Thor’s Arsenal

User Manual

Greenface Labs -- 2024

Table of Contents

[Description 3](#_Toc162867099)

[Quick Start 5](#_Toc162867100)

[Overview 7](#_Toc162867101)

[Signals 7](#_Toc162867102)

[Functions 7](#_Toc162867103)

[Controls 7](#_Toc162867104)

[Features 8](#_Toc162867105)

[Front Panel 9](#_Toc162867106)

[Open-Source Project 10](#_Toc162867107)

[Operation 11](#_Toc162867108)

[General Operation 11](#_Toc162867109)

[Screen Saver 12](#_Toc162867110)

[Output Details 13](#_Toc162867111)

[Waveform Details 16](#_Toc162867112)

[Settings Details 17](#_Toc162867113)

[Bounce Function 18](#_Toc162867114)

[Remote Modes 19](#_Toc162867115)

[WiFi 19](#_Toc162867116)

[USB Direct Mode 21](#_Toc162867117)

[Terminal Mode 25](#_Toc162867118)

[Software Update 30](#_Toc162867119)

[Disaster Recovery 30](#_Toc162867120)

# Description

Thor’s Arsenal is a 14HP Eurorack module that generates signals that can be used to modulate voltage-controlled functions such as VCO, VCF, VCA, etc... Eight independent outputs can be triggered by any of four trigger inputs and can also be modulated by the two input signals. In addition to the front panel controls, Thor’s Arsenal provides GUI and Terminal interfaces via USB. These interfaces enhance the user experience by making it easier to control the module and by adding some useful features. *The Bonkulator also has a WiFi interface that is similar to the GUI desktop app.*

#### 8 Outputs

* In general, the Output value follows the selected waveform.
* Waveforms can be from the factory list or user defined.
* Factory waveforms include ramp, sine, haystack and others.
* Waveforms can be scaled and offset by UI setting or by CV input.
* Waveform period can be set via UI or by CV input.
* Waveform repetitions can be set from 1 to 65,535.
* Waveform quantization can be enabled or disabled.
* Waveform can be randomized from 0-99%.
* Outputs can be programmed to be triggered or clocked by any of the trigger inputs.
* Outputs have LED level indicators

#### 4 Trigger/Clock Inputs

* Triggers activate output functions.
* When Output is set to Ext Clk mode, trigger inputs increment the waveform step.
* The trigger inputs respond to a positive-going signal or a button press.
* Each trigger can be independently enabled or disabled.
* An LED indicates when any attached output is active.

#### 2 CV Inputs

* Input Range: +/-10V (+/-5V for the Bonkulator).
* CV inputs can control many of the output parameters.
* LED level indicators.

# Installation

**Thor’s Arsenal** has an “either way” power connector. A 10-pin connector is typically used, but a 16-pin connector will work if the red line is lined up with either end of the mating connector.

A close-up of a circuit board

Description automatically generated

**For the Bonkulator**, care must be taken to connect the power to the right connector. The power connection is “either way”. Refer to the following image for the location of the power connector.

A close-up of a circuit board

Description automatically generated

# Quick Start

Thor’s Arsenal can be used in an endless variety of ways, but here is a great way to start exploring. These instructions are for use with a Moog Mother-32 but can be easily adapted to most modular synthesizers. Addition of an Echo module is highly recommended.

#### Mother-32 Set-up

* VCA Mode – ON (do not use the envelope generator)
* VCO Mod Amount – 50%
* VCO Mod Dest - Frequency
* VCF Mode – Low Pass
* VCF Mod Polarity – +
* LFO Wave – triangle
* LFO Rate – 30%
* Cutoff – 50%
* Resonance – 20%
* Sustain – Off
* Frequency – 50%
* Pulse Width – 50%
* VCO Wave - Pulse
* Octave – 4
* Note – 8

#### Thor’s Arsenal Set-Up

* Output 0 and Output 1 (common settings)
  + Waveform: Ramp
  + Init Delay: 00000
  + Active Time: 03010
  + Idle Time: 00000
  + Repeat: 00000
  + T0: Enabled
  + T1: Disabled
  + T2: Disabled
  + T3: Disabled
  + CV0: Off
  + CV1: Off
  + Scale: see below
  + Offset: 000
  + Randomness: 00
  + Quantize: No
  + Idle Value: 2048
  + Clock: Internal
  + Range: +/-5V
* Output 0
  + Scale: 100
* Output 1
  + Scale: -100

Connect Thor’s Arsenal Output 0 to the M-32 VCO 1V/Oct input and the VCF Cutoff input. Connect Output 1 to the M-32 LFO Rate input. Press Trigger 0 on Thor’s Arsenal and you should hear a sound much like that found [here](https://www.greenfacelabs.com/posts/bonkulator-quick-start-example/). Experiment with the various Output parameters. Try increasing the Randomness on Output 0 to 20.

# Usage Overview

Thor’s Arsenal has a button for each trigger and output as well as four ‘arrow’ buttons for navigating the display.

Pressing the output buttons brings up the configuration menu for each output. Outputs have many parameters. Among them are the Waveform to be sent and its Active Time. Select which triggers activate the output in the Output configuration menu. Complete information on all the parameters can be found in the Output Details section of this document.

Pressing the Adjust Knob activates various functions of Thor’s Arsenal. The use of this button is context-sensitive and is only active within the Settings function. Pressing and holding this knob for longer than 2 seconds brings up the Settings function.

In addition to the front panel controls, Thor’s Arsenal provides Desktop and Terminal interfaces. *The Bonkulator also has a WiFi interface*. You can do everything through these interfaces that you can do from the front panel and more. This benefit derived from using these interfaces cannot be fully appreciated until it is experienced directly.

## Signals

* Outputs (8) – Analog signal output. Range: +/- 10 volt. *Bonkulator: +/-5V or 0-10V*.
* Built-in and programmable waveforms.
* Triggers (4) – Digital signal that is activated by a 0 to +5V transition.
* CV (2) – Analog signal input. Range: +/- 10 volt

## Functions

* Output (0-7) – For configuring the 8 outputs
* Settings – For module configuration, factory reset, and extra functions like Bounce.

## Controls

* Output Buttons (0-7) – For selecting from the 8 outputs
* Trigger Buttons (0-3) – For manual triggering
* Move Buttons (^v<>) – For navigating the display.
* Adjust Rotary Encoder – For adjusting selected parameter
* Activate Push Button – Push the Adjust knob to activate functions
* More controls are available via the remote interfaces

## Features

* 128x64 two-color OLED graphic display
* Web, Desktop and Terminal interfaces provide rich user experiences with the addition of computer resources like a keyboard, mouse, display and scripting.
* All settings are stored in FRAM memory so that they are retained when power is off
* Inputs and Outputs are protected against damage during typical use
* Output and Input LED indicators show voltage levels present at the jacks
* Trigger Indicator LEDs show if any Outputs have been activated by that trigger
* *Bonkulator output range is switchable: +/-5V or 0-10V (board models 4.0 and later)*
* *An Expansion module is available for the Bonkulator that provides an external USB interface with some interesting extra features. See “The BonkDaddy”.*

## Front Panel

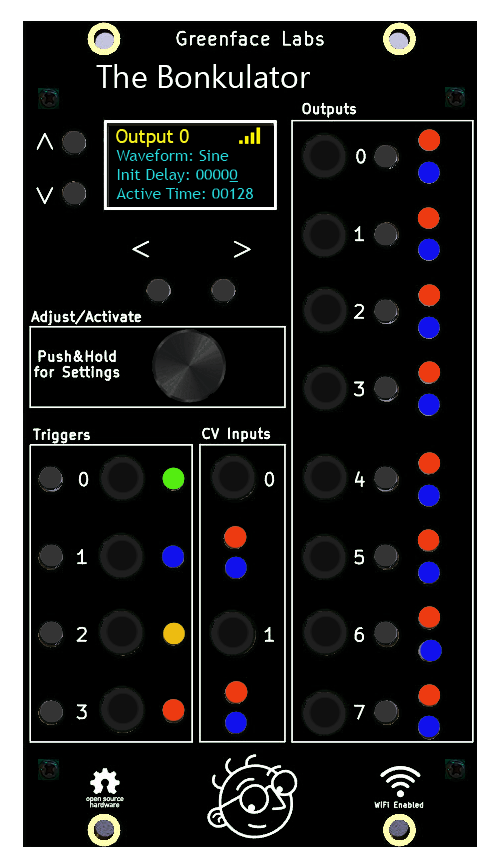


Figure 1 - Front Panel

## Open-Source Project

Thor’s Arsenal is an Open-Source project that has an Arduino RP2040 at its core. The code is written in C++ and makes use of several public domain code libraries for managing the various purchased hardware components such as the display and the FRAM non-volatile memory.

Because it is Open Source, in addition to being used by musicians and sound designers to create new and amazing sounds, Thor’s Arsenal can be used by students and entrepreneurs to create new, amazing modules. And, of course, the open-source community will generally improve the product as time goes along. Links to the design repositories can be found on Thor’s Arsenal web page.

# Operation

## General Operation

The typical procedure for using Thor’s Arsenal involves choosing a set of waveforms that are needed for a particular application. Then a set of triggers are defined and the various parameters for each waveform are entered. Once configured, Thor’s Arsenal reacts to trigger signals, sending the waveform for each Output that was triggered.

* The Output is selected using the Output buttons.
  + Terminal uses the **‘f’** command. E.g. F3
  + The app has dedicated Function buttons.
  + Each Output has several parameters.
  + Waveforms are selected as one of the parameters
  + Choose from 7 pre-installed waveforms and 8 user-defined waveforms.
* Parameters are adjusted using the arrow keys and the Adjust knob.
  + Parameters are selected using the **up** and **down** arrows.
  + Digit to be adjusted is selected by using the **right** and **left** arrows.
  + When entering strings, holding the right or left arrow down for 2 seconds inserts (right-arrow) or deletes (left-arrow) a character.
  + Use the Adjust knob to increment and decrement the selected digit.
  + Parameter values may be directly entered using the app or Terminal Interface.
* Enable Triggers for each Output that is to be used
  + Outputs can be triggered by any or all of the four triggers.
  + Triggers T0 – T3 can be either ENABLED or DISABLED. They are set as above.
* Operation of each Output is triggered by a variety of methods.
  + Pressing the selected trigger button. (Manual trigger)
  + Receiving a low to high transition on the selected Trigger jack.
  + Terminal uses the ‘**!**’ key to trigger the selected trigger.
  + Web Interface has a dedicated Trigger button for each trigger.
* If an Output is already active and REPEAT is non-zero, receiving a trigger command will re-trigger the output.
* If an Output is already active and REPEAT is zero, receiving a trigger command will stop the output.
* Trigger signals merely advance the waveform when in External Clock mode.
* Many parameters can be modulated by the two input signals.
  + Set CV0 and CV1 to select the modulation target parameter (scale, offset, active time, idle value, randomness or off).
  + CV In jacks have a range of -10V to +10V (Bonkulator: -5V to +5V)
  + The sample period for the CV inputs is approximately 20ms. There is no filtering at the input to the converter, so frequencies above 5Hz will produce unpredictable modulation. Could be interesting…
  + If both CVs are specified, the effect is to cascade CV1 after CV0

## Screen Saver

In order to maximize display life, the display will turn off after a set time. The default time is 15 minutes. The time can be set in the Settings function with a range of 1 to 9999 minutes.

When the screen saver activates, the screen will show an animation of falling stars. If no key is pressed within 15 minutes after the animation starts, the display will turn blank.

Press any button or send the ‘z’ command to bring the display back.

# Output Details

The Output signal ranges from +10.67V to -10.67V.

Bonkulator: The Output signal ranges from -5.33V to +5.33V or from 0 to 10.66V. (see Range setting)

When the Output isn’t active, the **Idle Value** is sent. Use Output parameters to select the Idle Value.

Web mode has a dedicated slider control for several parameters, and Terminal mode uses the *S, O* and *c* keys followed by a number. See the table of remote commands for more details.

Thor’s Arsenal Outputs have several parameters. Details for each parameter follow. A star(\*) next to the name indicates that output is not calibrated.

1. Waveform – These are the signals that can be assigned to the outputs
   1. Fifteen waveforms are available, seven are factory-installed and eight are user-defined. Waveforms are stored as 128 values that range from 0-1023. Depending on the chosen Active Time, 10 to 128 of these values will be used to create the output signal.
   2. The factory-installed waveforms are: Sine, Haystack, Ramp, Pyramid, Pulse, Maytag, and Toggle.
   3. The user-defined waveforms can be entered by the following methods:
      1. Manually via the front panel controls
      2. Drawing the waveform using the app.
      3. Via script using either Terminal mode or an app macro.
      4. Recorded from the CV0 input
2. Init Delay – The time in milliseconds to delay before sending the Waveform
   1. Range 0 to 65535 msecs
3. Active Time/Parts – The length in milliseconds of the waveform when in Internal clock mode or the number of samples to be used when in External clock mode
   1. Active Time ranges from 10 msecs – 65535 msecs
   2. Parts ranges from 10 to 128
   3. When in Internal clock mode, the Active Time affects the number of samples used and thus the sample rate.
      1. When the Active Time is a multiple of 128, the number of samples used will be the full 128.
      2. When the Active Time is less than 128, the number of samples will equal the Active Time.
      3. When the Active Time is greater than 128, the number of samples will be the result of a best-fit algorithm.
   4. When in External clock mode, the Delay and Repeat settings are ignored
4. Idle Time – The time in milliseconds to delay before repeating the Waveform
   1. Range 0 to 65535 msecs
5. Repeat – The number of times to repeat the waveform
   1. Range 1 to 65535
   2. Set Repeat to 0 to repeat forever.
6. T0 thru T4 – These are the Trigger/Clock inputs
   1. Each of the four triggers can be either Enabled or Disabled. That means that every output can be triggered by as many as four different trigger signals.
   2. Each trigger can be manually activated by pressing the associated button.
   3. Triggers are activated by a rising edge 5V signal at the input.
   4. In Internal clock mode, triggers re-trigger the trigger state of the output. So, if an output is Idle, receiving a trigger will activate the output. If the output is already active, a trigger will start sending the waveform from the beginning.
   5. When in External clock mode, every trigger merely sends the next output sample.
7. CV0 & CV1 – These are the control voltage inputs.
   1. They can be set to modulate the following Output parameters:
      1. Scale – Range .01x – 100x (Bonkulator: .1x – 10x)
         1. Signal gain is = 100.2Vi, where Vi is the input voltage.  
            This results in a gain range of .01 - 100 for an input range of -10V - +10V.
      2. Offset – Range +/-10.66V
         1. Output is offset by the input voltage.
      3. Active Time
         1. It modulates the Active Time by modifying the time increment for each step of the output wave. The input range is:  
            0-10V for increments of 1 ms - 512ms.
      4. Idle Value
         1. The output signal level follows the level of the input.  
            This parameter only applies when the trigger is active and Idle Time is greater than 0.
      5. Randomness
         1. An input range of 0-10V results in a randomness of 0-100 per cent of the signal's amplitude.
   2. A star (\*) next to the name indicates that input is not calibrated.
   3. Additionally, a change to the CV input can cause a trigger. Select *Trigger* as the CV target to do this.
8. Scale
   1. Signal level range is +/-10.00V.
   2. Bonkulator: Signal level range is +/-5.00V or 0-10V depending on Output Range setting
   3. Scale ranges from -100% to 100%
   4. Negative values invert the signal
9. Offset
   1. Range - +/-10.0V%
   2. Bonkulator Range - +/-5.0V%
10. Randomness – 0 to 100%
11. Quantize - Yes/No
    1. Quantize yes forces the output to match the 1V/oct pattern.
12. Idle Value
    1. The Idle Value is sent when the Output is not active.
    2. It is also sent during the Idle Time period when the output is active.
13. Clock
    1. Internal – The next sample is sent based on the Active Time setting.
    2. External – The next sample is sent on the rising edge of the enabled trigger.
14. Range (Bonkulator Only)
    1. +/-5V (Default)
    2. 0-10V (board models 4.0 and later)
15. Trigger Control
    1. Trigger Hold-off
       1. Suppresses triggers for set time after receiving a trigger.
       2. Range 0-65535ms
    2. Trigger Skip
       1. Skips set number of triggers
       2. Range 0-65535ms
    3. Trigger Density
       1. Randomly accepts a percentage of triggers
       2. Range 0-100%

# Waveform Details

Thor’s Arsenal has seven factory-installed and eight user-defined waveforms. The user-defined waveforms are managed in the Settings section where the initial names shown below may be changed.

1. Sine – Classic Sine Wave
2. Haystack – First half of Sine wave
3. Ramp – Asymmetrical sawtooth
4. Pyramid – Symmetrical sawtooth
5. Pulse – The Pulse is formed by the waveform transitioning from the Active Value to the Idle Value
   1. **Idle Time** must be non-zero for a pulse to form
   2. **Offset** is disabled for this waveform
   3. A variety of pulse shapes can be formed by adjusting **Scale** and **Idle Value**
6. Maytag – A random sequence
7. Toggle – Toggle creates a scaled and offset square wave that toggles with every transition of the trigger input
8. User Wave 0-7
   1. User Waves are managed in Settings
   2. Default is flat line
   3. User Waves can be manually entered point-by-point or recorded at CV0
   4. User waves can be drawn and imported via the app.

# Settings Details

The Settings function displays information about Thor’s Arsenal as well as allowing some adjustments to various system parameters. The Bounce function also appears as one of the Settings parameters. Refer to the table below.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Min** | **Max** | **Sub Menu** |
| Version | -- | -- |  |
| Name | -- | -- |  |
| Encoder Type | Normal | Reverse |  |
| WiFi | Enabled | Disabled | Press Activate |
| Screen Saver | 1 | 9999 |  |
| Waveforms | Wave 0 | Wave 7 | Press Activate |
| Reset | -- | -- | Press Activate |
| Calibrate | Inputs | Outputs | Press Activate |
| Bounce | 0 | 1 | Press Activate |

Notes:

1. **Version** shows the software version number. This is useful when communicating about bugs and questions.
2. **Name**
   1. The Name is displayed on boot-up and is displayed in the remote screens.
   2. This helps keep track of which unit is associated with which screen when there are multiple units installed.
3. **Encoder Type** takes the two types of rotary encoder into account. This makes them easier to purchase if one type is unavailable. If your Adjust knob is working opposite as it should, this is the setting that corrects it.
4. **WiFi** can be disabled here if you want to turn it off but retain your password.
5. **Screen Saver** prolongs the life of the OLED display. The screen will blank after the set number of minutes. Press any button to bring the display back.
6. **Waveforms** is where User Waveforms are managed. Each waveform contains 128 samples. Waveforms can be entered manually, drawn or imported using WiFi or the Desktop app, or recorded from CV0. Press Activate after selecting the waveform to bring up the Manage Waveform menu.
   1. **Name** – Enter a meaningful name for your waveform
   2. **Index** – For manual entry. Range 0:128
   3. **Value** – For manual entry. Range 0:4095 (see Output Range)
      1. Displays current value at the element pointed to by Index.
      2. New values are entered by pressing Activate.
      3. Pressing Activate also increments Index.
      4. Value of 0 corresponds to negative full scale
      5. Value of 4095corresponds to positive full scale
      6. Value of 2048 corresponds to mid scale
   4. **Sample Time** – This is the time between samples in milliseconds. Calculate using this formula: length of waveform to be sampled in milliseconds / 128
   5. **Record On** – The event that starts the recording of a waveform
      1. **Immediate** – Starts recording when Activated
      2. **Trigger** – Waits for a trigger on any trigger input
      3. **+Change** - Waits for a positive change in the voltage to start recording
      4. **-Change** - Waits for a negative change in the voltage to start recording
   6. **Note** – The WiFi interface allows for point and click setting of the Index and Value. This greatly speeds up entering new waveforms.
7. **Reset** brings Thor’s Arsenal’s parameters back to their factory defaults. Note that WiFi parameters are unaffected.
   1. **Parameters**
   2. **Input Corrections**
   3. **Output Corrections**
   4. **All**
8. **Calibrate** is where the 8 output and 2 input signal paths are calibrated.
   1. **Inputs** – Requires DC voltage source, accuracy .1%
      1. First apply 0VDC to input and adjust Offset to get a correct reading.
      2. Apply 4VDC to input and adjust Scale to get a correct reading.
      3. Test other voltages to verify accuracy. (note that full scale is +/-4.967V)
      4. Set Calibrated to Yes when done.
   2. **Outputs** – Requires DC Voltmeter, accuracy .1%
      1. Adjust Offset until voltage at Output is as close to zero as possible.
      2. Adjust Scale until voltage at Output is as close to 5.333 as possible.
      3. Set Calibrated to Yes when done.
9. **Bounce Function** – An advanced Sample & Hold + DVM (see below)

# Bounce Function

The Bounce function measures the voltage present at the selected CV Input when triggered and displays it to the user. It also scales and offsets the measured voltage and “bounces” the result to the selected output.

This creates a scalable sample and hold function.

All the triggering modes apply.

The Bounce function is accessed via the **Settings** menu. Select which input you want then press Activate to enter the Bounce function.

# Remote Modes

## WiFi

Thor’s Arsenal is equipped with a 2.4GHz WiFi interface. When connected to your network, you can control Thor’s Arsenal using the [Web Interface](https://www.greenfacelabs.com/bonkulator-control/) (see example screen below).

WiFi can be disabled in the Settings function. Make sure it is enabled before proceeding in the connection process.

#### Connecting to WiFi

1. Enter WiFi function and press Activate to scan for networks.
2. Use the up and down arrow keys to select your network. Press Activate.
3. Use the right and left arrow keys with the Adjust control to select each character of your password. Press Activate when your password is entered. Note: It is much easier in Terminal mode. Type your password preceded by $, then press Enter.

Thor’s Arsenal will try to connect using the password that you entered. If successful, the screen will display the connection data. This includes the connection status, the IP it can be found at and the signal strength. The signal strength will also be displayed in the upper right corner of the display as 1-4 bars. The signal strength display is shown in most functions.

If the connection fails, Thor’s Arsenal will give its best explanation of why it failed.

Thor’s Arsenal remembers its connection and will reestablish it if the power is recycled.

Visiting the WiFi function again while connected will display the connection data. Press Activate to re-enter the scan for networks screen.

If Thor’s Arsenal connects to the network, but the Web app has trouble connecting to Thor’s Arsenal, it often helps to recycle the power to Thor’s Arsenal. Try the Web app again after Thor’s Arsenal restarts and re-connects to WiFi.

Important: Thor’s Arsenal only operates via http: Browsers often try to force https: For this reason, it is necessary to remove the ‘s’ from the URL in the address bar of your browser.



Figure Chrome Browser

#### Using WiFi

Thor’s Arsenal’s Web Interface was designed to emulate the physical front panel. However, because the Web Interface enjoys the added functionality a computer brings, it has significant differences. Please refer to Fig. 3 below.

While you can still adjust parameters a digit at a time, you can now enter parameters from the keyboard by first clicking on the parameter then using the dedicated entry field.

Adjusting parameters digit by digit is possible by using the right and left arrows to select the digit and using the Inc and Dec buttons to increment and decrement the digit.

When entering User Waveforms, you can click on the graph to set the Index or Value. This saves a lot of time over manual entry.A screenshot of a computer

Description automatically generated

Figure Web Interface Example

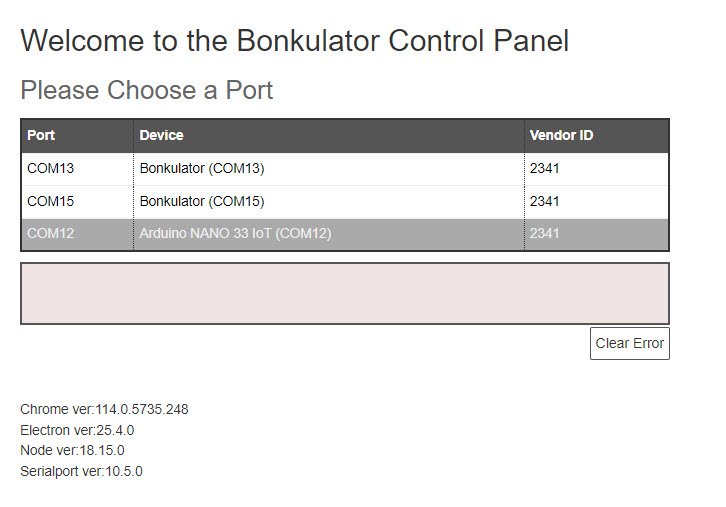
## USB Direct Mode

USB Direct Mode allows you to control Thor’s Arsenal via USB using a desktop app that emulates the Web Interface. It also has the added benefit of being able to display trigger events as they occur rather than having to request them as in the web app. USB Direct Mode is enabled on Thor’s Arsenal in two ways:

* In Settings
  + USB Direct Disabled
  + USB Direct Enabled
* Terminal Command
  + U0 - USB Direct Off
  + U1 - USB Direct On

Get the desktop app from the [website](https://www.greenfacelabs.com/bonkulator/#download_the_code). Unzip the folder into a convenient location and run the executable file contained in that folder. If you’re on a Mac, you’ll need to de-quarantine the app. See the section below if you need to do this.

Once you have the app running, you should see a screen like the following:



The control panel should recognize and enable your Thor’s Arsenals. (The Arduino Nano 33 IoT shown here goes to a Spankulator). Choose a Thor’s Arsenal by clicking on the port entry of the table. It will bring up a screen that looks like the WiFi app. Use it like you’d use the web app.

# De-quarantining on the Mac

If you download the app on the Mac and try to run it, it will say that the app is damaged and can’t be run. This is an attempt by Apple to get all apps to be distributed through their store. Until this app is available there, you’ll need to de-quarantine it to run.

To do this, using Finder navigate to the folder that contains the folder that contains the app. Select the folder that contains the app. Then click on Finder->Services->New Terminal at Folder.

This will open the Terminal. Once there issue this command:

xattr -d com.apple.quarantine bonk\_cp\_app.app

The app will now be ready to use.

# Drawing Waveforms

It is possible to draw your own waveforms in the Desktop and WiFi apps. Just select Settings->User Waveforms. Choose a user waveform from the dropdown menu. Then click Draw Waveform.

Use the mouse or touchpad to draw. Once you’re done, click Save Waveform. Or you may click Cancel Draw Waveform to abandon the edits.

A screenshot of a computer

Description automatically generated

Figure Draw Waveform Example

# Importing Waveforms

Using the Wav2Bonk app, it is possible to import waveforms from .wav file. The example below shows the Wav2Bonk app positioned over the Desktop app. The contents of the clipboard have been pasted into the User Waveform text box and the Send Waveform button has been pressed. The imported waveform is shown in the User 0 display.

A screenshot of a computer

Description automatically generated

Figure Impot Waveform Example

# Terminal Mode

Terminal mode is accessed via the MicroUSB connector on the Arduino that is on the bottom of Thor’s Arsenal. It is always available and cannot be turned on or off.

Use a terminal emulation program like PuTTY that has VT-100 mode. Thor’s Arsenal is set to run at 115200 baud but will adapt to the PuTTY settings.

On the Mac use this procedure:

List ports: ls /dev/tty.\*

Invoke screen: screen <port> 15200

Quit screen: ctrl-a ctrl-\

#### Terminal Example Screen

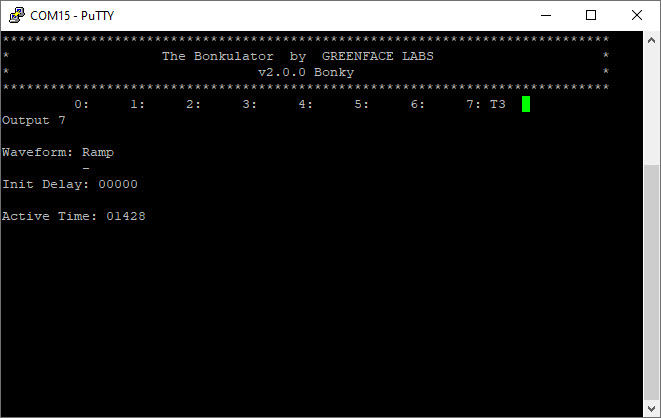


Figure 6 – Example Terminal Interface Screen

The Terminal screen essentially shows a copy of the user display with the addition of a product identification header that includes the software version and the unit’s name.

Below the product identification header is a status line that usually shows the state of the outputs. If an output is active, the status line will show which trigger activated that output.

Important! Only connect USB to a powered-up system. Otherwise, the 5V power from the USB will try to power the entire rack. And that’s not good. So, please disconnect the USB cable before you turn off the power.

You can get around this by using the BonkDaddy which disconnects the USB power when Thor’s Arsenal is turned off.

#### Thor’s Arsenal Remote Commands

|  |  |  |  |
| --- | --- | --- | --- |
| **Command** | **Front Panel** | **Terminal** | **Web** |
| Previous Parameter | ^ | [A | ^ |
| Next Parameter | v | [B | v |
| Next Digit | > | [C | > |
| Previous Digit | < | [D | < |
| Increment Digit | Adjust clockwise | u | Inc |
| Decrement Digit | Counter-clockwise | d | Dec |
| Settings Function | Press & Hold ADJ | \* | Settings |
| Disable Displays |  | Z |  |
| Restore Displays |  | z | Refresh Browser |
| Trigger Selected |  | ! | Use T0-T3 |
| Go To Function | Press Output | fx | Click Fxn Name |
| Select Parameter |  | px | Click on Param |
| Select Digit# |  | :xx |  |
| Enter Parameter |  | xxx… | Use Param Input |
| Enter Parameter (alt) |  | #xxx… | Use Param Input |
| Enter String |  | $xx… | Use Param Input |
| Idle Value |  | Vxxxx |  |
| Offset | Use Adj | Oxxxx | Use Offset slider |
| Scale | Use Adj | Sxxx | Use Scale slider |
| Randomness | Use Adj | Rxx | Use Random slider |
| Active Time | Use Adj | Axx | Use Active T slider |
| Group | N/A | Gxx | Use Group Controls |
| Group Active | N/A | gx | Use Grp Active Button |
| Set DAC |  | =xxxx |  |
| Quantization Off/On |  | Q0/1 |  |
| Print WiFi Status |  | W |  |
| Display Off/On |  | J0/1 |  |
| USB Direct Off/On |  | U0/1 |  |
| Select Trigger |  | tx |  |
| Clear Trigger |  | cx |  |
| Disable Selected Trig |  | T0x |  |
| Enable Selected Trig |  | T1x |  |
| Toggle Selected Trig |  | T2x |  |
| Disable All Triggers |  | T3 |  |
| Trigger All Triggers |  | T4 |  |
| Clear All Triggers |  | T5 |  |
| Trigger Report |  | T6 |  |
| Gen Macro Selected |  | M0 |  |
| Gen Macro Input Cal |  | M1 |  |
| Gen Macro Output Cal |  | M2 |  |
| Gen Macro User Wave |  | M1x |  |
| Dump Waveform |  | Dxy |  |
| Set Phase |  | Px |  |
| Set Default Value |  | I |  |
| Comment (for scripts) |  | / |  |

Notes:

* Under the hood, the Web interface uses the same command characters as the Terminal
* Commands with yellow background require the user to press Enter when using a terminal. **Not** **Required** when sending via Web interface.
* Parameters and digits may be navigated in Terminal mode using the arrow keys.
* Go To Function – x means to enter a single digit in the range of 0-8. The numbers 0 – 7 denote Outputs and 8 denotes Settings.
* Select Parameter – x means a single digit in the range of 0-n, where n = the number of parameters for a chosen function minus 1.
* Enter Parameter – xxxx means up to 4 digits depending on the parameter. The range also depends on the parameter.
* Enter Parameter Alt – Start with # followed by desired number. Terminate with carriage return.
* Enter String – Start with $ followed by desired string. Terminate with carriage return.
* Disable Displays - Z is used to speed up remote scripting.
* Restore Displays – z restores display that has been disabled by command or has been turned off due to inactivity.
* Idle Value – xxxx ranges from -5478 to +5478 or 0 to 10956 depending on output range setting. This number, divided by 1000, is the actual output voltage.
* Offset – xxxx ranges from -100 to +100
* Scale – xxxx ranges from -100 to +100
* Randomness – xx ranges from 0 to 99
* Set DAC – xxxx ranges from 0 – 4095. This sets the Idle Value of the selected output.
* Group – xx ranges from 0-255. Sets bitmap indicating which outputs are grouped.
* Group Active – x ranges from 0-1. Sets whether the group is active or not.
* Display On (J1) and Display Off (J0) just affect the OLED display. Turning off the OLED display noticeably speeds up response when using the terminal interface.
* Select Trigger – x denotes trigger number (0 – 3). This command affects other trigger commands.
* Clear Trigger – x denotes Output (0-7)
* Disable/Enable/Toggle Selected Trigger – x denotes Output (0-7)
* Gen Macro – These commands are used to capture the parameter settings for scripting. See below section for more details.
* Gen Macro User Wave - x denotes wave number.
* Dump Waveform - x denotes Output (0-7). This sends the waveform values to the terminal. If y is present, it dumps the reference waveform, which is the waveform before parameters are applied (scale, offset, etc.) Eg. D6 dumps wave 6. D61 dumps the reference wave 6.
* Set Phase – Sets the phase of the current output waveform. Units are in per cent. Eg. P25 changes a sine wave into a cosine wave.
* Comment – the / character causes the following text up until the carriage return to be ignored.

#### Grouping Outputs

When using the remote apps, it is possible to control several outputs with one command by **grouping** the outputs. The commands affected are: Offset, Scale and Randomness, Active Time, and Idle Value.

For example, the Offset slider can simultaneously control the *offset* parameter of several outputs. Just select which outputs you want by clicking on them.

#### Generating Parameter Macros

Software updates often contain new features that affect EEPROM allocation and thus will require initialization of the EEPROM when installed. This means that all parameters, user waveforms and calibration data will be lost and must be re-entered.

Parameter Macros are a way of minimizing the impact of this condition. Parameter Macros are created using Terminal Mode. When a Macro is created, it will be sent to the terminal screen where it can be copied and pasted into a text file for later use.

To run the Macro, merely paste it into the Terminal screen. Do this after a software update and whenever you need to set parameters to a known state. It is recommended that Calibration data macros only be run after a software update or if calibration data is lost or corrupted.

Macros are initiated by the ‘M’ character followed by a number that signifies the operation.

* M0 generates a macro that contains the parameters for the currently selected function.
* M1 generates a macro that contains the Input calibration data
* M2 generates a macro that contains the Output calibration data
* M10 – M17 generate macros that contain the User Waveform data. The selected waveform will equal the entered number – 10. I.e., M15 denotes User Waveform 5.

# Software Update

When a software update becomes available, it will be posted on the website as a zipped package. The package contains 6 files. Unzip these into an empty folder on your machine.

1. arduino-cli.exe
2. Bonkulator.ino.bin
3. Bonkulator.ino.elf
4. Release.txt
5. Update.bat
6. Blink.ino.elf.uf2 (for disaster recovery)

These files work for the Windows OS.

Follow these steps to update the code. This procedure ensures that your rack won’t be inadvertently powered by the USB +5 volt supply.

1. Note: If the update changes the EEPROM storage mapping, the unit will automatically initiate a Software Reset. Any stored settings will be lost, so make sure to save these settings before updating if they are important. See [Generating Parameter Macros](#_Generating_Parameter_Macros)
2. Turn off power to the rack and remove Thor’s Arsenal.
3. Leaving the power cable connected, place Thor’s Arsenal on a non-conductive surface. Make sure nothing touches Thor’s Arsenal that might cause a short.
4. Re-power the rack and plug micro-USB cable into Thor’s Arsenal.
5. Run update.bat. It will show available COM ports. Choose by entering the number of the port.
6. If it connects properly, you will see Windows announce a new drive has been mounted. The code is copied to this drive. If the update succeeds, Thor’s Arsenal will reboot as normal with the new software installed and the drive will be removed. You can see the new version number in the Settings function.
7. Note that an upload generates a file named: Bonkulator.ino.elf.uf2

## Disaster Recovery

When things go wrong, as they sometimes do, Thor’s Arsenal might become unresponsive. When this happens, the program must be reset. Usually recycling the power\* will fix the problem, but occasionally things have gone so bad that the code must be reloaded.

Usually, following the above update process will work. But occasionally things might have gone so bad that a hardware reset must be performed. There is a reset button on the Arduino 33 IoT. Pressing this twice quickly will force the Arduino into bootloader mode. You will see the yellow LED pulsate to indicate when the Arduino is in bootloader mode. The update process should work at this time.

If pressing the reset button doesn’t work, put a jumper onto J20 and press reset again. This should open a drive on your computer. When this happens, remove the jumper. Drag the file Blink.ino.elf.uf2 to the drive. Once loaded, you should see the Arduino blink some LEDs. The normal update process should now work.

Please contact Greenface Labs directly if this doesn’t work.

\*Make sure to unplug the USB cable before removing power.

**Detailed Explanation of Each Case**

1. **Case 'Z'**:
   * **Description**: Disables the display and terminal.
   * **Action**: Calls [ui.all\_off()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to turn off all UI elements.
2. **Case 'z'**:
   * **Description**: Restores the display.
   * **Action**: Calls [restore\_display()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to bring the display back to its previous state.
3. **Case '!'**:
   * **Description**: Activates a specific function.
   * **Action**: Calls [activate()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to perform the activation.
4. **Case 'u'**:
   * **Description**: Adjusts the group upwards.
   * **Action**: Calls [adjust\_group(1)](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to move the group up by one.
5. **Case 'd'**:
   * **Description**: Adjusts the group downwards.
   * **Action**: Calls [adjust\_group(-1)](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to move the group down by one.
6. **Case '\*'**:
   * **Description**: Selects the settings function.
   * **Action**: Calls [select\_fxn(SETTINGS\_FXN)](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to switch to the settings function.
7. **Case 'i'**:
   * **Description**: Inserts a space character.
   * **Action**: Calls [selected\_fxn->insert\_char(' ')](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to insert a space character at the current position.
8. **Case 127**:
   * **Description**: Removes a character (typically the delete key).
   * **Action**: Calls [selected\_fxn->remove\_char()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to remove the character at the current position.
9. **Case 'A'**:
   * **Description**: Moves up a parameter in escape mode.
   * **Condition**: Only executes if [esc\_mode](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is true.
   * **Action**: Calls [dec\_param\_num()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to decrement the parameter number.
10. **Case 'B'**:
    * **Description**: Moves down a parameter in escape mode.
    * **Condition**: Only executes if [esc\_mode](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is true.
    * **Action**: Calls [inc\_param\_num()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to increment the parameter number.
11. **Case 'C'**:
    * **Description**: Moves right in escape mode.
    * **Condition**: Only executes if [esc\_mode](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is true.
    * **Action**: Calls [inc\_dig\_num()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to increment the digit number.
12. **Case 'D'**:
    * **Description**: Moves left in escape mode.
    * **Condition**: Only executes if [esc\_mode](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is true.
    * **Action**: Calls [dec\_dig\_num()](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to decrement the digit number.

**Notes**

* The commented-out cases for '+' and '-' are not currently active but suggest functionality for selecting functions based on a remembered function index.
* The [esc\_mode](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) condition is used to determine if certain actions should be taken when specific keys are pressed.

## Commands that take an argument

**Detailed Explanation of Each Command**

Note: int\_param is the integer following the command. Example. Int\_param for ‘=403’ would be 403, and the command would be ‘=’

1. **Command '['**:
   * **Description**: Enters escape mode and sets the command to the next character in the input string.
   * **Action**: Sets [esc\_mode](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to true and updates [cmd](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to the second character of [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * **Example:** [A sends the up arrow, which goes to the previous parameter.
2. **Command '='**:
   * **Description**: Handles idle value or DAC output based on the second character of the input string.
   * **Action**: If the command is ‘=I’ (second character is 'I'), it prints the idle DAC value of the selected output. Otherwise, it sets the DAC output to [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * **Example**: =403 sets the current output DAC to 403.
3. **Command ':'**:
   * **Description**: Updates the digit number and prints parameters.
   * **Action**: Calls [put\_dig\_num](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and prints the parameters.
   * **Example**: :0 sets the digit number of the selected parameter to 0.
     1. Digit 0 is the MSD (most significant digit)
     2. The LSD (least significant digit) is the number of digits minus one.
     3. Setting the digit number to anything beyond the number of digits minus one just selects the LSD.
4. **Command 'f'**:
   * **Description**: Selects a function based on the integer parameter.
   * **Action**: Calls [select\_fxn](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * **Example**: f6 selects Output 6, and f8 selects Settings.
5. **Command 'p'**:
   * **Description**: Updates the parameter number and prints parameters.
   * **Action**: Calls [put\_param\_num](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and prints the parameters.
   * **Example**: p0 selects parameter 0. If in an Output function this would select Waveform.
6. **Command 'c'**:
   * **Description**: Clears a trigger for the specified output.
   * **Action**: Calls [clear\_trigger](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and the trigger number of [selected\_trigger](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * **Example**: c3 will stop the Output 3 waveform if active and return it to the Idle Value. It will not disable the trigger for that output.
7. **Command 't'**:
   * **Description**: Selects a trigger and optionally triggers it.
   * **Action**: Calls [select\_trigger](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [dig1](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html). If the third character of [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is '!', it triggers [selected\_trigger](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and restores the parameter number.
   * **Example**: t3! Selects Trigger 3 and activates that trigger.
8. **Command 'w'**:
   * **Description**: Receives a user waveform.
   * **Action**: Calls [recv\_user\_waveform](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * **Example**: w0,2080,2662,3503,… 128 values between 0 and 4095.
9. **Command 'I'**:
   * **Description**: Initializes the selected parameter and that parameter of every output that is in the current group. If the current function is not in a group, only the current selected parameter will be initialized.
   * **Action**: Calls [init\_group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * **Example**: I initializes the Waveform to Sine when the selected parameter is Waveform.
10. **Command 'T'**:
    * **Description**: Modifies triggers based on [dig1](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Action**: Depending on the value of [dig1](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), it updates, toggles, disables, or triggers all triggers, or displays a trigger report.
    * **Sub-Command 0:** Disable the selected trigger for the specified output.
      1. **Example:** T02 will disable the selected trigger for Output 2
      2. **Note:** use Command ‘t’ to select a trigger.
    * **Sub-Command 1:** Enable the trigger for the specified output.
      1. **Example:** T12 will enable the selected trigger for Output 2
      2. **Note:** use Command ‘t’ to select a trigger.
    * **Sub-Command 2:** Toggle the trigger for the specified output.
      1. **Example:** T22 will toggle the selected trigger for Output 2
      2. **Note:** use Command ‘t’ to select a trigger.
    * **Sub-Command 3:** Disable all triggers for all outputs.
      1. **Example:** T3 will disable all triggers. This will leave the Trigger buttons in the disabled state. Re-enable a trigger for an output to re-enable the trigger.
    * **Sub-Command 4:** Trigger all triggers.
      1. **Example:** T4 will trigger all enabled triggers for all outputs.
    * **Sub-Command 5:** Clear all triggers.
      1. **Example:** T5 will stop all outputs that have active waveforms and return those outputs to the Idle Value.
    * **Sub-Command 6:** Issue a Trigger Report.
      1. **Example**: T6 issues a Trigger Report
      2. **Note**: This command only works in Terminal Remote mode.
11. **Command 'U'**:
    * **Description**: Sets USB direct settings.
    * **Action**: Calls [settings\_put\_usb\_direct](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: U0 turns USB Direct OFF. U1 turns it ON.
    * **Note**: Selecting a Port from the app will automatically turn USB Direct ON. There is no reason to use U1 in normal operation.
    * **Note**: If you should connect via Terminal while the module is still in USB Direct mode, you can turn it OFF and return to Terminal Mode by using command: U0
12. **Command 'J'**:
    * **Description**: Turns the front panel display on or off based on [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Action**: If [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is 0, it turns the display off. If [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is 1, it turns the display on.
    * **Example**: J0 turns off the display and J1 turns it back on.
13. **Command 'M'**:
    * **Description**: Generates macros to the Terminal screen based on [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Action**: Depending on the value of [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), it generates different macros for input calibration, output calibration, user waveform parameters, or general parameters.
      1. **Sub-Command 1:** Generate Input calibration parameters macro.
      2. **Sub-Command 2:** Generate Output calibration parameters macro.
      3. **Sub-Commands 10 to 17:** Generate User waveform parameters macro.
      4. **Default:** Generate parameters macro for the currently selected function. (Output, Settings, etc.)
    * **Example**: M1 generates a macro that sets the input calibration values. This is good for remembering these values in case of a software update or Reset that would erase them.
    * **Example**: M14 would generate a macro for User Waveform 4
    * **Note**: The Default behavior occurs when M is anything other than the above sub-commands.
    * **Example:** M by itself or M paired with any integer above 17 while in an Output function would generate a macro for the current Output parameters.
14. **Command 'P'**:
    * **Description**: Updates the output phase if in an Output function.
    * **Action**: Calls [param\_put](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param % 100](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_PHASE](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html), then calls [update\_phase](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: P50 shifts the selected waveform by 180 degrees. This command only works with the standard waveforms.
15. **Command 'D'**:
    * **Description**: Prints the values of one of the standard waveforms to the Terminal.
    * **Action**: Calls [dump\_waveform](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [dig1](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and a boolean indicating if [dig2](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is greater than 0.
    * **Example**: D1 prints a Waveform Report for Output 1’s selected waveform.
    * **Example**: D11 prints the *reference* waveform for Output 1's selected waveform. This is the waveform before it is modified by the parameters.
    * **Note**: Command D only works in Terminal mode.
16. **Command 'G'**:
    * **Description**: Sets the group. The parameter is interpreted as a binary number whose bit positions indicate which outputs are to be grouped.
    * **Action**: Updates the [group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) variable with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: G3 groups Output 0 and Output 1
    * **Example**: G192 groups Output 6 and Output 7
    * **Note**: This works in Terminal mode. But, at present, there is no indication of grouped outputs like there is in the app.
    * **Note**: If no outputs are grouped, adjustments to the group merely adjust the selected parameter of the currently selected function.
17. **Command 'g'**:
    * **Description**: Activates or deactivates the group.
    * **Action**: Sets [group\_active](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to true if [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is greater than 0, otherwise sets it to false.
    * **Example**: g0 disables the group. Disabling the group unlinks the outputs assigned to the group without unassigning the outputs. This is a convenience when you want to adjust a parameter of a single output without affecting other outputs in the group. Issuing g1 will re-enable the group so that all outputs in the group are linked again.
18. **Command 'O'**:
    * **Description**: Sets the output offset for the group. The value ranges from -10.00V to +10.00V.
    * **Action**: Calls [put\_group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_OFFSET](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: O500 sets the Offset to +5V. O-123 sets the Offset to -1.23V.
    * **Note**: This command only affects the waveform. It does not affect the Idle Value.
19. **Command 'S'**:
    * **Description**: Sets the output scale for the group. The value ranges from -100 to +100. The units are per cent.
    * **Action**: Calls [put\_group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_SCALE](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: S-50 sets the scale of the output waveform to 50% full-scale. The negative sign inverts the signal around its middle point.
20. **Command 'A'**:
    * **Description**: Sets the active time for the group.
    * **Action**: Calls [put\_group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_ACTIVE\_TIME](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: A3423 sets the Active Time for the group to be 3423ms.
21. **Command 's'**:
    * **Description**: Reserved.
    * **Action**: Does nothing.
22. **Command 'R'**:
    * **Description**: Sets the randomness for the group. Randomness ranges from 0 to 99% of the selected signal level.
    * **Action**: Calls [put\_group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_RANDOMNESS](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: R43 sets the Randomness to 43%
23. **Command 'V'**:
    * **Description**: Sets the idle value for the group. The Idle Value is the voltage at the output when it isn’t active (sending a waveform). It ranges from plus full scale to minus full scale.
    * **Action**: Calls [put\_group](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_IDLE\_VALUE](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: V500 sets the Idle Value to 5.00V.
24. **Command 'Q'**:
    * **Description**: Sets the quantize parameter for the group.
    * **Action**: Calls [put\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [OUTPUT\_QUANTIZE](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: Q1 turns quantize on. Q0 turns it off.
25. **Command 'W'** (if [WIFI\_INSTALLED](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)):
    * **Description**: Clears the terminal and prints WiFi status. This command does not take a parameter. It might in the future.
    * **Action**: Calls [clearTerminal](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [printWifiStatus](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: If WiFi is installed (the device is a Bonkulator), W will print a WiFi status report.
26. **Command '$'**:
    * **Description**: Sets a string variable.
    * **Action**: Calls [put\_string\_var](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with the URL-decoded substring of [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
    * **Example**: $Mandy sets the string variable to Mandy.
    * **Note**: This only works on string variables. An example is the device Name in Settings.
    * **Note**: This is the preferred method for entering string variables in Terminal mode.
27. **Command '/'**:
    * **Description**: Comment character, does nothing.
    * **Action**: No action taken.
28. **Command '#'**:
    * **Description**: Updates a parameter and applies it to the waveform and graphs it if in output function.
    * **Action**: Calls [put\_param\_w\_offset](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html). If in output function, it applies parameters to the waveform and graphs it based on the parameter number.
    * **Example**: #780 updates the current parameter to 780.
    * **Note**: This works for numeric variables and also for select variables. For select variables just enter the index as the integer argument.
    * **Example**: #0 sets a select variable to the first menu item.
    * **Note**: The # is optional. Just entering a number will work. This command is reserved for apps that require a command character when building their instruction strings.
29. **Command ‘=’**:
    * **Description**: The = command is typically used for development but might be useful in scripting. It has two modes. I and O. I mode prints information to the app. Or it prints to the terminal if the app isn’t active. O mode takes a parameter and outputs it to the module. Currently the = command only deals with the DAC (digital-to-analog-converter) which determines the output voltage.
    * **I Mode**:
      1. **Description**: Reads the DAC setting given the output number.
      2. **Example**: =I3 reads the DAC setting for Output 3.
      3. **Example**: = by itself reads the DAC setting of the selected output.
    * **O Mode**:
      + **Description**: Sets the DAC given the output number.
      + **Example**: =O35643 sets the DAC for Output 3 to 5643.
      + **Example**: =5643 by itself sets the DAC for the selected output to 5643.

* **Command '=':**
  + Description: If the second character is 'I', print the idle value of the selected output.
  + Otherwise, output the DAC value.
* **Command 'M':**
* **Command '#':**
  + Set the parameter with an offset using the provided integer parameter.
  + If the selected function is an output function:
    - **Sub-Command OUTPUT\_SCALE:** Apply parameters to the waveform and graph the waveform.
    - **Sub-Command OUTPUT\_OFFSET:** Apply parameters to the waveform and graph the waveform.
    - **Sub-Command OUTPUT\_RANDOMNESS:** Apply parameters to the waveform and graph the waveform.
    - **Sub-Command OUTPUT\_QUANTIZE:** Apply parameters to the waveform and graph the waveform.

1. **Check if**[**in\_str**](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**is "=" or if the second character of**[**in\_str**](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html)**is 'I':**
   * If true:
     + Determine [temp\_int](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html):
       - If the second character of [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is 'I', convert the substring starting from the third character to an integer.
       - Otherwise, use [selected\_output\_temp](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
     + Constrain [temp\_int](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to be within the range [0, NUM\_OUTPUTS - 1].
     + Create a string [temp\_string](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) that includes the DAC value for the output specified by [temp\_int](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
     + If USB direct mode is enabled:
       - Format [temp\_string](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) as HTML and include the output number.
       - Send the formatted string to USB.
     + Otherwise:
       - Print [temp\_string](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to the terminal with the output number.
2. **If the first condition is false:**
   * Check if the second character of [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) is 'O':
     + If true:
       - Convert the third character of [in\_str](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to an integer and assign it to [selected\_output\_temp](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
       - Convert the substring starting from the fourth character to an integer and assign it to [temp\_int](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
     + Otherwise:
       - Use [int\_param](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) as [temp\_int](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html).
   * Call [dac\_out](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) with [selected\_output\_temp](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) and [temp\_int](vscode-file://vscode-app/c:/Users/shann/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) to output the DAC value.