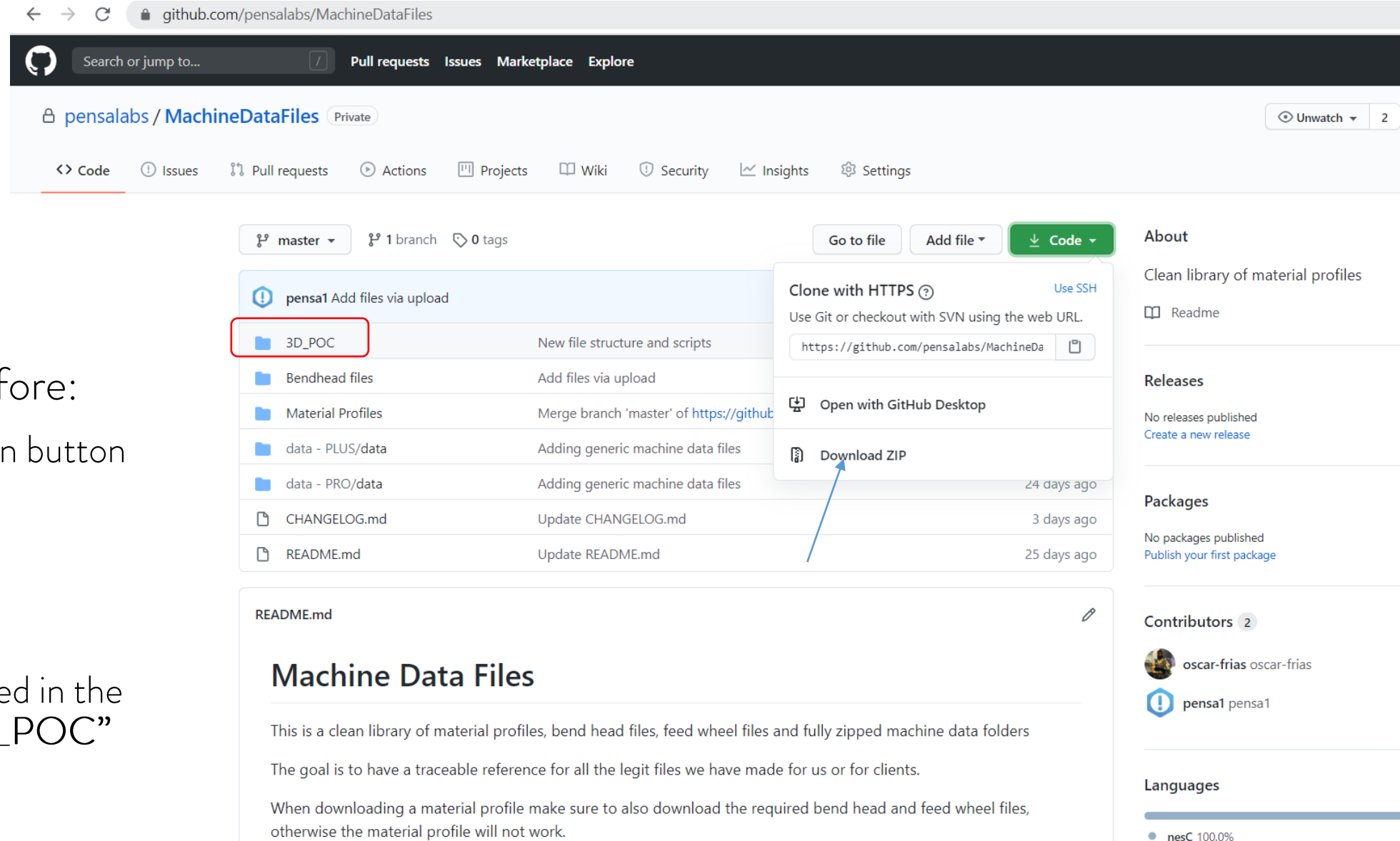# TinyTerm V2

Usage with the 3D POC

07/24/20

Using TinyTerm V2 with 3D POC

1. Download and unzip Processing on your computer
2. Grab the tinyTerm files from Github
3. Open sketch in processing
4. Connect the tinyG to your PC via USB cable
5. Boot tinyG
6. Run sketch [CTRL+R] (or click the arrow)
7. Select second COM port on the list
8. Load init file (3D_INIT.json)
9. Home the machine
10. Enjoy

Grab the init file and sample scripts from [Pensa's github](#)

Same deal as before:

- Click on the green button
- Download zip
- Extract files

You're <u>only</u> interested in the files inside the "3D_POC" folder

Grab tinyTerm V2 [from github](#)

Click the green button

Download zip and extract files

Extract files and run main sketch

- Run the sketch called "TinyTerm.pde"
- You'll need to have [Processing](#) in your computer, and the controlP5 library installed
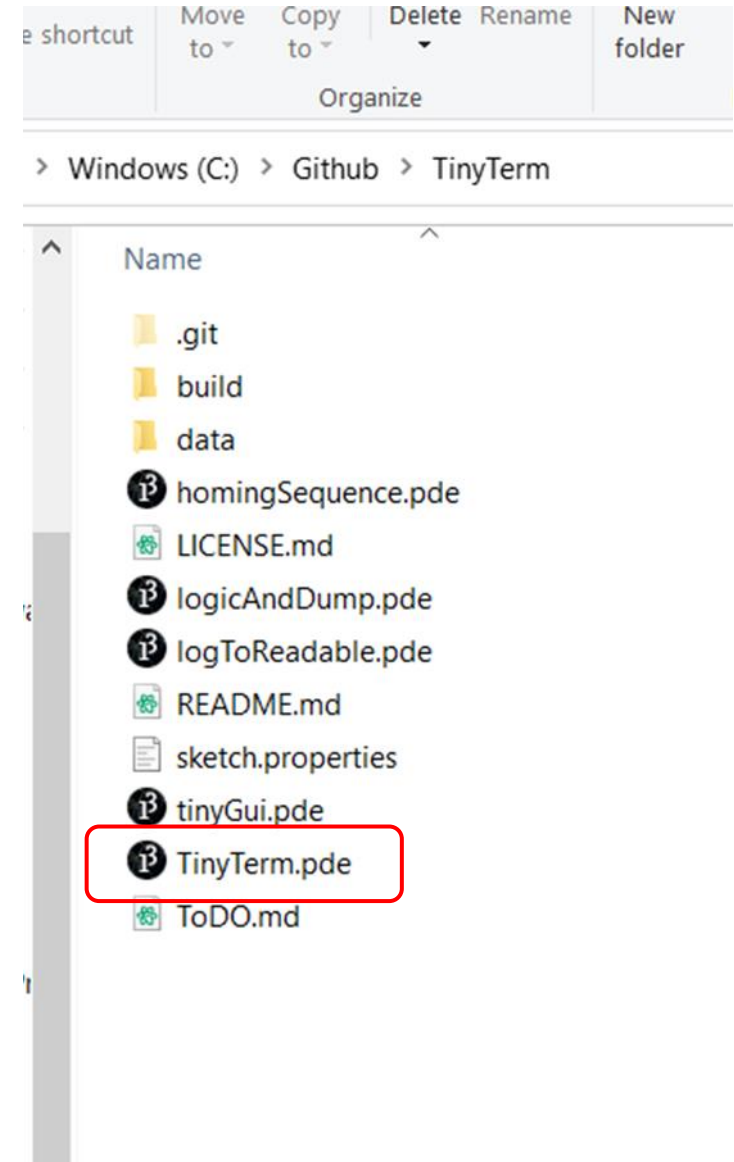
Run the sketch

Make sure the tinyG is ON and connected to your computer before running the sketch

- Run the sketch by clicking on the green arrow on the top left, or by using the CTRL+R keyboard shortcut

TinyTerm | Processing 3.5.4

File Edit Sketch Debug Tools Help

▶ ⬛ Run

| TinyTerm | homingSequence | logToReadable | logicAndDump | tinyGui |

```
1  // TinyTerm for DIWire
2  // By Oscar Frias (@_frix_) and Dinesh Durai 2020
3  // www.oscarfrias.com
4  //
5  // TinyTerm is a simple interface for a serial terminal to control the TinyG
6  // We needed a quick way to send both tinyG commands and GCode to the tinyG,
7  // while at the same time have a quick and easy way to dump the init settings
8  // to the board without having to type them every time.
9  //
10 // TinyTerm loads a JSON file called "init.json" from the /data folder contai
11 // the init settings that we want to dump on the tinyG, and extracts the arra
12 // json objects called "commands" as a long string. See the JSON file to lear
13 // properly format this init file.
14 //
15 // Then it opens a serial connection, searches for the tinyG, and
16 // then dumps the JSON-turned-into-strings into the tinyG
17 //
18 // The complete list of tinyG commands and settings can be found @
19 // https://github.com/synthetos/TinyG/wiki/TinyG-Configuration-for-Firmware-V
20 // Or if you have a tinyG connected to the TinyTerm, send the command "$$"
21 //
22 // The GUI gives you a textArea to show the history of what has been sent and
23 // and a text box to type and inject GCode or tinyG Commands on the fly into
24 // The interface is created using the controlP5 library.
25 //
26 // Thanks to http://stackoverflow.com/questions/29107544/serial-com-port-sele
27 // for the help with adding the Serial port dropdown list
```
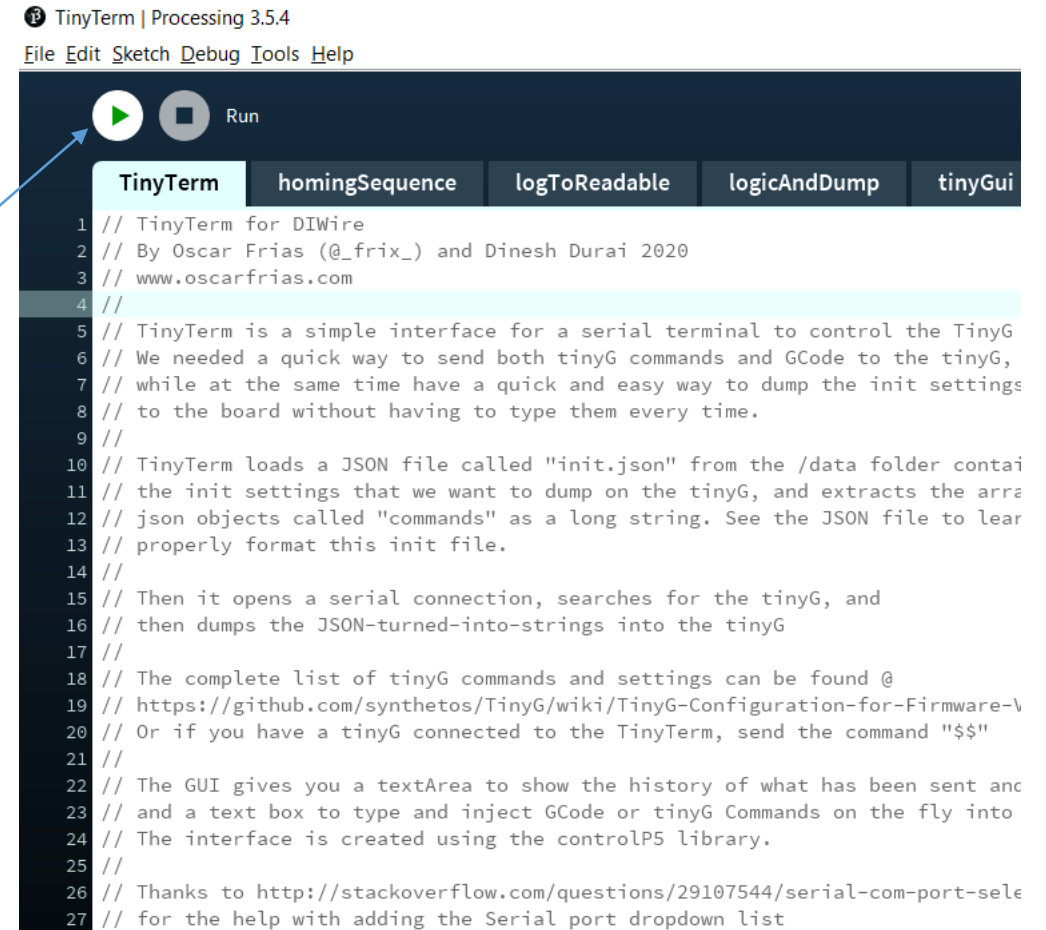
Connect to the tinyG

- Once the applet loads, select <u>the second</u> COM port on the list. You should see some gibberish pop on the terminal, and the rest of the buttons appear.

It's important to select the second port, otherwise you'll be trying to talk to a ghost port opened by the tinyG, and nothing will happen.
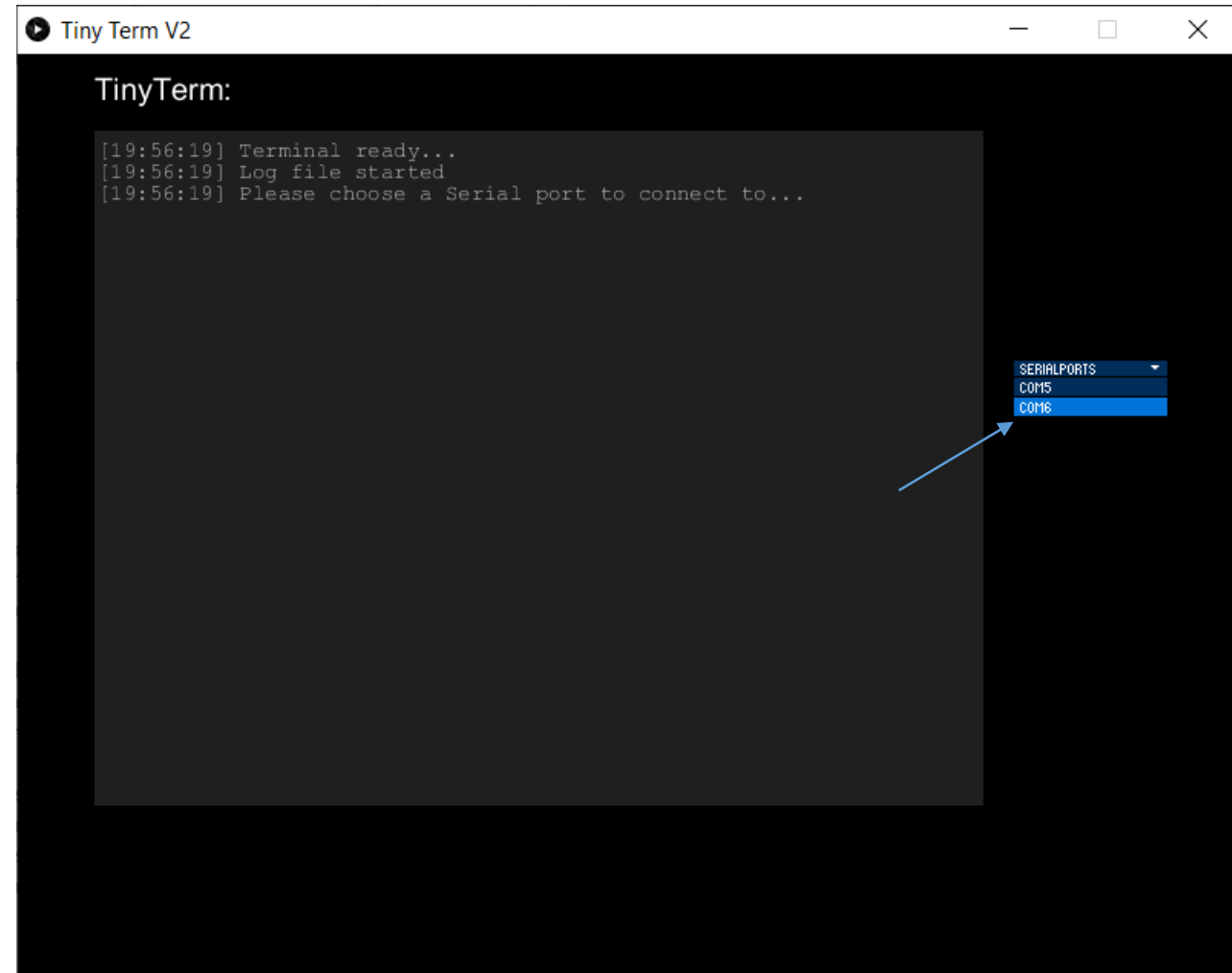
Load the init file

- Click on the "Load File" button and browse for the init file.

- You'll see lots of lines get filled in the terminal, that's OK.
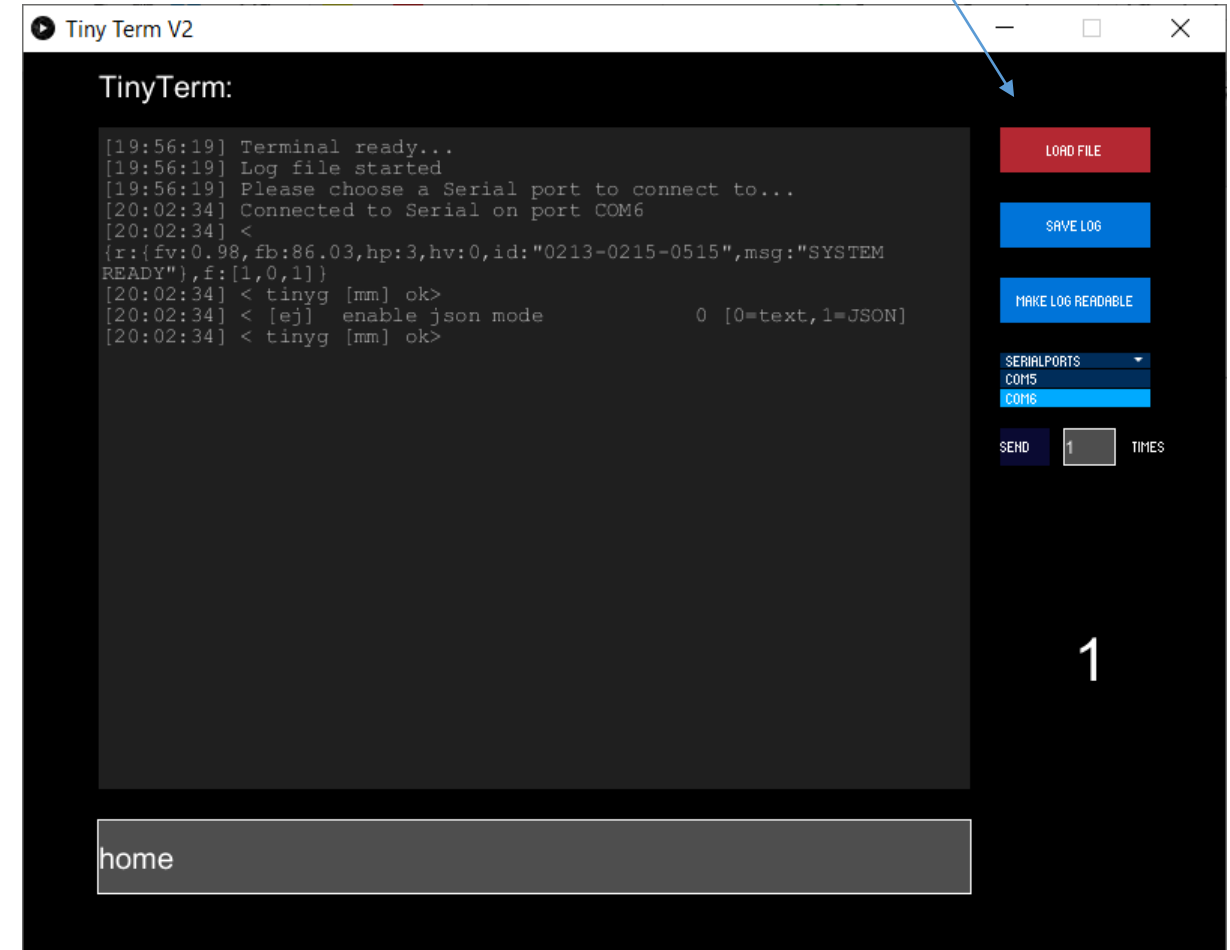
Only load the 3D POC init!

Make damn sure to open only the 3D_INIT.JSON file. Any other init file will make the machine misbehave and crash.
Try not to damage the POC by mistake.
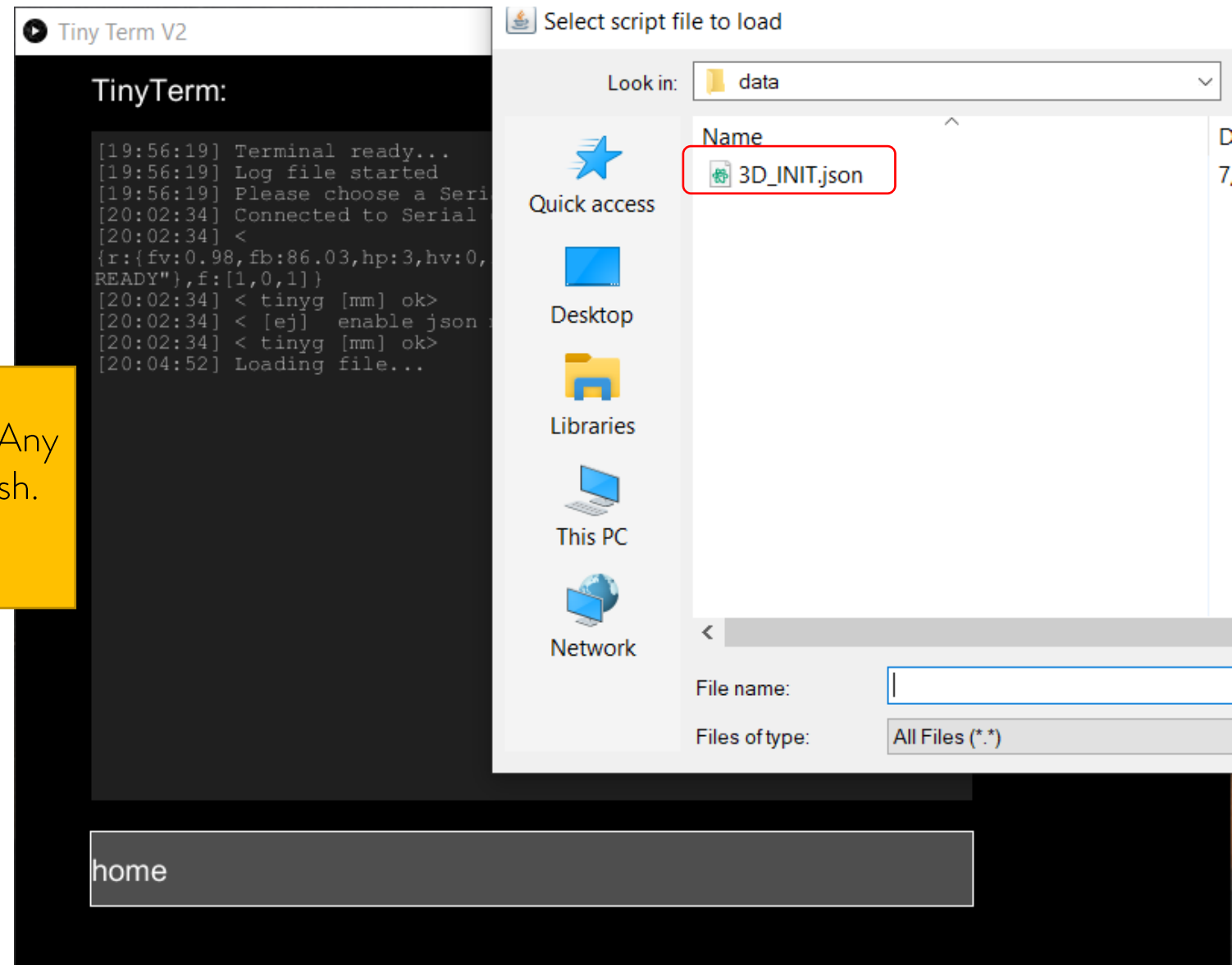
Home the machine

- To home the machine simply type HOME (lower or upper caps ok) in the textfield and hit [Enter]
- The machine will go through the motions:
  - Home Z axis
  - Home X axis
  - Home A axis
  - Go to A20 Z0

- Profit!

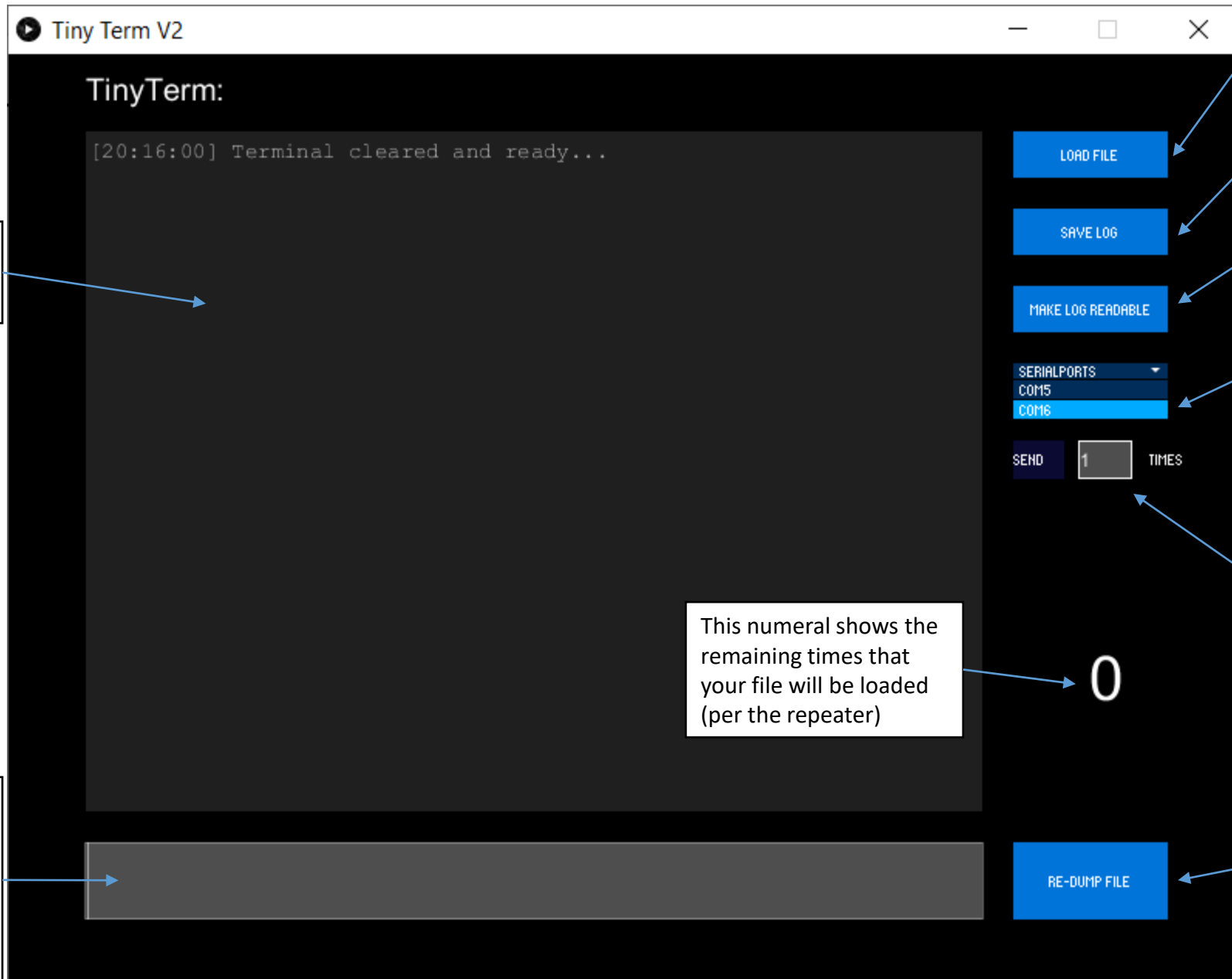# TinyTerm Usage and commands

Interface

Main terminal, shows all the incoming data from the tinyG

To load the init file or a gcode script

Saves the content of the terminal to a log file

Makes the log files human readable

Serial port selector. Use only the 2nd port on the list

Repeater: Type the number of times you want your script file to be sent. Very useful if you're making many copies of the same part. **THE FILE'S LAST LINE MUST SAY "(EOF)" FOR THE REPEATER TO WORK.** Just add it as a comment at the very end of your script.

This numeral shows the remaining times that your file will be loaded (per the repeater)

Text input field. Here you can type tinyG commands, raw Gcode or tinyTerm commands (see cheat sheet next slide). Type and hit [ENTER] to execute

To re-send the last file you loaded. Useful if you're doing quick edits to your file. Simply save the file and click re-dump, no need to load again
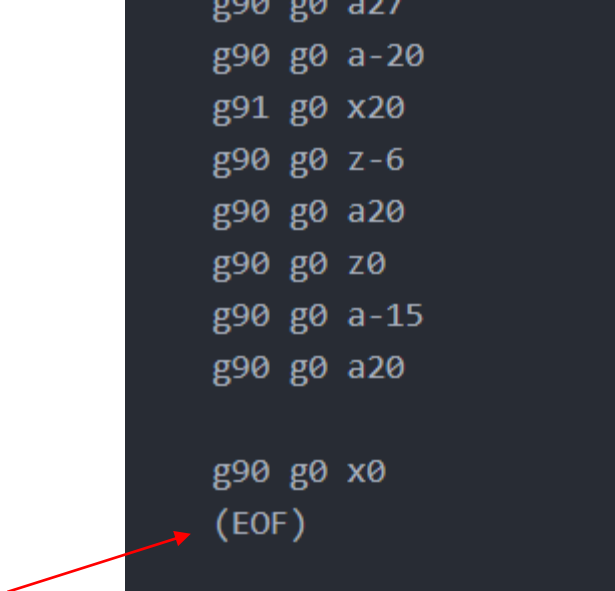
TinyTerm command cheat sheet

Command/ key	Action	Notes
Home	Home machine	Homes all 3 axes
Homea	Homes A axis only	Z axis must be homed, the machine could crash or simply miss the sensor
Homex	Homes X axis only	
Homez	Homes Z axis only	The machine will search for the flag downwards. So the carriage needs to be above the sensor before homing. You can move it with G91 moves
CLS	Clears the terminal	Saves the contents of the terminal in a log file and then clears all the text. Very useful if you want to catch a particular line or if it's too busy, or if your computer starts lagging (this terminal hogs memory, sorry. Will figure it out in the future)
Exit	Closes the app	Will save a log file and close the applet.
[ESC]	Exit prompt	The terminal will ask you if you want to exit or not, simply type 'Y' or 'N'
!	Feedhold	tinyG will stop moving, but the queue is still full. Send a '~' to continue
~	Feedhold release	Continue running the program
%	Flush	Empties the queue both on tinyTerm and on the tinyG
!%	Stop and Flush	Send both characters together to perform a hard stop and flush

File types

- Only the init files are treated specially, because they are comprised of at least two arrays that need to be extracted in a particular way. To be recognized by tinyTerm, they must have two conditions:
  - The filename incudes INIT
  - The file extension is JSON

- Otherwise, TinyTerm is inclusive and promiscuous. It will load any file extension you open. The tinyG will complain or try to make sense of it if it's not Gcode, but for most of it, it doesn't really matter.

- That said, I'd recommend you save your scripts as .txt, .nc, .gc, .tap, or whatever other format you can open with a text editor. And don't use any special Unicode, just let Windows do its thing.

Notes about the repeater

- As mentioned before, for the repeater to work, the file to be repeated MUST END WITH "(EOF)" (no quotation marks). I put it in parenthesis so the tinyG doesn't complain of a wrong command (it's seen as a comment).

- The algorithm running the repeater looks for this three letters to determine that the file is done. I may think of a different way to do it in the future, but right now this is the way. So make sure that the last line of your file is (EOF)

```
g90 g0 a27
g90 g0 a-20
g91 g0 x20
g90 g0 z-6
g90 g0 a20
g90 g0 z0
g90 g0 a-15
g90 g0 a20

g90 g0 x0
(EOF)
```

Troubleshooting

- Sometimes the homing sequence for the A axis won't finish. The solution is to re-load the init file and re-run the homing routine. This normally works. Otherwise a good reboot will do too.

- If the machine stops unexpectedly, or if the terminal stops running your commands, you can flush everything with a '%' or a '!%', and then send a '~'. This normally does the job.

- Otherwise there's not much troubleshooting to do. But this is a very simple sketch that tries to communicate with quite a snowflake of a board. So if everything fails, nothing as simple as to stopping the sketch, rebooting the tinyG and restarting the sketch (load init file, home, profit) will do it.

Thanks!

Let me know if you have any questions!