

Chimera Marlin Configuration

Set up your Marlin Firmware to support your new Chimera

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 V6 HotEnd, 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



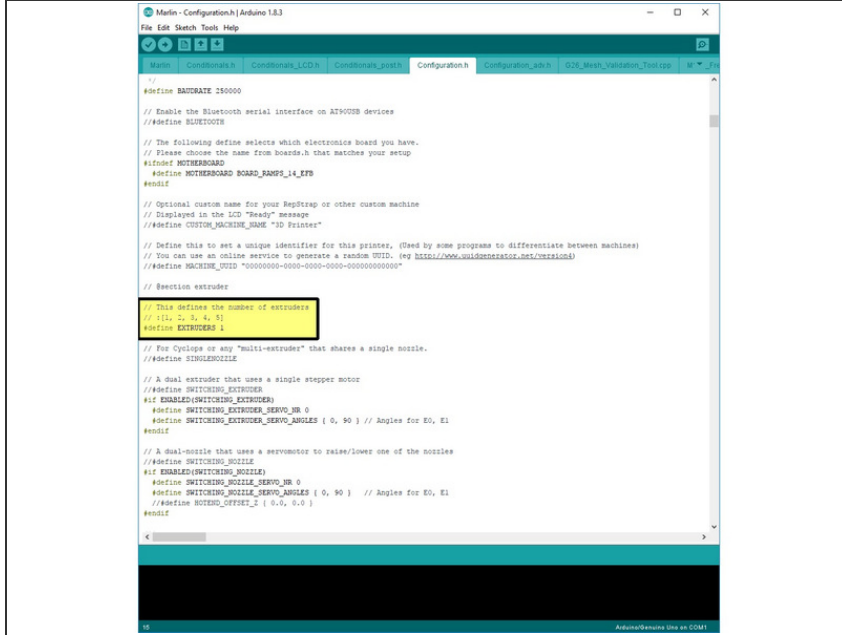
- Almost all printers use Arduino IDE to upload fresh firmware, so download it at <https://www.arduino.cc/en/Main/Software>

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

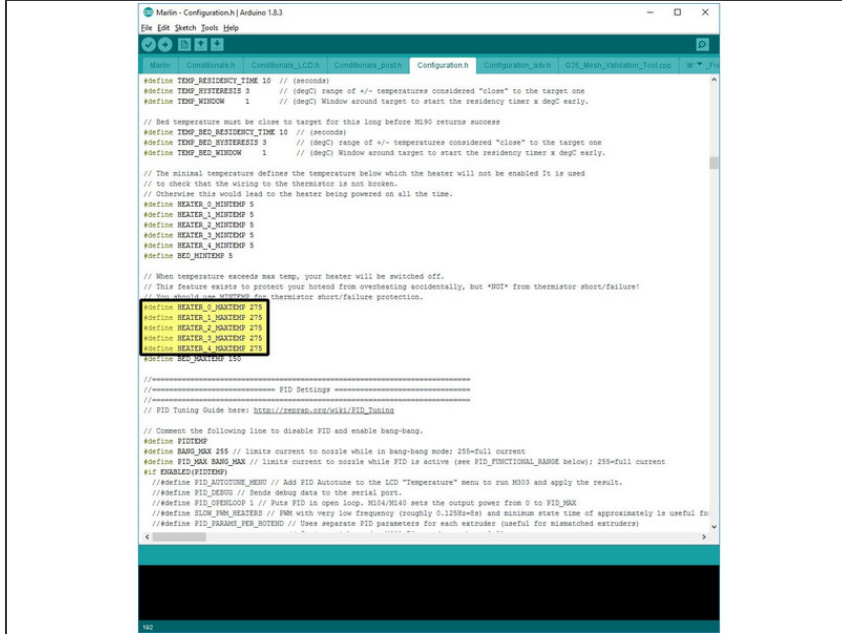
Step 4 — Update Number of Extruders



```
Marlin - Configuration.h [Arduino 1.8.3]
File Edit Sketch Tools Help
Main Conditional_LCD Conditional_post Configuration.h Configuration_Adv.h G2D_Mesh_Validation_Tool.cpp M...
-//
#define BAUDRATE 250000
// Enable the Bluetooth serial interface on AT90USB devices
// #define BLUETOOTH
// The following define selects which electronics board you have.
// Please choose the name from boards.h that matches your setup
#define MOTHERBOARD
#define MOTHERBOARD BOARD_RAMPF_14_EFB
#endif
// Optional custom name for your RepRap or other custom machine
// Displayed in the LCD "Ready" message
// #define CUSTOM_MACHINE_NAME "3D Printer"
// Define this to set a unique identifier for this printer. (Used by some programs to differentiate between machines)
// You can use an online service to generate a random UUID. (see https://www.uuidgenerator.net/version4)
// #define MACHINE_UUID "00000000-0000-0000-0000-000000000000"
// Section extruder
// This defines the number of extruders
// {1, 2, 3, 4, 5}
#define EXTRUDERS 1
// For Cyclops or any "multi-extruder" that shares a single nozzle.
// #define SINGLENOZZLE
// A dual extruder that uses a single stepper motor
// #define SWITCHING_EXTRUDER
// #define SWITCHING_EXTRUDER_SERVO_0
// #define SWITCHING_EXTRUDER_SERVO_ANGLES { 0, 90 } // Angles for E0, E1
// #endif
// A dual-nozzle that uses a servomotor to raise/lower one of the nozzles
// #define SWITCHING_NOZZLE
// #define SWITCHING_NOZZLE_SERVO_0
// #define SWITCHING_NOZZLE_SERVO_ANGLES { 0, 90 } // Angles for E0, E1
// #define NOZZLE_OFFSET_Z { 0.0, 0.0 }
// #endif
13
Arduino/Genuino Uno on COM1
```

- In the configuration file, find the extruder section, and set the number of extruders to two, by updating the following line: `#define EXTRUDERS 2`

Step 6 — Set Maximum Temperature



```

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

Main.h  Configuration.h  Conditionals.h  Conditionals_LCD.h  Conditionals_posh  Configuration.h  Configuration_adv.h  G29_Mesh_Valuation_Tool.cpp

// 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 3 // (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.

// 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 3 // (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 exists to protect your hotend from overheating accidentally, but *NOT* from thermistor short/failure!
// You should use MINTEMP 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

//===== 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 BANG_MAX 255 // limits current to nozzle while in bang-bang mode; 255=full current
#define PID_MAX_BANG_MAX // limits current to nozzle while PID is active (see PID_FUNCTIONAL_RANGE below); 255=full current
#define BANG_MINTEMP
// #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. M303 sets the output power from 0 to PID_MAX
// #define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.125Hz) and minimum state time of approximately 1s 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 Chimera 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.)

Step 7 — Set Minimum Temperature



```

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

Main.h  Configuration.h  Conditionals.h  Conditionals_LCD.h  Conditionals_posh  Configuration.h  Configuration_adv.h  G29_Mesh_Valuation_Tool.cpp

// 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 3 // (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.

// 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 3 // (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 exists to protect your hotend from overheating accidentally, but *NOT* from thermistor short/failure!
// You should use MINTEMP 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

//===== 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 BANG_MAX 255 // limits current to nozzle while in bang-bang mode; 255=full current
#define PID_MAX_BANG_MAX // limits current to nozzle while PID is active (see PID_FUNCTIONAL_RANGE below); 255=full current
#define BANG_MINTEMP
// #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. M303 sets the output power from 0 to PID_MAX
// #define SLOW_PWM_HEATERS // PWM with very low frequency (roughly 0.125Hz) and minimum state time of approximately 1s useful fo
// #define PID_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 two highlighted lines to: `#define HEATER_0_MINTEMP 5` and `#define HEATER_1_MINTEMP 5` respectively
- 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 8 — 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.

Head back to the [Chimera page](#) to finish the last few steps before you start printing.

This document was last generated on 2017-10-19 10:41:15 PM.