2021

TMC2209 UART for the Anycubic Chiron

4 WIRE TMC2209 UART CONNECTION FOR THE TRIGORILLA

NICK WELLS

Contents

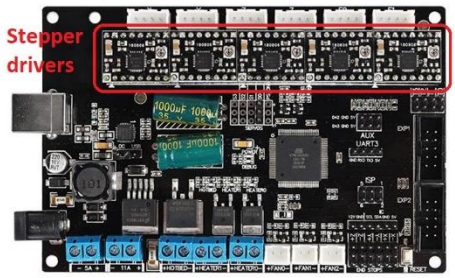
Overview

NOTE: This configuration requires 2209 drivers, as it uses MS1, MS2 slave addressing.
It will not work with 2208 drivers.

What is a TMC2209 and why should I use it?

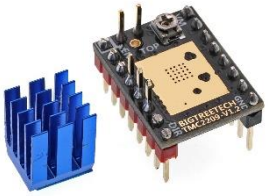
Your printer uses 5 plug-in boards that control the stepper motors called stepper drivers. The 2209 is an advanced stepper driver made by Trinamic. [TMC2209-LA - Trinamic]

It does the same job as the stock A4988 drivers in the printer, but it runs much smoother and quieter.



Trigorilla with A4988 drivers


A4988 Driver


2209 Driver

What is UART mode and why should I use it?

The 2209 driver has different operating modes, that are pre-set by the manufacturer or by soldered links on the driver board. You can use a 2209 in standalone mode (no UART connection) but you are stuck with the preconfigured mode and you need to set the driver current using the little trim pot on the board.
To get the best from the drivers, you need to allow the firmware to talk to the driver, using the UART port. This requires an extra wire connection to the Trigorilla board, and some small changes to the firmware. Once configured it is much easier to set things.

For example, to set the X driver current to 1 Amp you can send a GCode command `M906 X1000`
It's that easy!

How much quieter is the 2209?

It does depend on the movement being made, but as an example, when you probe the bed on the Chiron, the Z-Axis is so quiet you can hear the probe microswitch clicking!

How do I do this?

This guide will take you through the things you need to do to install and configure TMC 2209 drivers on an AnyCubic Chiron.
You will need to do the following things:

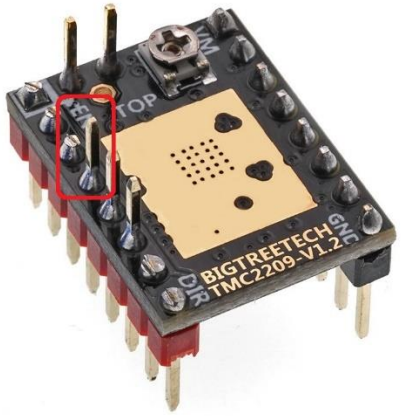
1. Solder pin headers into the servo ports on the Trigorilla board.
2. Create two cables cable using header jumper wires and two 1kΩ resistors.
3. Cut selected pins off the 2209 drivers
4. Modify settings and build a new version of Marlin firmware.
5. Send commands to the printer using a terminal program.

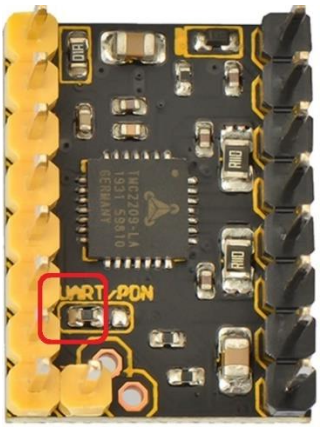
Parts List

You will need the following parts.

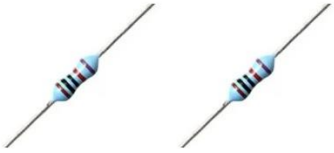
Five, TMC2209 drivers

Make sure the drivers you purchase have long pins at the top for the UART connection and they are set for UART mode.





Two, 1kΩ 1/4W resistors



Four, 3 pin headers

Six, jumper wires

A Soldering iron & solder

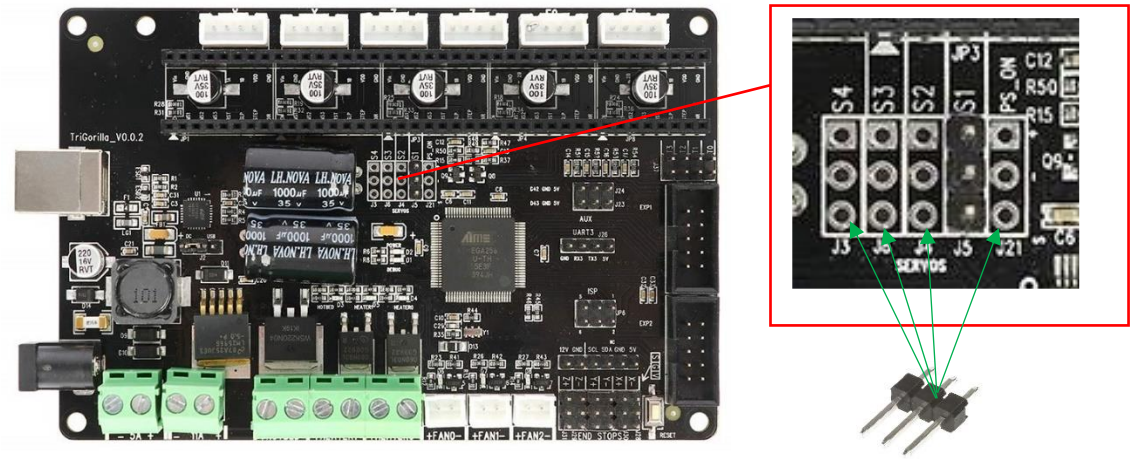
Heat shrink sleeve

Wire Cutters

Construction

Solder in Pin Headers

Solder the pin headers into all the ports (12 pins).
You can just populate the bottom row it you like, we will only be using the bottom row of pins.

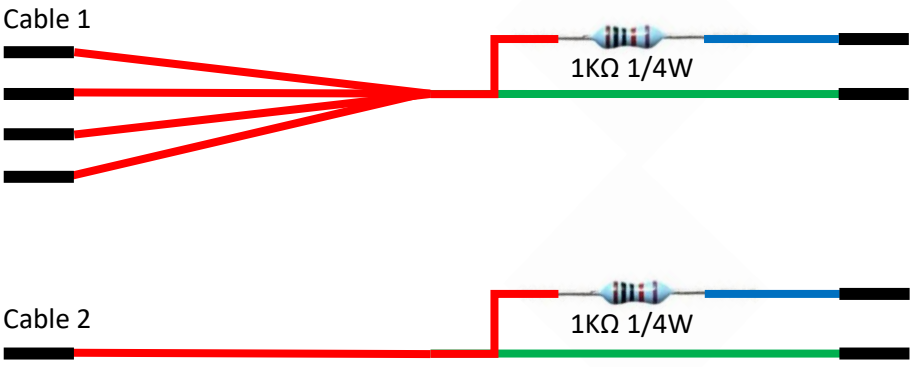


Make the Cables

Make these two cables and cover the resistors and solder joins with heat shrink insulation.
Use green and blue as shown that other colours don't matter. It will make the wiring easier!

Tip: Just cut the green and blue wires in half and use one half for each cable.

There should be a female header socket on each end of the cable.



Cut the pins on the drivers

Next, we need to cut pins on the stepper drivers to set the access address.
The TMC 2209 driver address is set using the MS1 and MS2 pins. On the Trigorilla board, pins MS1 & MS2 are connected to 5v. Internally each pin on the 2209 chip has a pulldown resistor, so we can set the address by cutting pins on the driver board.

A cut pin will be 0v, a pin connected to the board will be 5v.

Address	MS1	MS2
0	0v	0v
1	0v	5v
2	5v	0v
3	5v	0v

Some drivers have PDN/UART positions reversed so please check before cutting pins!

Note: Once the pins have been cut, each driver must be fitted in a specific location on the Trigorilla board, so we will mark each one with a stripe.

Now cut the pins on the stepper drivers as shown in the table and mark the side of each stepper using a sharpie, so you know where to put them! We need to cut off the lower UART pin as we will be connecting a cable to the top.

X
Mark 1 stripe

Cut MS1, MS2 & UART
Cable 1 address 0

Y
Mark 2 stripes

Cut MS2 & UART
Cable 1 address 1

Z1
Mark 3 stripes

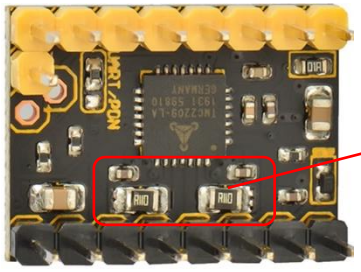
Cut MS1 & UART
Cable 1 address 2

E
Mark 4 Stripes

Cut UART only
Cable 1 address 3

Z2
Mark 5 Stripes

Cut MS1, MS2 & UART
Cable 2 address 0

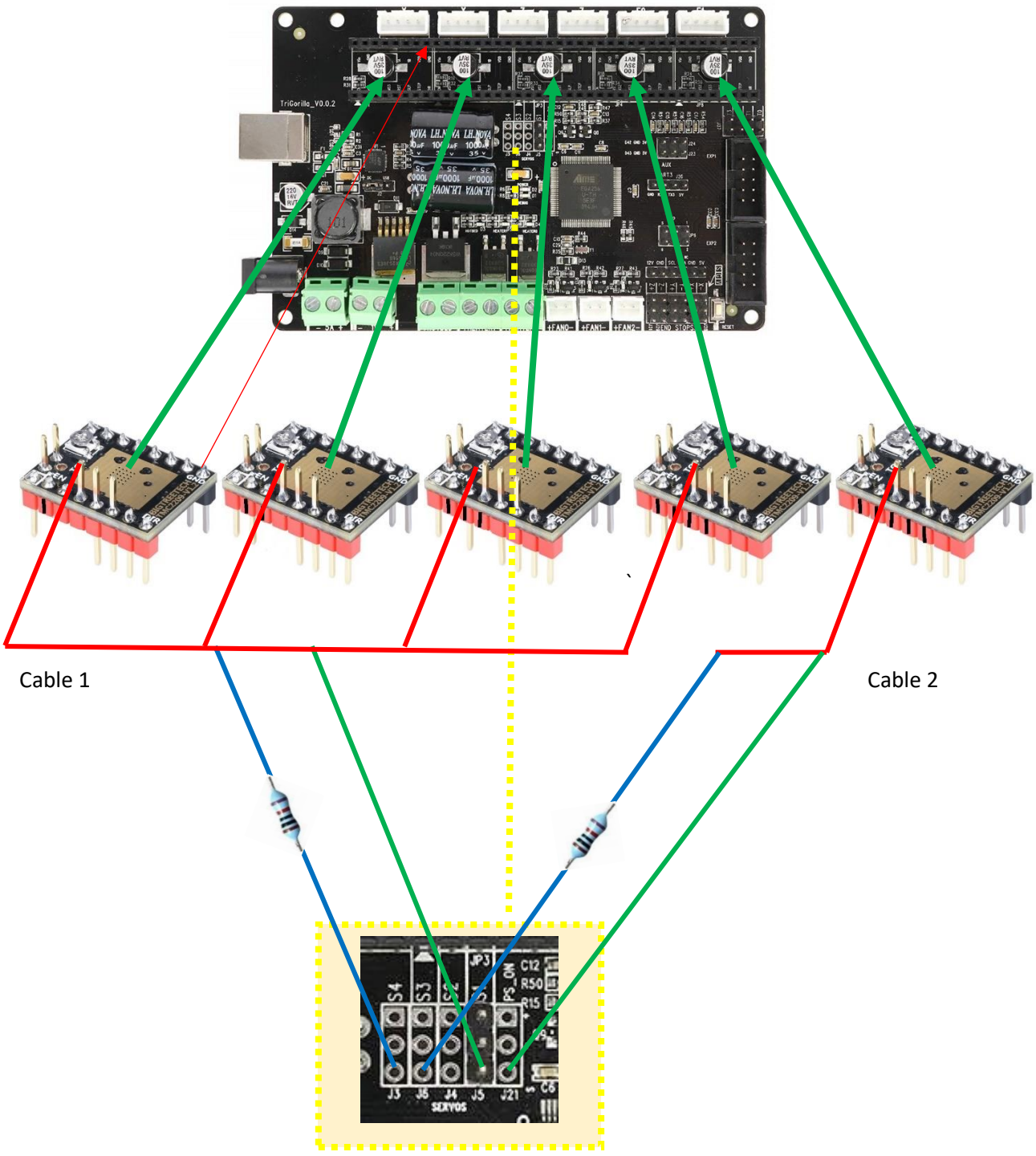


Make a note of the Rsense resistor value.
You will need to set this in firmware.
On this driver the value is R110 = 110mΩ = 0.11Ω
it may be R68 = 68mΩ = 0.068Ω
or something else.

Installation

Install the drivers as shown below, make sure you fit them correct way round and carefully fit the heatsinks.

Now connect the cables you made earlier as shown below.



Well done, that's the building part done! Next comes the firmware…

Firmware Configuration

You need to edit the two Marlin configurations files to configure the new drivers.
The changes made will do the following:

1) Configure the UART Ports
2) Assign the TMC2209 drivers to each stepper motor
3) Set the stepper direction
4) Set the default driver current
5) Enable TMC debug commands.

Start with the example configuration files for the Chiron for the version of Marlin you are using.

Configure the UART ports

Open configuration.h and look for the MOTHERBOARD definition lines.

```
140  // Choose the name from boards.h that matches your setup
141  #ifndef MOTHERBOARD
142    #define MOTHERBOARD BOARD_TRIGORILLA_14
143  #endif
144
145  // Anycubic Chiron custom pin defs for the Trigorilla board fastio_1280.h
146  #define OUTAGECON_PIN   58  // p93 F4
147  #define X_MAX_PIN       43
```

Copy and paste these lines below the X_MAX_PIN definition.

```
// TMC pin Definitions for full UART support

#define TMC_PORT1_RX   11  // Must be interrupt capable
#define TMC_PORT1_TX   4

#define TMC_PORT2_RX   12  // Must be interrupt capable
#define TMC_PORT2_TX   5

#define  X_SLAVE_ADDRESS 0
#define  Y_SLAVE_ADDRESS 1
#define  Z_SLAVE_ADDRESS 2
#define E0_SLAVE_ADDRESS 3

#define Z2_SLAVE_ADDRESS 0

#define X_SERIAL_RX_PIN TMC_PORT1_RX
#define X_SERIAL_TX_PIN TMC_PORT1_TX

#define Y_SERIAL_RX_PIN TMC_PORT1_RX
#define Y_SERIAL_TX_PIN TMC_PORT1_TX

#define Z_SERIAL_RX_PIN TMC_PORT1_RX
#define Z_SERIAL_TX_PIN TMC_PORT1_TX

#define E0_SERIAL_RX_PIN TMC_PORT1_RX
#define E0_SERIAL_TX_PIN TMC_PORT1_TX

#define Z2_SERIAL_RX_PIN TMC_PORT2_RX
#define Z2_SERIAL_TX_PIN TMC_PORT2_TX
```

These changes tell Marlin which ports to use to communicate with the stepper drivers.

We have connected X,Y,Z & E to Servo0 (RX Arduino pin 11) and Servo 3 (TX Arduino pin 4) using addresses 0,1,2,&3 and Z2 to PS_ON(RX Arduino pin 12) and Servo2(TX Arduino pin 5)

Assign TMC2209 drivers to each stepper

Locate the Stepper section in Configuration.h and set as shown below.

```
887   * Stepper Drivers
888   *
889   * These settings allow Marlin to tune stepper driver timing and enable advanced options for
890   * stepper drivers that support them. You may also override timing options in Configuration_adv.h.
891   *
892   * A4988 is assumed for unspecified drivers.
893   *
894   * Use TMC2208/TMC2208_STANDALONE for TMC2225 drivers and TMC2209/TMC2209_STANDALONE for TMC2226 drivers.
895   *
896   * Options: A4988, A5984, DRV8825, LV8729, L6470, L6474, POWERSTEP01,
897   *          TB6560, TB6600, TMC2100,
898   *          TMC2130, TMC2130_STANDALONE, TMC2160, TMC2160_STANDALONE,
899   *          TMC2208, TMC2208_STANDALONE, TMC2209, TMC2209_STANDALONE,
900   *          TMC26X,  TMC26X_STANDALONE,  TMC2660, TMC2660_STANDALONE,
901   *          TMC5130, TMC5130_STANDALONE, TMC5160, TMC5160_STANDALONE
902   * :['A4988', 'A5984', 'DRV8825', 'LV8729', 'L6470', 'L6474', 'POWERSTEP01', 'TB6560', 'TB6600', 'TMC2100',
903   */
904  #define X_DRIVER_TYPE  TMC2209
905  #define Y_DRIVER_TYPE  TMC2209
906  #define Z_DRIVER_TYPE  TMC2209
907  //#define X2_DRIVER_TYPE A4988
908  //#define Y2_DRIVER_TYPE A4988
909  #define Z2_DRIVER_TYPE TMC2209
910  //#define Z3_DRIVER_TYPE A4988
911  //#define Z4_DRIVER_TYPE A4988
912  //#define I_DRIVER_TYPE  A4988
913  //#define J_DRIVER_TYPE  A4988
914  //#define K_DRIVER_TYPE  A4988
915  #define E0_DRIVER_TYPE TMC2209
916  //#define E1_DRIVER_TYPE A4988
```

Set the stepper direction

You will also need to invert the stepper direction, so look for the machine section and set the stepper direction as shown.

```
1361  // @section machine
1362
1363  // Invert the stepper direction. Change (or reverse the motor connector) if an axis goes the wrong way.
1364  #define INVERT_X_DIR false
1365  #define INVERT_Y_DIR false
1366  #define INVERT_Z_DIR false
1367  //#define INVERT_I_DIR false
1368  //#define INVERT_J_DIR false
1369  //#define INVERT_K_DIR false
1370
1371  // @section extruder
1372
1373  // For direct drive extruder v9 set to true, for geared extruder set to false.
1374  #define INVERT_E0_DIR true
```

Set the Default driver Current

Open Configuration_adv.h and locate `@section tmc_smart`

Set the X driver current to 1 Amp = 1000mA
Set the X-RSENSE value to the value you wrote down earlier.
The driver in this document has R110 on the resistor. (110mΩ enter this as 0.11) R68 would be entered as 0.068

```
2582  #if AXIS_IS_TMC(X)
2583    #define X_CURRENT       1000        // (mA) RMS current. Multiply by 1.414 for peak current.
2584    #define X_CURRENT_HOME  X_CURRENT  // (mA) RMS current for sensorless homing
2585    #define X_MICROSTEPS     16        // 0..256
2586    #define X_RSENSE          0.11
2587    #define X_CHAIN_POS      -1        // -1..0: Not chained. 1: MCU MOSI connected. 2: Next in chain, ...
2588    //#define X_INTERPOLATE  true      // Enable to override 'INTERPOLATE' for the X axis
2589  #endif
```

Set Y current to 1.2Amps and set Y_RSENSE

```
2600  #if AXIS_IS_TMC(Y)
2601    #define Y_CURRENT       1200
2602    #define Y_CURRENT_HOME  Y_CURRENT
2603    #define Y_MICROSTEPS     16
2604    #define Y_RSENSE          0.11
2605    #define Y_CHAIN_POS      -1
2606    //#define Y_INTERPOLATE  true
2607  #endif
```

Set Z & Z2 current to 1 Amp and set Z & Z2 RSENSE values

```
2618  #if AXIS_IS_TMC(Z)
2619    #define Z_CURRENT       1000
2620    #define Z_CURRENT_HOME  Z_CURRENT
2621    #define Z_MICROSTEPS     16
2622    #define Z_RSENSE          0.11
2623    #define Z_CHAIN_POS      -1
2624    //#define Z_INTERPOLATE  true
2625  #endif
2626
2627  #if AXIS_IS_TMC(Z2)
2628    #define Z2_CURRENT      1000
2629    #define Z2_CURRENT_HOME Z2_CURRENT
2630    #define Z2_MICROSTEPS    Z_MICROSTEPS
2631    #define Z2_RSENSE         0.11
2632    #define Z2_CHAIN_POS     -1
2633    //#define Z2_INTERPOLATE true
2634  #endif
```

Finally set the E0 current to 1.2A and also RSENSE

```
2681  #if AXIS_IS_TMC(E0)
2682    #define E0_CURRENT      1200
2683    #define E0_MICROSTEPS    16
2684    #define E0_RSENSE         0.11
2685    #define E0_CHAIN_POS     -1
2686    //#define E0_INTERPOLATE true
2687  #endif
```

These are the stock current values set by Anycubic which do run the motors a bit hot.
Once this is all done you can easily experiment with lower values.

Set the chopper voltage to 24v (only used in SpreadCycle mode)

```
2843      * Define your own with:
2844      * { <off_time[1..15]>, <hysteresis_end[-3..12]>, hysteresis_start[1..8] }
2845      */
2846    #define CHOPPER_TIMING CHOPPER_DEFAULT_24V        // All axes (override below)
```

Enable TMC Debug features

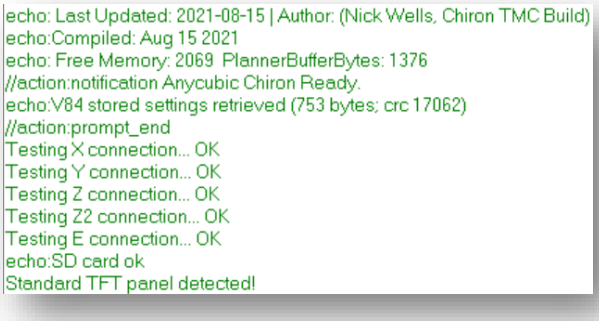
```
2974    /**
2975     * Enable M122 debugging command for TMC stepper drivers.
2976     * M122 S0/1 will enable continuous reporting.
2977     */
2978    #define TMC_DEBUG
2979
```

That's it!
Now build and flash the firmware and do a factory reset (M502 then M500) to initialise the eeprom.

Testing

Switching on for the first time

Connect a terminal to the printer and power up. If everything is working you should see something like this:



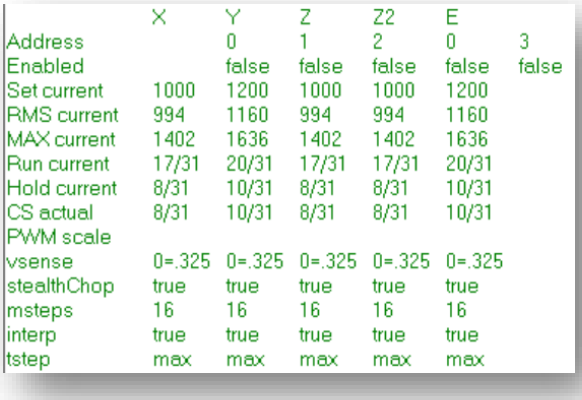
When the printer starts it will test each of the drivers and respond with OK.

If you see this then you can proceed with testing.

If not, skip to the troubleshooting section.

Check the driver currents

From the terminal Send M122 and press enter.



	X	Y	Z	Z2	E	
Address		0	1	2	0	3
Enabled		false	false	false	false	false
Set current	1000	1200	1000	1000	1200	
RMS current	994	1160	994	994	1160	
MAX current	1402	1636	1402	1402	1636	
Run current	17/31	20/31	17/31	17/31	20/31	
Hold current	8/31	10/31	8/31	8/31	10/31	
CS actual	8/31	10/31	8/31	8/31	10/31	
PWM scale						
vsense	0=.325	0=.325	0=.325	0=.325	0=.325	
stealthChop	true	true	true	true	true	
msteps	16	16	16	16	16	
interp	true	true	true	true	true	
tstep	max	max	max	max	max	

You should get a message showing the TMC settings.

If you see this then you can continue with motor testing.

Motor Testing

Manually move the head and bed to a middle position.
Now using the front panel axis menu, press the +10 button for X, Y and Z and check the printer moves in the correct direction.

If it moves correctly, you can home the machine and start printing!

If not, have a look at the troubleshooting section.

Settings

G-Code Commands for TMC drivers

All G-Code information for Marlin can be found here. Gcode | Marlin Firmware (marlinfw.org)

These commands relate to TMC drivers

- M122 TMC Debugging
- M569 Set TMC stepping mode
- M906 TMC Motor Current

To set the stepper currents use:

M906 X1000	Sets X axis to 1.0A	
M906 Y1200	Sets Y axis to 1.2A	
M906 T0 E1200	Sets Extruder to 1.2A	
M906 Z1000	Sets Z1 to 1.0A	(Left side motor)
M906 I1 E1000	Sets Z2 to 1.0A	(Right side motor)

To set stepper modes use:
S0 = SpreadCycle, S1 = StealthChop

M569 X S0	X axis
M569 Y S0	Y axis
M569 T0 E S0	Extruder
M569 Z S0	Z axis (left)
M569 I1 Z S0	Z axis (right)

Remember to save your changes, or they will be lost on a reboot.

M500	Saves changes.

Troubleshooting

Here's a few troubleshooting tips if you cant get things working.

TMC Connection Error

1) Check you have placed the stepper drivers in the correct location.
2) Check you have wired the UART pins correctly
3) Check you have cut the correct pins.
4) Check the driver boards are set to UART mode (refer to your supplier)
5) Make sure you are using TMC2209 drivers.

There are different versions of the Trigorilla board with different pin assignments for the Servo ports
If there is no communication, you can try changing the port definition lines on page 7.

```
#define TMC_PORT1_RX   11  // Must be interrupt capable
#define TMC_PORT1_TX   4
#define TMC_PORT2_RX   12  // Must be interrupt capable
#define TMC_PORT2_TX   5
```

	Trigorilla	Trigorilla 1.1
TMC_PORT1_RX	11	5
TMC_PORT1_TX	4	11
TMC_PORT2_RX	12	12
TMC_PORT2_TX	5	4

Layer Skip

If you experience skipped steps form movements or extrusion, try increasing the motor current in 0.1A increments.

The maximum ratings for the Chiron Stepper motors are:
X,Y & Z Axis: 1.5A
Extruder: 1.2A

TMC2208 drivers needed to run in SpreadCycle mode on the extruder but StealthChop is usually ok on the 2209.