Near Field Scanner Documentation

Version 0.1

Jacob Stewart

May 5, 2025

Contents

List of Code Listings

1 Introduction

This documentation will cover the basics of how to use the near field scanner, how the hardware is configured, and how the software is configured. The hardware section will cover important things to note about the construction of the machine. Information that can likely be quickly understood by some quick analysis of the machine will probably be left out of the documentation.

2 Quickstart

2.1 Common GRBL Commands

3 Hardware

3.1 General Comments

Much of the design is based around CNC machines. The custom parts that deviate from this basic design (the panel and its connections, for example) will be covered in this manual.

The primary functions/aspects of the hardware are the GRBL Arduino Uno R3 controller, which controls the hardware for the gantry system, which holds and controls the position of the probe. The AUT is simply attached to an acrylic table.

3.2 Gantry System

The majority of the hardware components were purchased from OpenBuilds[1]. The remaining parts were likely custom designed. Parts can be found in the Microsoft Teams' Near Field Scanner > Models folder.

Each of the linear rails can be constructed following this guide[2], which demonstrates how to build a smaller version of the 1 meter linear actuators.

There are only two unique connections in the gantry system. The first is between the machine y-axis (the bar running between the parallel x-axis).

[1] `https://openbuildspartstore.com/`
[2] `https://youtu.be/tLozVt_CjXQ?si=yefkdUC8tTgpIad6`

Before making this connection, assemble the gantry plates, and attach them to the y-axis C-Beam. With how the x-axis C-Beams are attached to the wooden base, the gantry plates should be connected approximately 5 mm from the edges. Instead of tightening the attachment between the gantry plates and the y-axis C-Beam, keep them loose as you line up the gantry plates into the channels of the x-axis C-Beams. Unfortunately, the x-axis C-Beams are not perfectly parallel, so it is likely best to line up the gantry plates near the center of the x-axis C-Beams. Once the gantry plates are lined up to a satisfactory degree, you can tighten them in and run the lead screws through the gantry plates.

The other unique connection is between the gantry plate of the y-axis and the z-axis C-Beam. This requires a custom part, which is 3D printed from the "2 Become 1" model. There is a design for the smaller gantry plate and the larger gantry plate. For either plate, it is recommended to 3D print using and FDM printer. Though an SLA printer is more accurate, it also has a tendency to warp large flat surfaces. Additionally, the cost of SLA resin is much greater than FDM PLA filament.

Once the "2 Become 1" connector is printed, you must first attach the connector to the z-axis C-Beam. The screwing holes and method is the same as the C-Beam end mount. After the connector and the C-Beam are connected, you can attach the connector the y-axis gantry plate. You may want to ask for an assistant to hold the z-axis C-Beam while you are attaching the connector to the y-axis gantry plate.

3.3 Panel Connection

Many of the components for the panel system are removable. Unfortunately, in order to remove the panel completely from the z-axis gantry plate, nearly all the components must be disassembled one after the other. Those components include the "connect to panel" connector, the absorber, the acrylic panel, and the panel to z-axis connector.

In order to remove the panel, the absorber will need to be removed first. It is advised that two people work together to remove the absorber. It is also advised to wear gloves when handling absorber, as it can act as an irritant. To remove the absorber, carefully peel the velcro along the edges. When removing the absorber, something to avoid is damaging the absorber spikes.

These are fragile, and vital to the absorber's performance. With this in mind, it is slightly better to start peeling the absorber from the velcro along the bottom edge and work your way up. If you were to drop the absorber while removing it from bottm to top, it is more likely to fall on the flat back of the absorber rather than the spiky front.

After the absorber has been removed, the acrylic panel can be unscrewed from the panel to z-axis connector. Once the acrylic panel has been removed and set aside, the panel to z-axis connector can be unscrewed and removed.

In order to remove the "connect to panel" connector, the absorber will need to be removed first, as described above. Though you can remove this part, it isn't necessary. It is designed to remain attached, and instead provide screw holes for different probe to panel mounts. Each of these probe to panel mounts will have to be custom designed around the probe being connected. Each of these screw holes are evenly spaced 1.125 inches away from each other. An example 3D model of this part can be found in the "Probe to Panel Mount" file on Microsoft Teams.

Reassembly of the panel is mostly the same as removal in reverse. In particular, it is easier to install the absorber by connecting the velcro at the top of the panel, and work your way down, allowing the top connection to act as a hinge to swing the absorber into the remaining velcro connections.

3.4 Circuit Design

A wiring diagram can be found in Figure 1. Though the wiring may look like a complicated mess, if you look closely, the mess is a consequence of the amount of wires more than the complexity of the circuit. Most of the connections between the stepper drivers (the TB6600s) and the CNC Shield are for the CNC Shield's ground pin.

As for the PCB design, the complete design file can be found in the Microsoft Teams folder in the file "PCB Wiring".

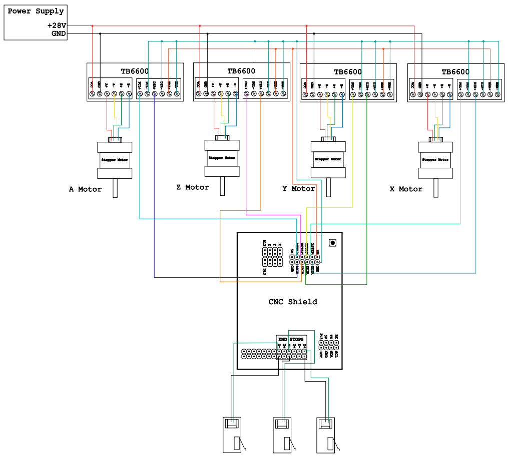

Figure 1: The CNC Wiring Schematic. Wire colors loosely match real wire colors, but are not perfectly equivalent.

3.5 The Arduino Uno (GRBL)

The Arduino Uno is running a firmware called GRBL[3], which is a CNC firmware that is practically the standard firmware for maker movement devices. Many 3D printers run GRBL under the hood. It is strongly advised to use an Arduino Uno R3 with a 16U2 communication chip rather than a CH340 chip. If you have the official Arduino Uno R3, you have the 16U2 communication chip. If you are not sure which chip your board has, it can be visually determined. On an official Arduino Uno R3, the 16U2 is the square surface mounted chip just behind the USB type-B port. On a board with a CH340 communication chip, in a similar location to the 16U2 chip, there will be a larger rectangular surface mounted chip.

When flashing the firmware to the Arduino Uno R3, there are two essential changes that need to be made to the GRBL configuration file. To make these changes, first follow the offical Compiling Grbl[4] instructions up until step 5: “Compile and upload Grbl to your Arduino". At this point, the Grbl code has been added to your Arduino library. In this library is the compilation configuration file. The essential changes will be made in here. This file can be accessed by going to WindowsUser/Documents/Arduino/libraries/grbl/config.h. (Note

[3] `https://github.com/gnea/grbl`
[4] `https://github.com/gnea/grbl/wiki/Compiling-Grbl`

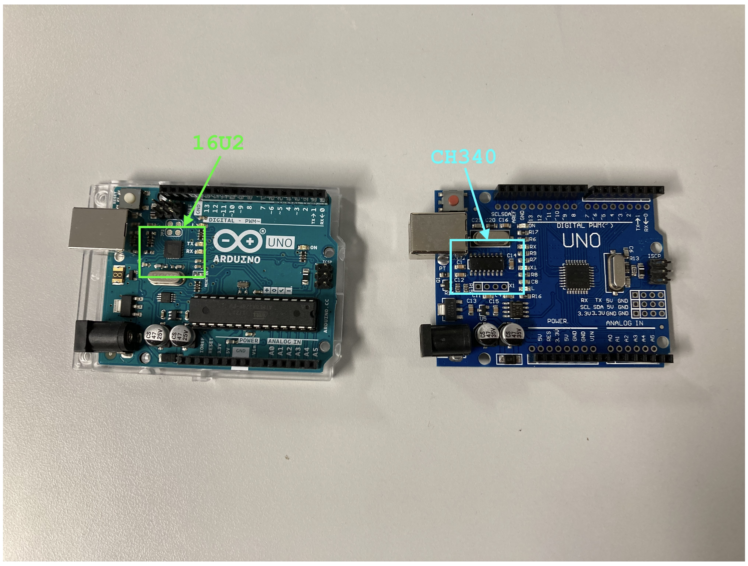

Figure 2: A visual comparison between the 16U2 chip and the CH340 chip. It is strongly advised to use an Arduino Uno R3 with the 16U2 chip.

that this location may change between your computer's OS, the version of Arduino, etc. If you are unable to find the library, an internet search should allow for you to find the library location). For the first essential change, go to line 124, and **uncomment** `#define HOMING_SINGLE_AXIS_COMMANDS`. For the second essential change, go to line 339 and **comment out** `#define VARABLE_SPINDLE`.

The first change is necessary because it enables commands that allow for each axis to be homed one at a time ($HX, $HY, $HZ),. Homing each axis one at a time is more reliable than using the default in homing command $H. Thus, the Matlab code uses and requires these commands. The second change is necessary due to the CNC Shield in use. The newer version of GRBL uses pin D12 of the Arduino Uno rather than pin D11 for the Z-Axis limit switch. This is because the newer version of GRBL needs the PWM variable voltage output of D11 for a new feature to control a variable spindle. However, older versions of GRBL (which this CNC Shield had in mind) use pin D11 for the Z-Axis limit switch. Because this machine does not have a variable spindle and needs the Z-Axis limit switch, line 339 of the configuration file should be commented out to reenable pin D11 as the limit switch pin.

Once these configuration comments have been altered, you can flash the firmware.

After the firmware has been flashed, the GRBL settings need to be correctly configured. The user can check what settings are currently in use by the GRBL by entering ?? into a serial connection with GRBL. The current settings can be found in Appendix 6.1. More information about the settings and how to change them can be found in the GRBL wiki[5].

There is also a local GRBL also saves its configuration in the the Arduino's persistent memory. The configuration can be queried with .

3.6 Cable Connections

There are two different connections that need to be made in order for the gantry system to be controlled. The first is the USB connection between the Arduino Uno R3 and the computer. This allows for the serial connection between the Arduino Uno and the computer for GRBL command communication. This also powers the Arduino Uno.

The second connection is the Cincon 28V AC power adapter. This connects between a wall outlet and the DC power jack on the wiring PCB. This powers the four TB6600 stepper drivers.

3.7 Comments and Improvements

Though the linear actuators[6] used for the gantry system are a cost effective method of getting a (approximately) millimeter precise machine, they are not reliable as other methods. For example, a linear actuator combined with linear rails would be significantly more stable and reliable compared to just the linear actuator. With the inclusion of a feedback systems such as a PID control system, such a setup would likely be able to reach hundreds of micrometers of precision. Possibly even tens of micrometers. Such precision would allow for higher frequency measurements.

However, it is worth noting that such precision will not be necessary for the delta x criterion until frequencies above 200 GHz are being explored. Considering the hardware limitations of the VNA in use,

[5] https://github.com/gnea/grbl/wiki/Grbl-v1.1-Configuration#grbl-settings

[6] https://openbuildspartstore.com/c-beam-linear-actuator-bundle/

Another improvement would be the PCB design. The current design was made in house. For the sake of manufacturing convenience, vias were not added between the pins. Instead, the metal pins themselves were to act as the vias. To solder the top of the pin, solder paste was used. Unfortunately, the application and solidification of this solder paste was not very effective. This means the pins are incredibly susceptible to spontaneous disconnections. In other words, the PCB is not very reliable. This problem could be rectified by having a professional manufacture the PCB. As for a redesign, a PCB hat for the CNC shield would reduce the number of wire connections, keeping the package cleaner, more compact, and more reliable.

The AUT table is primarily lacking in one of the most important features for the table: its rigidity. The current design can easily be deflected, causing the entire table to oscillate. Fortunately, this oscillation is mostly caused by external forces, so while the machine is moving around when taking measurements, little oscillation occurs

The Z-Axis C-Beam also experiences a significant amount of wobble. The source of this wobble appears to be the gantry plate. A potential solution to this wobble would be to use a linear rail along the machine y-axis.

The power delivery method for the machine requires the user to manually plug and unplug the AC connector to wall power. It would be more convenient to power on the device with a switch. Along with this switch should be an emergency stop button which cuts power to the motors as soon as it is activated.

4 Software

4.1 Definitions

A confusing connection between the software and the hardware is the axis. In the software, you will see mentions of “machine x" or “code x". This is because what is the x-axis to the hardware (i.e., to grbl) is different from the x-axis of the software. This is because much of the theory around transforming the near to far field of a planar near field scanner treats the distance between the probe and AUT as the z-axis, the horizontal motion between the probe and the AUT as the x-axis, and the height difference between the probe and the AUT as the y-axis. However, with how the hardware is configured, it is the x-axis that determines the distance between the probe and the AUT, the y-axis that determines the horizontal motion between the probe and the AUT, and the z-axis that determines the height difference between the probe and the AUT.

4.2 GRBL Commands

GRBL provides an explaination of the commands on the GRBL Wiki[7], and though this manual will not cover all the commands, it will cover the most common ones used for this machine.

The first command the user will probably need to know is `$X`. This will unlock the machine, allowing the user to use subsequent commands to move the machine. The only other way to unlock the machine is with the homing commands.

If the user has followed the firmware flashing instructions in Hardware Section 3.5, then the special homing commands `$HX`, `$HY`, and `$HZ` can be used. These will home just their respective axis. Be careful when using this form of homing, as it will unlock the movement capabilities of the machine (same effect as `$X`) even if the other axis have not been homed. Thus, if your program requires all axis to be homed, either call all homing commands one after the other, or use the `$H` command (which has had spotty results for the current machine).

There are two basic homing commands: `$G90` and `$G91`. Both of these commands have syntax in the form of

```
$G91 X1 Y-2 Z3
```

(Quick Note: you don't need to specify every axis every time you use the `$G90` and `$G91` commands. For example, `$G91 X1` will only move the X axis, and will leave the Y and Z axis alone.)

The major difference between `$G90` and `$G91` is how it interprets the inputs for `X1`, `Y2`, and `Z3`. In the case of `$G91`, the machine is put into relative mode. This means if the user inputs `X1`, the machine will move the gantry system 1 unit in the positive X direction from its current X position. `$G90`, on the other hand, will move the machine to the absolute position where X = 1 unit. In the case of this particular machine, after the X axis has been homed, the

[7] https://github.com/gnea/grbl/wiki/Grbl-v1.1-Commands

absolute position is at -898.8 mm. This means that if the user were to input the command `$G90 X1`, the machine will attempt to ram the x-axis off the edge until it has “reached" the position at X = 1. In other words, it is generally much safer and easier for a person to use `$G91` than it is to use `$G90`. However, absolute position is useful for programs and other aspects that don't change between program reboots, such as the AUT position or the maximum and minimum positions of the axis.

Unfortunately, there doesn't seem to be a command to stop the machine from making the current movement. Instead, the user will have to resort to hardware solutions, such as the ESTOP button on the PCB, or simply pulling the Arduino USB connection or pulling the power connection to the AC power supply connected to the PCB.

5 Background Theory

5.1 Near to Far Field Criteria

6 Appendix

6.1 GRBL Settings

The settings of the current GRBL setup after typing $$ into the GRBL serial prompt. More info on the GRBL settings can be found in the GRBL wiki[8].

```
Grbl 1.1h ['$' for help]
[MSG:'$H'|'$X' to unlock]
$0=10
$1=25
$2=0
$3=6
$4=0
$5=0
$6=0
$10=2
$11=0.010
```

[8] https://github.com/gnea/grbl/wiki/Grbl-v1.1-Configuration#grbl-settings

```
$12=0.002
$13=0
$20=0
$21=0
$22=1
$23=3
$24=25.000
$25=500.000
$26=250
$27=1.000
$30=1000
$31=0
$32=0
$100=200.000
$101=200.000
$102=200.000
$110=500.000
$111=500.000
$112=500.000
$120=10.000
$121=10.000
$122=10.000
$130=900.000
$131=900.000
$132=900.000
ok
```