Workbee CNC Router

I recently bought a 1500×1500 Workbee CNC router Kit from Bulkman 3d and i bought both the leadscrew and belt versions (technically i bought the leadscrew version and a belt kit)

i think i'm going to be happier with the belt because it can supposedly go faster.

leadscrew wabble really is an issue on this size of the machine. except for Bulkman nobody sells the 1500 kits with leadscrews.. there is a video out there suggesting to put tension on the leadscrews rather than pressure to eliminate wobble.. i might try that but i don't want to shorten my extrusions, so i'll need to figure out another way to put tension on the leadscrews.

i've got the grbl controller and am running grbl v1.1 these are my settings for the leadscrew so far.. limit switches haven not been configured yet.

```
$0 = 10    (Step pulse time, microseconds)
$1 = 25    (Step idle delay, milliseconds)
$2 = 0    (Step pulse invert, mask)
$3 = 0    (Step direction invert, mask)
$4 = 0    (Invert step enable pin, boolean)
$5 = 1    (Invert limit pins, boolean)
$6 = 0    (Invert probe pin, boolean)
$10 = 1    (Status report options, mask)
$11 = 0.010    (Junction deviation, millimeters)
$12 = 0.002    (Arc tolerance, millimeters)
$13 = 0    (Report in inches, boolean)
$20 = 0    (Soft limits enable, boolean)
$21 = 1    (Hard limits enable, boolean)
$22 = 0    (Homing cycle enable, boolean)
$23 = 0    (Homing direction invert, mask)
$24 = 25.000    (Homing locate feed rate, mm/min)
$25 = 500.000    (Homing search seek rate, mm/min)
$26 = 250    (Homing switch debounce delay, milliseconds)
$27 = 1.000    (Homing switch pull-off distance, millimeters)
$30 = 1000    (Maximum spindle speed, RPM)
$31 = 0    (Minimum spindle speed, RPM)
$32 = 0    (Laser-mode enable, boolean)
$100 = 100.000    (X-axis travel resolution, step/mm)
$101 = 100.000    (Y-axis travel resolution, step/mm)
$102 = 100.000    (Z-axis travel resolution, step/mm)
$110 = 4000.000    (X-axis maximum rate, mm/min)
$111 = 4000.000    (Y-axis maximum rate, mm/min)
$112 = 500.000    (Z-axis maximum rate, mm/min)
$120 = 300.000    (X-axis acceleration, mm/sec^2)
$121 = 300.000    (Y-axis acceleration, mm/sec^2)
$122 = 200.000    (Z-axis acceleration, mm/sec^2)
$130 = 200.000    (X-axis maximum travel, millimeters)
$131 = 200.000    (Y-axis maximum travel, millimeters)
$132 = 200.000    (Z-axis maximum travel, millimeters)
```

the above maximum speeds where reached doing aircuts only!

Here are some configurations that worked so far:

material	bit	depth	spindle RPM	feed rate (mm/min)	result
OSB	1/4 upcut	6mm	19200	1700 mm/min	very rough cut, a little smoke every now and then
OSB	1/4 upcut	3mm	19200	3000 mm/min	much smoother, no smoke
MDF	1/8 upcut 2-flute	spiral ?? bore	3000	500	no smoke but vibrations
MDF	1/8 upcut 2-flute	spiralpocket	7000	500	no smoke
Valchromat (Colored MDF, slightly harder than regular MDF)	6mm straight 2-flute	7mm	10'000	2000 mm/min	some vibrations but acceptable quality that only needs slight sanding on visible edges
Aluminum	1/8 2-flute straight	0.1mm	6000	1000mm/min	slow but decent :)

G-Code Sender

my Workbee is using the GRBL firmware on a arduino together with a CNCshield on top of it. In order to send gcode to the arduino, a gcode-sender is used on a PC that is connected via USB.

currently i am using UGS (Univseral Gcode Sender) Platform edition. The reason why i chose this software is simple: i used it on my shapeoko and already kenw it. It seems though, that UGS crashes every now and then, leaving the machine stuck. i then need stop the job, disconnect from the arduino, reconnect and start over again. since the ardino still knows the machine position etc. I have a feeling this crash might happen on the UGS side. I therefore want to try different gcode senders in the near future. the two most interesting ones seem to be:

- CNC.js - nice web gui, touch probe support, “pendants” which are small ui's for small devices like phones etc. server can be run on a raspberry pi
- bCNC - basic CAM built in, accepts not only gcode but also svg and dxf. curretly i do the CAM work on fusion 360 which is nice, but it might be simpler to have it integrated into the gcode sender, so it can be modified easily while standing at the machine.. the Fusion 360 interface needs quite a big screen to work with it smoothly and simple changes like changing the cutting speed means saving re-opening files etc. all the time.

CNC.js

i've decided to try CNC.js. I am installing it on a Raspberry PI 3 (because i had one laying around). the idea is, to get the raspberry as close as possible to the grbl to keep the USB cable as short as possible. I've read that communications issues on the USB link cause by interference could cause the hang-ups i saw with UGS. in any case, the size of the raspberry makes it a practical choice anyway :)

so first i installed raspberry OS, there is pre-built rasbperry image from CNC.js but it has not been updated in the last two years, so I decided to go with a manual install.

here are the steps i went through:

install_raspbian_on_f2fs_root with my script, set wifi and hostname and enable ssh so we don't need to connect a screen to it ever :)

install git

```
sudo apt update
sudo apt install git
```

install cncjs with npm which is installed via nvm. basically, just follow the instructions on the CNC.js github page

create a systemd service that will start cncjs automatically:

```
sudo nano /etc/systemd/system/cncjs.service
```

here are the contents of the startup script:

```
[Unit]
Description=CNCjs

[Service]
ExecStart=/bin/bash -c -l cncjs
Restart=always
RestartSec=10
 # Output to syslog
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=cncjs
User=pi
Group=pi

[Install]
WantedBy=multi-user.target
```

nwo start it:

```
sudo systemctl daemon-reload
sudo systemctl --now enable cncjs
systemctl status cncjs
```

this should show that it's running now and you should be able to access it on port `8000` of your raspberry

i'm lazy, so i don't want to type that port 8000 all the time, let's add a redirect using `lightthpd`: