

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

27 September 2022

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
Problem Statement & Scope	4
Goal	4
Requirements	5
Install Visual Studio Code	5
Download the Firmware from GitHub	5
Prepare the Visual Studio Code Environment	5
1. Introduction	6
2. Firmware Architecture Changes	6
2.1. marlin-firmware\Marlin\Configuration.h	6
2.2. marlin-firmware\Marlin\Configuration_adv.h	11
2.3. marlin-firmware\Marlin\Version.h	13
2.4. marlin-firmware\Marlin_Bootscreen.h	14
2.5. marlin-firmware\Marlin\src\lcd\extui\ftdi_eve_touch_ui\generic\leveling_menu.cpp	14
2.6. marlin-firmware\Marlin\src\lcd\language\language_en.h	14
3. Compiling and Uploading Firmware to Printer	15
3.1. Compiling the Firmware	15
3.2. Uploading the Firmware	15
3.2.1. Troubleshooting: Upload Failed because of port detection	15
References	16

List of Figures

Fig. 1. Location of the ERASE button.	15
---	----

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

Problem Statement & Scope

The purpose of this documentation is to provide an overview of the changes made to 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 changes to the architecture of the firmware that runs the soft material printer and where future adaptions may occur.

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.

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.

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.

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.

1. Introduction

The Soft Robotics Lab (SRL) at NIST has developed a soft material printer (Coral) capable of printing 2-part elastomers using direct ink writing techniques [1–3]. Similar to the work of Walker *et al.* [4], this printer is adapted from an off-the-shelf open-source fused deposition modeling 3D printer. The 3D printer (TAZ Pro, LulzBot) runs an adapted version of Marlin firmware specific to the printer. The adapted Marlin firmware is available through the company’s GitLab (<https://gitlab.com/lulzbot3d>) and the original Marlin firmware is available through Github (<https://github.com/MarlinFirmware/Marlin>). As the Marlin firmware is continuously updated, these updates should eventually be incorporated into SRL’s Fiddler software to add functionalities.

In this document, we explain the changes made to the following files within the Marlin code:

- `marlin-firmware\Marlin\Configuration.h`,
- `marlin-firmware\Marlin\Configuration_adv.h`,
- `marlin-firmware\Marlin\Version.h`,
- `marlin-firmware\Marlin_Bootscreen.h`,
- `marlin-firmware\Marlin\src\lcd\extui\ftdi_eve_touch_ui\generic\leveling_menu.cpp`,
- `marlin-firmware\Marlin\src\lcd\language\language_en.h`,

and changes made to gcode generation for the printer.

2. Firmware Architecture Changes

In this section, the changes to the Marlin firmware are explained.

2.1. `marlin-firmware\Marlin\Configuration.h`

The original Marlin documentation for this file is available at <https://marlinfw.org/docs/configuration/configuration.html#configuration.h>. Although the changes made were to the Lulzbot version of this file (found in their GitLab under the Marlin folder), there is no additional documentation available through Lulzbot. The original Marlin documentation should provide sufficient guidance to understand what is contained in this file. The printer uses the existing extruder settings/code to run the syringe pump.

The LulzBot Configuration.h file has pre-built printer settings. All the printer options were removed so only the TAZ Pro options were left since the printer is designed for the TAZ Pro printer.

All tool heads were removed besides the LulzBot universal tool head and the dual extruder tool head which serve as examples for defining different tool heads. A NIST-specific toolhead was selected:

```
#define NIST_SINGLE_TOOLHEAD
```

The author information was updated:

```
#define STRING_CONFIG_H_AUTHOR "Lulzbot -> NIST - J. Case" // Who made the changes.
```

The version of code was updated where 2.0.9.0 is the Marlin version and the .1 at the end indicates that it is the first released version of the soft material printer code:

```
#define NIST_VERSION "2.0.9.0.1"
```

The serial port was defined:

```
#define SERIAL_PORT -1
```

The motherboard was defined:

```
#define MOTHERBOARD BOARD_ARCHIM2
```

The custom machine name and uid was defined:

```
#define CUSTOM_MACHINE_NAME "NIST Coral"
#define MACHINE_UUID "354b54cd-c287-4daf-adb5-7e190ce2e23e"
```

The NIST toolhead was added based on the Lulzbot universal toolhead:

```
#if defined(NIST_SINGLE_TOOLHEAD)
#define NIST_UNIVERSAL_MOUNT
#define NIST_EXTRUDERS 1
#define NIST_M115_EXTRUDER_TYPE "Universal"
#define NIST_TOOLHEAD_X_MAX_ADJ 0
#define NIST_TOOLHEAD_X_MIN_ADJ 0
#define NIST_TOOLHEAD_Y_MAX_ADJ 0
#define NIST_TOOLHEAD_Y_MIN_ADJ 0
#define NIST_TOOLHEAD_Z_MAX_ADJ 0
#define NIST_TOOLHEAD_Z_MIN_ADJ 0
#define NIST_MOTOR_CURRENT_E0 0 // mA
#endif
```

Change the number of extruders to point to the NIST-defined extruders:

```
#define EXTRUDERS NIST_EXTRUDERS
```

Change the hotend offsets to point to NIST-defined offsets (note this is only required for more than one extruder):

```
#define HOTEND_OFFSET_X NIST_HOTEND_OFFSET_X // (mm) relative X-offset for
each nozzle
#define HOTEND_OFFSET_Y NIST_HOTEND_OFFSET_Y // (mm) relative Y-offset for
each nozzle
```

Since the syringe pump does not require temperature settings to run nor does it have a temperature sensor, the temperature sensors for the extruders were changed to either a dummy sensor that always read 25°C or turned the sensor off entirely:

```
#define TEMP_SENSOR_0 998 // change this to a dummy sensor
#define TEMP_SENSOR_1 0 // change this to 'not used'
```

If statements around defining the temperature settle time were removed:

```
#define TEMP_RESIDENCY_TIME 10 // (seconds) Time to wait for hotend
to "settle" in M109
```

PID gains were removed for tool heads to remove references to other printers leaving a single definition:

```
#define DEFAULT_Kp SLSEHE_DEFAULT_Kp
#define DEFAULT_Ki SLSEHE_DEFAULT_Ki
#define DEFAULT_Kd SLSEHE_DEFAULT_Kd
```

If statements around the maximum power to the heated bed were removed to remove references to other printers:

```
#define MAX_BED_POWER 255 // limits duty cycle to bed; 255=full current
```

PID gains were removed for the heated bed that are not specific to this printer.

The feature that prevents filament extrusion at cold temperatures was adjusted since printing soft materials does not require elevated temperatures:

```
///#define PREVENT_COLD_EXTRUSION  
#define EXTRUDE_MINTEMP 20 // 140
```

The feature that prevents lengthy extrusions was turned off since the soft material printer requires a lengthy print at the beginning of every print in order to prime the nozzle:

```
///#define PREVENT_LENGTHY_EXTRUDE  
///#define EXTRUDE_MAXLENGTH 200
```

The endstop locations and communication methods were specified to remove references to other printers:

```
#define USE_XMIN_PLUG  
///#define USE_XMAX_PLUG  
///#define USE_ZMIN_PLUG - turned off because no probe  
///#define USE_YMIN_PLUG  
#define USE_YMAX_PLUG  
#define USE_ZMAX_PLUG  
  
#define X_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of  
    the endstop.  
#define Y_MIN_ENDSTOP_INVERTING false // Set to true to invert the logic of  
    the endstop.  
#define Y_MAX_ENDSTOP_INVERTING true // Set to true to invert the logic of  
    the endstop.  
#define Z_MIN_PROBE_ENDSTOP_INVERTING true // Set to true to invert the  
    logic of the probe.  
#define Z_MIN_ENDSTOP_INVERTING true // Set to true to invert the logic of  
    the endstop.
```

The if statement around the motor drivers were removed to remove references to other printers:

```
#define X_DRIVER_TYPE TMC2130  
#define Y_DRIVER_TYPE TMC2130  
#define Z_DRIVER_TYPE TMC2130  
#define E0_DRIVER_TYPE TMC2130  
#define E1_DRIVER_TYPE TMC2130  
#define E2_DRIVER_TYPE TMC2130
```

The endstop noise threshold is turned off to remove references to other printers:

```
///#define ENDSTOP_NOISE_THRESHOLD 2
```

The feature that detects broken endstops is turned off to remove references to other printers:

```
///#define DETECT_BROKEN_ENDSTOP
```

The default steps per unit along an axis is specified to remove references to other printers:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT { 100, 100, 500, 420 }
```

The default maximum feedrate for the syringe pump was increased given that the design of the syringe pump reduces the speed significantly through gears:

```
#define DEFAULT_MAX_FEEDRATE { 300, 300, 25, 10*60 } // adjust the
    default_max_feedrate from 40 to 10*60
```

The default maximum acceleration for the syringe pump was decreased to prevent skipping in the belt during start-up for the syringe pump:

```
#define DEFAULT_MAX_ACCELERATION { 9000, 9000, 100, 5 } // reduce
    default_max_acceleration from 9000
```

The default acceleration and travel acceleration were specified to remove references to other printers:

```
#define DEFAULT_ACCELERATION 500 // X, Y, Z and E acceleration
    for printing moves
#define DEFAULT_TRAVEL_ACCELERATION 500 // X, Y, Z acceleration for
    travel (non printing) moves
```

The default x- and y-axes jerks were specified to remove references to other printers:

```
#define DEFAULT_XJERK 8.0
#define DEFAULT_YJERK 8.0
```

The option to use the probe as a z-axis minimum endstop is turned off:

```
//#define Z_MIN_PROBEUSES_Z_MIN_ENDSTOP_PIN
```

The Lulzbot TAZ Pro printer uses the extruder nozzle as a probe. Since the nozzle for printing soft materials is not conductive and, therefore, not capable of acting as a probe, this feature was turned off and the option to probe manually was turned on:

```
#define PROBE_MANUALLY
//#define NOZZLE_AS_PROBE
```

The if statement around the BLTouch probe was removed to remove references to other printers:

```
//#define BLTOUCH
```

If statements around the nozzle-to-probe offset were removed to remove references to other printers. While an offset is specified, there is no actual probe on the printer:

```
#define NOZZLE_TO_PROBE_OFFSET { 0, 0, -1.102 } // there is no probe
    currently
```

The probing margins were arbitrarily changed to 10 since the defined probing margins of -9 only works if the nozzle is being used as a probe since the code is intended to prevent the nozzle from moving outside of the print area. The probe safe point was adjusted arbitrarily to the center of the bed rather than -1, -9 for a similar reason:

```
#define PROBING_MARGIN 10 // -9
#define PROBE_SAFE_POINT { X_CENTER, Y_CENTER } // { -1, -9 } //safe probe
    point for single extruder toolheads on Pro/XT
```

Since probing is no longer automatic, the repeatability test feature to test probe accuracy was turned off via commenting:

```
//#define Z_MIN_PROBE_REPEATABILITY_TEST
```

The stepper directions for driving the axes were specified based on how the physical system behaves to remove references to other printers:

```
#define INVERT_X_DIR false
#define INVERT_Y_DIR true
#define INVERT_Z_DIR false
```

The stepper direction for the syringe pump motor is reversed from how the Lulzbot extruders behaved based on the design of the syringe pump:

```
#define INVERT_E0_DIR false // true
#define INVERT_E1_DIR false // true
```

The home directions for the y- and z-axes were specified to remove references to other printers:

```
#define Y_HOME_DIR 1
#define Z_HOME_DIR 1
```

The printable areas for beds other than the TAZ Pro were removed and travel limits were adjusted for the NIST tool head:

```
#define X_MAX_POS (LULZBOT_X_MAX_POS + NIST_TOOLHEAD_X_MAX_ADJ)
#define X_MIN_POS (LULZBOT_X_MIN_POS + NIST_TOOLHEAD_X_MIN_ADJ)
#define Y_MAX_POS (LULZBOT_Y_MAX_POS + NIST_TOOLHEAD_Y_MAX_ADJ)
#define Y_MIN_POS (LULZBOT_Y_MIN_POS + NIST_TOOLHEAD_Y_MIN_ADJ)
#define Z_MAX_POS (LULZBOT_Z_MAX_POS + NIST_TOOLHEAD_Z_MAX_ADJ)
#define Z_MIN_POS (LULZBOT_Z_MIN_POS + NIST_TOOLHEAD_Z_MIN_ADJ)
```

The filament runout sensor feature was turned off via commenting out the code since the syringe pump does not have integrated sensing to detect when it is out of material; although the code has been adjusted to support that feature if runout sensing is added in later:

```
/*#define FILAMENT_RUNOUT_SENSOR
#endif FILAMENT_RUNOUT_SENSOR)
#define FIL_RUNOUT_ENABLED_DEFAULT true // Enable the sensor on startup.
// Override with M412 followed by M500.
#define NUM_RUNOUT_SENSORS NIST_EXTRUDERS // Number of sensors,
// up to one per extruder. Define a FIL_RUNOUT#_PIN for each.
```

Since probing is no longer automatic, bed leveling was turned off. The code specifies mesh bed leveling which is technically possible for manual probing; however, there is no pre-built manual leveling menu for the LCD used on the TAZ Pro, so there is currently no easy way to use this feature:

```
/*#define AUTO_BED_LEVELING_BILINEAR
#define MESH_BED_LEVELING
/*#define RESTORE_LEVELING_AFTER_G28
/*#define DEBUG_LEVELING_FEATURE
```

If statements around the leveling removed to remove references to other printers:

```
#define GRID_MAX_POINTS_X 2 //2x2 grid of mounted washers
/*#define PROBE_Y_FIRST
/*#define LCD_BED_LEVELING
/*#define Z_SAFE_HOMING
#define Z_SAFE_HOMING_X_POINT (X_CENTER) // X point for Z homing
#define Z_SAFE_HOMING_Y_POINT (Y_BED_SIZE/2) // Y point for Z homing
```

If statements around homing speeds were removed to remove references to other printers:

```
#define HOMING_FEEDRATE_Z 1800
```

If statements around nozzle parking were removed to remove references to other printers:

```
#define NOZZLE_PARK_POINT { X_CENTER, (Y_MAX_POS - 5), (Z_MAX_POS/2) }
```

The nozzle clean feature has been commented out because it is not used for printing soft materials since the nozzle will need to manually cleaned after each print:

```
//#define NOZZLE_CLEAN_FEATURE
```

If statements around the encoder direction were removed to remove references to other printers:

```
//#define REVERSE_ENCODER_DIRECTION
```

If statements around the LCD interface were adjusted to remove references to other printers:

```
//#define REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER
#define TOUCH_UI_FTDI_EVE
```

If statements around fans were adjusted to remove references to other printers:

```
//#define FAST_PWM_FAN
#define FAN_SOFT_PWM
```

2.2. marlin-firmware\Marlin\Configuration_adv.h

The original Marlin documentation for this file is available at https://marlinfw.org/docs/configuration/configuration.html#configuration_adv.h. Although the changes made were to the Lulzbot version of this file (found in their GitLab under the Marlin folder), there is no additional documentation available through Lulzbot. The original Marlin documentation should provide sufficient guidance to understand what is contained in this file.

If statements around the fan control pin were removed to remove references to other printers:

```
#define CONTROLLER_FAN_PIN FAN1_PIN
```

If statements around sensorless backoff were removed to remove references to other printers:

```
#define SENSORLESS_BACKOFF_MM { 4, 4, 0} // (mm) Backoff from endstops
before sensorless homing
```

If statements around stepper shutdowns for inactivity were removed to remove references to other printers:

```
#define DISABLE_INACTIVE_Y true
#define DISABLE_INACTIVE_Z false
```

If statements around backlash compensation were removed to remove references to other printers:

```
#define BACKLASH_DISTANCE_MM {0.252, 0.183, 0.075} // (mm)
#define BACKLASH_CORRECTION 1.0 // 0.0 = no correction; 1.0 = full
correction
```

The if statement around calibration gcode was removed to remove references to other printers:

```
#define CALIBRATION_GCODE
```

The calibration specifications were adjusted to remove references to other printers:

```

#if defined(TOOLHEAD_Quiver_DualExtruder)
  #define CALIBRATION_OBJECT_CENTER {261.5, -18, -2.0} // mm
  #define CALIBRATION_OBJECT_DIMENSIONS {10.0, 10.0, 10.0} // mm
  #define CALIBRATION_MEASURE_FRONT
#else
  #define CALIBRATION_OBJECT_CENTER {267.5, -9.5, -2.0} // mm
  #define CALIBRATION_OBJECT_DIMENSIONS {10.0, 1.0, 10.0} // mm
#endif

```

The motor current for the second extruder is set to 0mA if there is no second extruder:

```

#if !defined(NIST_MOTOR_CURRENT_E1)
  #define NIST_MOTOR_CURRENT_E1 0
#endif

```

The speed at which the syringe pump motor runs during manual control through the printer's LCD menu was increased:

```

#define MANUAL_FEEDRATE { 50*60, 50*60, 50*60, 10*60 } // (mm/min) Feedrates
  for manual moves along X, Y, Z, E from panel - increase extruder speed
  from 2*60
// 50*60 extruder speed is too fast, 10*60 is fine but skips at beginning (
  adjusted max acceleration), 20*60 unhappy motor sounds

```

The if statement around the SD card speed was removed to remove references to other printers:

```
#define SD_SPI_SPEED SPI_HALF_SPEED
```

The if statement around USB support was removed to remove references to other printers:

```
#define USB_FLASH_DRIVE_SUPPORT
```

The if statement around MarlinUI U8GLIB was adjusted to remove references to other printers:

```
#if ENABLED(HAS_MARLINUI_U8GLIB)
```

Babystepping for the z-probe is turned off because there is no automatic probe anymore and this feature is incompatible with a manual probe:

```
//#define BABYSTEP_ZPROBE_OFFSET
//#define BABYSTEP_ZPROBE_GFX_OVERLAY
```

The G29 command is turned off because the printer is no longer able to perform automatic bed leveling and if statements pertaining to other printers were removed:

```
//#define G29_RETRY_AND_RECOVER
```

The if statement around z-axis microsteps were removed to remove references to other printers:

```
#define Z_MICROSTEPS 16
```

The currents for the syringe pump and second theoretical syringe pump were specified as:

```
#define E0_CURRENT      NIST_MOTOR_CURRENT_E0
#define E1_CURRENT      NIST_MOTOR_CURRENT_E1
```

The if statement around SPI communication were removed to remove references to other printer:

```
#define TMC_USE_SW_SPI
```

If statements around x- and y-axes stall sensitivities were removed to remove reference to other printers.

The start up command for the printer was specified to home all axes via:

```
#define STARTUP_COMMANDS "G28"
```

If statements are PID gains and menu items were removed to remove reference to other printers.

If statements around mechanical gantry calibration were adjusted to remove reference to other printers:

```
//#define MECHANICAL_GANTRY_CALIBRATION
#if ENABLED(MECHANICAL_GANTRY_CALIBRATION)
#define GANTRY_CALIBRATION_TO_MIN
#define GANTRY_CALIBRATION_SAFE_POSITION {150, 192} // Safe position for
nozzle
#define GANTRY_CALIBRATION_EXTRA_HEIGHT -10 // Extra distance in mm
past Z_###_POS to move
```

The if statement around x-axis leveling was removed to remove reference to other printers:

```
#define X_LEVEL_SEQUENCE
```

2.3. marlin-firmware\Marlin\Version.h

Marlin developers are currently working on creating documentation for the code (<https://github.com/MarlinFirmware/MarlinDocumentation>). There is currently no existing documentation on this particular file.

Information about the versions have been changed to be specific to the printer:

```
#define SHORT_BUILD_VERSION NIST_VERSION
#define DETAILED_BUILD_VERSION " FIRMWARE_VERSION:" SHORT_BUILD_VERSION "
EXTRUDER_TYPE:" NIST_M115_EXTRUDER_TYPE
```

The distribution date has been adjusted to the release date:

```
#define STRING_DISTRIBUTION_DATE "2022-09-24"
```

The machine name and uuid is specified for the printer:

```
\textbf{\#define MACHINE_NAME NIST_CUSTOM_MACHINE_NAME}
#define DEFAULT_MACHINE_UUID NIST_MACHINE_UUID
```

The source code url and website url points to the appropriate github directory:

```
#define SOURCE_CODE_URL "https://github.com/jcase-robots/exploratory-soft-
printer"
#define WEBSITE_URL "https://github.com/jcase-robots/exploratory-soft-
printer"
```

The if statement for the USB device is removed to remove reference to other printers:

```
#define USB_DEVICE_VENDOR_ID 0x27b1
#define USB_DEVICE_PRODUCT_ID 0x0001
```

2.4. marlin-firmware\Marlin__Bootscreen.h

Marlin developers are currently working on creating documentation for the code (<https://github.com/MarlinFirmware/MarlinDocumentation>). There is currently no existing documentation on this particular file.

This file is used to store the bootscreen image, but only for specific LCD displays. It has been adjusted to show "NIST"; however, the LCD display for this printer does not reference this file. Since the changed code is just 64 lines of hex and the file is not used by the firmware itself, the altered code is not included in this documentation.

2.5. marlin-firmware\Marlin\src\lcd\extui\ftdi_eve_touch_ui\generic\leveling_menu.cpp

Marlin developers are currently working on creating documentation for the code (<https://github.com/MarlinFirmware/MarlinDocumentation>). There is currently no existing documentation on this particular file.

The leveling menu for the printer's LCD was reverted back to the original Marlin code in an attempt to enable mesh bed leveling. It was unsuccessful and additional work is needed:

```
#define TEST_MESH_POS      BTN_POS(2,4), BTN_SIZE(1,1)
//#define M48_TEST_POS      BTN_POS(2,4), BTN_SIZE(1,1)

.enabled(EITHER(Z_STEPPER_AUTO_ALIGN, MECHANICAL_GANTRY_CALIBRATION))
.tag(2).button(LEVEL_AXIS_POS, GET_TEXT_F(MSG_LEVEL_X_AXIS))
.enabled(ENABLED(HAS_BED_PROBE))
.tag(3).button(PROBE_BED_POS, GET_TEXT_F(MSG_PROBE_BED))
.enabled(ENABLED(HAS_MESH))
.tag(4).button(SHOW_MESH_POS, GET_TEXT_F(MSG_MESH_VIEW))
.enabled(ENABLED(HAS_MESH))
.tag(5).button(EDIT_MESH_POS, GET_TEXT_F(MSG_EDIT_MESH))
.enabled(ENABLED(G26_MESH_VALIDATION))
.tag(6).button(TEST_MESH_POS, GET_TEXT_F(MSG_PRINT_TEST))
//.enabled(ANY(Z_STEPPER_AUTO_ALIGN, MECHANICAL_GANTRY_CALIBRATION,
//  X_LEVEL_SEQUENCE))
//.tag(2).button(LEVEL_AXIS_POS, GET_TEXT_F(MSG_LEVEL_X_AXIS))
//.tag(3).button(PROBE_BED_POS, GET_TEXT_F(MSG_PROBE_BED))
//.enabled(ENABLED(Z_MIN_PROBE_REPEATABILITY_TEST))           //Taking over mesh
//  view option due to using Bilinear leveling and not being able to edit
//  the mesh
//.tag(4).button(M48_TEST_POS, GET_TEXT_F(MSG_M48_TEST))
//#if DISABLED(AUTO_BED_LEVELING_BILINEAR)
//  .enabled(ENABLED(HAS_MESH))
//  .tag(5).button(EDIT_MESH_POS, GET_TEXT_F(MSG_EDIT_MESH))
//  .enabled(ENABLED(G26_MESH_VALIDATION))
//  .tag(6).button(TEST_MESH_POS, GET_TEXT_F(MSG_PRINT_TEST))
//#endif
```

2.6. marlin-firmware\Marlin\src\lcd\language\language_en.h

Marlin developers are currently working on creating documentation for the code (<https://github.com/MarlinFirmware/MarlinDocumentation>). There is currently no existing documentation on this particular file.

When adding the leveling menu back into the code, it was necessary to add a couple of additional definitions in this file:

```
PROGMEM Language_Str MSG_MESH_VIEWER = _UxGT("Mesh Viewer");
PROGMEM Language_Str MSG_MESH_VIEW = _UxGT("View Mesh");
```

3. Compiling and Uploading Firmware to Printer

3.1. Compiling the Firmware

This procedure assumes you have followed the requirements documented in Section 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.

3.2. Uploading the Firmware

This procedure assumes you have followed the requirements documented in Section 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”).

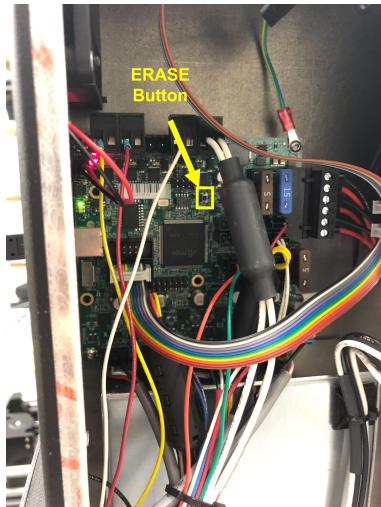


Fig. 1. Location of the ERASE button.

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.

3.2.1. 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]` section¹:

¹https://docs.platformio.org/en/latest/projectconf/section_env_upload.html

```
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.

References

- [1] R. L. Truby and J. A. Lewis, “Printing soft matter in three dimensions,” *Nature*, vol. 540, no. 7633, pp. 371–378, 2016.
- [2] J. Z. Gul, M. Sajid, M. M. Rehman, G. U. Siddiqui, I. Shah, K.-H. Kim, J.-W. Lee, and K. H. Choi, “3d printing for soft robotics—a review,” *Science and technology of advanced materials*, vol. 19, no. 1, pp. 243–262, 2018.
- [3] Y. L. Yap, S. L. Sing, and W. Y. Yeong, “A review of 3d printing processes and materials for soft robotics,” *Rapid Prototyping Journal*, 2020.
- [4] S. Walker, U. Daalkhaijav, D. Thrush, C. Branyan, O. D. Yirmibesoglu, G. Olson, and Y. Menguc, “Zero-support 3d printing of thermoset silicone via simultaneous control of both reaction kinetics and transient rheology,” *3D Printing and Additive Manufacturing*, vol. 6, no. 3, pp. 139–147, 2019.