is the idea that Q [power/cut rate] = Sd [surface density] x f-factor and if in the course of your testing you find a combination of power and cut rate that makes a really nice job then measure the size of the balsa panel, weigh it and calculate it's surface density. Armed with power, cut rate and surface density you can calculate the f-factor that fits your setup:

since     f-factor = Q /Sd

I ran more than 100 tests and kept not only a paper record but many of the balsa test samples too and would encourage you to do the same.

FliarPhil@gmail.com