Marlin 2.0 Firmware configured for Ender 3 (8-bit) + BL Touch

1 year ago | By Daniel (/p/en/user/1) | 206 comments

This is the vanilla version of **Marlin Firmware 2.x** pre-configured for Creality Ender 3 printers with **original or generic** BL Touch sensors. The standard configuration file from Marlin distribution was carefully reviewed to include the latest Ender 3 specific settings from the official Creality firmware and from Antclabs.

Please note that this **is not the version I recommend for 8-bit boards** like the one used in the Ender 3. The most stable and feature rich version of the firmware for this architecture is the 1.1.9 that I made available in another article (/p/en/node/136). Marlin 2.x is aimed to a new generation of boards using 32-bit processors, the built in libraries are bigger and there is less space left for features. Anyway, if you want to try this you can be sure I did the best I could to configure it properly and fit the maximum amount of features possible. It was downloaded and installed more than 10000 times and I believe most of the users will find no problems with it.

Features

- Official "vanilla "Marlin firmware with all hardware settings from Creality latest official firmware.

- The maximum amount of Marlin features that I was able to fit in the small 128KB memory.
- BLTouch original/generic versions enabled by default. Compatible with BLTouch versions from 1.0 up to 3.x
- No need to remove any capacitors from the board
- Installs from the PlatformIO IDE that runs in several operating systems
- Source code modified to prevent releasing the BLTouch sensor during "beep" events triggered by LCD menus
- Thermal runway protection enabled
- Does not reset the bed level settings after the G28 (Home) command
- Custom menu with maintenance mode, emergency mode and full auto bed leveling sequence (warmup, wait, home, level and then save settings)
- Ultra-precise bilinerar bed leveling algorithm with 25 probing points, 3 slow probes per point (takes about 10 minutes to complete)
- Run and toggle the bed level and BLTouch tools from LCD menus
- Tuned PIDs for faster heating
- M503 enabled (reports all printer settings)
- Baby steps enabled and configured for the stepper driver "Magic Numbers"
- Fully compatible with OctoPrint
- All changes from the original Marlin config file were marked with the comment tag "Customized DBP"
- Due memory size limitations, the power resume function was disabled as well as the arc support
- LCD menus were changed to "slim mode" to save memory space
- Optional compiled files ready to upload to the board (no configuration, no compilation, no installation of any tools):
    - Creality standard board
    - Creality upgraded boards with TMC silent drivers (like the 1.1.5)

Note about the bed leveling strategy

There are two ways to perform the automatic bed leveling, the first is before every print and the second is only when necessary. Based on my experience with the Ender 3, it's better to do an extremely precise (slower) bed leveling when necessary than a simpler one before every print. This firmware configuration reflects this choice.

If you want to do quick/imprecise bed leveling before every print you will need to reconfigure the firmware.

Requirements

- BLTouch installed on pin 27
- Z stop switch removed (it can block the Z axis movement)

- **IMPORTANT!** BLTouch tip height must be adjusted according to the official instruction manuals (https://www.antclabs.com/manual). There are different BLTouch sensor clearences and the YouTube videos tips are wrong most of the times
- Bootloader installed
- Microsoft Visual Studio Code (https://code.visualstudio.com/) if you plan to install from the source code
- PlatformIO IDE (https://platformio.org/) extension installed on Visual Code if you plan to install from the source code

Instructions for installation from the Source Code

1. Download and install Visual Studio Code. (https://code.visualstudio.com/)
2. From Visual Studio Code extension menu, install the PlatformIO IDE. (https://platformio.org/)
3. With all requirements running, unpack this entire Marlin source code downloaded from the link below to a folder.
4. From Visual Studio Code select File->Open folder and select the folder where the *platformio.ini* file is located. **Attention! Use the "open folder" command, not the"open file".**
5. Look for the *configuration.h* inside the *"Marlin"* sub-folder and edit the following configuration to the relative coordinates of your BLTouch mount to the Nozzle. This does not need to be too precise, the only purpose is avoiding the probe to test outside the bed. The default settings X = -42mm, Y= -5mm and Z=0 are for this mount (https://www.thingiverse.com/thing:3003725):

   ```
   #define NOZZLE_TO_PROBE_OFFSET { -42, -5, 0 }
   ```

6. Save the file and then select Terminal->Run Task->Build . It will take a few minutes for the system to download all dependencies and compile the code.
7. If the code compiled properly you will see something like this in the terminal: *"=== 1 succeeded in 00:00:30.050 ==="*.
8. Connect your printer to the computer using the USB cable.
9. Go to the Visual Code menu Terminal->Run Task->Upload .
10. If everything goes well another message like this *"=== 1 succeeded in 00:00:55.010 ==="* will be printed.

Installation using the pre-compiled HEX file

The pre-compiled firmware is also available here. You can simply burn the compiled HEX file directly to your Ender3 board **without installing the Visual Studio Code or PlataformIO**.

To upload the pre-compiled firmware you will need the AVRDUDE 6.3 (https://www.nongnu.org/avrdude/) tool. For your convenience I included the Windows version of this tool in the zip file

that contains the HEX firmware but the tool is also shipped with the Arduino IDE and PlatformIO. The upload command in Windows PowerShell should look like this:

```
./avrdude -p atmega1284p -c arduino -P COM3 -b 115200 -v -U flash:w:firmware.h
```

The serial port is something between COM1 and COM8, depending on where the USB driver is pointing to, and *firmware.hex* is the file you are uploading. If you are using other operating system like Linux the port address will be somehting like this */dev/ttyUSB0*.

After initializing the board with the new firmware don't forget to reconfigure the probe offsets running the following command from any serial terminal. Replace the default offset values in the command (-42, -5, 0) with the ones for your probe mount.

```
M851 X-42 Y-5 Z0
M500
```

Post-installation configuration

1. **Manually pre-level the bed**! This is important because the BLTouch has limited max/min detection ranges and the print quality can be affected if the Z-Axis moves too much.

    1. Cool down the bed and the nozzle. Heating should have little to no effect on this specific procedure.
    2. Disable stepper motors using the LCD menu command.
    3. Place a credit card on one the bed, the thickness of the card will be used as a reference distance.
    4. Turn the Z axis with your hand until the nozzle hits the credit card.
    5. Move hot end to several positions of the bed, starting from the 4 corners. For each position, adjust the bed level knobs to keep the nozzle at the credit card thickness distance from the nozzle. This is what you would do in manual calibration but instead of using a paper you will use the card for convenience. The card thickness does not matter because the BLTouch probe will redefine the Z reference.

2. Heat the bed to the printing temperature and run the bed level command from the LCD (Motion -> Bed Leveling -> Level bed) or using the G29 gcode or using the *Heat and Level* command from the custom menu (recommended). Note that the *Heat and Level* command waits 2 minutes after the bed is hot to allow the mounted surfaces to expand/contract properly.
3. Save the settings using the LCD menu or the M500 gcode (if you used the *Heat and Level* custom command, they will be automatically saved).
4. Add the following gcode to the startup code of you slicer, right after the line of the G28 command to load the bed level configuration from the memory before every print:

```
M420 S1
```

5. After enabling the bed level, print something large and flat to adjust the Z offset. The nozzle will always be far from the bed until you adjust the offset **to a negative value** (LCD -> Configuration -> Probe Z Offset). Don't forget to save the settings after the print to keep the offset in the memory.

Custom pre-compiled firmware versions for different BLTouch mounting brackets

If you have a different bracket for BLTouch and would like me to pre-compile the firmware for you please let me know in the comments below. The compiled firmware will be available for everyone so I need detailed information about the bracket manufacturer/model or the link to the site hosting the STL files.

Troubleshooting tips

- **Disable serial port clients before flashing**: If the COM ports are used by any other app, the flashing process will fail. Shutdown Cura, OctoPrint, any other slicer and any other Serial Monitor program like the Arduino IDE before compiling/flahsing the new ROM.
- **Correct fuse settings:** If you are using other software than AVRDUDE to upload the firmware make sure you are using the correct fuse settings. Apparently the Creality tutorial has the wrong settings and the board fails to boot afterwards.
- **Reset the EEPROM**: Sometimes the EEPROM memory is not reset during the Marlin reflash and this can cause problems. To reset manually use the following G-Codes from any Serial Terminal: *M502* followed by *M500*.
- **Blank screen after flashing**: You did not flashed the firmware properly. Please use AVRDude according to the instructions above and if this does not work you might have to reflash the bootloader.
- **Optiboot bootloader:** To compile/flash the firmware forcing optiboot support, edit the file *platformio.ini* and replace the lines *default_envs = melzi* or *default_envs = sanguino_atmega1284p* with *default_envs = melzi_optiboot* . If you flashed your bootloader with Arduino IDE, that uses optiboot by default, you don't need to update this setting, it works perfectly out of the box.
- **Serial Port Monitor for Windows**: This small program (https://github.com/whitestone-no/open-serial-port-monitor) allows you to send/receive GCode to the printer. If the characters look strange you are probably using the wrong baud rate (start with 115200bps).
- ***avrdude: stk500_getsync() attempt 1 of 10: not in sync*:** Wrong baud rate for the serial port

or serial port already in use (OctoPrint, Cura, etc..). Try configuring the upload rate to 115200bps. If you are using PlatformIO to upload, make sure you are using the correct baud rate for your environment in the *platformio.ini* file.

- **Reconfigure the BLTouch probe offsets:** Use the following command replacing the -42, -5 and 0mm coordinates with the offsets of your actual mount:

```
M851 X-42 Y-5 Z0
M500
```

If you have problems compiling the firmware...

I can compile the firmware for you. Please see here (/p/en/node/152) for more details.

Other Marlin downloads for Ender 3

- Marlin 1.1.9 for Ender 3 (8-bit) with Mesh Bed Leveling (no BLTouch) (/p/en/node/146)
- Marlin 1.1.9 for Ender 3 (8-bit) with BLTouch (/p/en/node/136)
- Ultimate Marlin 2 for Ender 3 (8-bit) + BLTouch (advanced users only) (/p/en/node/143)
- Marlin 2.x for Ender 3 with SKR Mini E3 v2 boards (/p/en/node/149)

FILES

Version: Marlin 2.0.1 (January/2020)
Platforms: Any operating system
File size: 5.1 MB (source), 450 KB (HEX)
Download link
Package 1: Source Code for Ender 3 + BLTouch (https://www.danbp.org/downloads/Marlin-2.0.1.zip)
Package 2: Compiled Marlin (HEX) for Ender 3 + BLTouch (https://www.danbp.org/downloads/Marlin-2.0.1_HEX.zip)
Package 3: Compiled Marlin (HEX) for Ender 3 TMC2208 + BLTouch (https://www.danbp.org/downloads/Marlin-2.0.1_HEX_TMC2208.zip)
Checksum
Package 1 SHA256:
770FA69E3F5A694C41521CBE307EA9B6CC73CCCAE7494669EF93B87D68D0DE9E
Package 2 SHA256:
E557C4E5CA17BE436BEE977DA3C864734C66516A0D5608841A6C4DBF47201509
Package 3 SHA256:
96242782112FBE50EE206EA826FBDB38B22DB342D020E83C947109EDEC36B61A