| 3D PRINTING AND DESIGN REFERENCE DOCUMENT | |
|---|---|
| **Document Title:** | Kinetic Sand Table |
| **Document No.:** | 1734601619 |
| **Author(s):** | jattie |
| **Contributor(s):** | |

**REVISION HISTORY**

| Revision | Details of Modification(s) | Reason for modification | Date | By |
|---|---|---|---|---|
| 0 | Draft release | Document Essential References and Resource for Building a Kinetic Sandtable | 2024/12/19 09:46 | jattie |

# Kinetic Sand Table Design and Build

The objectives for the project is to 3D print as many of the parts as possible and to build low budget linear stages to construct the basic system and build it up from there.

- Electronics/Controllers
  - GRBL Controller
  - CNC Shield
  - Stepper Drivers for shield
- Linear Hardware
  - Stepper motors
  - Linear Stages
  - GT2 drive belts
  - GT2 Idlers
  - GT2 Stepper Attachments
  - Belt Clamp
- Playlist [1)]

# Electronics

## GRBL Controller

After some extensive research I discovered grblHAL[2)]. grblHAL is the updated version of GRBL. GRBL is an open-source firmware that converts G-code commands into motion control signals for CNC (Computer Numerical Control) machines. It's widely used for controlling CNC routers, mills, lathes, laser cutters, and other automated machinery.

grblHAL essentially makes it possible to use a wide range of low cost 32 bit microcontrollers. Some popular grblHAL supported microcontrollers are :

- RP2040: Raspberry Pi Pico
- ESP32: Popular for IoT projects
- STM32: Various models like STM32F1xx, STM32F3xx, STM32F4xx, STM32F7xx, and STM32H7xx
- LPC176x: Used in many embedded systems
- SAM3X8E: Found in Arduino Due

- Teensy 4.x: High-performance microcontrollers
- NXP iMXRT1062: Used in Teensy 4.x boards

This allows for a wide range of options to avail of to build a very low cost grbl interface. There are handy web based tools to select the controller of choice and build the firmware code for you.[3]. The alternative route is to build the code using VSCode. The full tutorial is here.



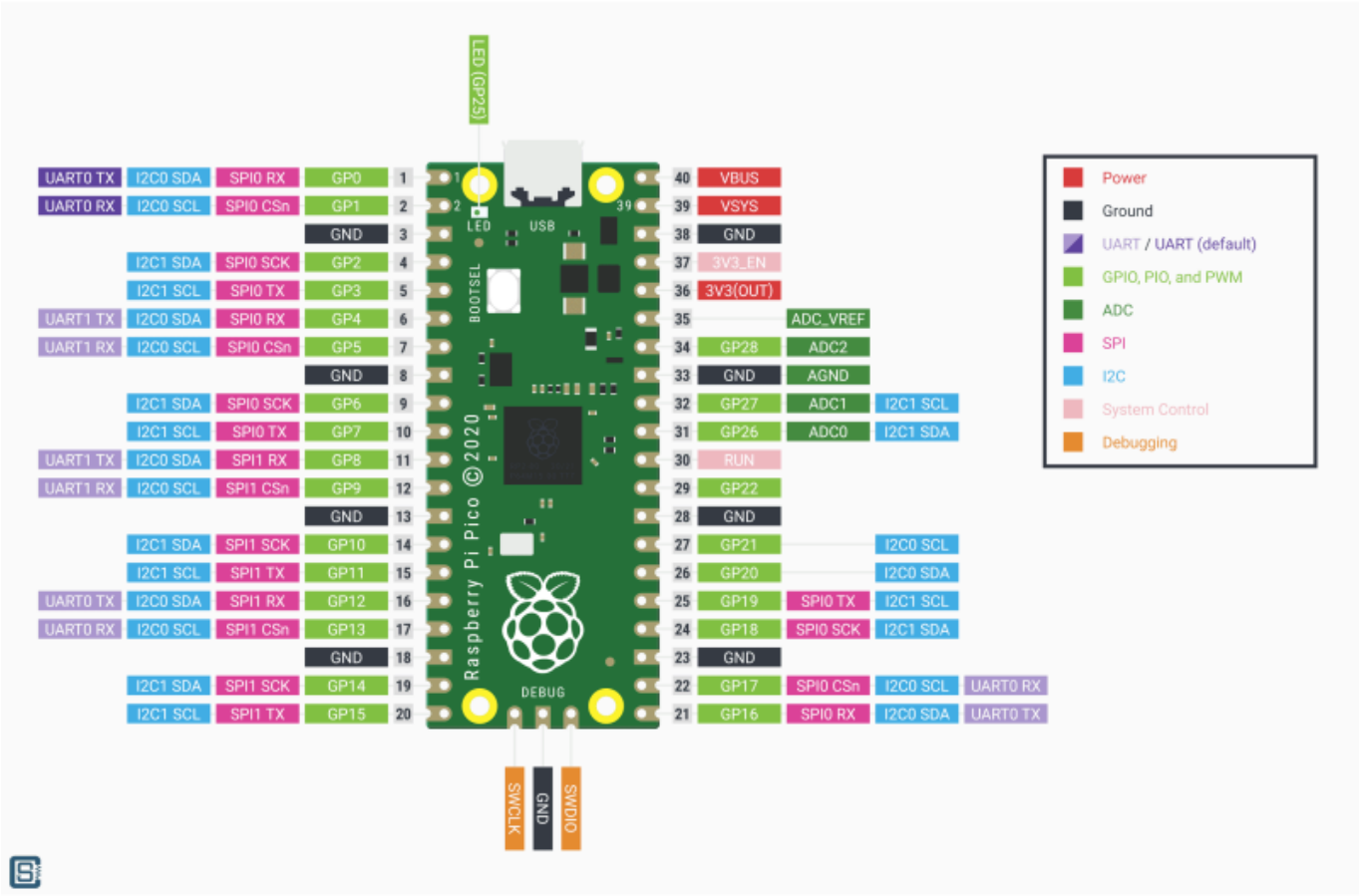
The basic steps are:

- Create firmware [4]
- Power off the Pico 2040 by unplugging the USB, hold in BOOTSEL and plug it back in. Upload the firmware.
- Connect to the unit using IOSender XL [5]

That's it, you now have a GRBL controller.

## Connecting Steppers to the GRBL controller

The pinouts for a Pico 2040 are as follow:




Following the grblHAL software mapping for the firmware [6] we deduce the following map to actual pinouts:

| GRBL Function | Pico GP Pin |
|---|---|
| Step Output X | 2 |
| Step Output Y | 3 |
| Step Output Z | 4 |

| GRBL Function | Pico GP Pin |
|---|---|
| Direction Output X | 5 |
| Direction Output Y | 6 |
| Direction Output Z | 7 |
| Steppers Enable | 8 |
| Limit X | 9 |
| Limit Y | 10 |
| Limit Z | 11 |

## Stepper Drivers

There are a few projects for Pico specific shields, however they are costly to get hold of and the Arduino community has very low cost and mature products that is compatible, or at least I believe it is and will explore this and test it for this purpose.

I will explore the CNC Shield that is discussed in detail here [7)]

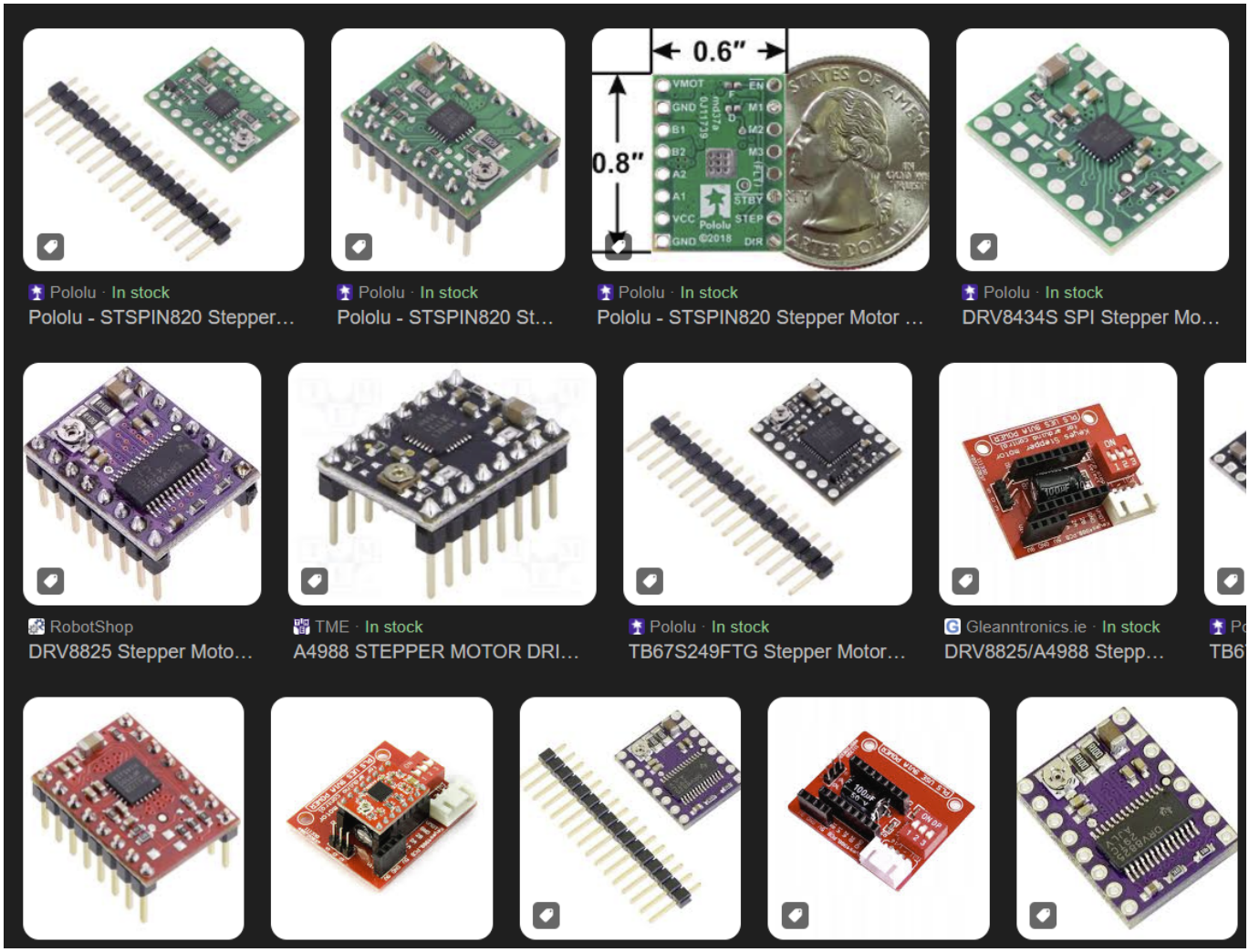
As an alternative we explored TB6600. [8)9)10)] These units are very low cost and available on Amazon.

The different models of these types of units are tabled below for comparison.




| Feature/Driver | TB6600 | DM556 | DM556T | DM542 | ERP60 | DM860H | DM860S | DM860T |
|---|---|---|---|---|---|---|---|---|
| **Input Voltage** | 9-42V | 20-50V | 20-50V | 20-50V | 20-50V | 20-50V | 20-50V | 20-50V |
| **Output Current** | 0.5-4A | 0.5-5.6A | 1.8-5.6A | 1.8-5.6A | 1.8-5.6A | 1.8-5.6A | 1.8-5.6A | 1.8-5.6A |
| **Microsteps** | 1, 2/A, 2/B, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 | 1, 2, 4, 8, 16, 32 |
| **Control Interface** | Digital | Digital | Digital | Digital | Digital | Digital | Digital | Digital |
| **Protection Features** | Overcurrent, Overheat | Overcurrent, Overheat | Overcurrent, Overheat | Overcurrent, Overheat | Overcurrent, Overheat | Overcurrent, Overheat | Overcurrent, Overheat | Overcurrent, Overheat |
| **Applications** | General use, CNC machines | General use, CNC machines | General use, CNC machines | General use, CNC machines | General use, CNC machines | General use, CNC machines | General use, CNC machines | General use, CNC machines |

The other option is smaller packaged driver like these:




| Feature/Driver | STSPIN820 | DRV8834 | A4988 | MP6500 | A5984 | TB67S249 | DRV8434 | TMCM-1260 |
|---|---|---|---|---|---|---|---|---|
| **Operating Voltage** | 7-45V | 2.5-10.8V | 8-35V | 8-40V | 8-40V | 8-40V | 8-40V | 8-40V |
| **Max Output Current** | 1.5 Arms | 1.5 A | 2 A | 2.5 A | 2.5 A | 2.5 A | 2.5 A | 2.5 A |
| **Microstepping** | Up to 1/256 | Up to 1/32 | Up to 1/16 | Up to 1/16 | Up to 1/16 | Up to 1/16 | Up to 1/16 | Up to 1/16 |
| **Protection Features** | Overcurrent, Overtemperature, Short-circuit, Undervoltage lockout, Thermal shutdown | Overcurrent, Short-circuit, Undervoltage lockout, Overtemperature, Low-power sleep mode | Overcurrent, Short-circuit, Thermal shutdown | Overcurrent, Short-circuit, Thermal shutdown | Overcurrent, Short-circuit, Thermal shutdown | Overcurrent, Short-circuit, Thermal shutdown | Overcurrent, Short-circuit, Thermal shutdown | Overcurrent, Short-circuit, Thermal shutdown |
| **Package Type** | QFN 4×4 mm | HTSSOP/VQFN 24-pin | DIP-16 | DIP-16 | DIP-16 | DIP-16 | DIP-16 | DIP-16 |
| **Applications** | 3D printers, Medical equipment, Industrial printers, Robotics | Toys, Printers, Cameras, Robotics | 3D printers, CNC machines, Robotics | 3D printers, CNC machines, Robotics | 3D printers, CNC machines, Robotics | 3D printers, CNC machines, Robotics | 3D printers, CNC machines, Robotics | 3D printers, CNC machines, Robotics |

The TMC2208 and TMC2209 are known for their silent operation due to their StealthChop technology.

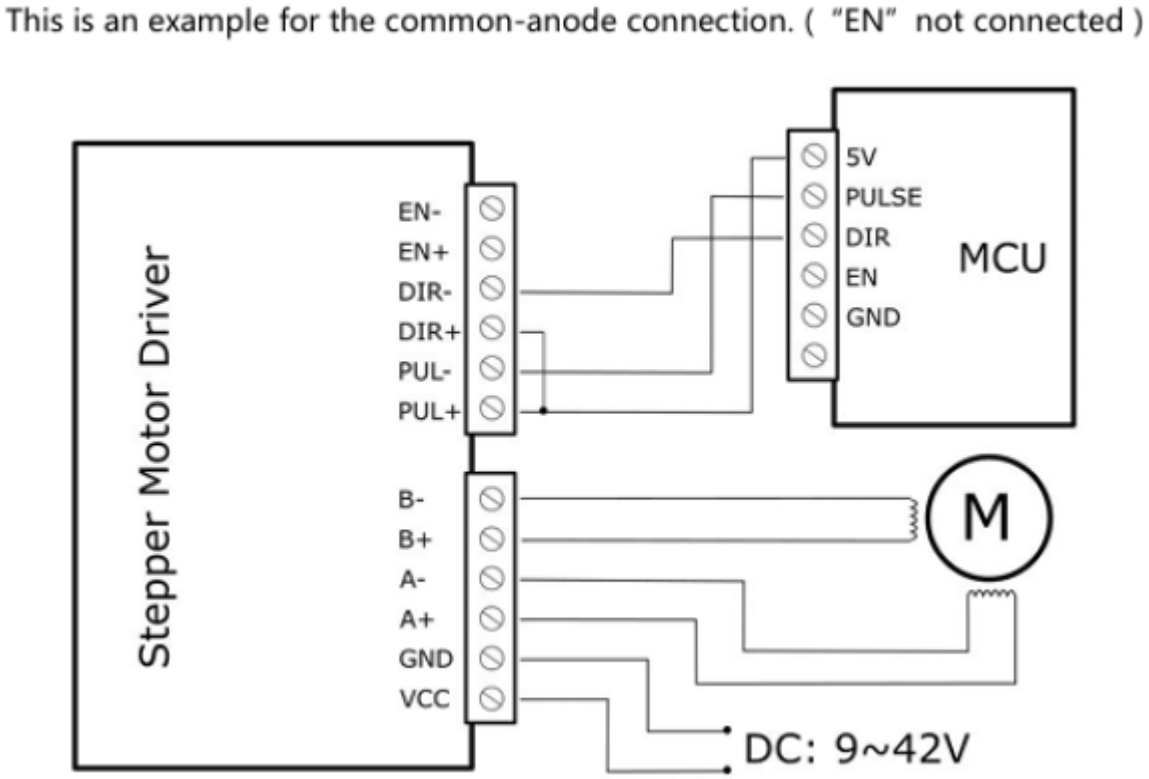


The final choice of driver came down to cost and effort to implement. At the time of writing TB6600 units were available from amazon for €11,10. With optical insulated inputs[11)] and the cost, it's a no brainer choice for me.

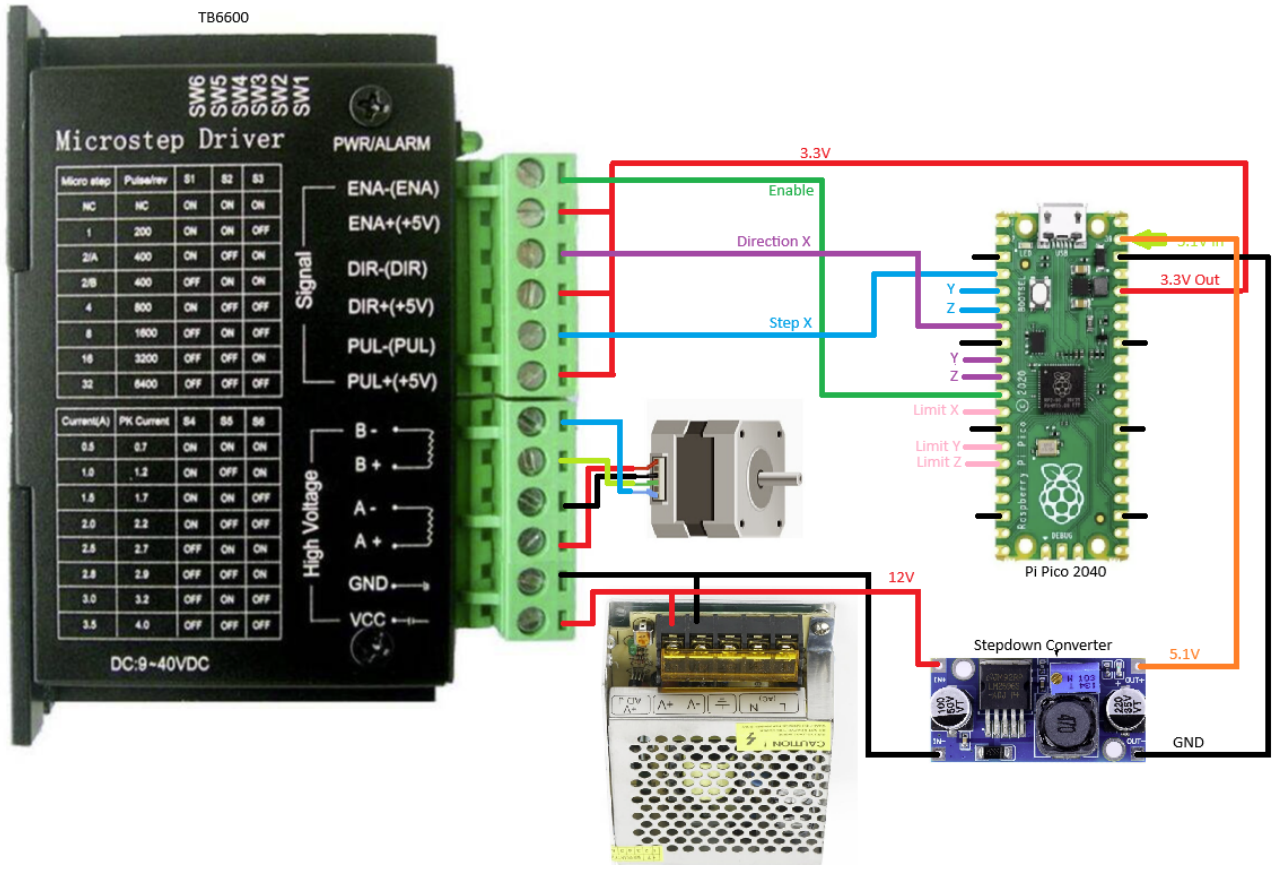
## Wiring

The pico datasheet[12] section 2.1 confirms use of the 3.3V output for signal purposes of up to 300mA. The TB6600 datasheet[13] confirms 15mA currents required to drive the optocouplers.

The proposed microcontroller wiring from the same datasheet proposes the following:




So in keeping with the proposed vendor Microcontroller wiring we then add the enable lines back in and it should look like this.




This schematic show the wiring for the X axis, the two Y axis connections for direction and step should be used instead.

# Linear Hardware

1)

https://github.com/texx00/sandypi

2)

https://github.com/grblHAL/core/blob/master/README.md

3) , 4)

http://svn.io-engineering.com:8080/?driver=RP2040

5)

https://github.com/terjeio/ioSender/releases/

6)

https://github.com/grblHAL/RP2040/blob/master/boards/generic_map.h

7)

https://all3dp.com/2/arduino-cnc-shield/

8)

https://www.amazon.co.uk/gp/product/B07SBZ9SM5/ref=ox_sc_act_title_1?smid=A3G751PYK8M98N&psc=1

9) , 13)

https://www.makerguides.com/wp-content/uploads/2019/10/TB6600-Manual.pdf

10)

https://www.watelectronics.com/tb6600-stepper-motor-driver-module/

11)

https://components101.com/sites/default/files/component_datasheet/TB6600-Datasheet.pdf

12)

https://datasheets.raspberrypi.com/pico/pico-datasheet.pdf

From:
http://3dfaq.net/ - **3D Printing Wiki**

Permanent link:
**http://3dfaq.net/04_projects/02_kinetic_sand_table?rev=1734725924**

Last update: **2024/12/20 20:18**