



# Titan Marlin Configuration

Set up your Marlin Firmware to support your new Titan

Written By: Gabe S.



## Step 1 — Download Marlin



- First things first: you're going to need a copy of Marlin.
  - If you are upgrading an existing 3D printer to use a Titan, you should try to get a copy of your current firmware from your printer's manufacturer.
  - If you're building a new printer, or simply want to upgrade to the latest version of Marlin, download it at <http://marlinfw.org/meta/download/>
- ⚠ If you download a fresh version of Marlin you'll have to configure more settings than the ones mentioned in this guide so that it will work well with your printer.

## Step 2 — Download Arduino



- Almost all printers use Arduino IDE to upload fresh firmware, so download it at <https://www.arduino.cc/en/Main/Software>

## Step 3 — Open Marlin in Arduino



- Unzip Marlin from the zip file you downloaded and put the resulting folder anywhere on your computer for safe keeping.
- Inside this folder, navigate to the Marlin sub-folder, and open the Marlin.ino file. This should open every file in Marlin.
- Find the Configuration.h file

## Step 4 — Extruder Direction

```

Marlin - Configuration.h | Arduino 1.8.13
File Edit Sketch Tools Help
Main | Configuration.h | Conditionals_LCD.h | Conditionals_post.h | Configuration.h | Configuration_Mk3 | G29_Mesh_Validation_Tool.cpp
#define INVERT_E0_DIR false
// Enable this option for Tachiba stepper drivers
// #define CONFIG_STEPPER2_TOSHIBA

// #section extruder
// For direct drive extruder v9 set to true, for geared extruder set to false.
#define INVERT_E0_DIR false
#define INVERT_E1_DIR false
#define INVERT_E2_DIR false
#define INVERT_E3_DIR false
#define INVERT_E4_DIR false

// #section homing
// #define X_HOME_OFFSET 4 // (in mm) Minimal a height before homing (Z0) for 1 clearance above the bed, clamps, ...
// // Be sure you have this distance over your Z_MAX_POS in case.

// Direction of endstops when homing: 1=MAX, -1=MIN
// (-1,-1)
#define X_HOME_DIR -1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1

// #section machine
// Travel limits after homing (units are in mm)
#define X_MIN_POS 0
#define Y_MIN_POS 0
#define Z_MIN_POS 0
#define X_MAX_POS 200
#define Y_MAX_POS 200
#define Z_MAX_POS 200

// If enabled, axes won't move below MIN_POS in response to movement commands.
#define MIN_SOFTWARE_ENDSTOPS
// If enabled, axes won't move above MAX_POS in response to movement commands.
#define MAX_SOFTWARE_ENDSTOPS

/*
 * Filament Runout Sensor
 * A mechanical or opto endstop is used to check for the presence of filament.
 */

```

- If your old extruder was ungeared, you'll notice that your new Titan extrudes backwards!
- Flip the following line in the Configuration.h file from true to false, or vice versa:  
**INVERT\_E0\_DIR**

## Step 5 — Extruder Steps-per-mm 1

$$\text{E-Steps-per-MM} = \frac{\text{Motor Steps} * \text{Micro-stepping} * \text{Gear Ratio}}{(\text{Hobb Diameter} * \pi)}$$

- Standard motor steps / rev = 400
- Standard micro-stepping = 16x
- Gear Ratio = 3
- Hobb Diameter (Effective) = 7.3

$$400 * 16 * 3 / (7.3 * 3.142) = \mathbf{837 \text{ E-steps-per-mm}}$$

(For the motor sold with the Titan)

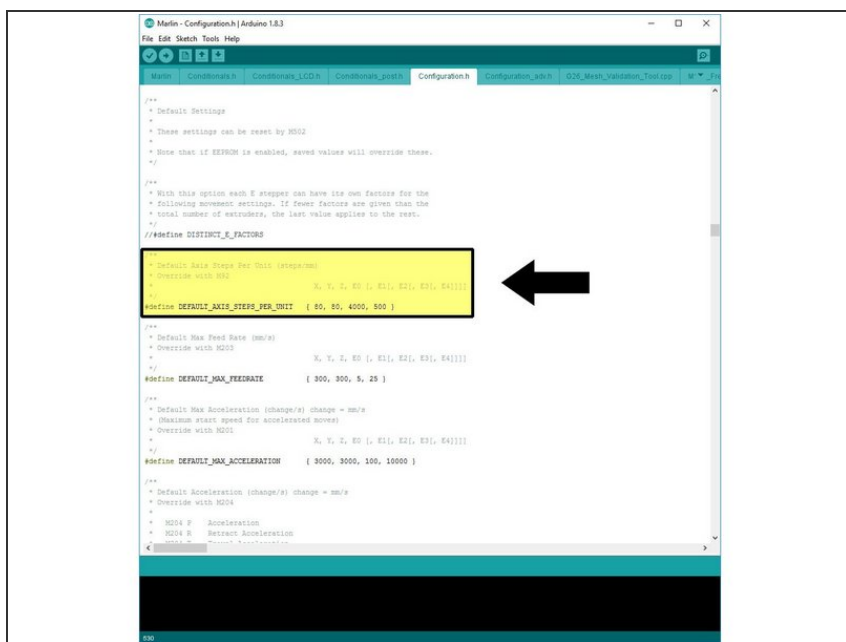
- The next thing we'll have to update is your printer's E-steps-per-mm.
- ⓘ Your slicer will generate G-Code for your printer, which will tell it to extrude a certain length (in millimeters) of filament. Your printer takes those lengths of filament and calculates how much it should rotate your Titan's stepper motor to push out the expected amount filament. This number is used to make that conversion
- First, we'll start with a ball-park estimate of your E-steps-per-mm, and then we'll fine-tune it.
- If you're using the standard Titan Slimline motor, start with 837 Steps-per-mm

## Step 6



- i** To set your new E-steps-per-mm you need to edit your firmware and EEPROM
- i** EEPROM are special settings that can be changed without re-uploading new firmware to your printer. Steps-per-mm settings for each axis are included in the EEPROM. If you update your firmware, your EEPROM will overwrite any changes your firmware might have tried to make.
- i** Not all printers have EEPROM settings, so if you can't update them, just update your firmware instead.

## Step 7 — Update E-Steps



```

Marlin - Configuration.h [Arduino 1.8.3]
File Edit Sketch Tools Help
Main Configuration LCD Configuration_pins Configuration.h Configuration_pins CNC_Mesh_Visualizer_Tutorial
/**
 * Default Settings
 * These settings can be reset by M502
 * Note that if EEPROM is enabled, saved values will override these.
 */
/**
 * With this option each E stepper can have its own factors for the
 * following movement settings. If these factors are given than the
 * total number of extruders, the last value applies to the rest.
 */
//define DISTINCT_E_FACTORS

/* Default axis steps per unit (steps/mm)
 * Override with M92
 * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 4000, 500 }

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_MAX_FEEDRATE { 300, 300, 9, 25 }

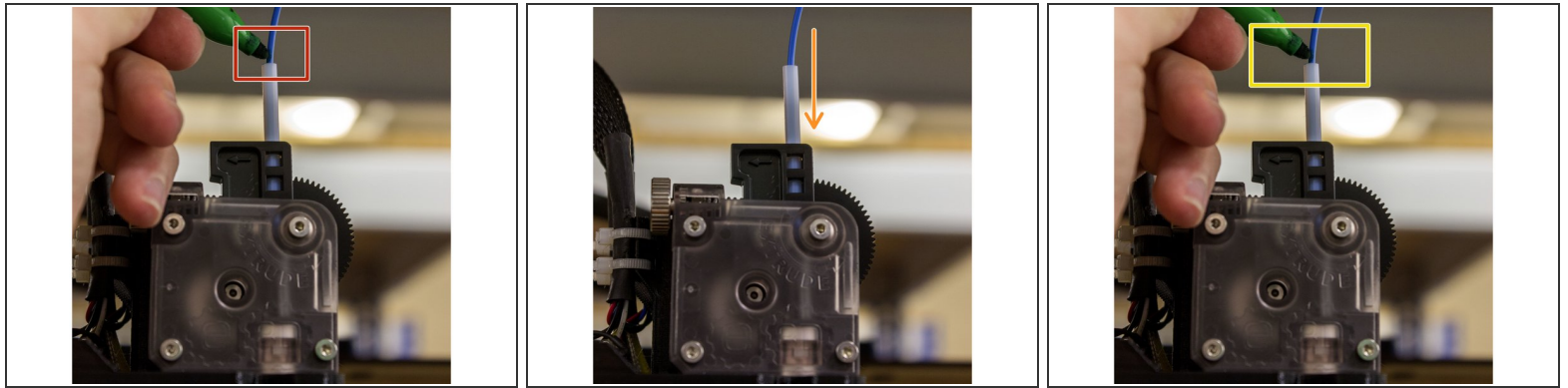
/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 * X, Y, Z, E0 [, E1[, E2[, E3[, E4]]]]
 */
#define DEFAULT_MAX_ACCELERATION { 3000, 3000, 100, 10000 }

/**
 * Default Acceleration (change/s) change = mm/s
 * Override with M201
 * M201 P Acceleration
 * M201 R Retract Acceleration
 */

```

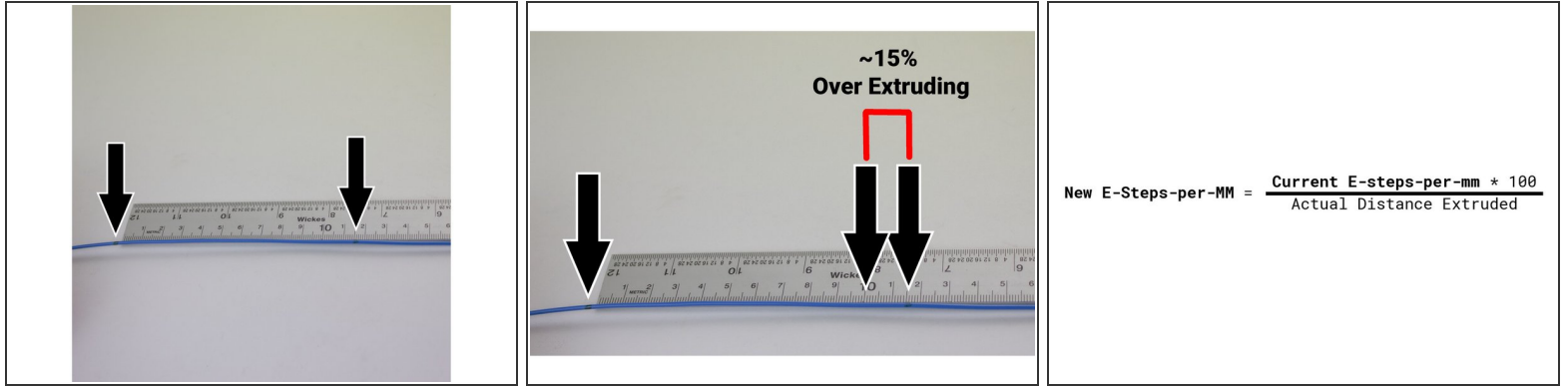
- If you have an LCD Screen:
  - Navigate to Control → Motion → Steps/mm → Esteps/m and enter your new E-steps-per-mm value.
  - Select Store Settings in the Control menu to save your settings.
- Over USB Connection
  - Use a printer control software to connect to your printer. Send the command M92 E<your number here> to your printer.
  - Then, send M500 to store your settings
- If you don't have EEPROM, or want your firmware to be consistent with your EEPROM:
  - Update the following line with your new value in the E-steps spot: `#define DEFAULT_AXIS_STEPS_PER_UNIT {<X-axis> <Y-steps> <Z-steps> <E-steps>}`. Upload your firmware as normal.

## Step 8 — Extruder Steps-per-mm Tune



- i To get a more exact value for your E-steps-per-mm, measure the exact amount of filament that is pushed out of your extruder.
  - Load filament into your extruder, just until it is gripped by the drive shaft (you can't pull it out without moving the large gear turning)
  - Mark your filament at the top of the idler arm or PTFE tubing with a pen or permanent marker .
  - Tell your printer to extrude 100mm of filament. Use your printer's LCD screen, or send it: `G92 E0`, then `G1 E100` via your printer control software.
- i You may need to heat your HotEnd before your printer allows you to extrude filament. You can use [M302](#) command to get around that.
  - Mark your filament again at the top of your idler arm or PTFE tubing
  - Eject your filament.

## Step 9 — Extruder Steps-per-mm Tune 2



- With a ruler or calipers, measure the distance between the two marks on your filament.
- If the distance wasn't exactly 100mm, use a proportion to calculate a more precise E-steps-per-mm value.
- Enter that new value into your firmware or EEPROM as you did before.

## Step 10 — Upload Firmware



- Upload the new firmware to your electronics as you normally would. Typically this means plugging in your printer to your computer, selecting the correct COM port and board type, and pressing the upload button.
- If you're unsure of how to update your printer's firmware, check with its manufacturer.

You're all done. Enjoy your new Titan!