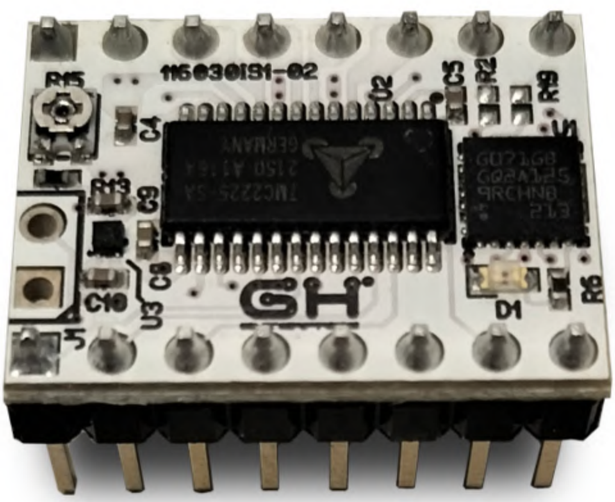
ENGLISH

User Manual



NAME: SMART SHAPER
MODEL: 116030-01

INDEX:

1. Introduction

Input shaping algorithms are designed to suppress the vibrations characterized by a particular frequency and dumping ratio.
Dumping ratio is difficult to estimate and it’s usually assumed to be 10%.
Frequency should be estimated experimentally for both axes.
The measure of the axes frequencies requires the following steps:

- Print of a reference model (see 5.1.)
- Measurement of distance between ringing peaks (see 5.2.)
- Frequency calculation by given formula (see 5.2.)

Next it will be explained how to configure the boards with the frequencies previously found and the preferred Input Shaping algorithm (see 6.1.)
Finally It will possible to print again the reference model but with Input Shaping Enabled (see 6.6.) in order to verify the correct values of frequencies and the performance of the algorithm.

2. Requirements

The following instructions are intended for the following setup:

- Cartesian and Core XY/YX printers.
- X/Y (or A/B for Core XY/YX printer) axes driven by Smart Shaper Boards.
- X/Y axes resolution 80 steps/mm (for axes with different resolutions see 6.5.)
- Max Step Rate 48Khz.
- Cartesian printer **max printing speed is 600 mm/s with 80 step/mm.**
- Core XY/YX printer **max printing speed is 300 mm/s with 80 step/mm.**
- TMC2225 configured with **16 microsteps** resolution and interpolation.
- TMC2225 configuration can be done via UART or bootstrap pins.

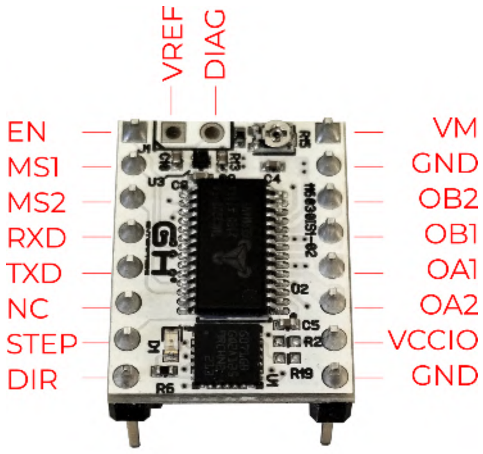
3. Hardware

3.1. TMC 2225 Driver Specifications

- TMC2225-SA stepper motor controller & driver
- Supply voltage 5.5-36V
- Continuous I_{phase} = $1.2A_{RMS}$
- Quiet operation with StealthChop
- Configuration and extended diagnostic via UART
- Control via Step&Dir interface
- Board width 0.6”. board height 0.8”
- 2x8 pin 0.1” head rows for pins/connectors

3.2. Pinout

	Power Supply
GND	Ground
VM	Motor Supply Voltage 5.5V-36V
VCCIO	Logic Supply Voltage 3V-5V
	Motor Outputs
OA2	Motor Coil 1
OA1	Motor Coil 1
OB1	Motor Coil 2
OB2	Motor Coil 2
	Control Inputs
STEP	STEP input (internal pull-down resistor)
DIR	DIR input (internal pull-down resistor)
	TMC2225
EN	Enable Motor Outputs: GND=on, VIO=off
MS1	Microsteps resolution configuration (internal pull-down resistors) MS2, MS1: 00: ¼, 01: ⅛, 10: 1/16, 11: 1/32
MS2	-
RXD	UART RX, Directly connected to the PDN
TXD	UART TX, Connected to the PDN via a 1K resistor on board
DIAG	Diagnostic output. Hi level upon driver error. Reset by ENN=high.
VREF	Analog Reference Voltage



4. Configuration and Installation SmartShaper

4.1. SmartShaper Configuration modes

SmartShaper board is based on TMC2225 driver. TMC2225 can be configured via bootstrap pins or via UART protocol.

According to your motion board and firmware you should select the one that is more suitable to your setup.

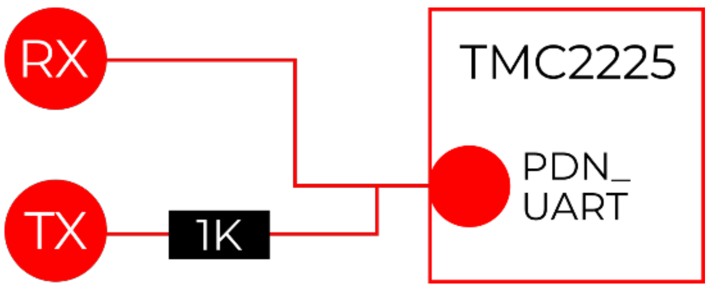
The parameters that need to be configured are:

- micro-stepping resolution (usually 16 microsteps)
- motor current

If you are able to use UART mode the parameters are programmed via UART by the firmware otherwise that should be configured using MS1/MS2 bootstrap pins and VRef potentiometer.

4.2. SERIAL UART mode

TMC2225 uart pin is mapped to SmartShaper board as described in the following picture



Each motion board has it custom way to enable and configure UART pins. Please refer to the user manual of your motion board. Usually instruction are referred to TMC2208. TMC2225 UART is equivalent to TMC2208.

Then, you need to enable UART mode in your firmware.

If you are using Marlin this is the configuration to do in Configuration.h

```
Marlin > C Configuration.h > Y_DRIVER_TYPE
 * These settings allow Marlin to tune stepper driver timing and enable advanc
 * stepper drivers that support them. You may also override timing options in
 *
 * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and TMC2209/TMC2209_STAN
 *
 * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
 *          TB6560, TB6600, TMC2100,
 *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
 *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
 *          TMC26X,  TMC26X_STANDALONE,  TMC2660, TMC2660_STANDALONE,
 *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
 * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', '
 */
#define X_DRIVER_TYPE  TMC2208
#define Y_DRIVER_TYPE  TMC2208
//#define Z_DRIVER_TYPE  A4988
//#define X2_DRIVER_TYPE A4988
//#define Y2_DRIVER_TYPE A4988
//#define Z2_DRIVER_TYPE A4988
//#define Z3_DRIVER_TYPE A4988
//#define Z4_DRIVER_TYPE A4988
//#define I_DRIVER_TYPE  A4988
```

The other parameters can be set as described in the following table

Marlin Parameters	
Micro-stepping	Default value is 16. Relative define entries in Configuration_adv.h are: • X_MICROSTEPS • Y_MICROSTEPS
Motor Current (mA RMS)	Relative define entries in Configuration_adv.h are: • X_CURRENT • Y_CURRENT or runtime via GCODE with command Gcode M906

4.3. STANDALONE mode

4.3.1. Micro-stepping configuration

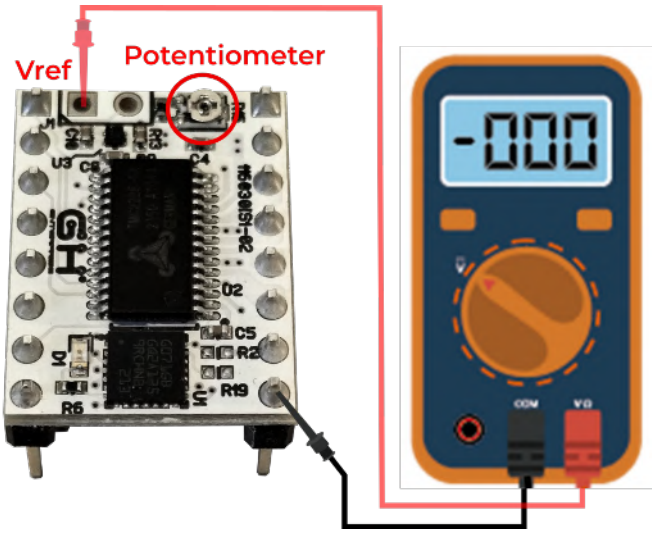
Configuration Microsteps with bootstraps pins is different between TMC 2208/2209 and TMC 2225

TMC 2208/2209 MS1/MS2 Configuration		
MS2	MS1	MICROSTEPS
GND	VCC_IO	1/2
VCC_IO	GND	1/4
GND	GND	1/8
VCC_IO	VCC_IO	1/16

TMC 2225 MS1/MS2 Configuration		
MS2	MS1	MICROSTEPS
GND	GND	1/4
GND	VCC_IO	1/8
VCC_IO	GND	1/16
VCC_IO	VCC_IO	1/32

MS2 and MS1 pins, if not connected, have an internal pull-down that keep them to GND.
If your board does not have jumpers to force MS1 or MS2 to GND, you can cut or extract the relative pin to set GND level.

4.3.2. Motor Current Regulation



Driver motor current should match the current tolerated from your step motors. The current can be adjusted the value of VREF using the onboard potentiometer.
VREF [V]= (Irms * 2.5V) / 1.77A = Irms * 1.41 = Imax [A]
The value of VREF in Volt corresponds to the value of Peak Current Imax in Ampere.
In the following picture are indicated how to measure VREF. TheSmartShaper board should be installed on the motion board and the VM power supply should be present.

Pay attention to avoid any short circuits when using screwdriver on potentiometer.

4.4. Dissipation recommendation

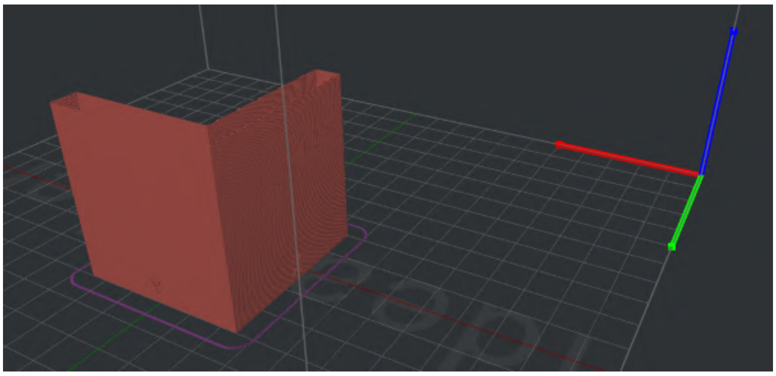
TMC2225 if not well cooled goes in to thermal protection state and stops to work. It’s highly recommended to:

- Install the Heat Sinks provided with installation set
- Use a 3D motion control board equipped with a cooling fan system.

5. Resonance Frequency Measurement

5.1. Print Calibration Model

For Cartesian printers the calibration model prepared for frequency estimation consists in a simple L Model printed with the walls aligned along X and Y axes.



The printed part has one wall marked with X label and another one marked with Y label. The X Label is on the wall aligned along the Y axes and the Y Label is on the wall aligned along the X axis. The reason is that the ringing due to Y acceleration/deceleration are visible on X axis and vice-versa.
The walls are printed at **100 mm/s** (**this speed value is fundamental for the correct frequency calculation using the formula in 5.2.**).

Acceleration is linearly increased from 500 mm/s^2 (Bottom) to 18000 mm/s^2 (Top) in order to increase the vibrations.
TMC2225 is configured in current mode to support high acceleration through G-code sequence.
X and Y Coordinates will not exceed 145 mm.
Material should be PLA (using a red color improves the measurement of ringings) and nozzle diameter 0.4 mm.
Print the model. It's possible that at some height the printer start to loose steps because acceleration is too high to be tolerated by mechanics. If it happens just stop the printer.
Now extract the printed part from the printer and analyze it.

For Core XY/YX printers the print calibration model is different from that for Cartesian printers, because the model is rotated of 45 degrees respect to the cartesian plane and the label to reference axes are called A and B, but it has the same characteristics as the model for cartesians printers.

The calibration models can be downloaded from the web site.
https://gh-enterprise.com/en/3d-printer-controller/

5.2. Frequency Measurement

To find the ringing frequency it should be measured the distance between two peaks of the oscillations.

The easiest way is:

- put the part under a light beam parallel to the surface

- mark each visible oscillation peaks
- measure with a caliper the total distance D in mm between the first mark and last mark

- calculate the oscillation frequency (Hz) using the formula

$$f=\frac{V(N-1)}{D}$$

where

- V =100 mm/s
- N is the number of marks

Example of a Y axis oscillation frequency measurement.

We marked 7 oscillation peaks. The measured distance was about 19.8 so the frequency was:

$$f=\frac{100(7-1)}{19.8}\approx 31\,Hz$$

6. Boards Configuration

6.1. G-Code Generator

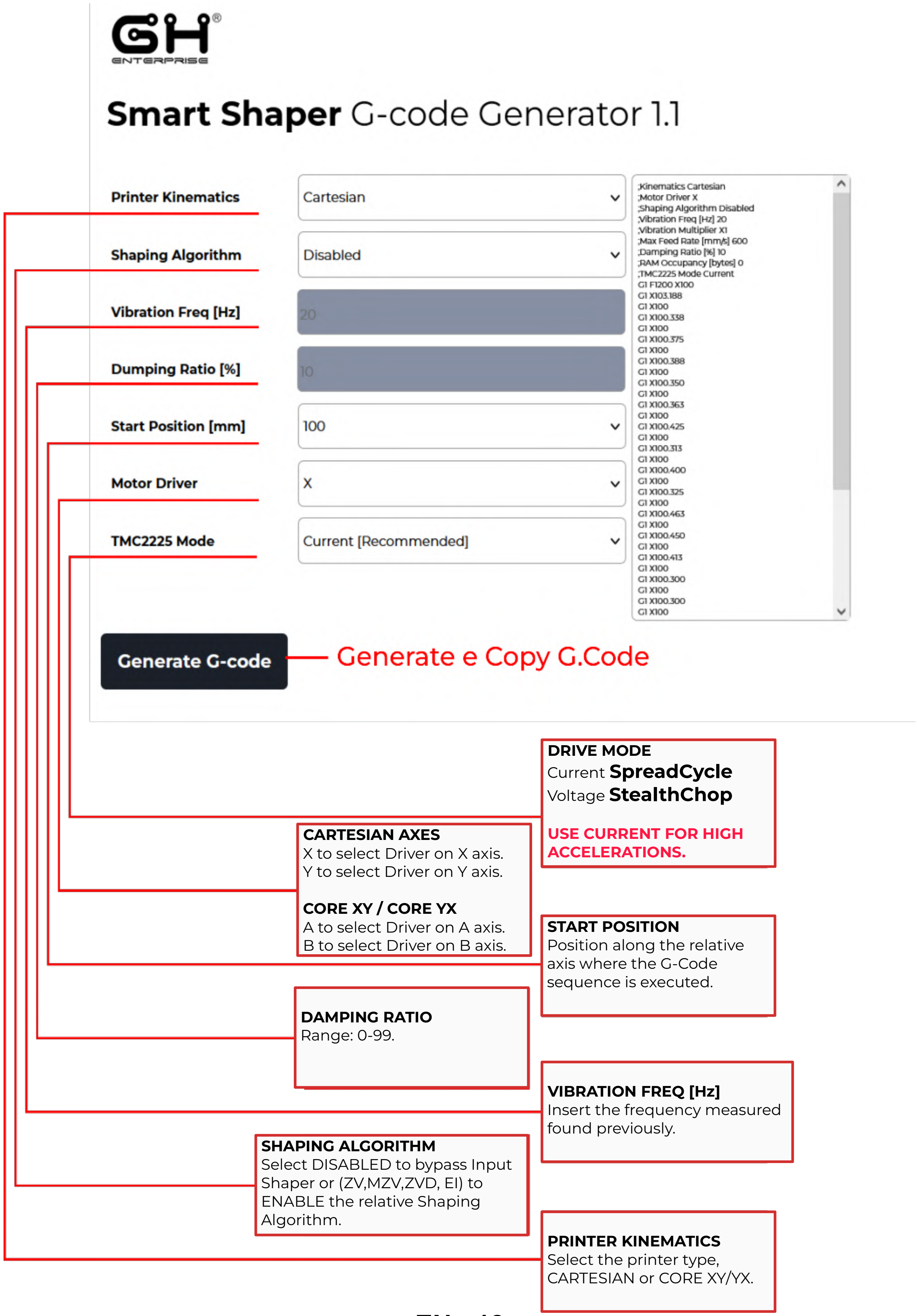
The G-Code generator is Online version on the GH Enterprise web site
(https://gh-enterprise.com/smart-shaper-generator-1.1/smartshaper_gcode.html)

6.2. G-Code configuration sequence

Each Smart Shaper board can be programmed via a specific G-Code sequence generated by **Smart Shaper G-Code Generator 1.1** (see 6.3. for description).

An example of such sequences can be found in the file **CartesianSmartShaperCalibration.gcode** (lines 32-129). In this case they have been inserted two sequences, one for X driver and one for the Y driver, that **DISABLE** the Input Shaping filter **to detect ringings.**

6.3. Smart Shaper G-Code Generator 1.1



6.4. Sequence Generator

The **Smart Shaper G-Code Generator 1.1** is the tool to generate the G-Code configuration sequence.

Once the correct settings have been entered, a Gcode will be created in the right column to be copied and pasted into the gcode to be printed.

6.5. Printer with different axes resolutions

For the programming with G-Code to be correct, the resolution of the axes must be 80 step/mm.

If the resolutions is different, for example 100 step/mm for the X axis, proceed as follows:

- First of all insert the **M92 X80** G-Code command before the programming part made with Smart Shaper G-Code Generator 1.1.
- After the programming G-Code insert the **M92 X100** G-Code command to set the correct resolution of the axis.

The maximum print speed will be slower with higher resolution.
MAX STEP RATE(Hz) / RESOLUTION (step/mm) = MAX SPEED (mm/s)
example 48000 / 80 = 600

6.6. Print with Input Shaping

Generate a G-Code configuration sequence for each axes according to the measured frequencies.

Due to the high accelerations of Calibration Model,
the TMC2225 SHOULD BE CONFIGURED IN CURRENT MODE (SPREADCYCLE).

Replace the two original sequences in the **SmartShaperCalibration.gcode** (lines 32-129) with the new generated ones. Print the modified G-Code.

Just after homing procedure the two configuration sequence will be executed.

6.7. Status Led Information

A red led is placed on each Smart Shaper board. The blinking frequency provide some information about the board.

Just after the startup the Led will blink as many times related to the version of the installed firmware (Ex. Firmware is 1.3 will blink 3 times).

After startup the Led will indicate the current status of the board as shown in the following table.

Frequency (Hz)	Meaning
0.5	Input Shaper Off
1	Input Shaper On
2	Configuration Failed (wrong parameters)
4	Input Step Rate too high

Led can be used to verify that the configuration has been correctly received and processed by the board.

If you experience “Input Step Rate too high” alarm it means that the step rate has exceed the hardware limit that Input Shaping micro controller can handle.

This could be caused by following reasons:

- Printer Feedrate has been set to a high value. Remember that maximum supported speed is 48000 / RESOLUTION (step/mm). Considering that typical resolutions on X/Y are 80 steps/mm, corresponding to 600 mm/s, it’s difficult that this condition could happen.
- Motion Board firmware generates steps not always respecting the requested feedrate. In that case, we found the limiting the Stepper Rate in the firmware solves the issue. In Marlin firmware, this could be achieved, defining MAXIMUM_STEPPER_RATE value equal to 64000 in Configuration_adv.h.