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

List of Figures

List of Tables

Disclaimer

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").

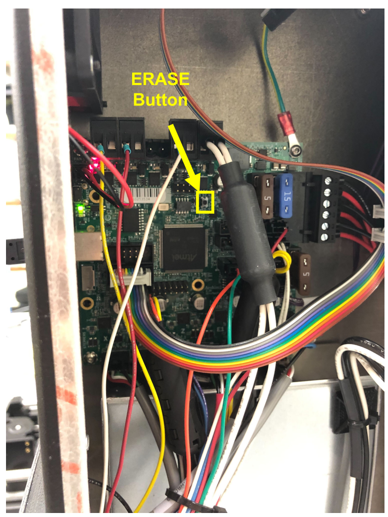

Fig. 1. Location of the ERASE button.

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

3. Configuration.h

3.1. General Printer Settings

This code uses pre-defined and adjusted settings for a Lulzbot Quiver TAZ Pro printer.

The original code provided by Lulzbot defines settings for all of their printers; however, all other printers have been removed from this code because the way that the printers were used to define the motherboard made it impossible for the Marlin Auto Builder to find the correct motherboard.

- Motherboard: Archim 2
- Serial port: -1 (USB emulated serial port)
- Baud rate: 250,000
- Linear axes: 3 (X, Y, Z)

3.2. Tool Head Settings

Lulzbot has pre-defined toolhead settings. These settings will need to be replaced when the soft material printer head is added.

For the time being, the Lulzbot Universal Toolhead will be used. All other tool heads besides the Quiver Dual Extruder were removed for simplicity since printers were already removed.

3.2.1. Universal tool head settings

- Number of extruders: 1
- Extruder type: Universal
- Tool head coordinate adjustments in X, Y, Z: 0 mm
- Motor current for 1st extruder: 0 mA (currently turned off because there is no motor attached)

3.2.2. Dual Extruder tool head settings

This list is not complete.

- Number of extruders: 2
- Extruder type: Dual Extruder
- Tool head coordinate adjustments in X, Y, Z
- Tool change Z-raise: 0 mm
- Number of servos: 2
- Servo delays
- Switching nozzles: true
- Switching nozzle servo angles
- Hot end offsets in X, Y
- Motor currents for both extruders

3.2.3. General tool head settings

- Number of extruders: defined by selected tool head
- Default filament diameter: 2.85 mm
- (If multiple extruders) X, Y offset for nozzles from the default nozzle

3.2.4. Other tool head types

These tool head types are not used, but code exists supporting them.

- A multi-extruder where multiple filament share a single nozzle
- A dual extruder that uses a single stepper motor
- A dual-nozzle that uses a servomotor to move one or both nozzles
- Two separate X-carriages for extruders with either solenoid or magnetic docking
- Swappable tool heads held in place with either servos or magnets
- Swappable tool heads where the other tool heads are placed on the side and the 3D printer can grab them as needed
- Mixing extruder that mixes multiple filaments together as it prints

3.3. Power Supply Control

This feature is not currently used.

3.4. Thermal Settings

The firmware has a large number of temperature sensors pre-defined to measure temperatures of the extruders, a heated bed, a probe, a heated chamber, and/or a cooler.

There is an option to use a redundant sensor that can be used to abort a print if two temperature readings differ by a set amount.

The firmware has pre-defined minimum temperature settings that will automatically shut off the heaters if the temperature sensors read low enough since this likely means the sensor is broken.

The firmware also has pre-defined maximum temperature settings that will also automatically shut off the heaters if the temperature is too high to prevent overheating.

Since heating often results in overshooting the target temperature, target temperatures that are too close to the maximum value will trigger a heater shutdown, so there are additional settings that prevent temperatures from being set too close to the maximum temperature to allow the overshoot to occur without triggering a shutdown.

- Type of temperature sensors for up to seven extruders, bed, probe, chamber, cooler, and a redundant sensor
- Two dummy thermistors with changeable values
- Times to wait for extruders, bed, and chamber to settle (M109, M190, M191): (10, 10, 10) s
- Temperature windows signaling when the temperature has been reached for the extruders, bed, and chamber: (1, 1, 1) C

- Temperature hysteresis allowed for being close enough to the target temperature for the extruders, bed, and chamber: (3, 3, 3) C
- Sensor that will provide a redundant sensor reading
- Sensor that the redundant reading is for
- Temperature difference that triggers a print abort
- Minimum temperatures for the extruders, bed, and chamber
- Maximum temperatures for the extruders, bed, and chamber
- Temperature difference from the maximum temperature to avoid an overshoot-triggered shutdown for the extruders, bed, and cooler.

3.5. PID Settings

The firmware allows both PID and Bang-bang controllers. If the PID controllers are turned off, it will default to bang-bang controllers.

PID gains can be tuned for the extruders, bed, and a chamber.

3.6. Additional Extruder Settings

There is a safety protocol that prevents cold extrusion given a minimum extruder temperature.

There is also a protocol to prevent lengthy extrusions given a maximimum extrusion length.

3.7. Thermal Runaway Protection

There are additional protocols in place to prevent thermal damage to printers. These protocols are turned on for the extruders, bed, chamber, and cooler.

[More information in Configuration_adv.h]

3.8. Mechanical Settings

This feature is not currently used.

3.9. Endstop Settings

Endstops indicate when the toolhead has reached the limit along one of the axes. The firmware can specify endstops at either the minimum or maximum along the axes.

Endstops can have either pullups or pulldowns. These can be specified for all endstops if they all have either pullups or pulldowns or can be specified individually if there is a mixture of pullups and pulldowns.

Endstops generally have three connection points: Common, Normally Open (NO), and Normally Closed (NC). NO means that it is connected to Common when the endstop is triggered; NC means that it is disconnected from Common when the endstop is triggered. The Marlin firmware assumes that Common is connected to ground and NC is used as the signal. There is an option to tell the firmware to invert the logic in case that assumption does not hold. This inversion is done on an individual end stop basis.

If all the endstop pins function as interrupts, there is a feature that can be turned on to run the firmware faster by removing endstop polling.

There are additional features to manage endstop noise or detect broken endstops. The noise function should only be used if the endstops falsely trigger.

3.10. Stepper Drivers

There are many built-in drivers in the Marlin firmware that can be used to control the motors controlling the linear axes and the extruders.

3.11. Movement Settings

The firmware specifies a default conversion for steps/mm for the stepper motors along the axes and the extruders. This default can be overridden with M92.

The firmware specifies a default maximum speed, or feedrate, for the stepper motors along the axes and the extruders. This default can be overriden with M203. A limit can be set for how much these defaults can be changed via M203 or through the LCD menu.

The firmware specifies a default maximum acceleration for the stepper motors along the axes and the extruders. This default can be overriden with M201. A limit can be set for how much these defaults can be changed via M201 or through the LCD menu.

The firmware specifies a default acceleration for the stepper motors along the axes and the extruders. This default can be overriden with M204 (M204 P - printing acceleration; M204 R - retraction acceleration; M204 T - travel acceleration).

The firmware specifies a default jerk limits for the stepper motors along the axes and the extruders. This default can be overriden with M205.

There is an optional feature for S-curve acceleration that is intended to reduce vibration and make smoother direction changes that is not currently used.

3.12. Z Probe Options

There are many built-in options for various probing methods:

- The Lulzbot sets the Z probe to be the nozzle and also sets it so the probe (i.e., nozzle) should be used as the Z endstop during homing.
- Manual probing allows bed leveling to be done manually using the LCD screen. However, there may be additional settings that have to be turned off if no probe exists.
- Fixed mount probes (e.g., inductive probes or probe-switches) are located at a fixed location from the nozzle.
- The nozzle can be used as the probe assuming there is a conductive nozzle or other sensor integrated with the nozzle to measure when it is close to the bed. This is the method that Lulzbot used.
- A Z-Servo probe deploys a probe via a servo motor. Angles are stored for the stowed and deployed angles.
- A BLTouch probe is a manufacturer specific probe. This probe uses a Hall effect sensor.
- A solenoid probe is a non-contact probe.
- A sled-mounted probe is docked along the x-axis and is grabbed by the printhead to perform a probing task and returned when when the task is complete.
- A rack-and-pinion probe uses a rack-and-pinion mechanism to deploy a probe.
- A duet smart effector probe is a manufacturer specific probe. This probe is built into an extruder.
- It is possible to probe the bed directly with the nozzle without the use of an additional sensor by monitoring the response of the stepper motor controlling the z-axis. However, it requires using a stepper

motor that is capable of using StallGuard measurement software. Additionally, using this method to probe can damage a Z-axis lead screw if present on the printer, so caution is required if this probing method is utilized.

- An allen key can be used as a probe to push a tactile sensor.

When using a probe, it is necessary to know the location of the probe with respect to the extruder nozzle. If there are multiple extruders, the probe offset is defined with respect to the first, or main, extruder. The offset should be measured along all three axes. A range is provided for the z-axis in case the z-offset needs to be changed via the printer menu. If a probe needs to be deployed, the clearance required for probing should also be provided. There is also a clearance specified when traveling between probe points and for probing multiple times at a given location. A minimum z-coordinate is also provided to ensure that a failure in the probe to trigger does not damage the printer.

The firmware specifies a probing margin that is used to keep probes away from the edge of the bed to avoid accidentally driving the tool head into the bed. There are certain probes, such as using the extruder nozzle as the probe, that can have negative probing margins. Note that for the Lulzbot printer, the probing locations are not on the bed, which is why a negative probing margin may be needed.

The firmware specifies a safe probing location, given in x- and y-coordinates, that is used to provide the printer with a location to probe during M48 operations or for the optional probe offset wizard.

The firmware specifies the travel speed for the x- and y-axes when probing. Additionally, there are two travel speeds given for the z-axis: a fast probing speed and a slow probing speed for increased accuracy.

The firmware has an optional probe activation switch that will signal when a probe has been properly deployed.

The firmware has an optional feature to tare the probe automatically before each probe. This feature is generally used for probes with strain gauges or piezo sensors.

The firmware has optional features to probe multiple times: (1) multiple probing and (2) extra probing. Both of these features will run the probe a specified number of times, but extra probing will attempt to remove outliers from the readings.

The firmware has an optional feature to perform a repeatability test to ensure the accuracy of the probe.

The firmware has an optional feature to wait for user input before deploying or stowing a probe.

The firmware has optional features to turn heaters, fans, or stepper motors (within reason) off during probing as well as put a delay in before pausing to remove electrical noise from readings.

The firmware has an optional feature to preheat the extruders and bed before probing to specified temperatures.

3.13. Motor Settings

The firmware specifies whether the stepper motors along the axes and for the extruders are enabled.

The firmware specifies whether the stepper motors along the axes and for the extruders should be disabled if they are not being moved; if the axes' motors are disabled, this can affect positional accuracy. For the extruders, there is an additional option to disable inactive extruders.

The firmware allows inversion of stepper motor directions if the axes move in the wrong direction.

3.14. Homing

There is an optional feature to prevent movement of the motors before homing is performed.

There is an optional feature to require homing after stepper motors are deactivated.

There is an optional feature to set a z-height to move to when motors are disabled, which is useful if the tool head moves along the z-axis when the stepper motors are disabled.

The firmware specifies a minimum z-height prior to homing.

There is an optional feature to specify a z-height to move to after homing.

The firmware specifies the location of end stops so it knows which direction to drive the axes when homing.

The firmware has an option to use Z safe homing that prevents homing outside of the bed area.

The firmware specifies speeds for homing.

3.15. Machine Settings

The firmware specifies the bed size in x- and y-directions.

The firmware specifies the minimum and maximum positions along each of the axes. It is possible to use these limits to define software end stops.

3.16. Filament Runout Sensors

The firmware has an option to use sensors to detect when filament runs out during a print.

3.17. Bed Leveling

There are several options for bed leveling:

- 3-point bed leveling: probe 3 non-collinear points to get a single titled plane.
- Linear bed leveling: probe points in a grid where the outer rectangle and density is specified to get a single tilted plane.
- Bilinear bed leveling: probe points in a grid where the outer rectangle and density is specified to get a mesh.
- Unified bed leveling: this method integrates the features of the other methods and includes mesh generation, mesh validation, and mesh editing.
- Mesh bed leveling: probe a grid manually if there is no probe.

Homing the printer disables leveling, so the firmware has optional settings to restore prior leveling or perform leveling after homing.

The firmware has an option to preheat the extruder and bed prior to leveling.

The firmware has an option to create a detailed log of leveling, but this feature requires a lot of program memory.

The firmware has an option to include a bed leveling menu if manual bed leveling is required.

3.18. Bed Skew Compensation

There is an optional feature to correct for alignment issues of the axes that is not currently used.

3.19. Additional Features

The firmware has an option to save settings in the eeprom.

The firmware has an option to let a host know when it cannot accept commands.

The firmware has an option to work in inches instead of millimeters.

The firmware is able to hold preset values for preheating the system for up to five materials.

The firmware has an optional feature to park the nozzle/extruder at a given location when the printer is idle.

The firmware has an optional feature to clean the nozzle.

The firmware has an optional feature to track how long a print job takes.

The firmware has an optional feature to track statistics about print success rates.

The firmware has an optional feature to lock the printer with a password.

3.20. LCD and SD Support

The language can be specified.

SD support can be turned on for printers that use an SD card.

4. Configuration_adv.h

4.1. Thermal Settings

The firmware allows a certain number of errors in thermocouple readings before failure occurs.

Custom thermistor settings are stored in this file.

There is an optional feature to use settings for a Hephestos 2 24V heated bed.

The settings for using a bang-bang controller for the heated bed are stored in this file.

The settings for running a heated chamber are stored in this file.

The settings for running a laser cooler are stored in this file.

The settings for thermal protection (e.g., thermal runaway, expected heating) are stored in this file.

There is an experimental optional feature to adjust heating according to extrusion speed using a PID controller.

There is an experimental optional feature to adjust the fan speeds according to heating using a PID controller.

There is an optional feature to adjust heating based on extrusion speed proportionally.

There is an optional feature to assist errors that come from using thermistors that are unreliable at low temperatures. It is better to source an appropriate temperature sensor rather than relying on this feature.

There is an optional feature for runout prevention where if the extruder temperature is above a certain temperature, it will extrude a specified amount of filament at specified time intervals. This prevents build-up of melted material in the extruder that can cause jams.

There is an optional feature to timeout the extruder if it is heated too long with no further commands. This prevents build-up of melted material in the extruder that can cause jams.

The settings for calibrating AD595 and AD8495 temperature sensors are stored in this file.

The settings for a controller fan for the drivers and transistors are stored in this file.

The settings for extruder cooling fans are stored in this file.

4.2. Case Lighting

The settings for controlling case lights are stored in this file.

4.3. Homing

There is an optional feature to keep the end stops on even when not homing. This would prevent running into the end stops in case positional errors occur.

There are optional settings if two stepper motors are used to drive the axes.

There is an optional setting to drive the E-axis (i.e., the extruder) with two synchronized stepper motors.

There are optional settings if there are two independently-driven tool heads.

The settings for the homing procedure are stored in this file.

There is a setting for assisted tramming which reads the bed corners to help adjust screws to level the bed. This setting requires a bed probe.

4.4. Motion

There are settings for the stepper drivers in this file.

The settings for disabling idle stepper motors are stored in this file.

The settings for minimum travel speeds are stored in this file.

There is an optional feature to slow down the printer if the buffer is half full.

There is an optional feature to limit the frequency of small zig-zag infill movements.

There is an optional feature for backlash compensation that adds movement for direction changes.

There is an optional feature that will calibrate positional offsets, backlash, and extruder offsets. To do the calibration, the shape of the nozzle must be defined. This feature operates on an electrically conductive cube/bolt/washer mounted to the bed.

There is an optional feature for adaptive step smoothing that increases resolution for multi-axis movement.

There are optional settings for microstepping which moves a motor at less than a full step.

The settings for stepper motor current are stored in this file.

4.5. Additional Features

Various settings for using the LCD screen to control the printer are stored in this file.

Various settings for SD card support are stored in this file.

There is an optional feature for babystepping that allows tiny movements along the axes without changing the current position values. This feature is mainly used when printing the first layer. It does not listen to endstops and certain settings in babystepping requires a probe.

There is an optional feature for linear pressure control of the extruder.

The settings for identifying probe points is stored in this file.

There are settings to overriding probing margins if needed.

There are settings to continue attempting leveling in case of leveling failure.

There are optional features for compensating probe measurements due to thermal distortion.

There is an optional feature to save a specified number of positions.

There is an optional feature to provide arc support.

There is an optional feature to provide bezier curve support.

There is an optional feature to direct step the motors, which can reduce motion calculations, increase printing speeds, and use less step aliasing, but requires changes to g-code generated from most softwares.

There is an optional feature to use a probe target.

There are settings for buffers in this file.

There are settings for retracting filament in this file.

There are settings for multiple extruders in this file.

There are settings for adjusting wait times for shutting off extruder temperature in the case of manual filament changes.

There are settings for various stepper drivers in this file.

There are settings for sensorless homing for StallGuard drivers in this file.

There are settings for CNC machines in this file.

There is an optional feature for a filament width sensor to detect and react to irregularities in material width.

The start up commands are defined in this file.

The LCD screen menu is defined in this file.