



# Titan Aero Marlin Configuration

Set up your Marlin Firmware to support your new Titan Aero.

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 1.8.3]
File Edit Sketch Tools Help
Marlin | Configuration.h | Conditional_LCD | Conditional_post | Configuration.h | Configuration_Mini | G29_Mesh_Validation_Tool | G29_Mesh_Validation_Tool

// 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)

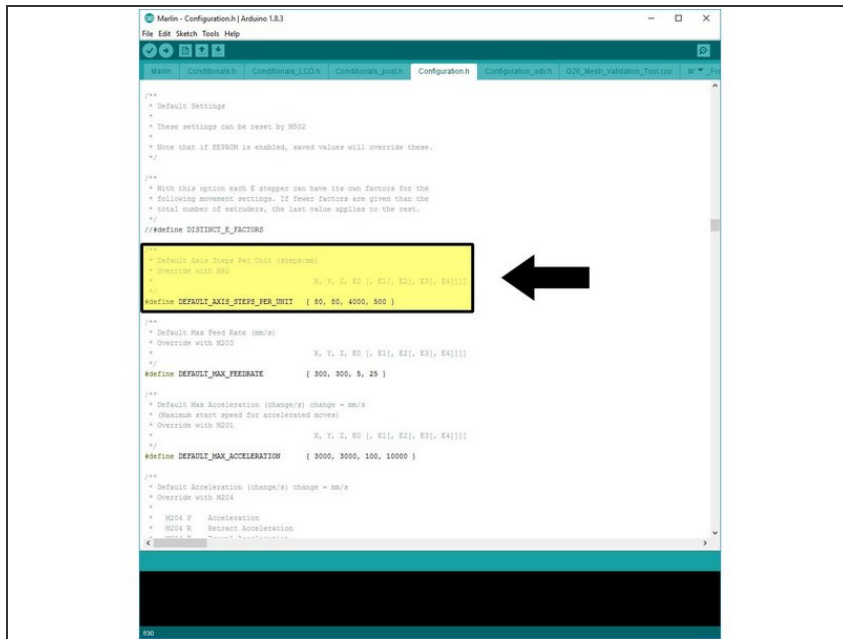
- The next thing we'll have to update is your printer's E-steps-per-mm.
- ⓘ Your slicer will generate G-Code for your printer, which will tell it to extrude a certain length (in millimeters) of filament. Your printer takes those lengths of filament and calculates how much it should rotate your Titan's 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 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



```

//#define DISTINCT_X_FACTORS
//#define DISTINCT_Y_FACTORS

/**
 * Default axis steps per unit (steps/mm)
 * Override with M92
 * X, Y, Z, E0 [X1, Y1, Z1, E1], E2[E3, E4]
 */
#define DEFAULT_AXIS_STEPS_PER_UNIT { 80, 80, 4000, 800 }

/**
 * Default Max Feed Rate (mm/s)
 * Override with M203
 * X, Y, Z, E0 [X1, Y1, Z1, E1], E2[E3, E4]
 */
#define DEFAULT_MAX_FEEDRATE { 300, 300, 9, 25 }

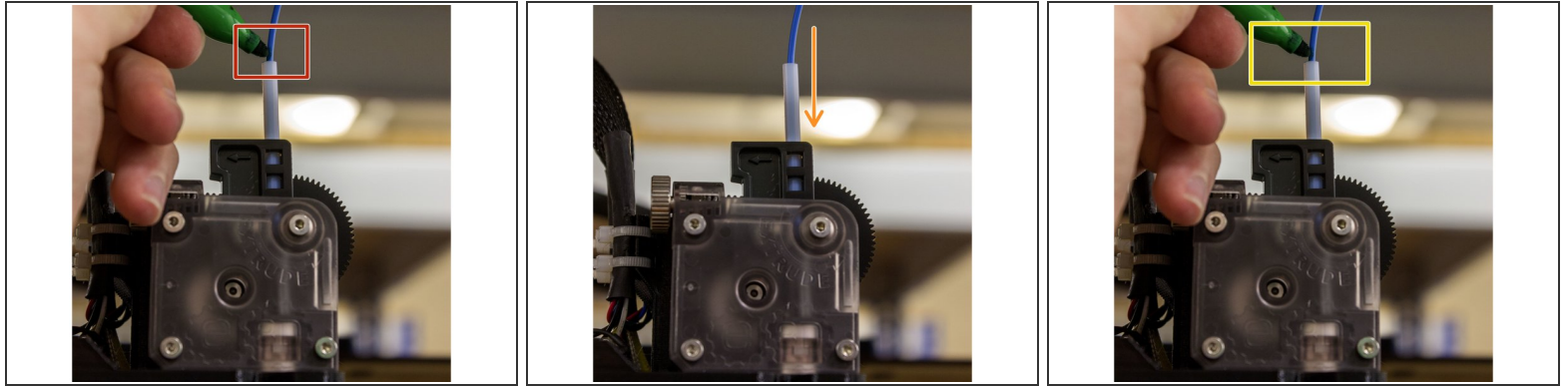
/**
 * Default Max Acceleration (change/s) change = mm/s
 * (Maximum start speed for accelerated moves)
 * Override with M201
 * X, Y, Z, E0 [X1, Y1, Z1, E1], E2[E3, E4]
 */
#define DEFAULT_MAX_ACCELERATION { 3000, 3000, 100, 10000 }

/**
 * 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

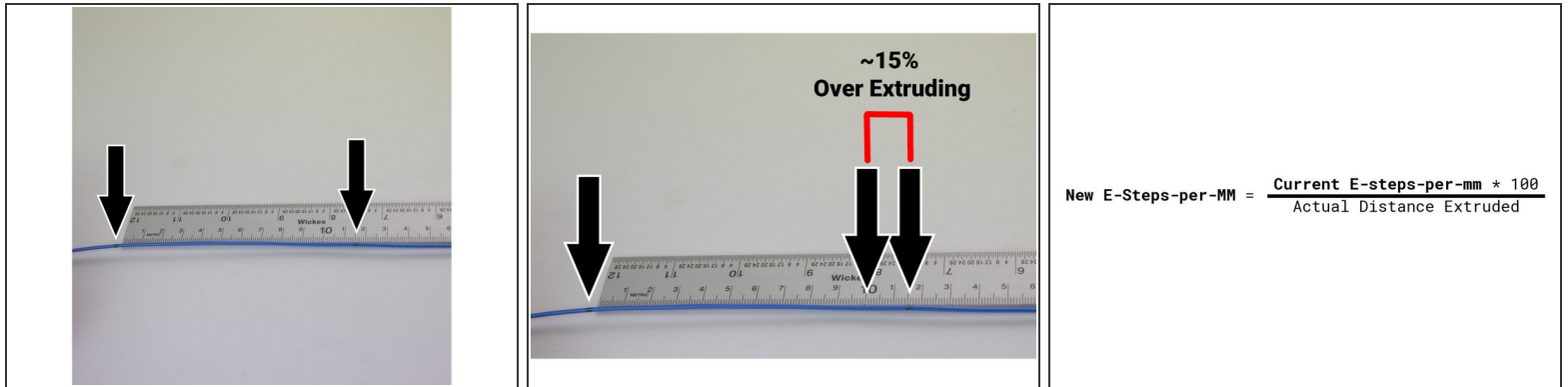


- i** To get a more exact value for your E-steps-per-mm, measure the exact amount of filament that is pushed out of your extruder.

  - 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



$$\text{New E-Steps-per-MM} = \frac{\text{Current E-steps-per-mm} * 100}{\text{Actual Distance Extruded}}$$

- 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 — Thermistor Settings

```

// Section temperature

//===== Thermal Settings =====

//-----NORMAL IS 4.7kOhm NTC100----- (this pullup can be used on hotend sensor, using correct resistor and table)

// Temperature sensor availability:
// -1 = thermocouple with MAX31855 (only for sensor 0)
// -2 = thermocouple with MAX6675 (only for sensor 1)
// -1 = thermocouple with AD595
// 0 = not used
// 1 = 100k thermistor - best choice for EPCOS 100K (4.7k pullup)
// 2 = 200k thermistor - ATC Semitec 2040T-2 (4.7k pullup)
// 3 = Mendel-parts thermistor (4.7k pullup)
// 4 = 100k thermistor - ATC Semitec 1040T-3 (used in PartC and J-head) (4.7k pullup)
// 5 = 100k thermistor - ATC Semitec 1040T-3 (used in PartC and J-head) (4.7k pullup)
// 6 = 100k Rosewell thermistor 130-10-1040-201 (4.7k pullup)
// 7 = 100k Rosewell thermistor 130-10-1040-201 (4.7k pullup)
// 8 = 100k M603 300 Tinkay WTC0603E01040E1 (4.7k pullup)
// 9 = 100k GE Heating M20000-01-100-01-01 (4.7k pullup)
// 10 = 100k K1 thermistor 100-041 (4.7k pullup)
// 11 = 100k beta 3950 1k thermistor (4.7k pullup)
// 12 = 100k 400 300 Tinkay WTC0603E01040E1 (4.7k pullup) (calibrated for Hakkon hot bed)
// 13 = 100k Klauer 3950 1k up to 200°C for hotend "Simple ONE" + "Revised "All in ONE"
// 14 = the PT100 sensor found in the Ultimaker 2 XL
// 60 = 100k Maker's Tool Works Kapton Bed Thermistor beta=3950
// 61 = 4.7k High Temperature thermistor from Dye Shop
// 70 = the 100k thermistor found in the bq Hephestor 2
// 75 = 100k Generic Silicon Heat Pad with WTC 100K M0815-1049305012 thermistor

// 1k ohm pullup tables - This is optional, and requires changing out the 4.7k pullup for 1k.
//      (but gives greater accuracy and more stable PID)
// 81 = 100k thermistor - EPCOS (1k pullup)
// 82 = 200k thermistor - ATC Semitec 2040T-2 (1k pullup)
// 83 = 100k thermistor - ATC Semitec 1040T-3 (used in PartC and J-head) (1k pullup)
// 1047 = PT100 with 4k7 pullup
// 1018 = PT100 with 1k pullup (non standard)
// 147 = PT100 with 4k7 pullup
// 110 = PT100 with 1k pullup (non standard)

// Use these for Testing or Development purposes, NEVER for production machine.
// 999 = Dummy Table that ALWAYS reads 200°C or the temperature defined below.
// 999 = Dummy Table that ALWAYS reads 100°C or the temperature defined below.
// 1 = "0": "Not used", "1": "100K / 4.7k - EPCOS", "2": "100K / 4.7k - ATC Semitec 2040T-2", "3": "Mendel-parts / 4.7k", "4": "100K / 4.7k"

//=====

```

```

//=====
// 80 = 100k Maker's Tool Works Kapton Bed Thermistor beta=3950
// 66 = 4.7k High Temperature thermistor from Dye Shop
// 70 = the 100k thermistor found in the bq Hephestor 2
// 75 = 100k Generic Silicon Heat Pad with WTC 100K M0815-1049305012 thermistor
//
// 1k ohm pullup tables - This is optional, and requires changing out the 4.7k pullup for 1k.
//      (but gives greater accuracy and more stable PID)
// 81 = 100k thermistor - EPCOS (1k pullup)
// 82 = 200k thermistor - ATC Semitec 2040T-2 (1k pullup)
// 83 = 100k thermistor - ATC Semitec 1040T-3 (used in PartC and J-head) (1k pullup)
// 1047 = PT100 with 4k7 pullup
// 1018 = PT100 with 1k pullup (non standard)
// 147 = PT100 with 4k7 pullup
// 110 = PT100 with 1k pullup (non standard)
//
// Use these for Testing or Development purposes, NEVER for production machine.
// 999 = Dummy Table that ALWAYS reads 200°C or the temperature defined below.
// 999 = Dummy Table that ALWAYS reads 100°C or the temperature defined below.
// 1 = "0": "Not used", "1": "100K / 4.7k - EPCOS", "2": "100K / 4.7k - ATC Semitec 2040T-2", "3": "Mendel-parts / 4.7k", "4": "100K / 4.7k"

//=====
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_3 0
#define TEMP_SENSOR_4 0
//=====

// Dummy thermistor constant temperature readings, for use with 999 and 999
#define DUMMY_THERMISTOR_999_VALUE 25
#define DUMMY_THERMISTOR_999_VALUE 100

// Use temp sensor 1 as a redundant sensor with sensor 0, if the readings
// from the two sensors differ too much the print will be aborted.
// #define TEMP_SENSOR_1_AS_REDUNDANT
// #define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10

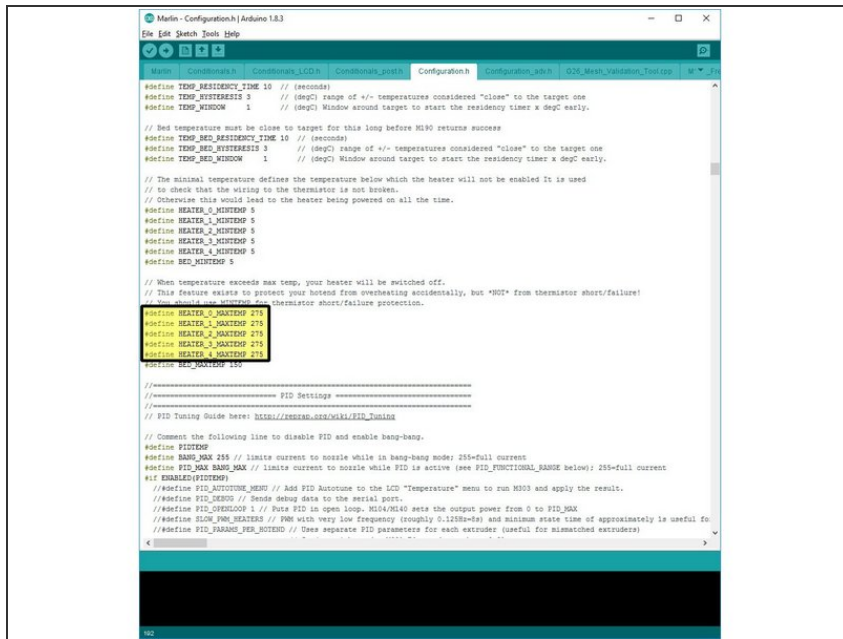
// Extruder temperature must be close to target for this long before M109 returns success
#define TEMP_RESPOND_TIME 10 // (seconds)
#define TEMP_HYSTERESIS 3 // (degC) range of +/- temperatures considered "close" to the target one
#define TEMP_WINDOW 1 // (degC) Window around target to start the residency timer a degC early.

// Bed temperature must be close to target for this long before M90 returns success
// #define BED_TEMP_WINDOW 0

```

- In the configuration.h file, find the Thermal Settings section. Below the comments you'll find the settings for the types of thermistors your printer uses. (Typically there will be one per hotend and one more if you have a heated bed.)
- If you're installing your V6 as your only hotend, change the first highlighted line to: `#define TEMP_SENSOR_0 5`
- If you're replacing an existing hotend or have multiple hotends, adjust whichever line corresponds to the tool number that you're changing (they start counting from 0)

## Step 11 – Set Maximum Temperature



```

Marlin-Configuration (Arduino 1.8.3)
File Edit Sketch Tools Help
Main Configuration.h ConfigurationLCD.h Configuration_pins Configuration.h Configuration_pins Config_Mesh_Visualizer_Tutorial.h
#define TEMP_RESIDENCY_TIME 10 // (seconds)
#define TEMP_HYSTERESIS 5 // (degC) range of +/- temperatures considered "close" to the target one
#define TEMP_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

// Bed temperature must be close to target for this long before M30 returns success
#define TEMP_BED_RESIDENCY_TIME 10 // (seconds)
#define TEMP_BED_HYSTERESIS 5 // (degC) range of +/- temperatures considered "close" to the target one
#define TEMP_BED_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

// The minimal temperature defines the temperature below which the heater will not be enabled It is used
// to check that the wiring to the thermistor is not broken.
// Otherwise this would lead to the heater being powered on all the time.
#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define HEATER_3_MINTEMP 5
#define HEATER_4_MINTEMP 5
#define BED_MINTEMP 5

// When temperature exceeds max temp, your heater will be switched off.
// This feature works to protect your bed from overheating accidentally, but "M05" from thermistor short/failure!
// Use the "M303 E" command to protect your bed from overheating accidentally, but "M05" from thermistor short/failure protection.
#define HEATER_0_MAXTEMP 275
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define HEATER_3_MAXTEMP 275
#define HEATER_4_MAXTEMP 275
#define BED_MAXTEMP 300

//===== PID Settings =====
//===== PID Tuning Guide here: https://reprap.org/wiki/PID_Tuning =====
// Comment the following line to disable PID and enable bang-bang.
#define PIDTEMP
#define PID_MAX 255 // limits current to max while in bang-bang mode: 255=full current
#define PID_MIN BANG_MAX // limits current to min while PID is active (see PID_FUNCTIONAL_RANGE below); 255=full current
#define ENABLED(PIDTEMP)
//#define PID_AUTOTUNE_MENU // Add PID Autotune to the LCD "Temperature" menu to run M303 and apply the result.
//#define PID_DEBUG // Sends debug data to the serial port.
//#define PID_OPENLOOP 1 // Puts PID in open loop. M304/M340 sets the output power from 0 to PID_MAX
//#define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.12Hz) and minimum state time of approximately 1s is useful fo
//#define PID_PARAMS_PER_HOTEND // Uses separate PID parameters for each extruder (useful for mismatched extruders)

```

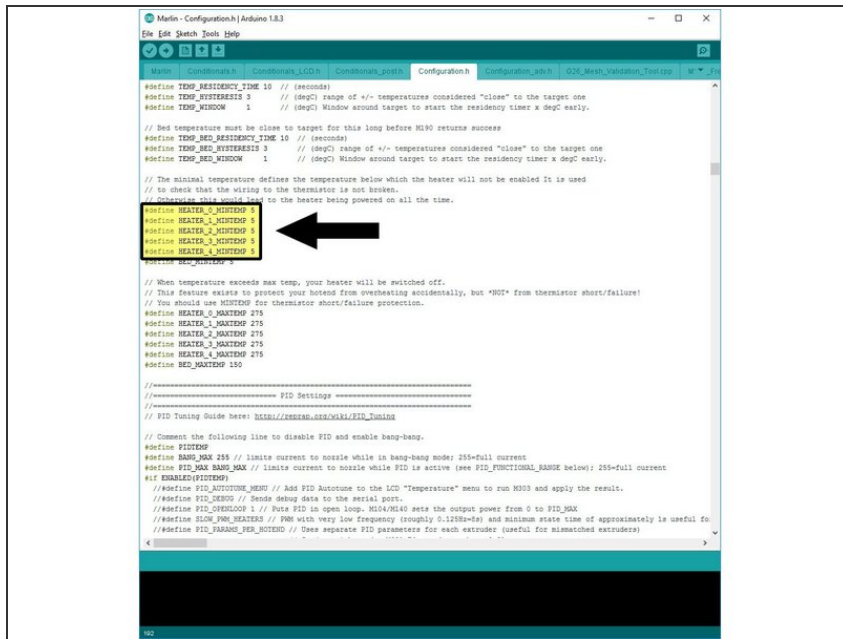
**i** While the metal of your new V6 hotend can withstand very high temperatures, your thermistor can only go up to around 285°C for extended periods.

- Set that as your maximum temperature by changing the first highlighted line to: `#define HEATER_0_MAXTEMP 285` (or adjust the line corresponding to the hotend you're changing.)

**i** The Max settable temperature on the LCD is 15 less than the max temperature, you will need to set the max temperature to 300°C in the firmware in order to hot tighten at 285°C.

**!** Remember to change the max temperature back to 285°C after you have hot tightened.

## Step 12 — Set Minimum Temperature



```

Marlin-Configuration.h [Arduino 1.8.3]
File Edit Sketch Tools Help

Main Configuration.h Conditionals_LCD.h Conditionals_post.h Configuration.h Configuration_adv.h G29_Mesh_Valuation_Tools.h

#define TEMP_RESISTENCY_TIME 10 // (seconds)
#define TEMP_HYSTERESIS 0 // (degC) range of +/- temperatures considered "close" to the target one
#define TEMP_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

// Bed temperature must be close to target for this long before M190 returns success
#define TEMP_BED_RESISTENCY_TIME 10 // (seconds)
#define TEMP_BED_HYSTERESIS 0 // (degC) range of +/- temperatures considered "close" to the target one
#define TEMP_BED_WINDOW 1 // (degC) Window around target to start the residency timer x degC early.

// The minimal temperature defines the temperature below which the heater will not be enabled. It is used
// to check that the wiring to the thermistor is not broken.
// Comment out this line to enable the heater to be powered on all the time.
#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define HEATER_3_MINTEMP 5
#define HEATER_4_MINTEMP 5

// When temperature exceeds max temp, your heater will be switched off.
// This feature exists to protect your heater from overheating accidentally, but "M03" from thermistor short/failure!
// You should use M30TEMP for thermistor short/failure protection.
#define HEATER_0_MAXTEMP 275
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define HEATER_3_MAXTEMP 275
#define HEATER_4_MAXTEMP 275
#define BED_MAXTEMP 150

//===== FID Settings =====
//===== FID Tuning Guide here: https://reprap.org/wiki/FID_Tuning

// Comment the following line to disable FID and enable bang-bang.
#define FIDTEMP
#define FID_MAX 255 // limits current to max while in bang-bang mode; 255=full current
#define FID_MIN BANG_MAX // limits current to min while FID is active (see FID_FUNCTIONAL_RANGE below); 255=full current
#define ENABLE(FIDTEMP)
// #define FID_AUTOTUNE_MENU // Add FID Autotune to the LCD "Temperature" menu to run M303 and apply the result.
// #define FID_SEND // Send temp data to the serial port.
// #define FID_OPEN_LOOP // Put FID in open loop. M304/M340 sets the output power from 0 to FID_MAX
// #define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.128Hz) and minimum state time of approximately 1ms is useful for
// #define FID_PARAMS_PER_HOTEND // Uses separate PID parameters for each extruder (useful for mismatched extruders)

```

- For safety it is strongly recommended to make sure that your printer detects if the thermistor stops sending correct temperatures for any reason.
- Set the first highlighted line to: **#define HEATER\_0\_MINTEMP 5** (or adjust the line corresponding to the hotend you're changing.)
- In newer versions of Marlin there are other features such as Thermal Runaway Protection that might be useful as well, though they are typically on by default.

## Step 13 — 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!

This document was last generated on 2019-07-09 04:28:01 AM.