

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

INTELLIGENT SYSTEMS DIVISION, ENGINEERING LAB

DEVELOPMENT OF A SOFT MATERIAL 3D PRINTER FOR ADVANCING CAPABILITIES IN SOFT
ROBOTICS AT NIST

Marlin Architecture Documentation

Jennifer Case, Vinh Nguyen, Jeremy Marvel

Initial Completion Date

Audience: Users who are required to make changes to the Marlin code to change or add additional capabilities to the printer.

Table of Contents

Disclaimer	2
Change Log	3
Executive Summary	4
Problem Statement & Scope	4
Goal	4
1. Requirements	5
1.1. Install Visual Studio Code	5
1.2. Download the Firmware from GitHub	5
1.3. Prepare the Visual Studio Code Environment	5
2. Compiling and Uploading Firmware to Printer	6
2.1. Compiling the Firmware	6
2.2. Uploading the Firmware	6
2.2.1. Troubleshooting: Upload Failed because of port detection	6

List of Figures

Fig. 1. Location of the ERASE button.	6
---	---

List of Tables

Disclaimer

Certain trade names and company products are mentioned in the text or identified in certain illustrations. In no case does such an identification imply recommendation or endorsement by the NIST, nor does it imply that the products are necessarily the best available for the purpose.

Change Log

Executive Summary

Problem Statement & Scope

The purpose of this documentation is to provide an overview of the Marlin firmware architecture and provide guidance for how to change code for future capabilities.

The scope shall be limited to describing the relevant Marlin firmware files required for the adapted Lulzbot printer including the additional files Lulzbot added and to assisting with troubleshooting anticipated changes to the code.

Goal

To familiarize users with the architecture of the firmware that runs the soft material printer such that they are able to make additional changes to the code quicker than would be possible without proper documentation.

1. Requirements

To change the Marlin firmware and upload the firmware to the Lulzbot Taz Pro's Archim2 motherboard, two applications are required: (1) Visual Studio Code and (2) Arduino IDE.

1.1. Install Visual Studio Code

The latest version of the Visual Studio Code software can be downloaded from:

<https://code.visualstudio.com/Download>

No specific extensions are required to edit the Marlin firmware.

1.2. Download the Firmware from GitHub

1. Contact Jennifer Case (jennifer.case@nist.gov) to get access to the GitHub repository.
2. [Optional] Download GitHub Desktop to manage repositories on the computer:

<https://desktop.github.com/>

3. Pull the exploratory-soft-printer using GitHub Desktop or a preferred Git manager to desired location.

1.3. Prepare the Visual Studio Code Environment

To open the Marlin firmware in Visual Studio Code:

1. Select File »Open Folder...
2. Navigate to the exploratory-soft-printer folder.
3. Select, but do not click into, the Marlin-firmware folder and hit the "Select Folder" button. The Marlin-firmware folder should now be navigable in the Explorer of Visual Studio Code.
4. Select View »Extensions.
5. Download the C/C++ and C/C++ Extension Pack extensions. This should enable coloring of the code to easily identify which code is not used with the current configuration settings. It will also enable easy navigation to files by CTRL+clicking on the file name in include statements.
6. Search and download PlatformIO IDE. This extension is necessary for the Auto Build Marlin extension and the Marlin devs apparently hate the Arduino IDE since it seems to introduce a number of bugs.
7. Search and download the Auto Build Marlin extension. This simplifies building the code for debugging and uploading the code to the board.

2. Compiling and Uploading Firmware to Printer

2.1. Compiling the Firmware

This procedure assumes you have followed the requirements documented in Section 1 and have the appropriate folder open.

1. Select the Auto Build Marlin option on the left bar of Visual Studio Code.
2. Select the Build button from one of the available environments.

2.2. Uploading the Firmware

This procedure assumes you have followed the requirements documented in Section 1 and have the appropriate folder open.

1. Remove the side panel (left of LCD when facing the front of the printer) by unscrewing the bolts.
2. Turn on the power to the Taz Pro.
3. Hold the ERASE button, shown in Fig. 1, for 1 second to clear the EEPROM memory (this switches the board from "Operating Mode" to "Uploading Mode").

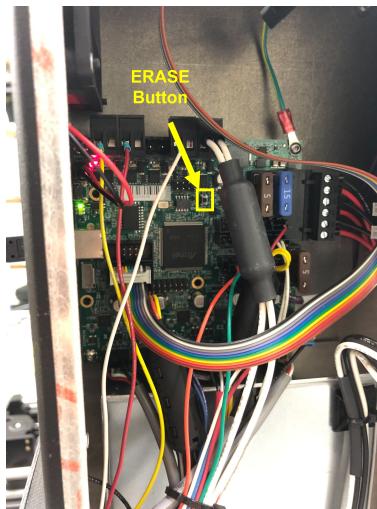


Fig. 1. Location of the ERASE button.

4. The LCD should be blank about 10 seconds after pressing the ERASE button.
5. Plug the USB cord from the printer into your computer and wait for any drivers to install.
6. Select the Auto Build Marlin option on the left bar of Visual Studio Code.
7. Select the Upload button from one of the available environments.
8. Cycle the power on the Taz Pro to make the uploaded firmware live.

2.2.1. Troubleshooting: Upload Failed because of port detection

The PlatformIO code should be able to autodetect the port to upload to, but if this fails, the following can be added to the `marlin-firmware\platformio.ini` file under the `[env]` section:

```
upload_port = COM[num]
```

The port number can be found using Device Manager.

If this line is already added in the code, make sure it is pointing to the correct port.

Reference: https://docs.platformio.org/en/latest/projectconf/section_env_upload.html