

Anycubic i3 Mega / Mega-S Marlin 1.1.9 by davidramiro

Donate PayPal downloads 6k total open issues 0 closed issues 25 license GPL-3.0 release v1.4.5

This is a custom version of the [Marlin Firmware](#) for the i3 Mega/Mega-S, gratefully based on [derhopp's repo](#) with his remarkable efforts to get the Anycubic TFT screen to work with the latest versions of Marlin.

Looking for a **BLtouch firmware**? Head [this way!](#) Mounting and configuration instructions are included.

Note: This is just a firmware, not magic. A big part of print quality still depends on your slicer settings and mechanical condition of your machine.

Make sure to take a look at the [Wiki](#), especially the [FAQ](#).

Why use this?

While the i3 Mega is a great printer for its price and produces fantastic results in stock, there are some improvements and additional features that this firmware provides:

- Many people have issues getting the Ultrabase leveled perfectly, using Manual Mesh Bed Leveling the printer generates a mesh of the flatness of the bed and compensates for it on the Z-axis for perfect prints without having to level with the screws.
- Much more efficient bed heating by using PID control. This uses less power and holds the temperature at a steady level. Highly recommended for printing ABS.
- Fairly loud fans, while almost every one of them is easily replaced, the stock FW only gives out 9V instead of 12V on the parts cooling fan so some fans like Noctua don't run like they should. This is fixed in this firmware.
- Even better print quality by adding Linear Advance, S-Curve Acceleration and some tweaks on jerk and acceleration.
- Thermal runaway protection: Reducing fire risk by detecting a faulty or misaligned thermistor.
- Very loud stock stepper motor drivers, easily replaced by Watterott or FYSETC TMC2208. To do that, you'd usually have to flip the connectors on the board, this is not necessary using this firmware.
- No need to slice and upload custom bed leveling tests, test it with a single GCode command
- Easily start an auto PID tune or mesh bed levelling via the special menu (insert SD card, select special menu and press the round arrow)
- Filament change feature enabled: Switch colors/material mid print (instructions below) and control it via display.
- The filament runout, pause and stop functionality have been overhauled and improved: The hotend now parks and retracts (on pause, stop or filament runout) and purges automatically (on resume).
- Added **M888** cooldown routine for the Anycubic Ultrabase (EXPERIMENTAL): This is meant to be placed at the end Gcode of your slicer. It hovers over the print bed and does circular movements while running the fan. Works best with custom fan ducts.
 - Optional parameters:
 - **T<temperature>** : Target bed temperature (min 15°C), 30°C if not specified (do not set this under room temperature)
 - **S<fan speed>** : Fan speed between 0 and 255, full speed if not specified
 - e.g. **M888 S191 T25** : run the fan at 75% until the bed has cooled down to 25°C

Known issues:

- Power outage support is not included
- Estimated print times from your slicer might be slightly off.
- Special characters on any file or folders name on the SD card will cause the file menu to freeze. Simply replace or remove every special character (Chinese, Arabic, Russian, accents, German & Scandinavian umlauts, ...) from the name. Symbols like dashes or underscores are no problem. **Important note: On the SD card that comes with the printer there is a folder with Chinese characters in it by default. Please rename or remove it.**

How to flash this?

I provided three different precompiled hex files: One for no modifications on the stepper motor drivers - good for people who didn't touch anything yet, one for boards with TMC2208 installed and where the connectors have been flipped and one with TMC2208 and the connectors in original orientation.

Choose your precompiled hex:

- Download the precompiled firmware here: [Releases](#)
- Choose the correct hex file:
- For TMC2208 with connectors in original orientation, use [Marlin-AI3M-vx.x.x-TMC2208.hex](#)
- If you use TMC2208 and already reversed your connectors, use [Marlin-AI3M-vx.x.x-TMC2208_reversed.hex](#)
- If you use a newer version of the TMC2208 that doesn't require the connector to be reversed (TMC2208 "v2.0" written on the PCB, chip on the top side), please also use [Marlin-AI3M-vx.x.x-TMC2208_reversed.hex](#) .
- If you use the original stepper motor drivers, use [Marlin-AI3M-vx.x.x-stock_drivers.hex](#) .

Or compile it yourself:

- Download and install [Arduino IDE](#)
- Clone or download this repo (or extract Source.zip from Thingiverse)
- Browse into the Marlin folder and run [Marlin.ino](#)

- In the IDE, under **Tools -> Board** select **Genuino Mega 2560** and **ATmega2560**
- **Customize if needed** (e.g. motor directions and type at line **559** to **566** and line **857** to **865** in **Configuration.h**)
- Under **Sketch**, select **Export compiled binary**
- Look for the .hex file in the Marlin directory (only use the **Marlin.ino.hex**, not the **Marlin.ino.with_bootloader.hex** !)

After obtaining the hex file:

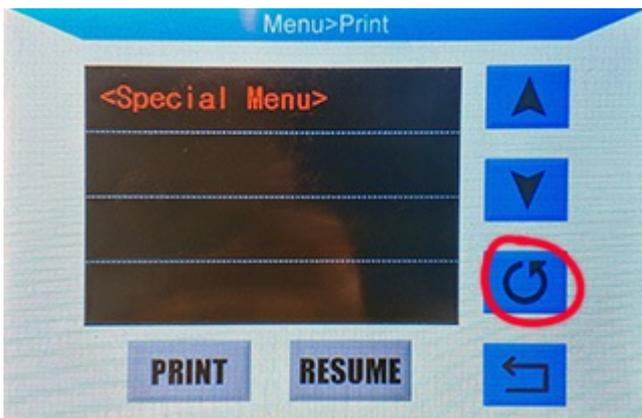
- Flash the hex with Cura, OctoPrint or similar (Use Google if you're not sure how)
 - e.g. in Cura: Settings -> Printer -> Manage Printers -> Update Firmware.
- Get a tool with a terminal (OctoPrint, Pronterface, Repetier Host, ...) to send commands to your printer.
 - Side note: Cura can also send commands via USB in the **Monitor** tab, but since you can't read responses from the printer, I'd recommend a tool with a proper terminal
- **Important:** Connect to the printer and send the following commands: (Only needed on major firmware flash - e.g. coming from Anycubics default FW - or getting an EEPROM error; not needed after flashing an update of this firmware unless mentioned in the changelog)
- **M502** - load hard coded default values
- **M500** - save them to EEPROM
- **If you are using this on a Mega-S, those two additional commands are necessary:**
- **M92 E384** - set correct steps for the new extruder
- **M500** - save them
- I highly recommend calibrating the extruder as written below.

Calibration & Tuning

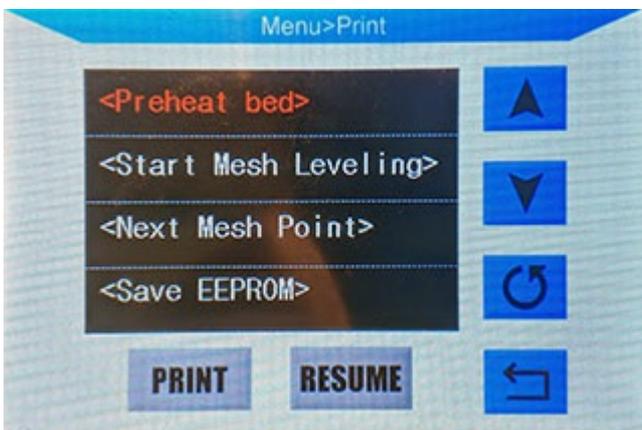
Manual Mesh Bed Leveling

If you have issues with an uneven bed, this is a great feature.

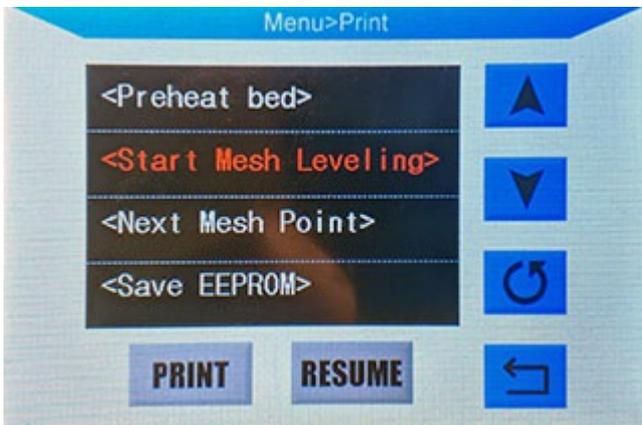
- Insert an SD card, enter the print menu.
- Enter the special menu by selecting it and pressing the round arrow:



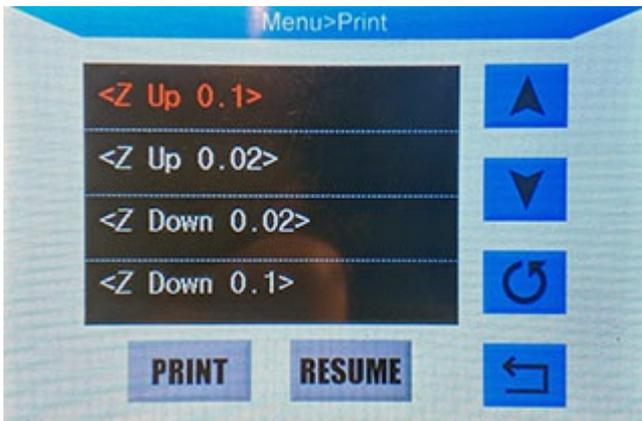
- In this menu, the round arrow is used to execute the command you selected.
- Preheat the bed to 60°C with this entry: (if you usually print with a hotter bed, use the Anycubic menu)



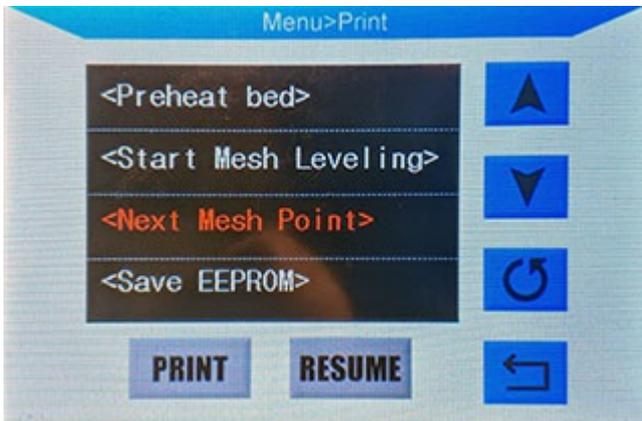
- Level your preheated bed as well as you can with the four screws.
- Start the mesh leveling:



- Your nozzle will now move to the first calibration position.
- Don't adjust the bed itself with screws, only use software from here on!
- Use a paper - I recommend using thermopaper like a receipt or baking paper
- Use the onscreen controls to lower or raise your nozzle until you feel a light resistance: **(If you want to send the same command multiple times, select the item again, even though it is still marked red.)**



- Once finished , move to the next point:



- Repeat the last two steps until all 25 points are done.
- Your printer will beep, wait 20 seconds and then save (the printer will beep once more to confirm):



After leveling:

- Reboot the printer.
- To ensure your mesh gets used on every print from now on, go into your slicer settings and look for the start GCode
- Look for the Z-homing (either just `G28` or `G28 Z0`) command and insert these two right underneath it:

```
M501
M420 S1
```

- Your printer should now correctly print first layers even on a warped bed.
- When working on the printer, installing a new hotend or nozzle or the bed warping over time, a new Mesh Leveling procedure is recommended.

Manual commands for use with OctoPrint etc.:

- `G29 S1` - Start MMBL
- `G29 S2` - Next Mesh point
- Raising Z: `G91`, `G1 Z0.02`, `G90` (one after another, not in one line)
- Lowering Z: `G91`, `G1 Z-0.02`, `G90`
- All 25 points done: After seeing `ok` in the console, send `M500` to save.

Testing your bed leveling

- I can recommend this [customizable leveling test](#) by kenwebart.

Alternatively via GCode:

- Send those commands to your printer:

```
G28
G26 C H200 P5 R25 Q4.2 Z4
```

- To adjust your filament's needed temperature, change the number of the `H` parameter
- Default bed temperature is 60°C, if you need another temperature, add e.g. `B80`
- `Q` parameter sets retraction length in mm, `Z` sets unretraction.
- If your leveling is good, you will have a complete pattern of your mesh on your bed that you can peel off in one piece
- Optional: Hang it up on a wall to display it as a trophy of how great your leveling skills are.

Calibrating extruder & PID

Extruder steps

- Get your old E-Steps with `M503`. Look for the line starting with `M92`, the value after the `E` are your current steps.
- Preheat the hotend with `M104 S220`
- Send `M83` to prepare the extruder
- Use a caliper or measuring tape and mark 120 mm (measured downwards from the extruder intake) with a pencil on the filament
- Send `G1 E100 F100`
- Your extruder will feed 100 mm of filament now (takes 60 seconds)
- Measure where your pencil marking is now. If it's exactly 20 mm to the extruder, it's perfectly calibrated
- If it's less or more than 20 mm, subtract that value from 120 mm, e.g.:
- If you measure 25 mm, your result would be 95 mm. If you measure 15 mm, your result would be 105 mm
- Calculate your new value: **(100 mm / actually extruded filament) * your current E-Steps** (default: i3 Mega 92.6, Mega-S 384)
- For example, if your markings are at 15 mm, you'd calculate: $(100/105) * 92.6 = 88.19$
- Put in the new value like this: `M92 X80.00 Y80.00 Z400.00 Exxx.xx`, replacing `x` with your value
- Save with `M500`
- Finish with `M82`
- You can repeat the process if you want to get even more precise, you'd have to replace your current E-steps with your newly calibrated value in the next calculation.

PID tuning

PID calibration is only necessary if you experience fluctuating temperatures.

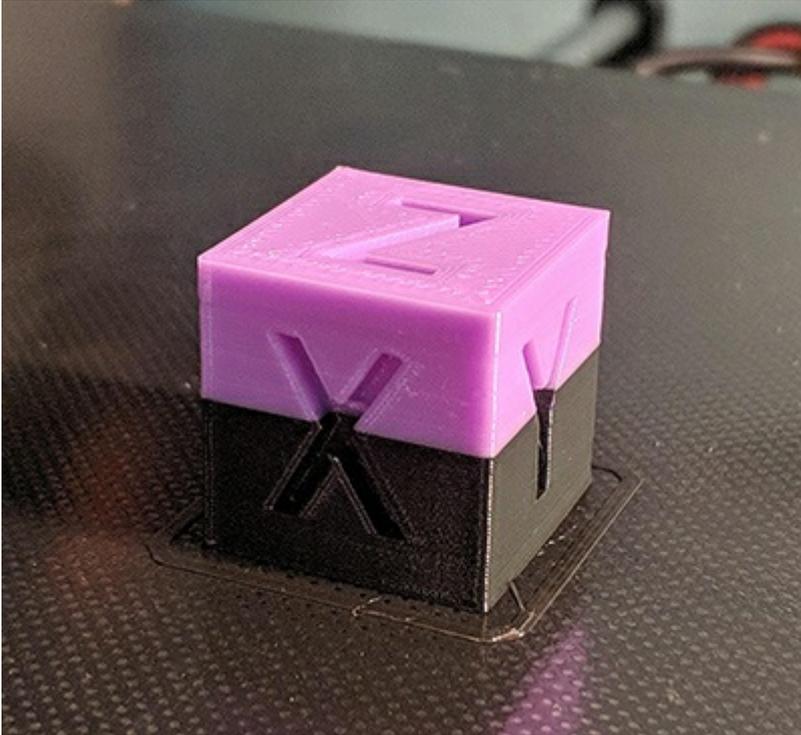
- Turn on parts cooling fan If you have a radial blower fan like the original one, I generally recommend running it at 70% because of the 12V mod (`M106 S191`). Remember to also limit it in your slicer.
- Send `M303 E0 S210 C15 U1` to start extruder PID auto tuning
- Wait for it to finish (Get a coffee! This can take a while.)
- Send `M303 E-1 S60 C6 U1` to start heatbed PID auto tuning

- Wait for it to finish
- Save with `M500`, turn off fan with `M106 S0`

Note: These commands are tweaked for PLA printing at up to 210/60 °C. If you run into issues at higher temperatures (e.g. PETG & ABS), simply change the `S` parameter to your desired temperature

Reminder: PID tuning sometimes fails. If you get fluctuating temperatures or the heater even fails to reach your desired temperature after tuning, you can always go back to the stock settings by sending `M301 P15.94 I1.17 D54.19` and save with `M500`. You can also try my values: `M301 P20.28 I1.45 D71.01 + M500`.

M600 Filament Change



BETA: This now also works without USB printing, via SD & display.

Configuration (only needed once):

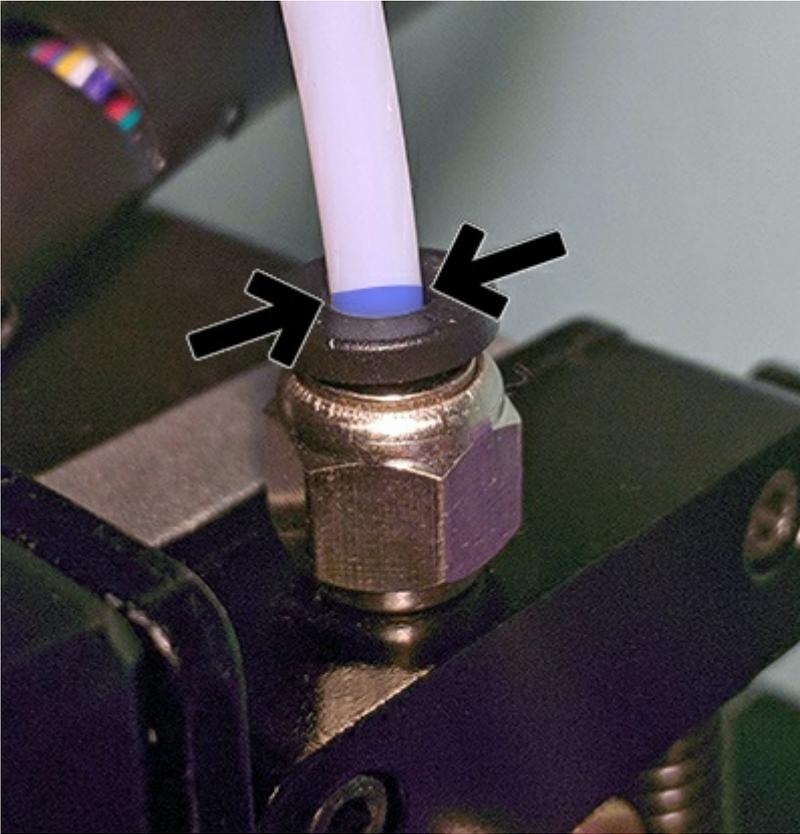
- Send `M603 L0 U0` to use manual loading & unloading.
- Send `M603 L538 U555` to use automatic loading & unloading
 - The `L` and `U` parameters define the load and unload length in mm. The values above work well on a stock setup, if you modded your extruder, bowden tube or hotend, you might need to adjust those.
- Save with `M500`

Filament change process (manual loading):

- For printing via SD:
 - Place `M600` in your GCode at the desired layer
- For printing via USB:
 - Place `M600` in your GCode at the desired layer or send it via terminal
 - Alternatively: Use `FilamentChange Pause` in the Special Menu
- The nozzle will park and your printer will beep
 - For safety reasons, the printer will turn off the hotend after 10 minutes. If you see the temperature being under the target:
 - SD printing: Click `CONTINUE` (**only once!**) on the screen and wait for the hotend to heat up again.
 - USB printing: Send `M108` and wait for the hotend to heat up again.
- Remove the filament from the bowden tube
- Insert the new filament right up to the nozzle, just until a bit of plastic oozes out
- Remove the excess filament from the nozzle with tweezers
- For printing via SD:
 - Click `CONTINUE` on the screen
- For printing via USB:
 - Send `M108` via your USB host or use `FilamentChange Resume` in the Special Menu
 - Note for OctoPrint users: After sending `M108`, enable the advanced options at the bottom of the terminal and press `Fake Acknowledgement`

Filament change process (automatic loading):

- For printing via SD:
 - Place `M600` in your GCode at the desired layer
- For printing via USB:
 - Place `M600` in your GCode at the desired layer or send it via terminal
 - Alternatively: Use `FilamentChange Pause` in the Special Menu
- The nozzle will park
- The printer will remove the filament right up to the extruder and beep when finished
 - For safety reasons, the printer will turn off the hotend after 10 minutes. If you see the temperature being under the target:
 - SD printing: Click `CONTINUE` (**only once!**) on the screen and wait for the hotend to heat up again.
 - USB printing: Send `M108` and wait for the hotend to heat up again.
- Insert the new filament just up to the end of the bowden fitting, as shown here:



- For printing via SD:
 - Click `CONTINUE` on the screen
- For printing via USB:
 - Send `M108` via your USB host or use `FilamentChange Resume` in the Special Menu
 - Note for OctoPrint users: After sending `M108`, enable the advanced options at the bottom of the terminal and press `Fake Acknowledgement`
- The printer will now pull in the new filament, watch out since it might ooze quite a bit from the nozzle
- Remove the excess filament from the nozzle with tweezers

Troubleshooting & Contribution

Back up & restore your settings

Some updates require the storage to be cleared (`M502`), if mentioned in the update log. In those cases, before updating, send `M503` and make a backup of all the lines starting with:

```
M92
G29
M301
M304
```

After flashing the new version, issue a `M502` and `M500`. After that, send every line you saved before and finish by saving with `M500`.

Something went wrong?

No worries. You can easily go back to the default firmware and restore the default settings.

- Flash the hex file from the [manufacturer's website](#) (in case it's offline, I have uploaded the stock firmwares [here](#) as well).
- After flashing, send `M502` and `M500`. Now your machine is exactly as it came out of the box.

Collaboration

Get your hands dirty and join in on the development! Your fork is just two clicks away. Don't hesitate to message me either via PM on here or by email (to be found on my GitHub profile) to get us starting to work on further improving this together.

Reporting bugs or issues

First: Read the [FAQ](#). I am very grateful for any feedback and constructive criticism. While I ran this firmware for over 800 print hours successfully, I am by no means a commercial user and this machine is just for private purposes. It's stable for me, but since every machine is different, feel free to chime in with your observations.

But please keep in mind that a simple two-liner with nothing but "x doesn't work" won't get us anywhere (let alone on Thingiverse, with a single comment feed at over 500 posts now - it's a mess to navigate). So I highly encourage you to go onto the [firmware's GitHub](#) and fill out the prepared bug report (or feature request) form.

Firmware details

Detailed changes:

- Thermal runaway protection thresholds tweaked
- Manual mesh bed leveling enabled ([check this link](#) to learn more about it)
- Heatbed PID mode enabled
- TMC2208 configured in standalone mode
- Stepper orientation flipped (you don't have to flip the connectors on the board anymore)
- Linear advance unlocked (Off by default. [Research](#), [calibrate](#) and then enable with `M900 Kx`)
- S-Curve Acceleration enabled
- G26 Mesh Validation enabled and tweaked
- Some redundant code removed to save memory
- Minor tweaks on default jerk and acceleration
- Print statistics enabled (send `M78` to read them)
- `M600` filament change feature enabled
 - Implemented easy resume via display
- Filament runout behaviour tweaked
 - Added purge and retract
 - Move nozzle to park position on runout
 - Prevent false positives by adding a small delay to the sensor
- Pause and stop behaviour tweaked
- Added `M888` cooldown routine for the Anycubic Ultrabase
- Startup chime and endstop beeps added as option, disabled by default

Changes by [derhopp](#):

- Ported and improved display code from Anycubic's Marlin 1.1.0 source (big kudos!)
- Removed 9V limit from FAN0 (parts cooling fan)
- Subdirectory support added
- Special menu added

Disclaimer

```
/*
 * Flashing a custom firmware happens at your own risk.
 *
 * THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS"
 * AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
 * THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
 * PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS
 * BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR
 * CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF
 * SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS
 * INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER
 * IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)
 * ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */
```

License

Marlin is published under the [GPLv3 license](#) because we believe in open development. The GPL comes with both rights and obligations. Whether you use Marlin firmware as the driver for your open or closed-source product, you must keep Marlin open, and you must provide your compatible Marlin source code to end users upon request. The most straightforward way to comply with the Marlin license is to make a fork of Marlin on Github, perform your modifications, and direct users to your modified fork.