

INTRODUCTION:

This is a basic user guide on how to verify and/or modify the settings in the Marlin firmware for use with the RepRap SmartRap 3D printer. There are several settings not covered in this manual and those will be reserved for another manual that can be used after the RepRap SmartRap 3D printer is built and ready for basic testing and calibration.

As with most of my detailed r/c projects, I like to create a user guide before I create any videos. In this manner I can discover what to do and what not to do before making a major mistake resulting in magic white smoke.

Please note that you can load, view and modify the Marlin firmware without first flashing it to the Arduino Mega 2560. It's the same as an author writing a book. The proof reading and editing is done before it's sent to a publisher.

Special thanks to Thomas Sanladerer and SolidUtopia for posting their how to guides.

ARDUINO IDE:

The Arduino IDE is used to view, modify and flash the Marlin firmware. It can be obtained for free from <http://www.arduino.cc/en/Main/Software>

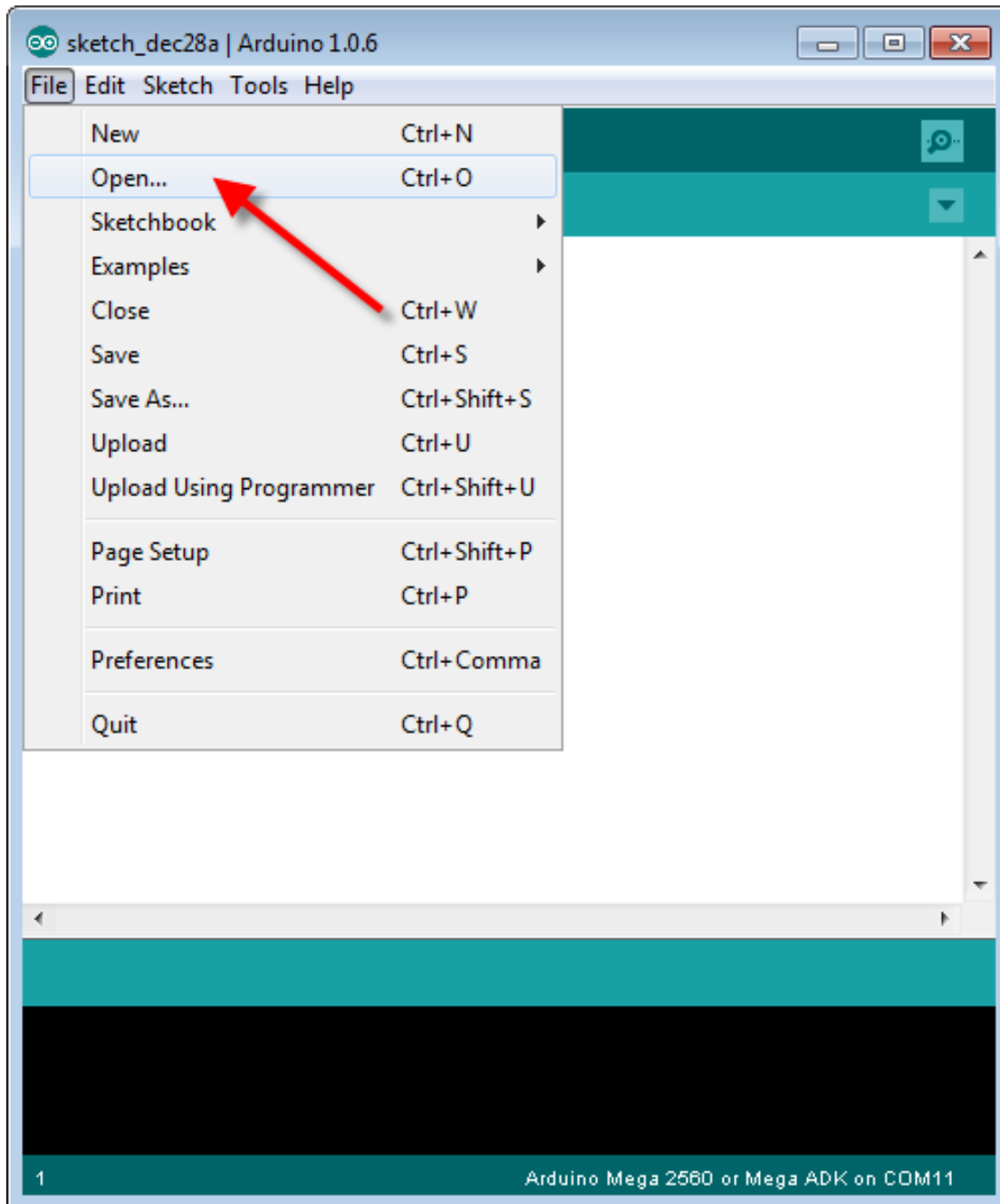
MARLIN FIRMWARE:

The Marlin firmware version specific for the RepRap SmartRap 3D printer can be downloaded for free from the SmartFriendz GitHub site <https://github.com/smartfriendz/smartrap>.

This version of the firmware has most of the settings defined already but it is beneficial to get familiar with the settings and how to change them.

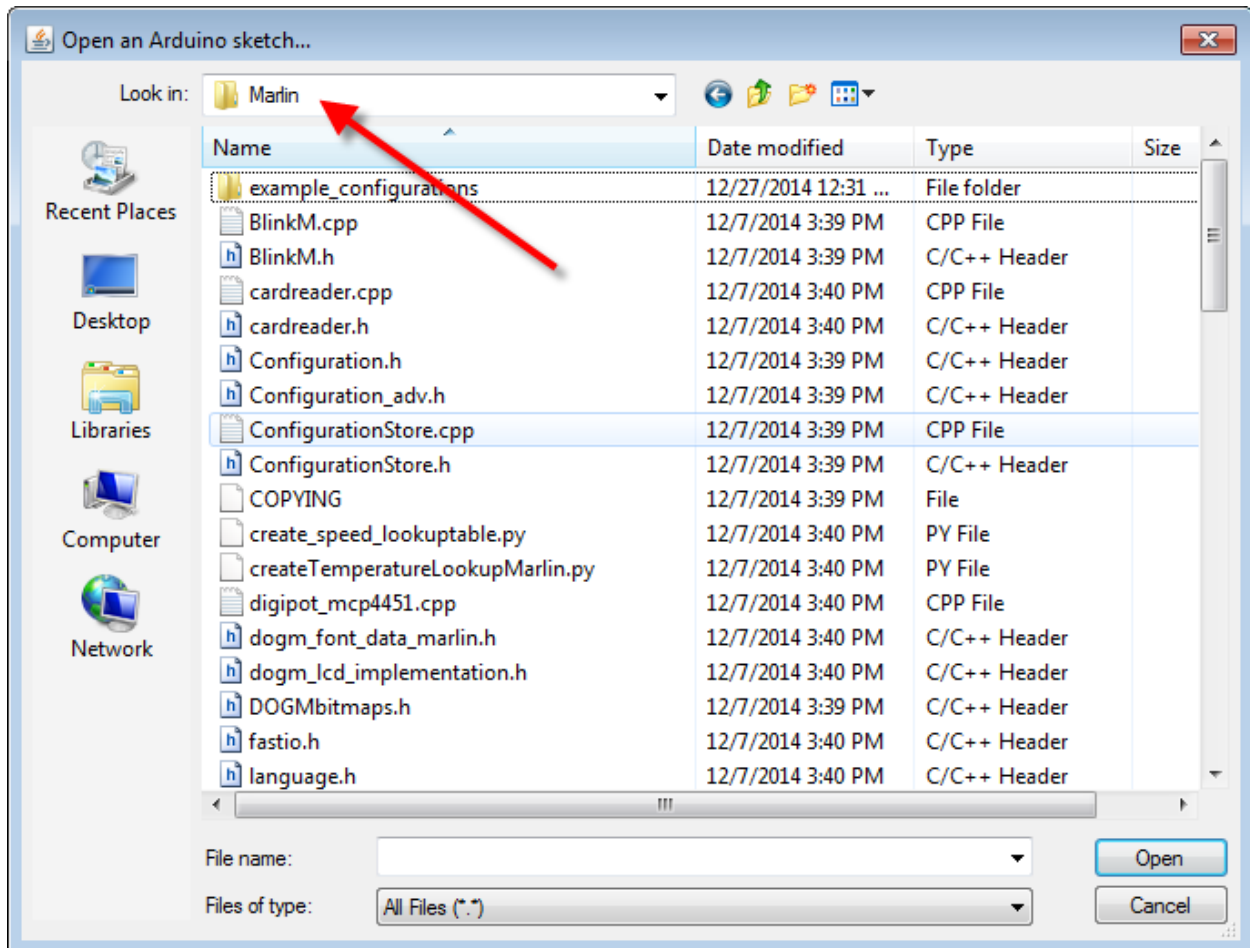
LOADING THE FIRMWARE:

- Start the Arduino IDE
- Click on File and then Open



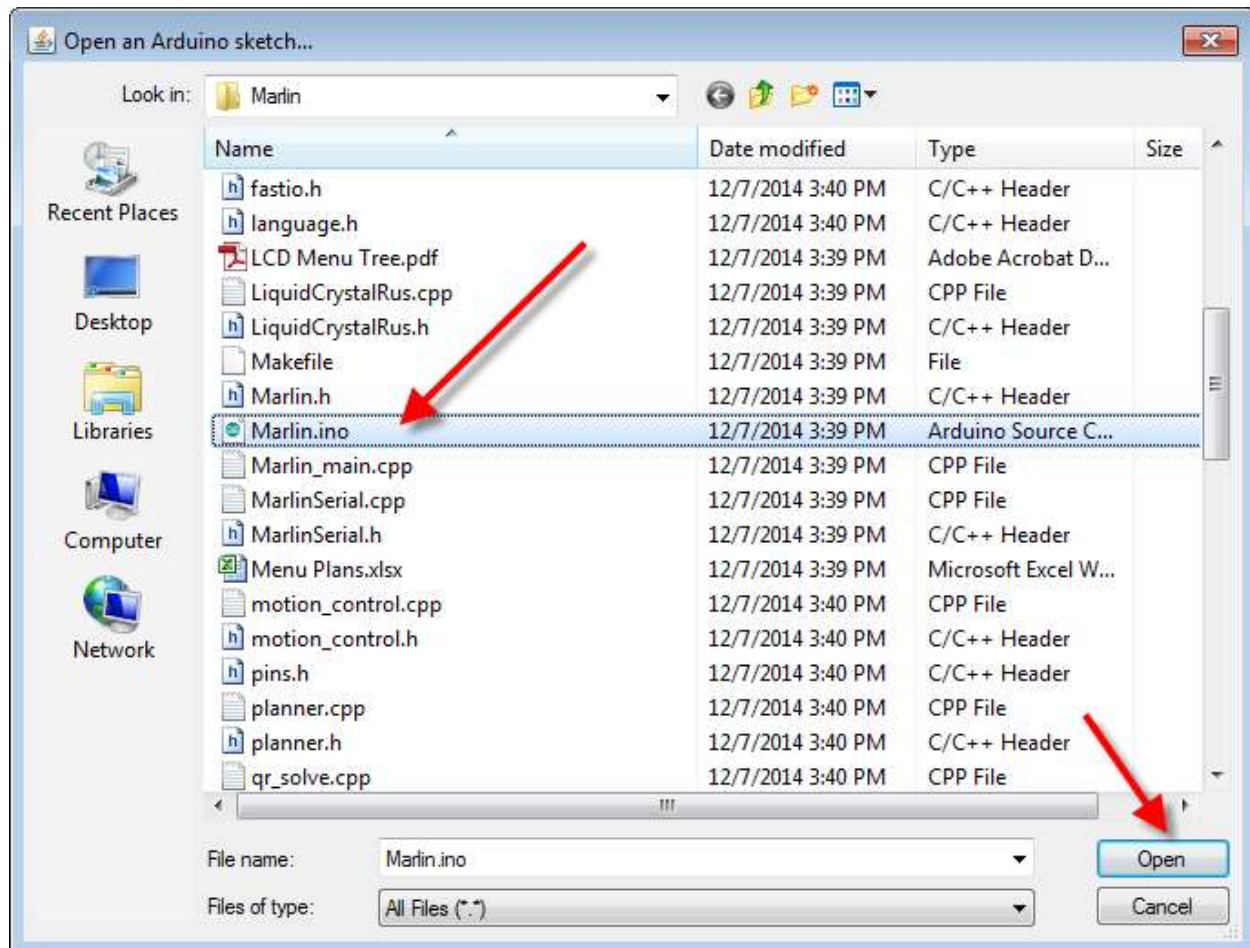
BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

- Browse to the Marlin folder where the firmware files are stored. Please note that the firmware files must be stored in a folder named Marlin (not case sensitive) or else you will get a warning message when trying to load the firmware.

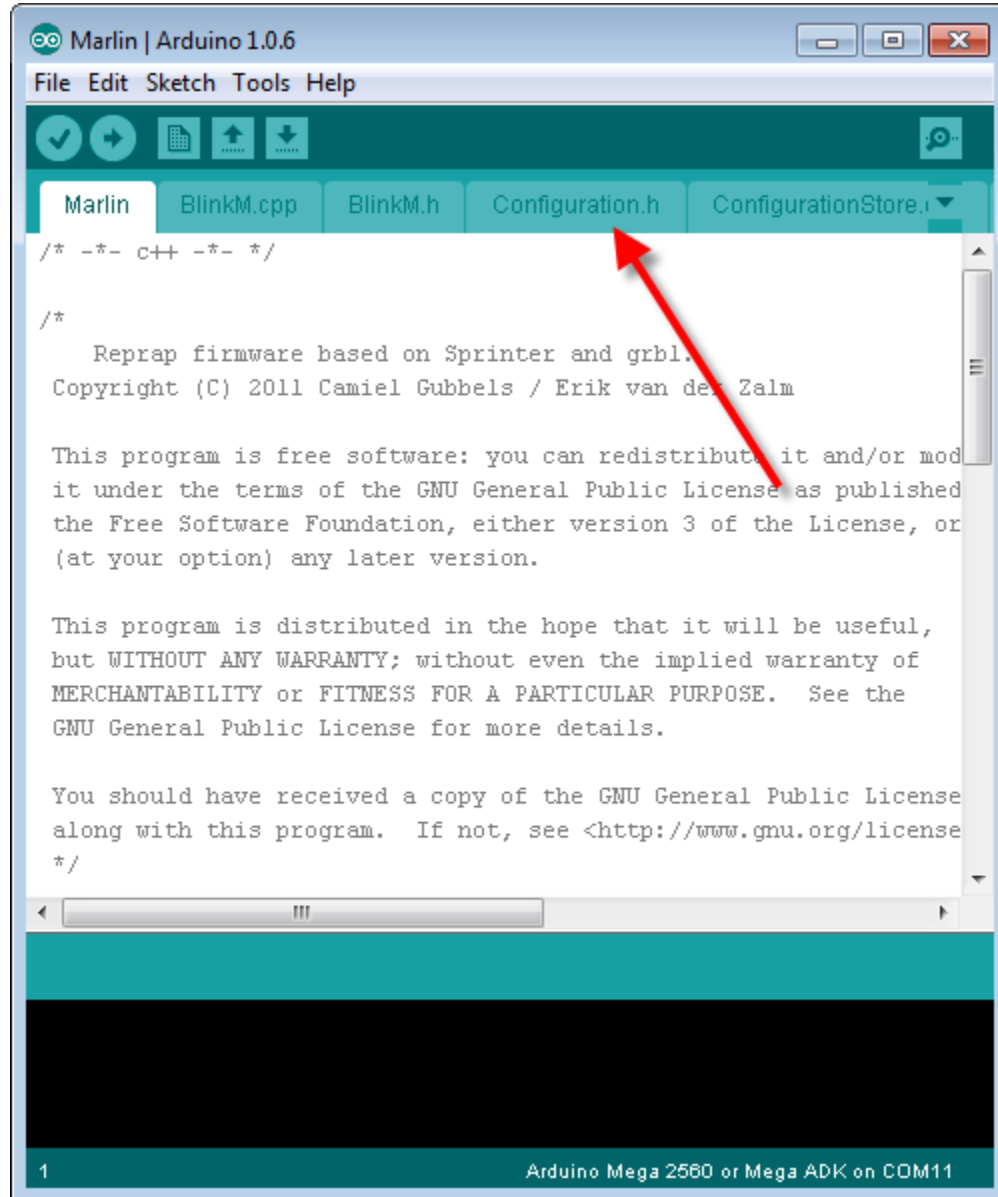


BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

- Locate the Marlin.ino file
- Click on it once
- Click on the Open button



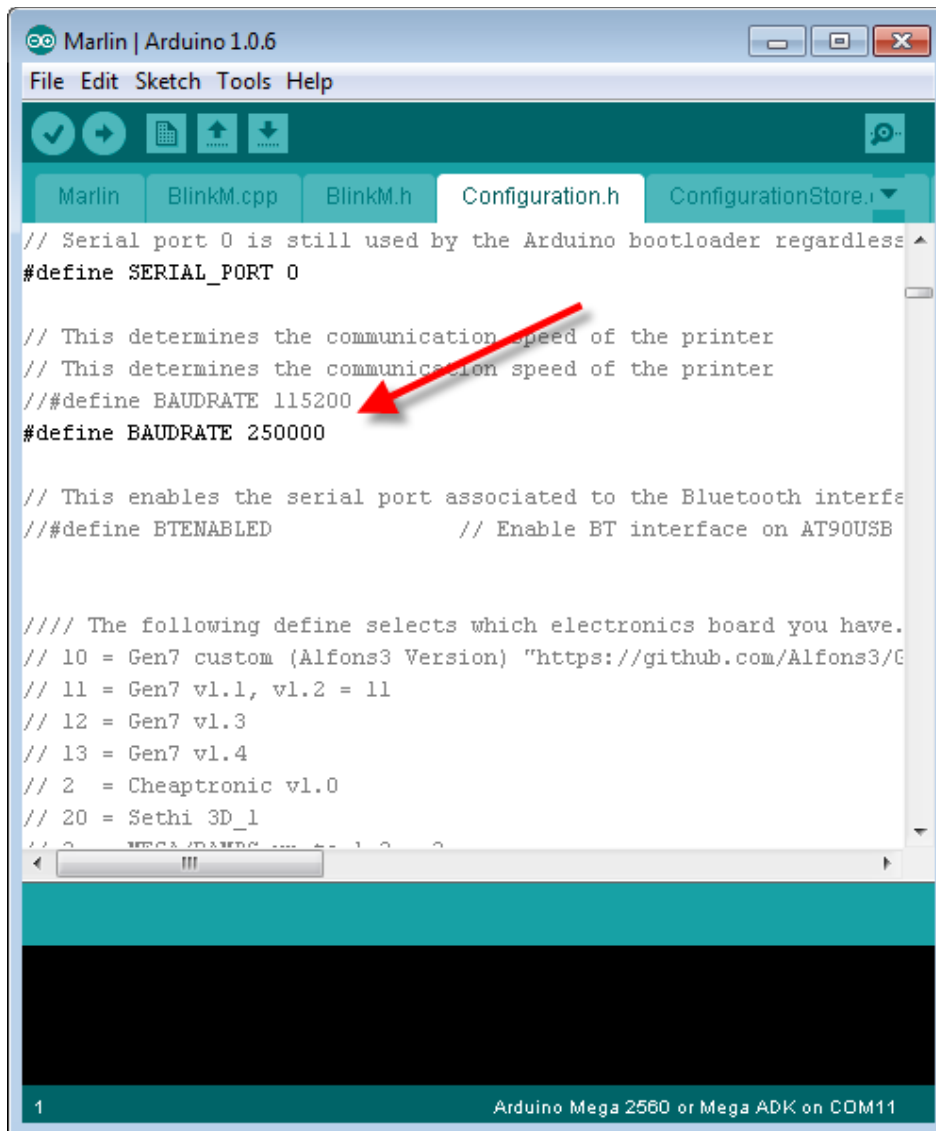
- Click on the Configuration.h tab



ARDUINO BAUD RATE:

This setting defines the speed at which the RepRap SmartRap 3D printer will communicate with your computer's USB port. The default setting is 250000 but can be changed to 115200 if your computer is having issues communicating with the RepRap SmartRap 3D printer.

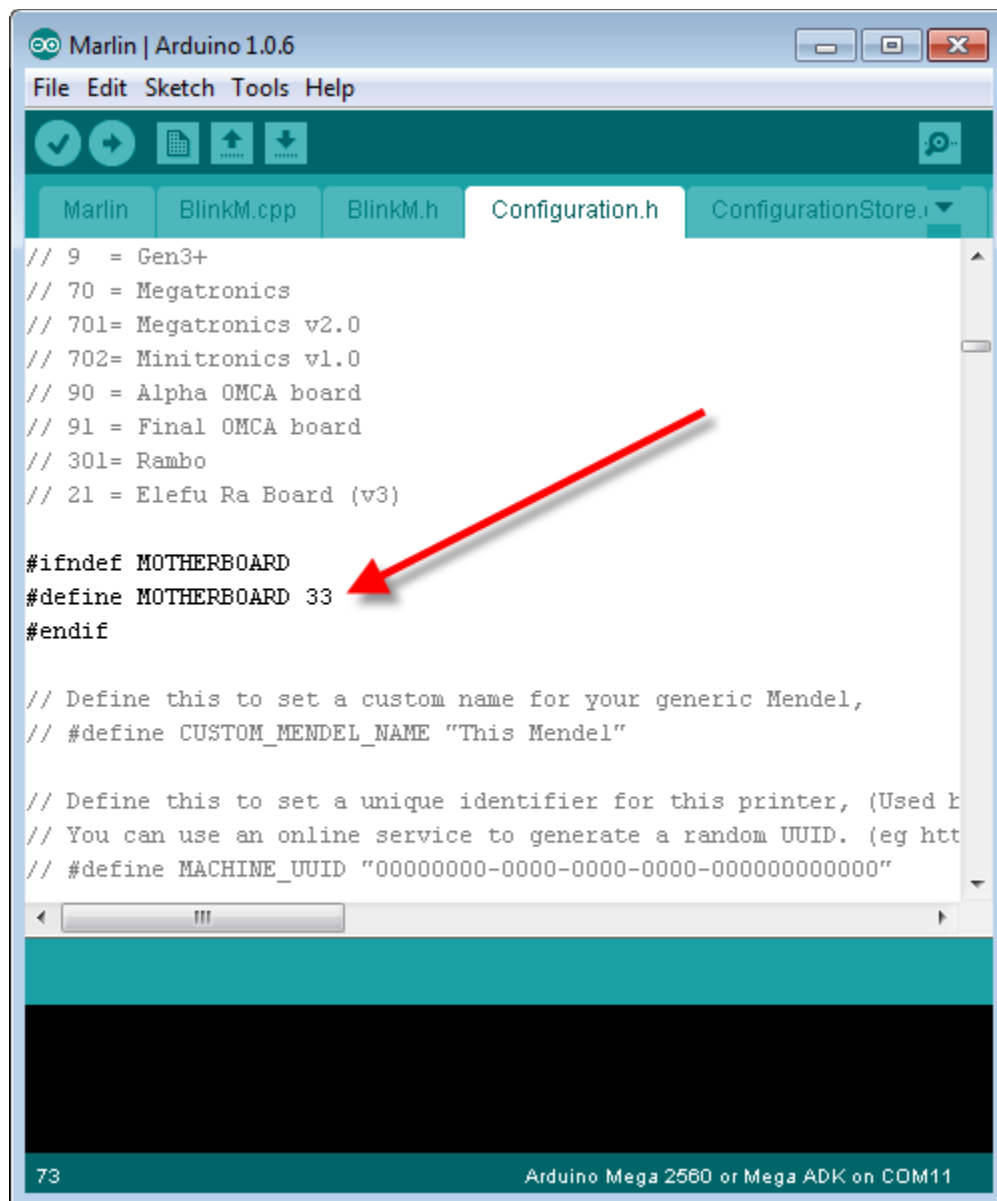
- Use the scroll bar to locate the line `#define BAUDRATE`
- If you wish to utilize the default baud rate leave it as is
- If you wish to change the baud rate to 115200
 - Remove the two slashes in front of the line `#define BAUDRATE 115200`
 - Add two dashes in front of the line `#define BAUDRATE 250000`



MOTHERBOARD:

This setting defines which RAMPS board is being used. The default setting is 33, which is for a RAMPS V1.3/V1.4 board with power outputs for an extruder, fan and heated bed. This setting can be used regardless if you are/are not using a heated bed or fan.

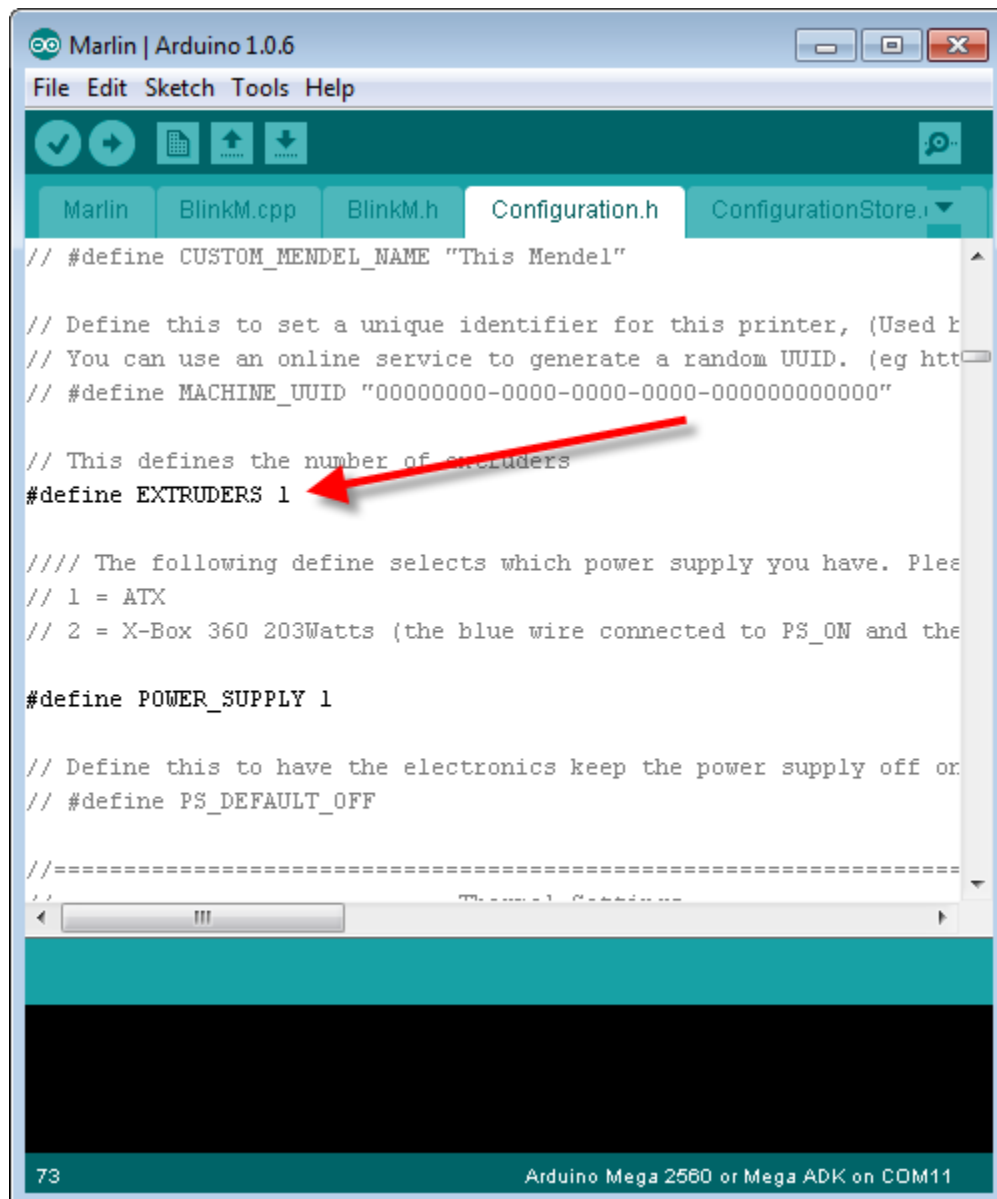
- Use the scroll bar to locate the line #define MOTHERBOARD
- If you wish to utilize the default setting leave it as is
- If you wish to change the setting to match your particular board
 - Use the board listing above the setting to determine the board's value
 - Change the 33 value to match that of your board



NUMBER OF EXTRUDERS:

This setting defines the number of extruders being used. The RepRap SmartRap 3D printer only uses one extruder so the default setting is 1.

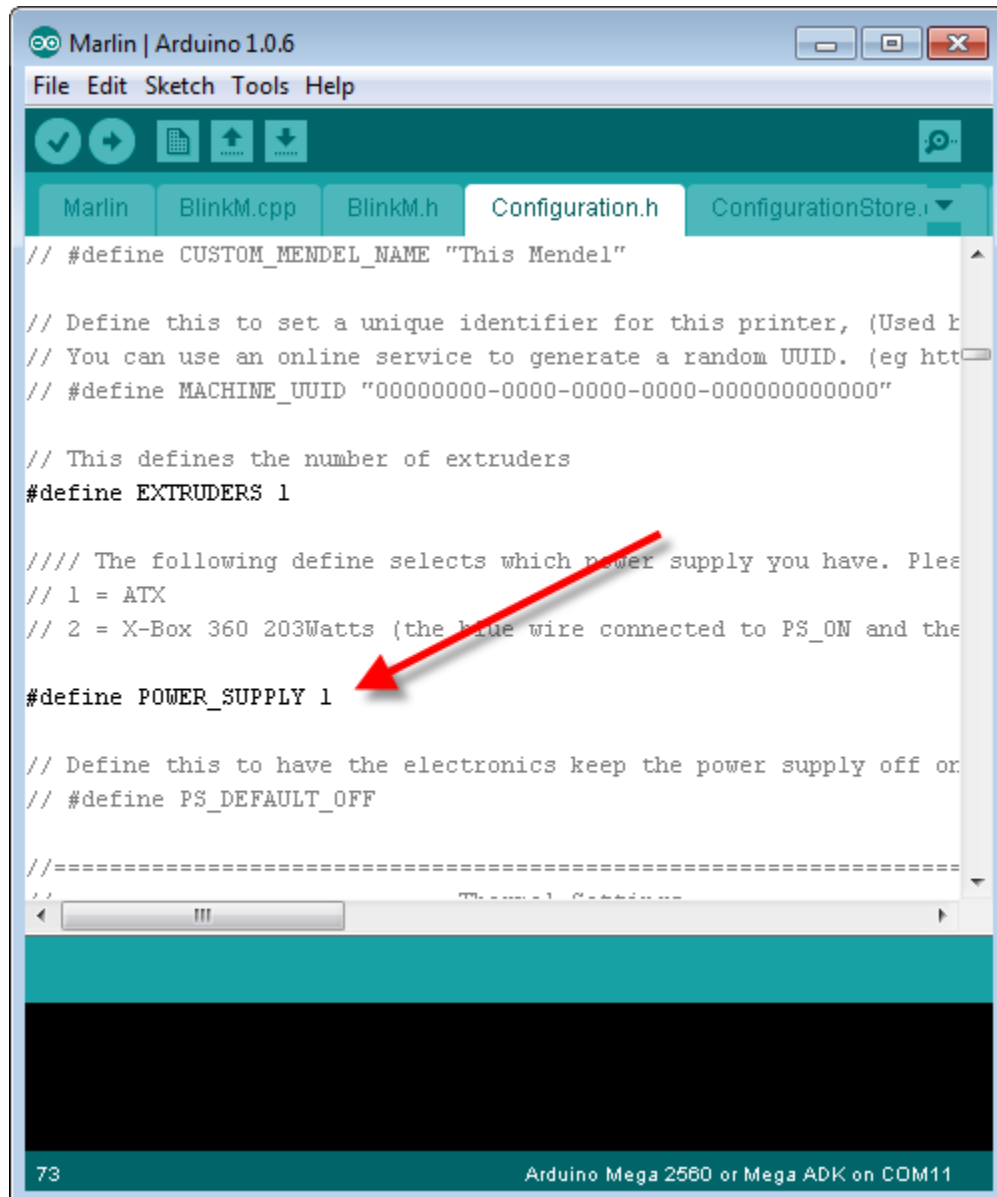
- Use the scroll bar to locate the line #define EXTRUDERS
- If you wish to utilize the default setting leave it as is
- If you wish to change the setting
 - Change the 1 to the number of extruders being used, i.e. 2 for 2 extruders



POWER SUPPLY:

This setting defines what type of power supply is being used. The default setting is 1 which represents a typical ATX power supply. Only change this setting if you are using an XBOX 360 power supply.

- Use the scroll bar to locate the line `#define POWER_SUPPLY`
- If you wish to utilize the default setting leave it as is
- If you wish to change the setting
 - Change the 1 to a 2, which is the only other option available



TEMPERATURE SENSOR SETTINGS:

These settings determine what type of temperature sensors are used for both the hot end and heated bed. It is very important that you review these settings and change them to match your setup.

#define TEMP_SENSOR 0 7

This defines the type of thermistor used for the first hotend. By default, the RepRap SmartRap 3D printer utilizes only one hotend and the default thermistor value is 7.

In order to change this setting you need to know what type of thermistor is utilized in your hotend. If you utilized the Parts Cost and Analysis spreadsheet from my blog and ordered a J-Head hotend from MakeFarm.COM, the thermistor value is 1.

#define TEMP_SENSOR 1 0

This defines the type of thermistor used for the second hotend. By default, the RepRap SmartRap 3D printer utilizes only one hotend so this setting is turned off.

#define TEMP_SENSOR 2 0

This defines the type of thermistor used for the third hotend. By default, the RepRap SmartRap 3D printer utilizes only one hotend so this setting will be turned off.

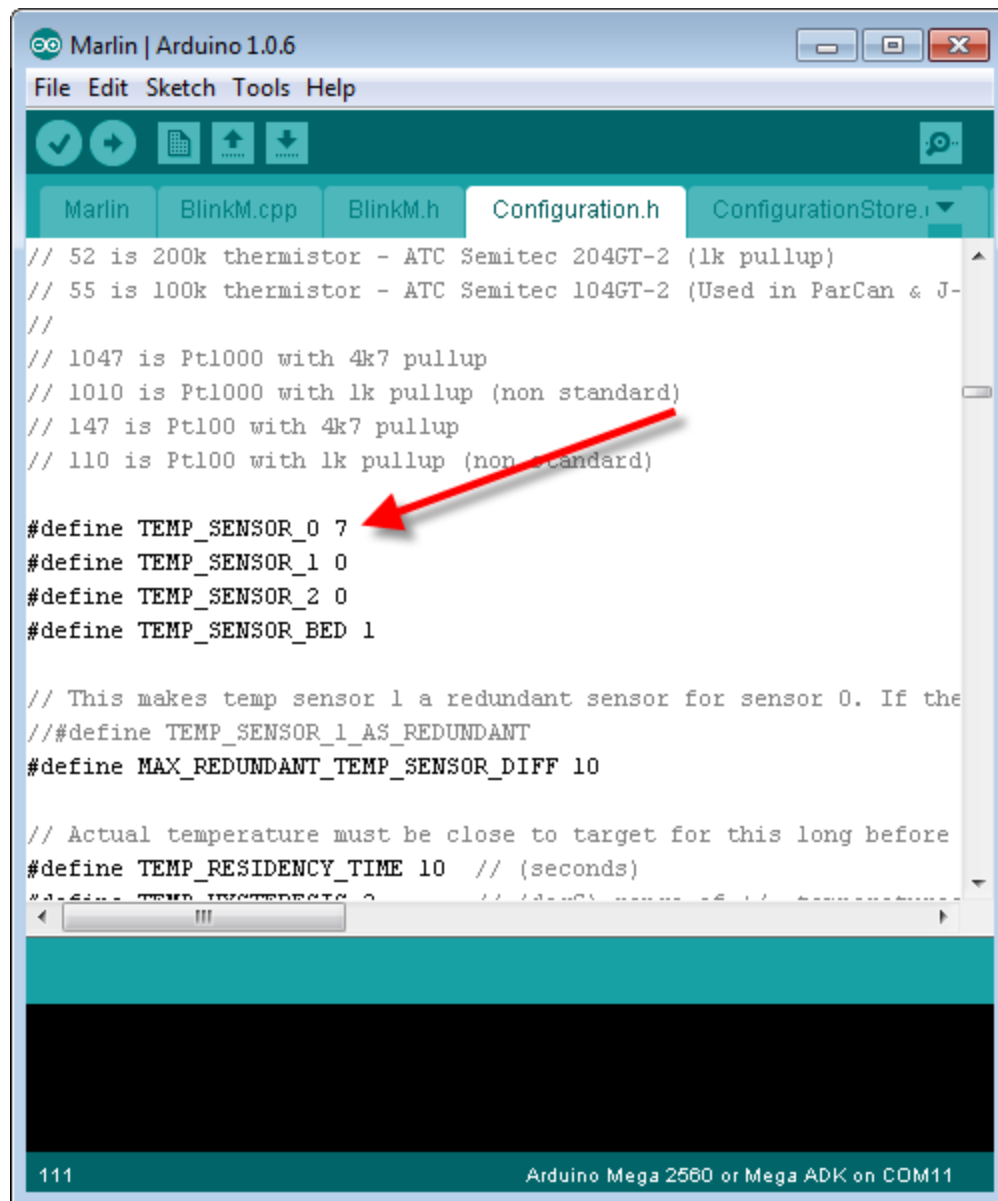
#define TEMP_SENSOR_BED 1

This defines the type of thermistor used for the heated bed. If you are not utilizing a heated bed, then this setting will be turned off.

In order to change this setting you need to know what type of thermistor is utilized in your heated bed.

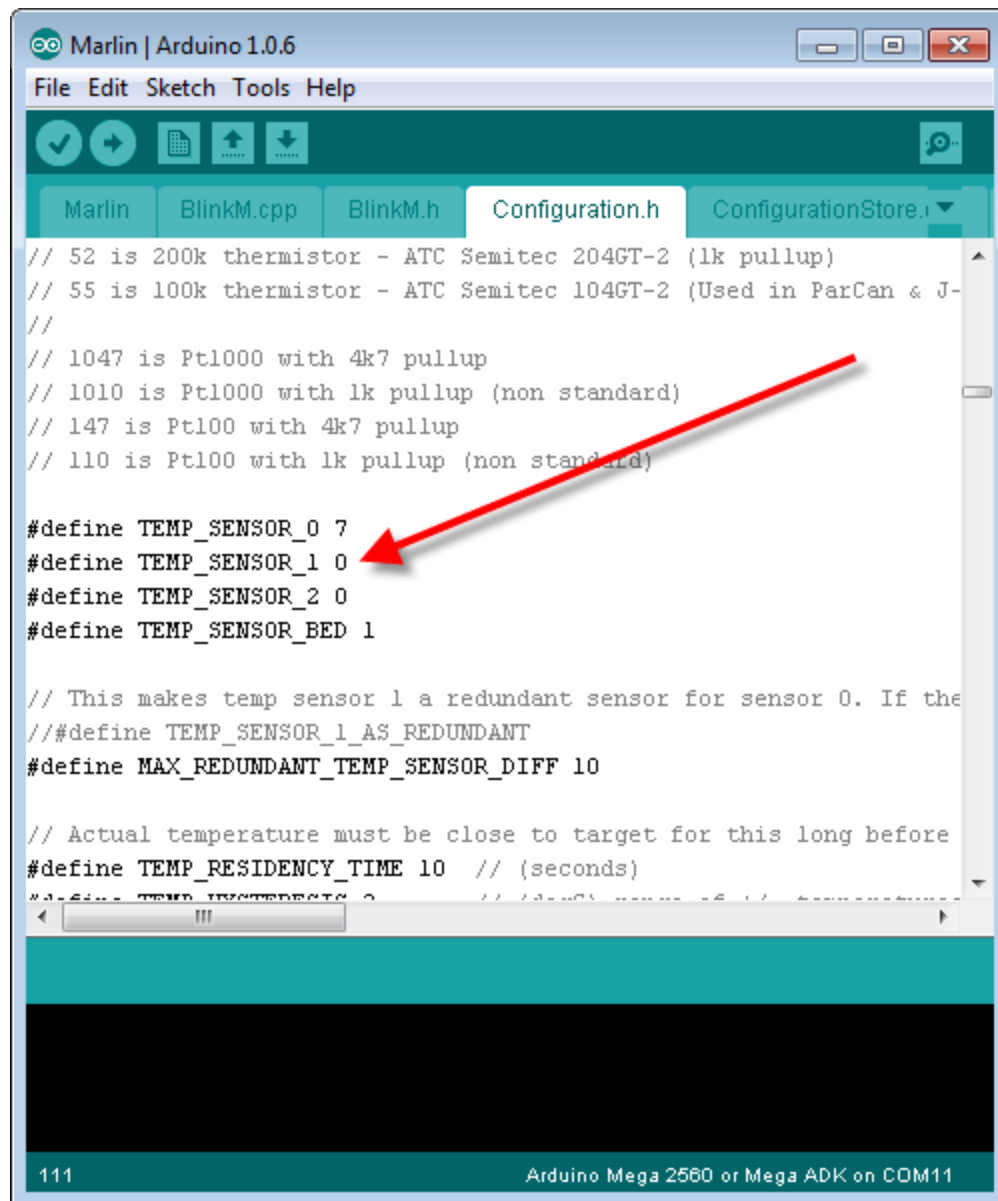
TEMP_SENSOR 0 7:

- Use the scroll bar to locate the line `#define TEMP_SENSOR_0 7`
- If using a MakerFarm.COM J-Head change the default value of 7 to 1
- If using another hotend
 - Use the thermistor listing above these setting to determine the thermistor's value
 - Change the 7 value to match that of your hotend



TEMP_SENSOR 1 0:

- Use the scroll bar to locate the line `#define TEMP_SENSOR 1 0`
- If not using as 2nd hotend, leave it as is
- If using a 2nd hotend
 - Use the thermistor listing above these setting to determine the thermistor's value
 - Change the 0 value to match that of your 2nd hotend



The screenshot shows the Marlin | Arduino 1.0.6 IDE with the Configuration.h file open. The file contains various thermistor definitions and sensor settings. A red arrow points to the line `#define TEMP_SENSOR_1 0`, which is part of a block of sensor definitions. The status bar at the bottom indicates '111' and 'Arduino Mega 2560 or Mega ADK on COM11'.

```

// 52 is 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
// 55 is 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-
//
// 1047 is Pt1000 with 4k7 pullup
// 1010 is Pt1000 with 1k pullup (non standard)
// 147 is Pt100 with 4k7 pullup
// 110 is Pt100 with 1k pullup (non standard)

#define TEMP_SENSOR_0 7
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_BED 1

// This makes temp sensor 1 a redundant sensor for sensor 0. If the
// #define TEMP_SENSOR_1_AS_REDUNDANT
#define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10

// Actual temperature must be close to target for this long before
#define TEMP_RESIDENCY_TIME 10 // (seconds)

```

TEMP_SENSOR 2 0:

- Use the scroll bar to locate the line `#define TEMP_SENSOR 2 0`
- If not using as third hotend, leave it as is
- If using a third hotend
 - Use the thermistor listing above these setting to determine the thermistor's value
 - Change the 0 value to match that of your third hotend

The screenshot shows the Marlin Arduino IDE interface. The 'Configuration.h' file is open, displaying various thermistor definitions and temperature sensor settings. A red arrow points to the line `#define TEMP_SENSOR_2 0`, which is the target for modification if a third hotend is being used.

```

// 52 is 200k thermistor - ATC Semitec 204GT-2 (1k pullup)
// 55 is 100k thermistor - ATC Semitec 104GT-2 (Used in ParCan & J-
//
// 1047 is Pt1000 with 4k7 pullup
// 1010 is Pt1000 with 1k pullup (non standard)
// 147 is Pt100 with 4k7 pullup
// 110 is Pt100 with 1k pullup (non standard)

#define TEMP_SENSOR_0 7
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_BED 1

// This makes temp sensor 1 a redundant sensor for sensor 0. If the
// #define TEMP_SENSOR_1_AS_REDUNDANT
#define MAX_REDUNDANT_TEMP_SENSOR_DIFF 10

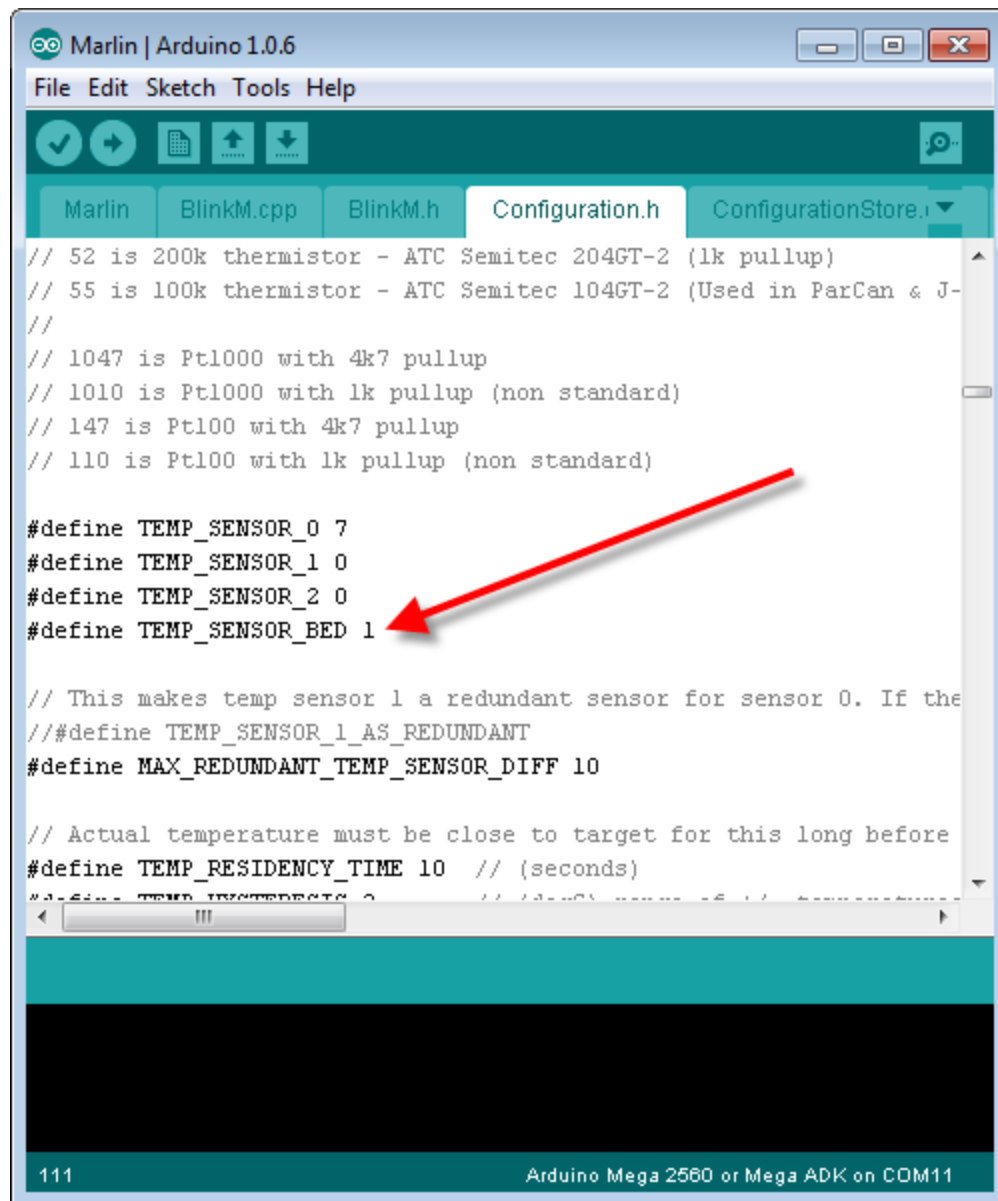
// Actual temperature must be close to target for this long before
#define TEMP_RESIDENCY_TIME 10 // (seconds)

```

At the bottom of the window, the status bar indicates '111' and 'Arduino Mega 2560 or Mega ADK on COM11'.

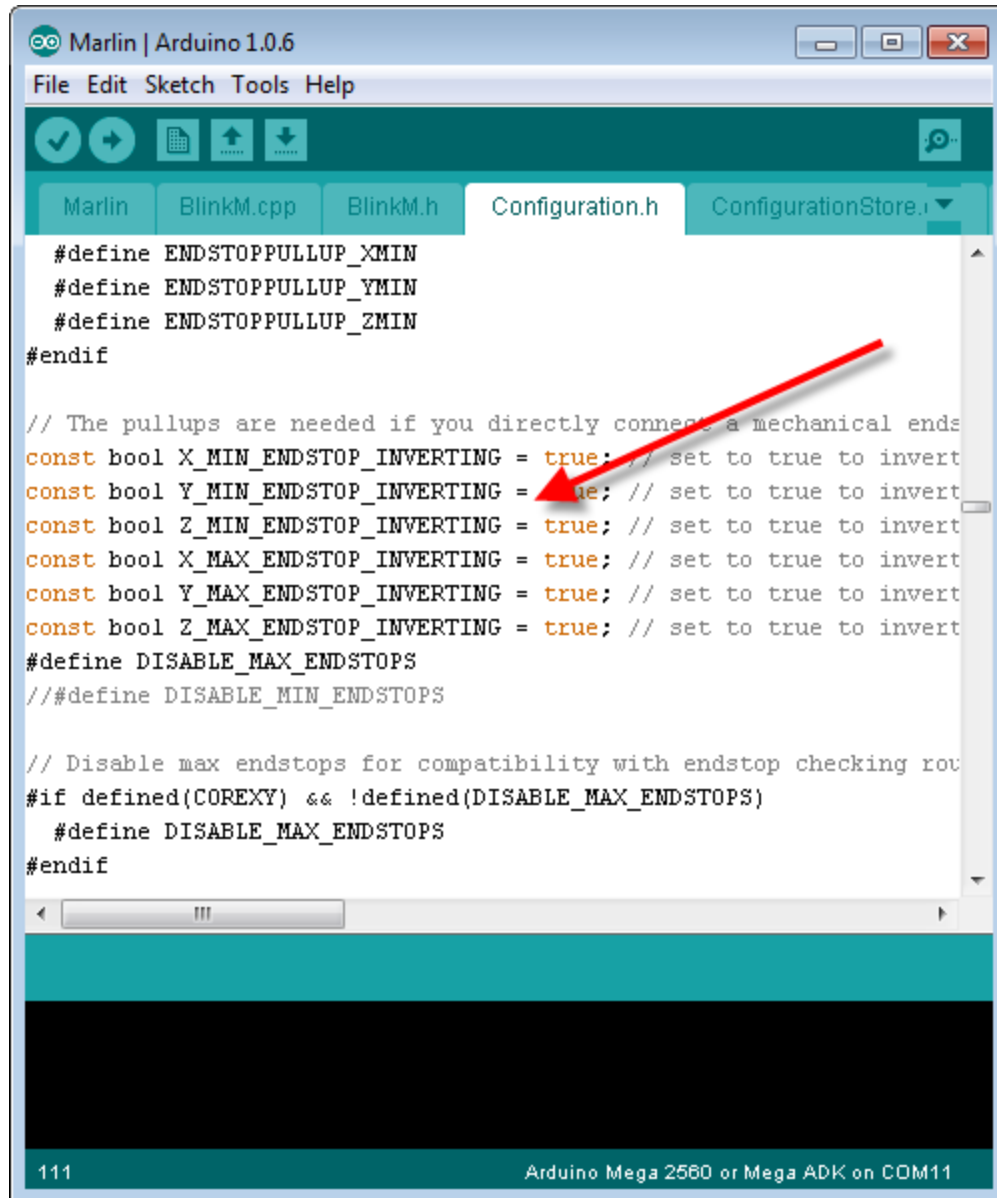
TEMP_SENSOR_BED 1:

- Use the scroll bar to locate the line `#define TEMP_SENSOR_BED 1`
- If not using a heated bed, change the 1 value to 0
- If using a heated bed
 - Use the thermistor listing above these setting to determine the thermistor's value
 - Change the 1 value to match that of your heated bed



LIMIT SWITCH/END STOP SWITCH LOGIC:

These settings determine if the switches are normally closed or normally open when not engaged. You will not know what their behavior is until they are tested so leave these settings alone for now.



```
Marlin | Arduino 1.0.6
File Edit Sketch Tools Help

Marlin BlinkM.cpp BlinkM.h Configuration.h ConfigurationStore.h

#define ENDSTOPPULLUP_XMIN
#define ENDSTOPPULLUP_YMIN
#define ENDSTOPPULLUP_ZMIN
#endif

// The pullups are needed if you directly connect a mechanical endstop
const bool X_MIN_ENDSTOP_INVERTING = true; // set to true to invert
const bool Y_MIN_ENDSTOP_INVERTING = true; // set to true to invert
const bool Z_MIN_ENDSTOP_INVERTING = true; // set to true to invert
const bool X_MAX_ENDSTOP_INVERTING = true; // set to true to invert
const bool Y_MAX_ENDSTOP_INVERTING = true; // set to true to invert
const bool Z_MAX_ENDSTOP_INVERTING = true; // set to true to invert
#define DISABLE_MAX_ENDSTOPS
// #define DISABLE_MIN_ENDSTOPS

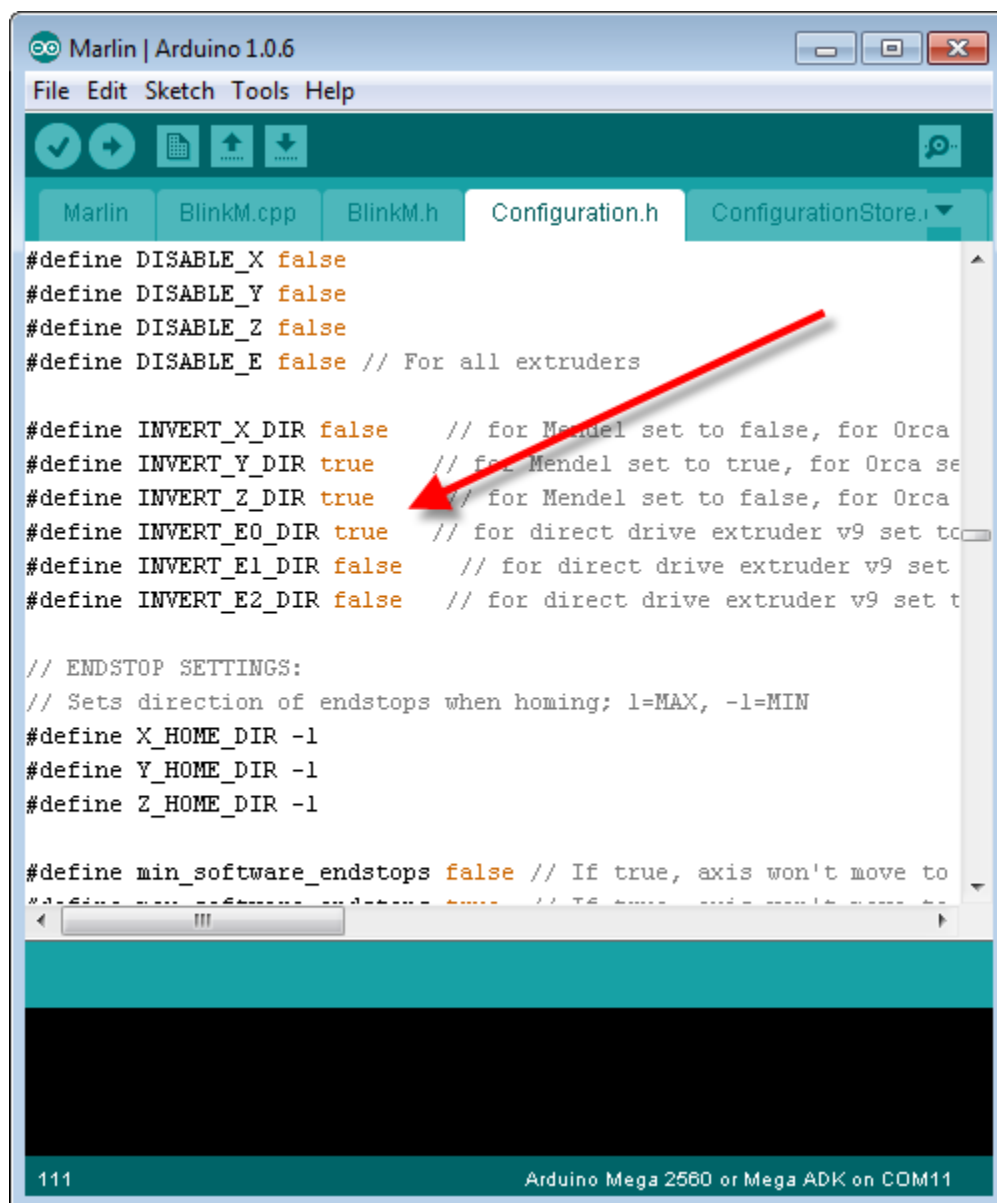
// Disable max endstops for compatibility with endstop checking routines
#if defined(COREXY) && !defined(DISABLE_MAX_ENDSTOPS)
  #define DISABLE_MAX_ENDSTOPS
#endif

111 Arduino Mega 2560 or Mega ADK on COM11
```

INVERT AXIS SETTINGS:

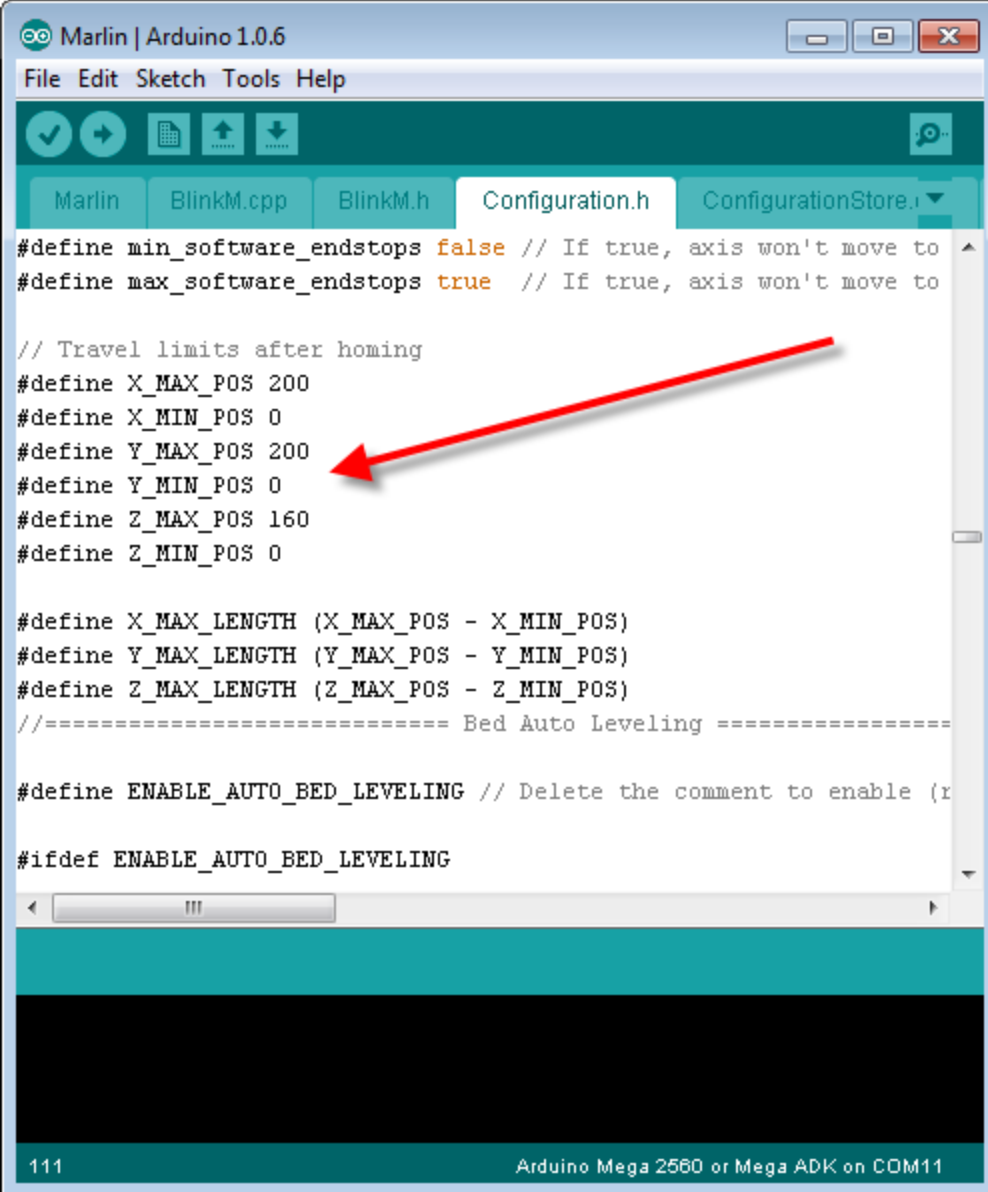
These settings define if the turning direction of each axis motor is versed or not. All of these settings can be set to FALSE because a motor's direction can be changed by simply reversing the connector on the RAMPS board.

- Use the scroll bar to locate the #define INVERT settings
- Change each setting to false



MIN MAX POSITIONS:

These settings define how large the print bed area is. Unless you have drastically changed the size of the RepRap SmartRap 3D printer, leave these settings alone.



```
Marlin | Arduino 1.0.6
File Edit Sketch Tools Help

Marlin BlinkM.cpp BlinkM.h Configuration.h ConfigurationStore.h

#define min_software_endstops false // If true, axis won't move to
#define max_software_endstops true // If true, axis won't move to

// Travel limits after homing
#define X_MAX_POS 200
#define X_MIN_POS 0
#define Y_MAX_POS 200
#define Y_MIN_POS 0
#define Z_MAX_POS 160
#define Z_MIN_POS 0

#define X_MAX_LENGTH (X_MAX_POS - X_MIN_POS)
#define Y_MAX_LENGTH (Y_MAX_POS - Y_MIN_POS)
#define Z_MAX_LENGTH (Z_MAX_POS - Z_MIN_POS)
//===== Bed Auto Leveling =====

#define ENABLE_AUTO_BED_LEVELING // Delete the comment to enable (r

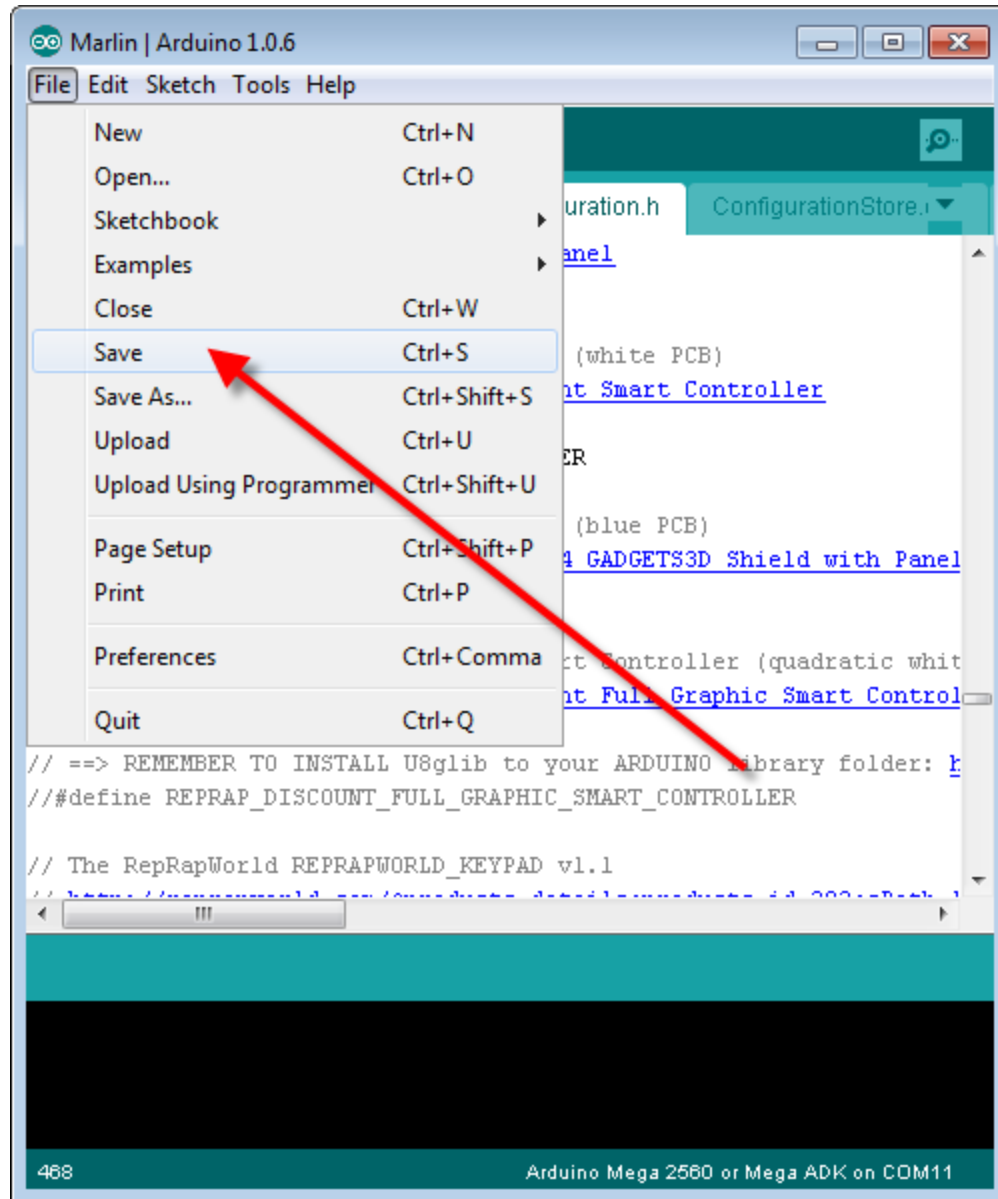
#ifdef ENABLE_AUTO_BED_LEVELING

111 Arduino Mega 2560 or Mega ADK on COM11
```

LCD SCREEN SETUP - LIBRARY:

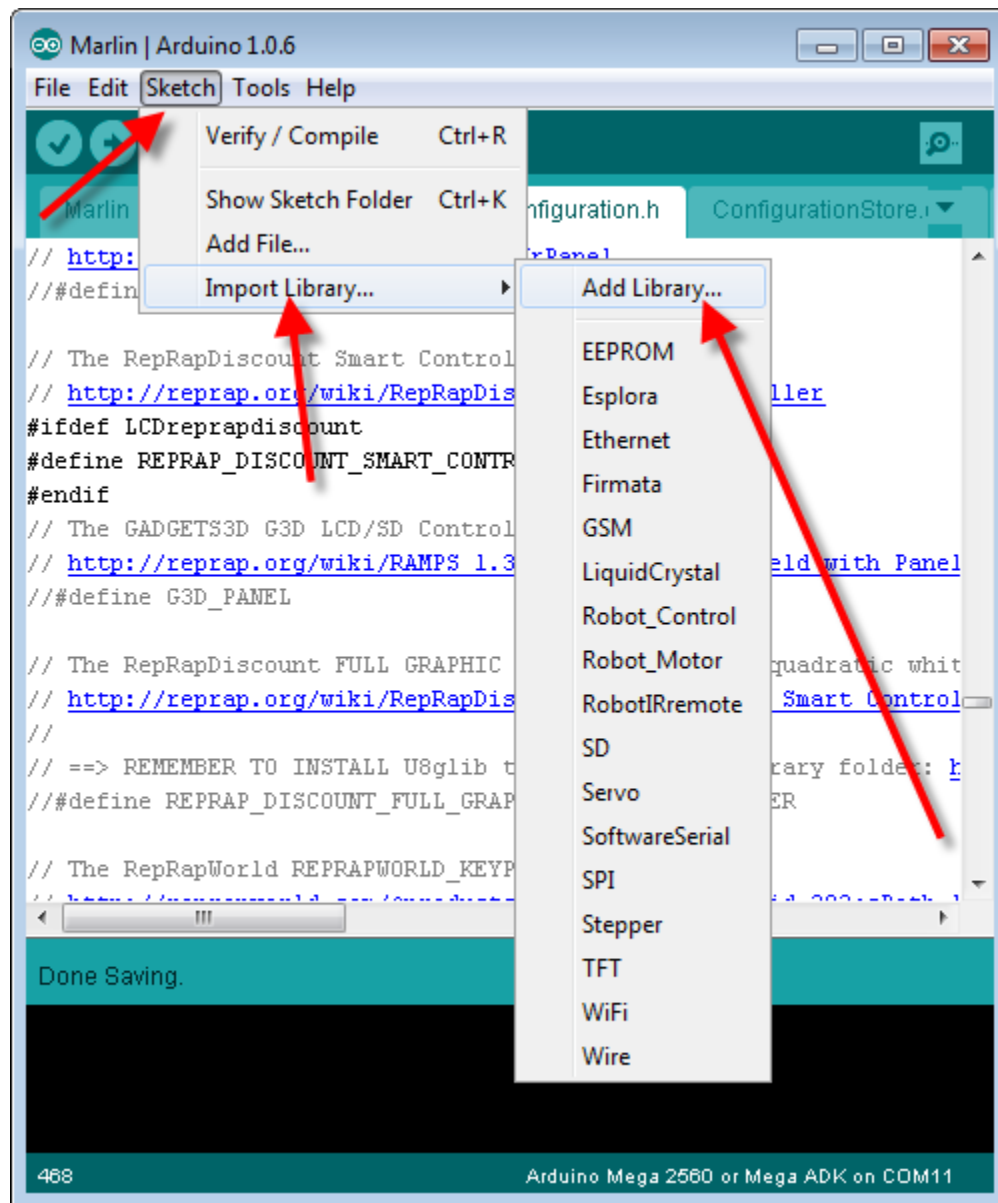
The LCD screen requires that a special library be added to the Marlin firmware and that the type of LCD screen be defined in the configuration. Let's save the modifications made so far.

- Click on File
- Click on Save



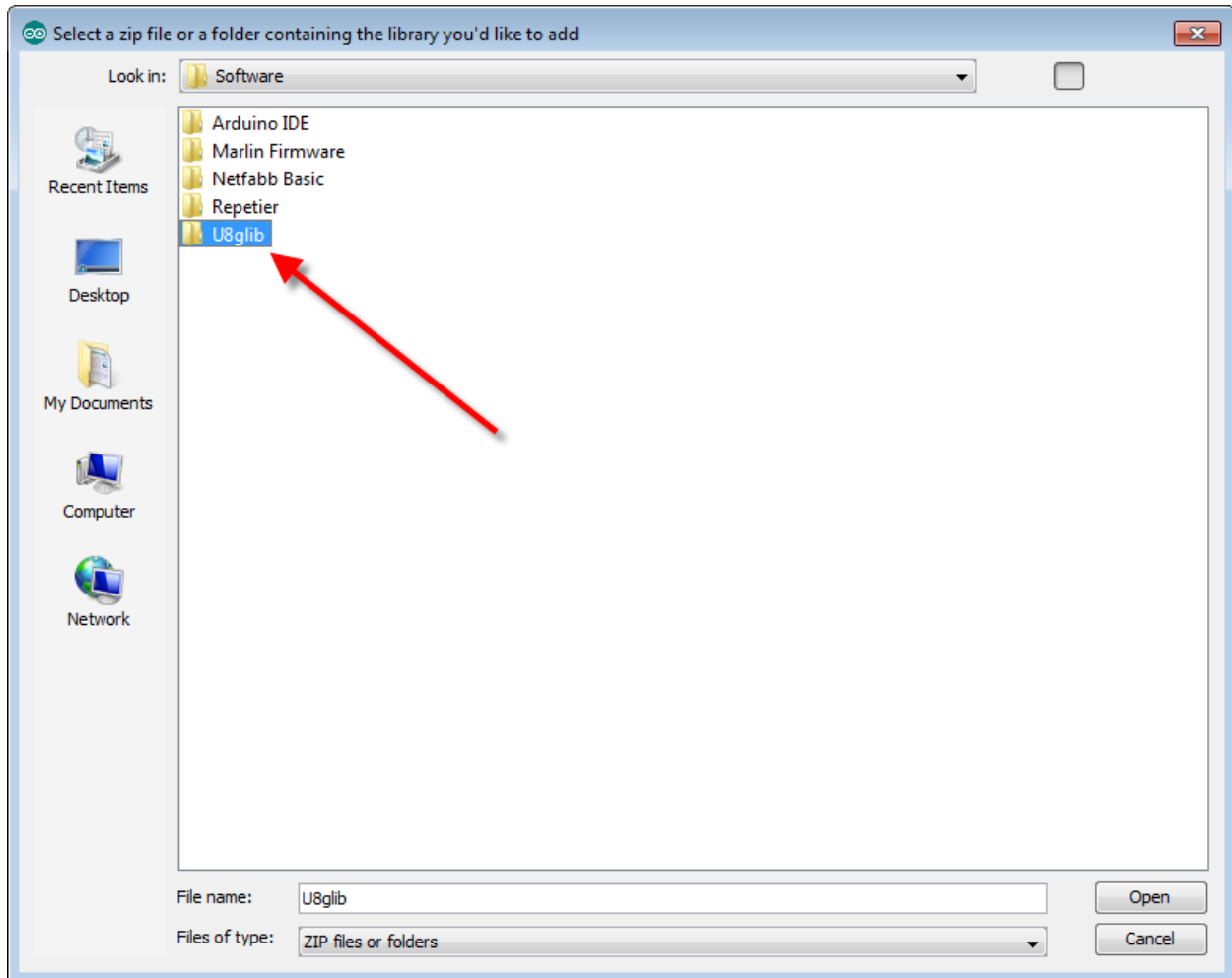
BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

- Download the U8G library in zip file format from <https://bintray.com/olikraus/u8glib/Arduino>
- Extract the zip file to a directory
- Click on Sketch
- Click on Import Library
- Click on Add Library



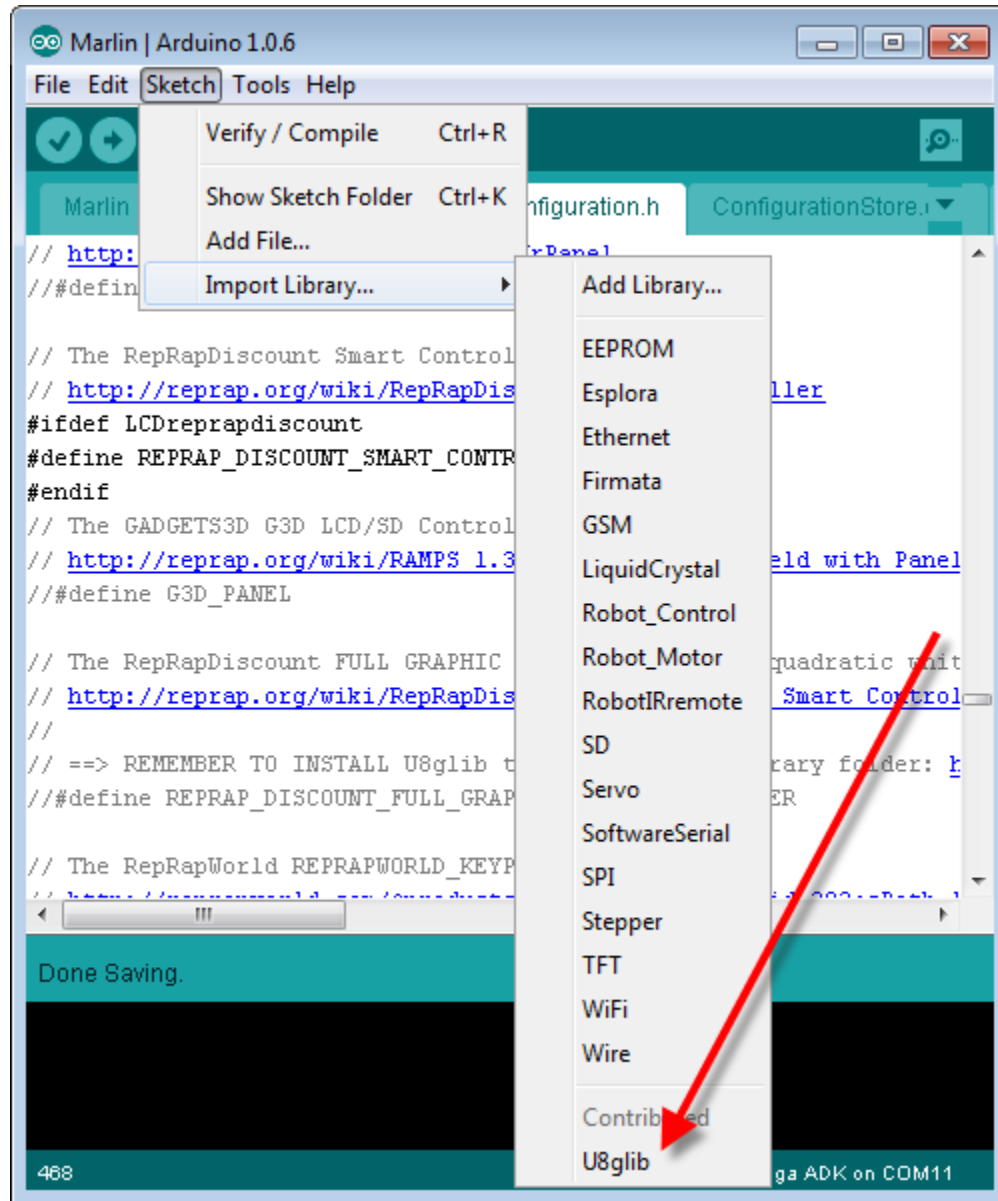
BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

- Browse to the folder where you extracted the zip file
- Click on the Open button



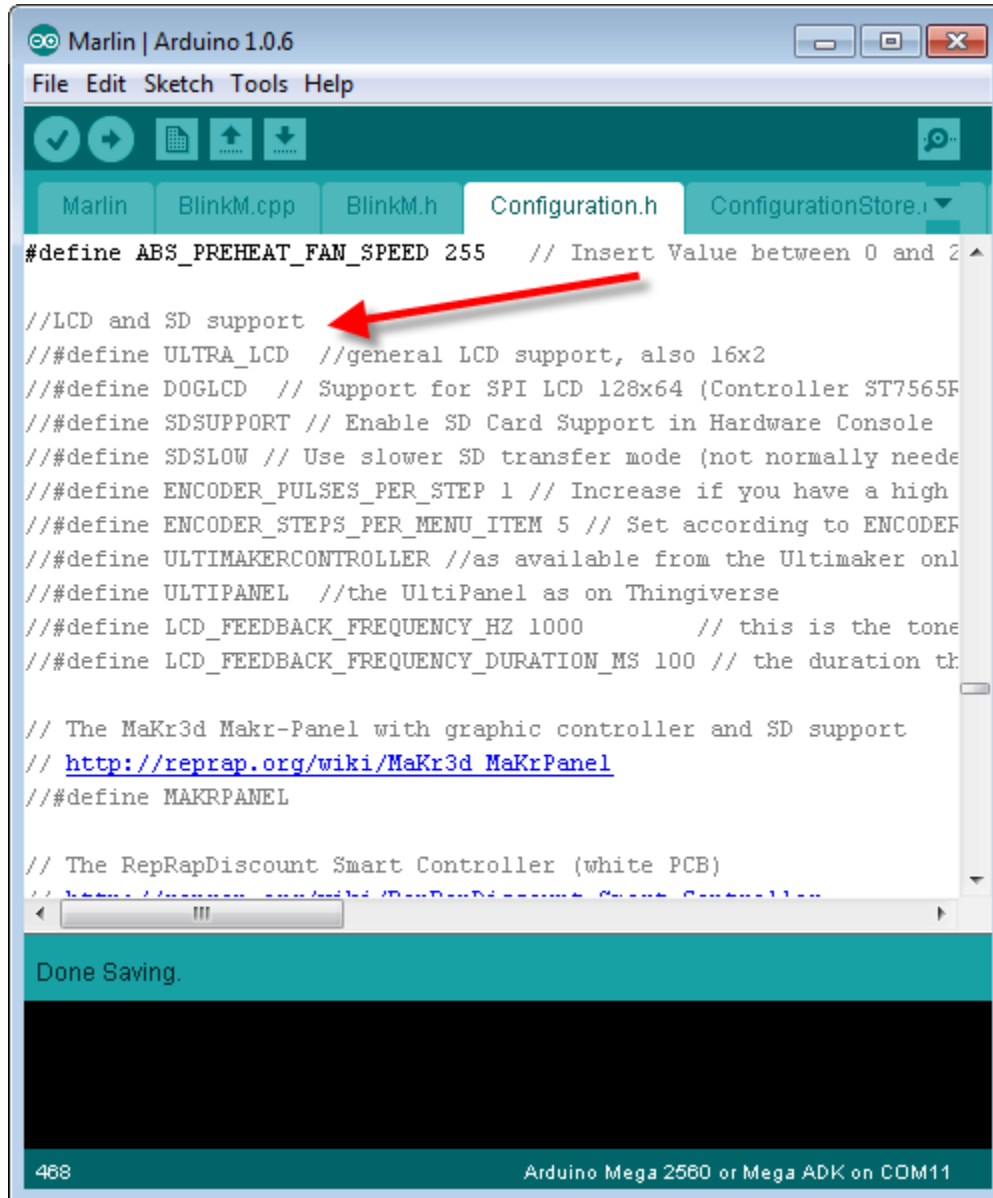
You can confirm that the library has been added successfully:

- Click on Sketch
- Click on Import Library
- U8glib or the name of the folder you opened is displayed at the bottom of the list



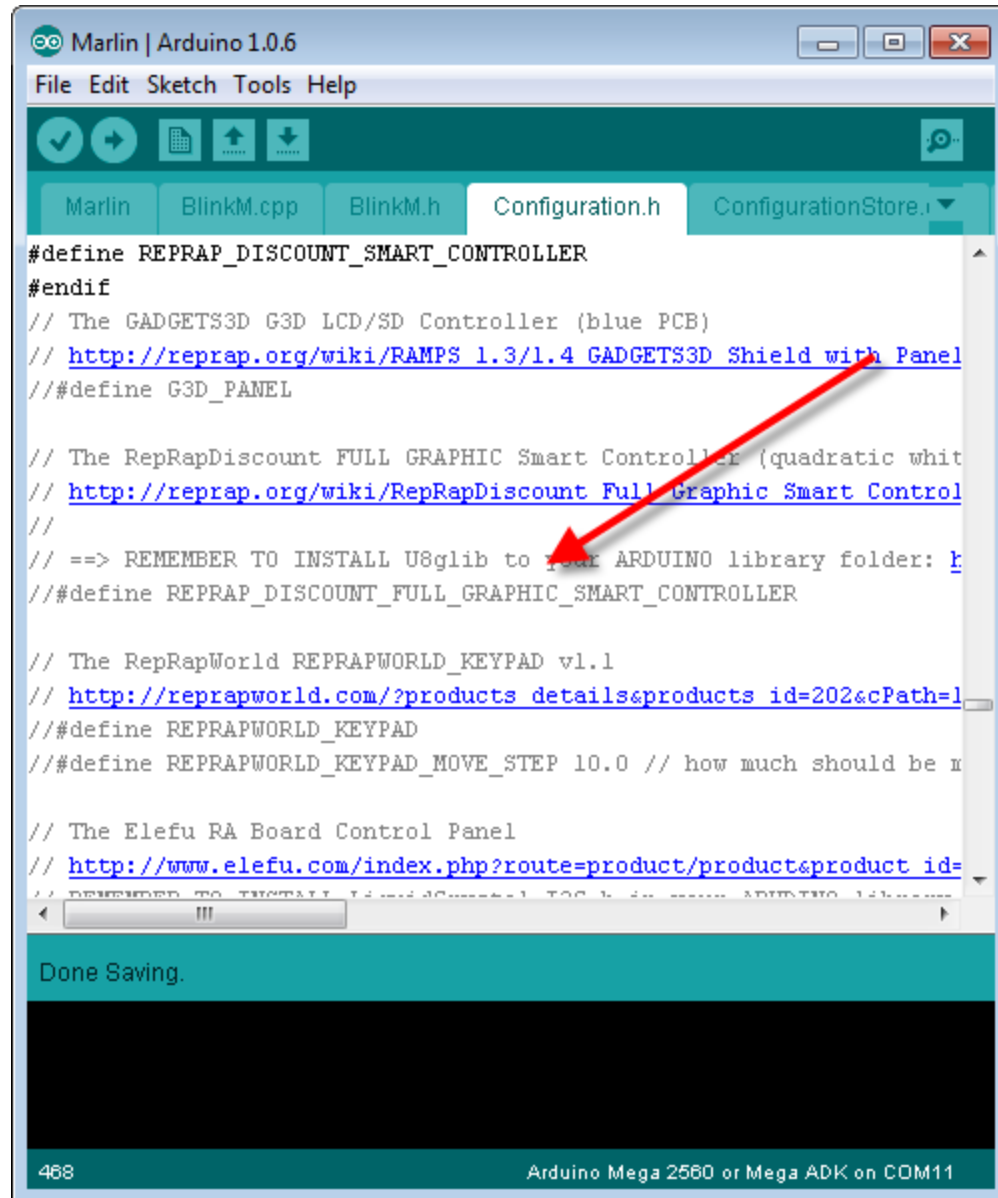
LCD SCREEN SETUP – LCD SELECTION:

- Use the scrollbar to locate the LCD section
- Locate the #define setting for your LCD screen and remove the dashes for that setting



BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

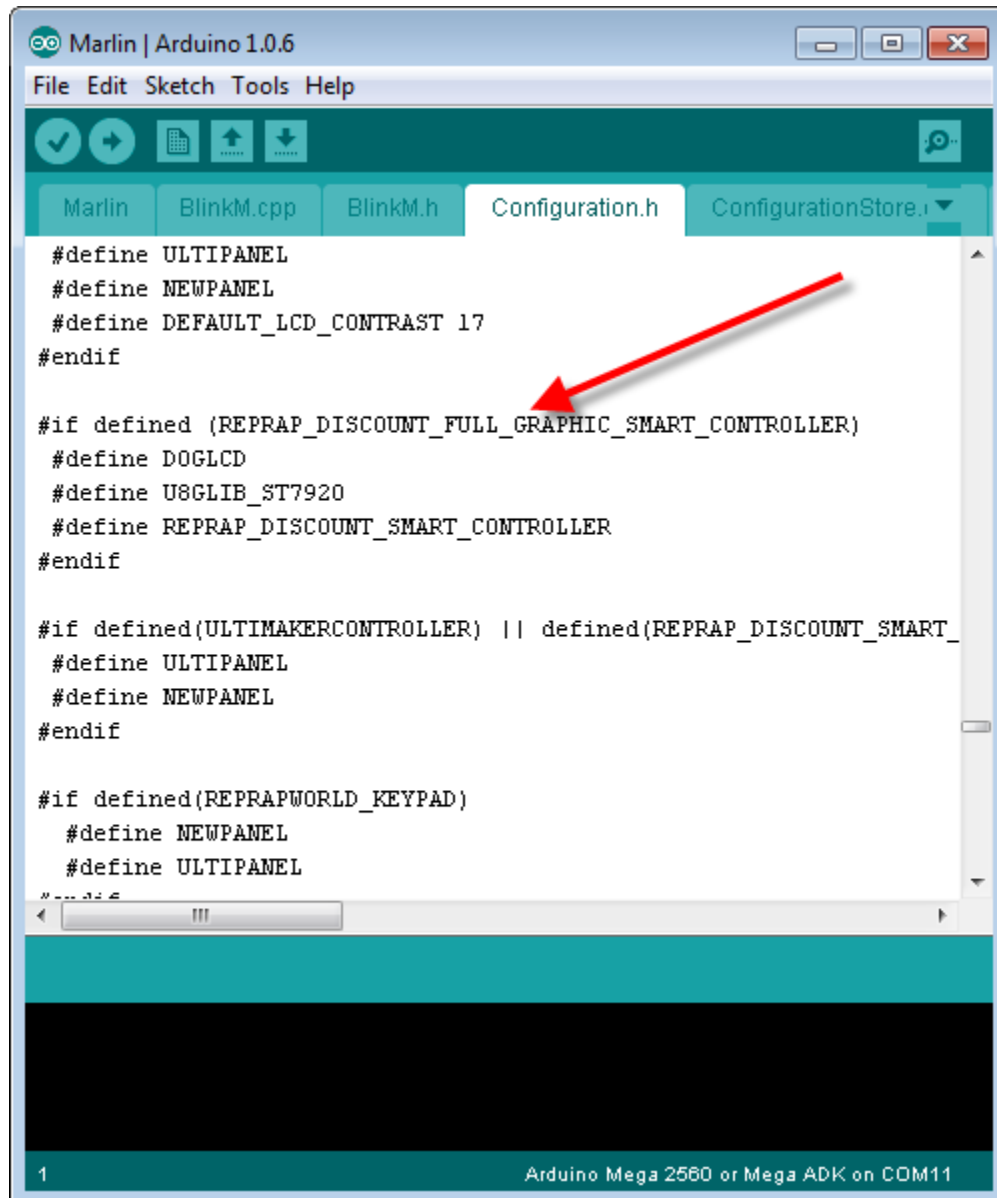
- If you utilized the Parts Cost and Analysis spreadsheet from my blog and ordered the RAMPS V1.4 kit from AliExpress
 - Locate the `//#define REPRAP_DISCOUNT_FULL_GRAPHICS_SMART_CONTROLLER` setting
 - Remove the dashes



LCD SCREEN SETUP – CONTROL KNOB INCREMENT SPEED:

In order to ensure that the control knob on the LCD screen increments values one at a time when changing values, two define statements need to be added.

- Use the scrollbar to locate the #if defined section for the LCD screen that you selected
- Add the following two lines before the #endif statement
 - #define ENCODER_PULSES_PER_STEP 4
 - #define ENCODER_STEPS_PER_MENU_ITEM 1



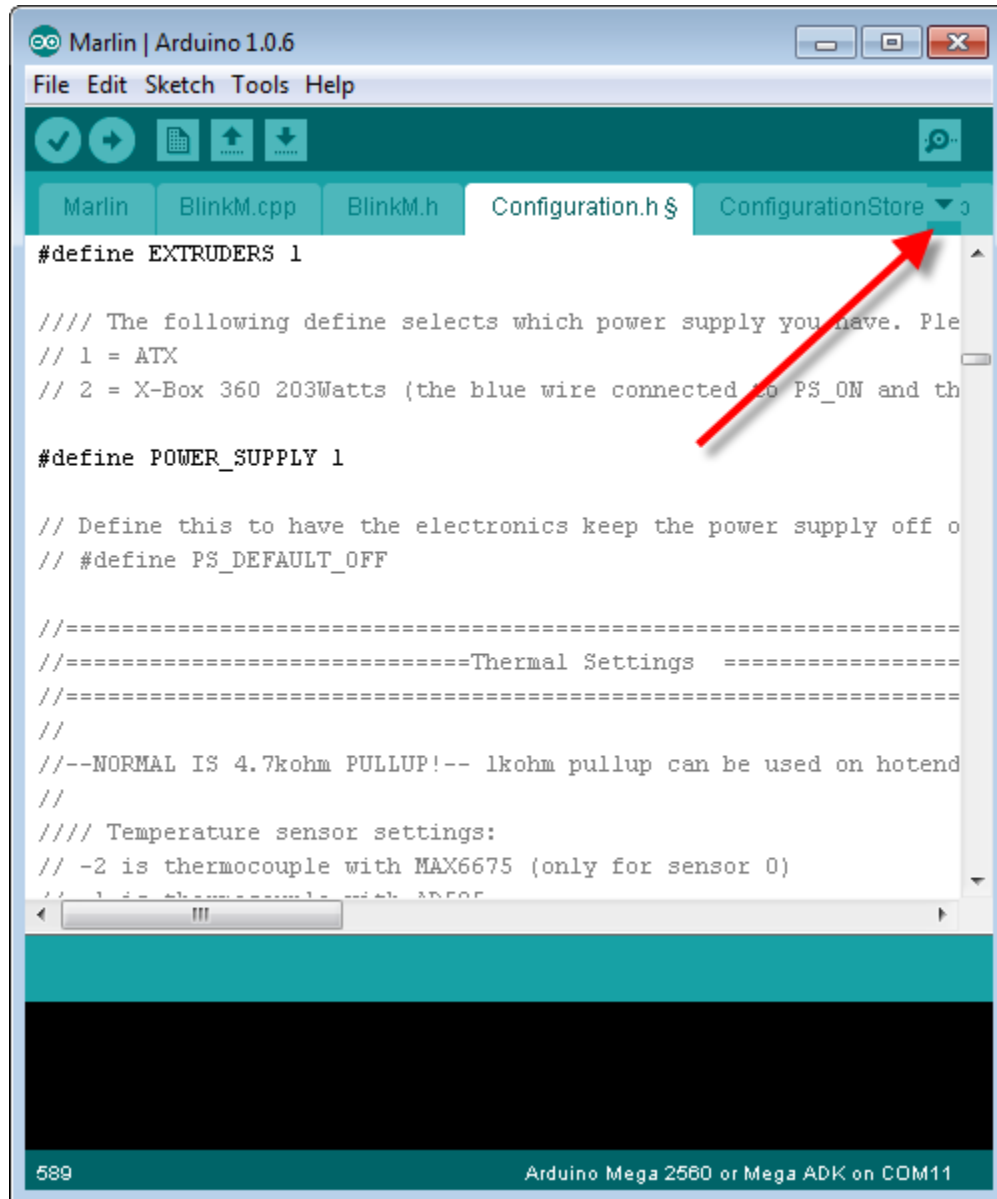
Depending on the LCD screen you selected, the section should now look like something like this

```
#if defined (REPRAP_DISCOUNT_FULL_GRAPHIC_SMART_CONTROLLER)
#define DOGLCD
#define U8GLIB_ST7920
#define REPRAP_DISCOUNT_SMART_CONTROLLER
#define ENCODER_PULSES_PER_STEP 4
#define ENCODER_STEPS_PER_MENU_ITEM 1
#endif
```

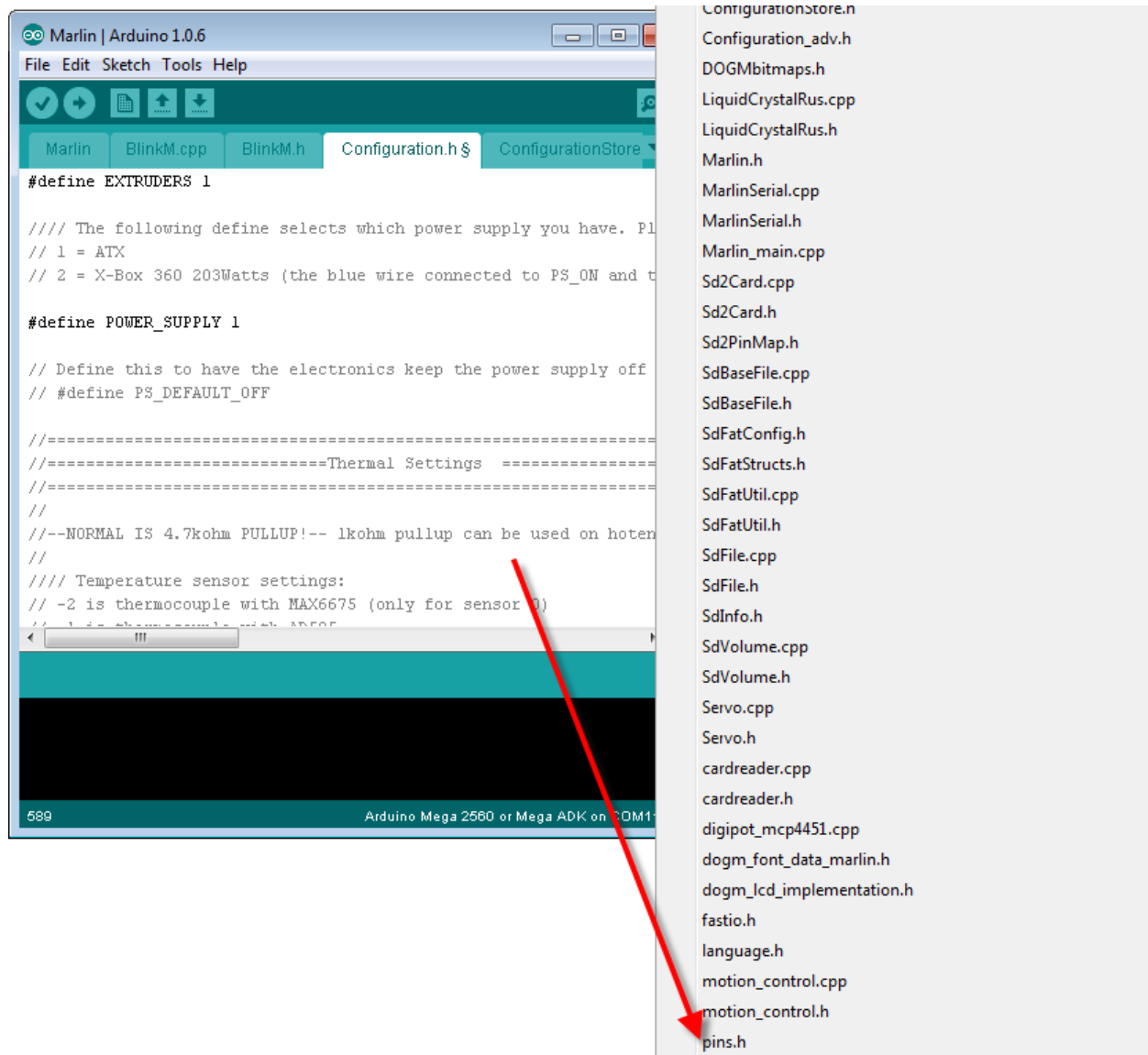
LCD SCREEN SETUP – CONTROL KNOB SCROLL DIRECTION:

In order to ensure that the control knob on the LCD screen increments values in the right direction, i.e. turning to the left decreases a value while turning to the right increases a value, there are two #define statements that need to be changed in the Pins.h tab.

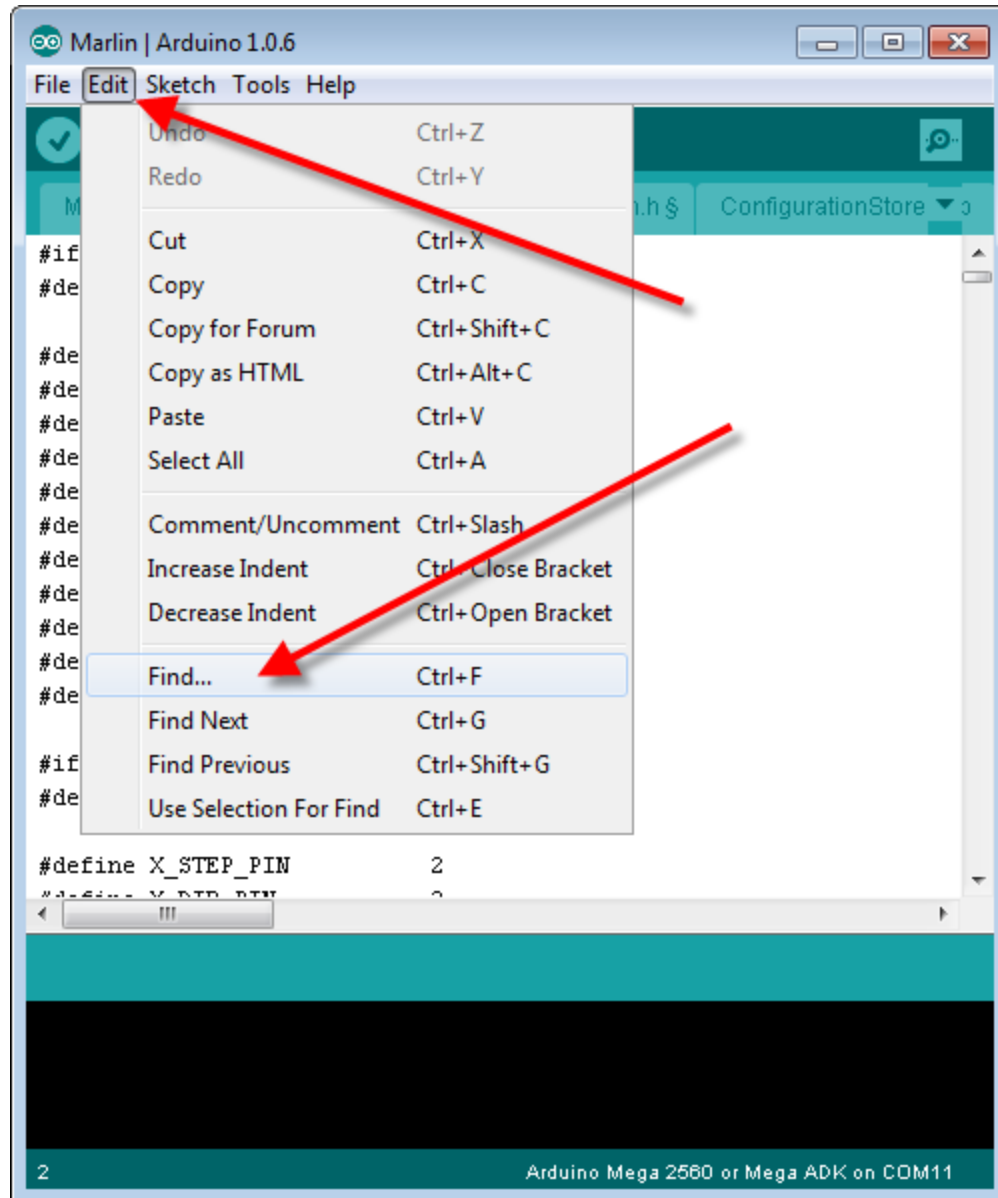
- Click on the small down arrow to the right of the tabs



- Click on the Pins.h item from the drop down list

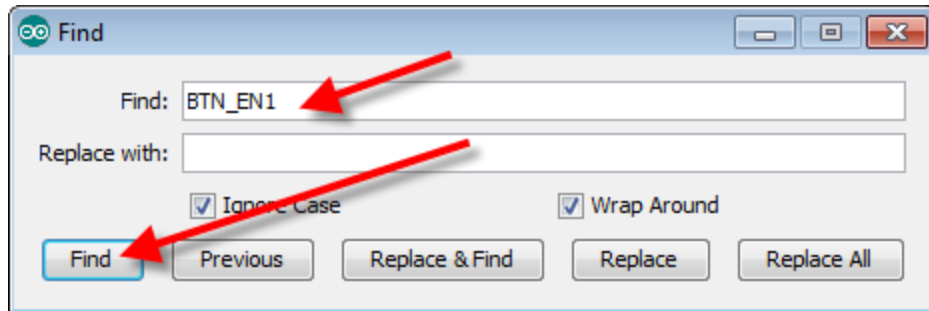


- Click on Edit
- Click on Find

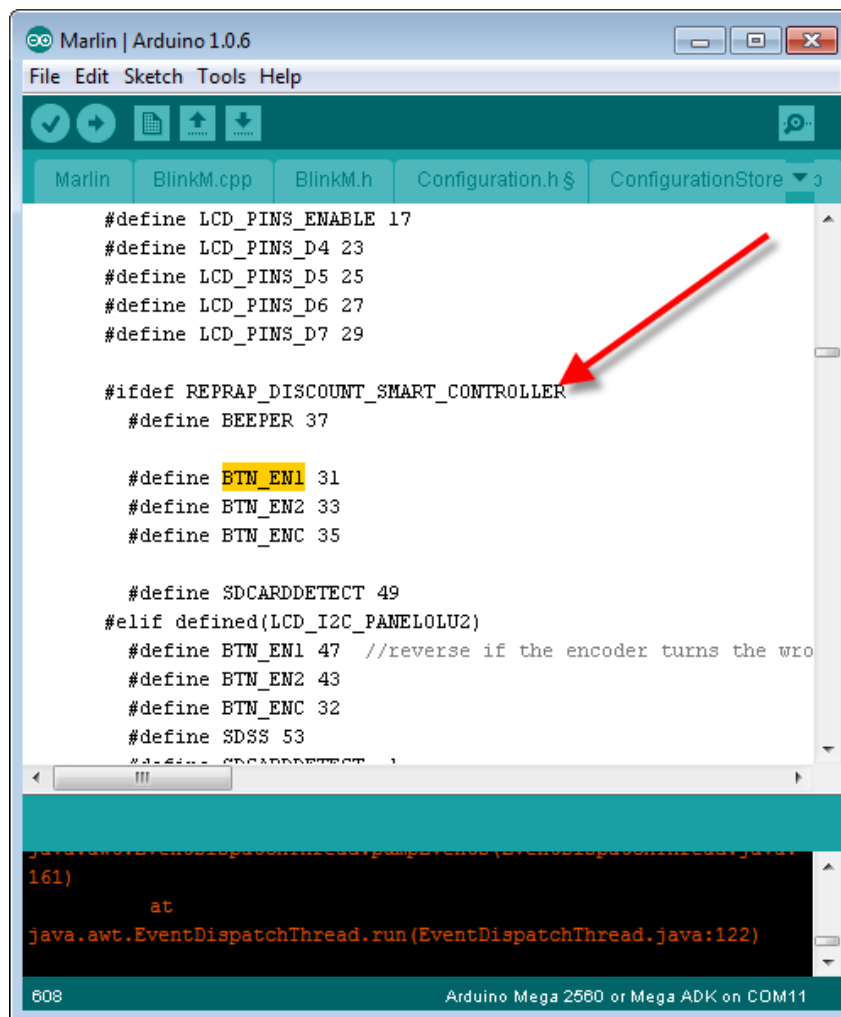


BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

- In the search field, type in BTN_EN1
- Click on the Find button



- Search until you locate the #define BTN_EN1 statement under the type of LCD screen that you previously defined

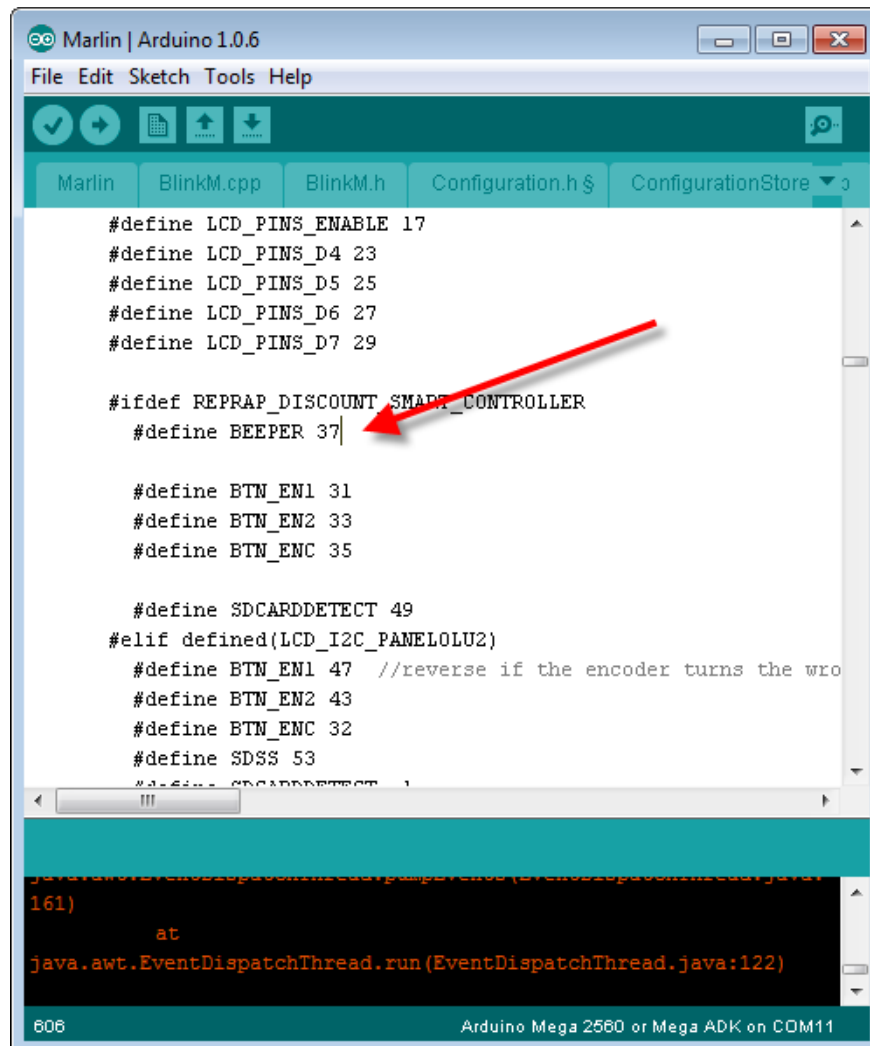


- Swap the values between EN1 and EN2
- For example
 - `#define BTN_EN1 31`
 - `#define BTN_EN2 33`
- Is changed to
 - `#define BTN_EN1 33`
 - `#define BTN_EN2 31`

If after changing these values you find that the control knob does not scroll values properly up or down, then revert these settings back to what they were and test again. If the control knob still does not scroll properly, ensure that the correct LCD was selected.

LCD SCREEN SETUP – BEEPER ON/OFF:

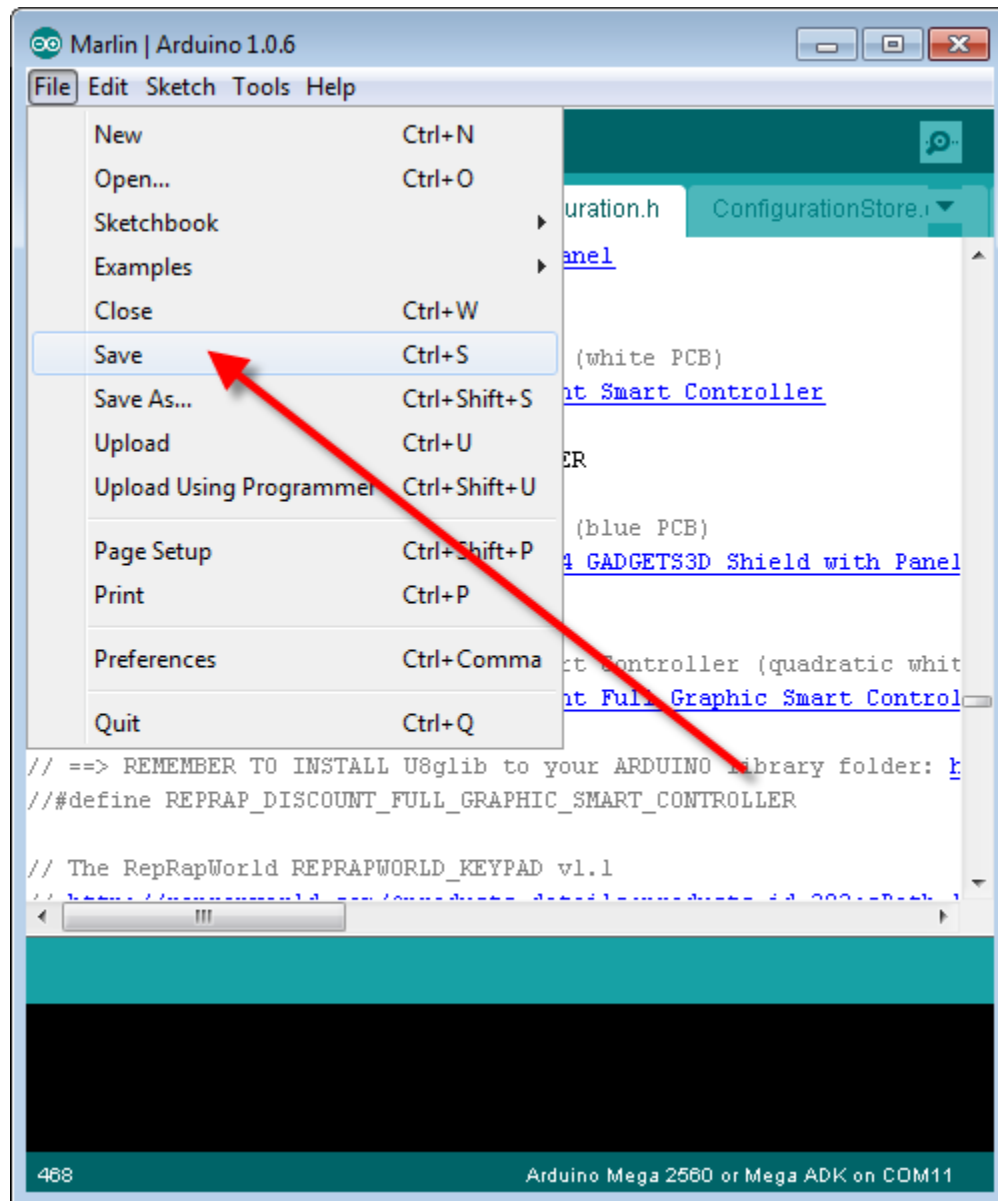
If you wish to disable the beeper on the LCD, change the #define BEEPER value to -1



SAVE YOUR SETTINGS:

You should always save your settings before flashing the firmware.

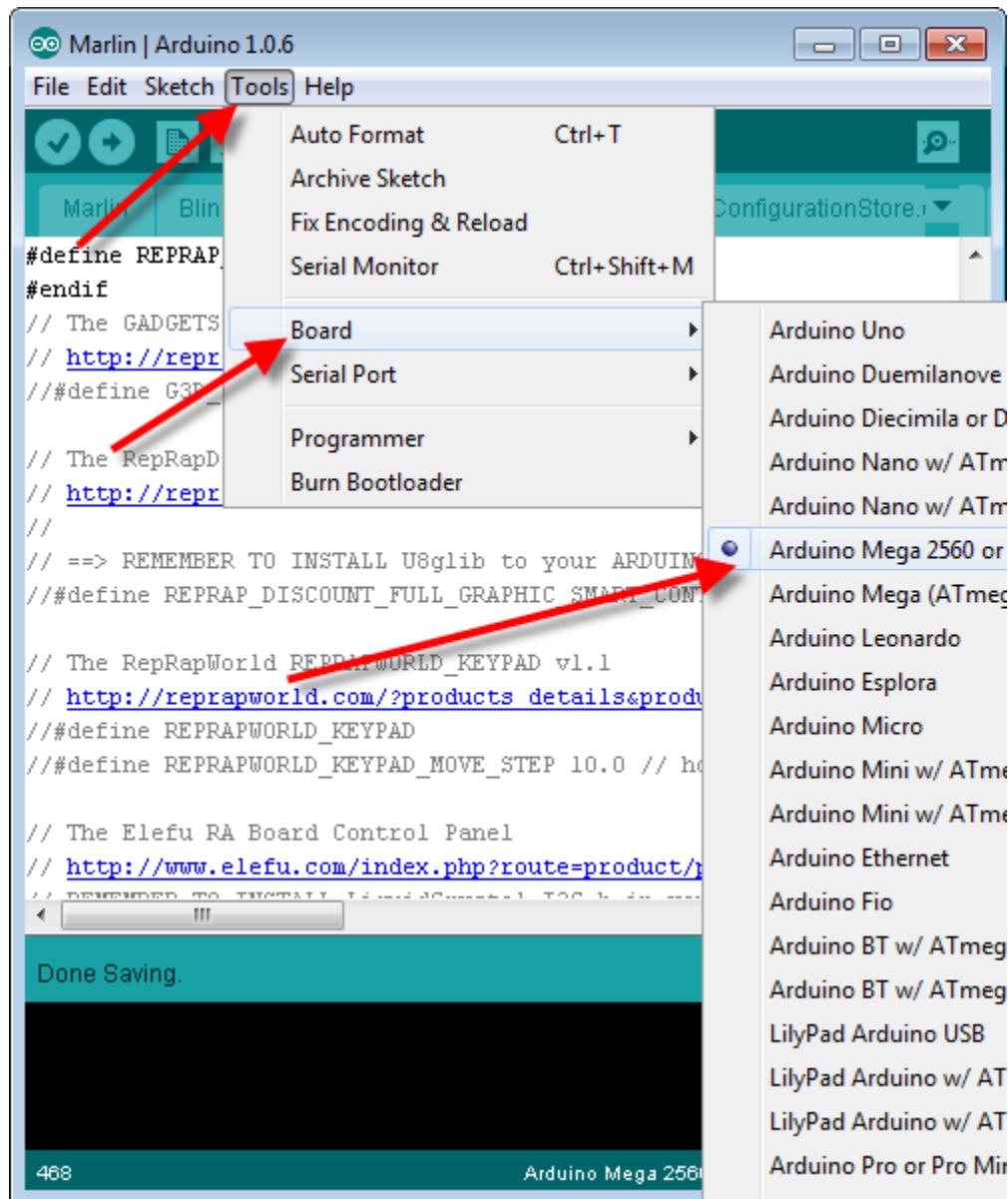
- Click on File
- Click on Save



FLASHING THE FIRMWARE:

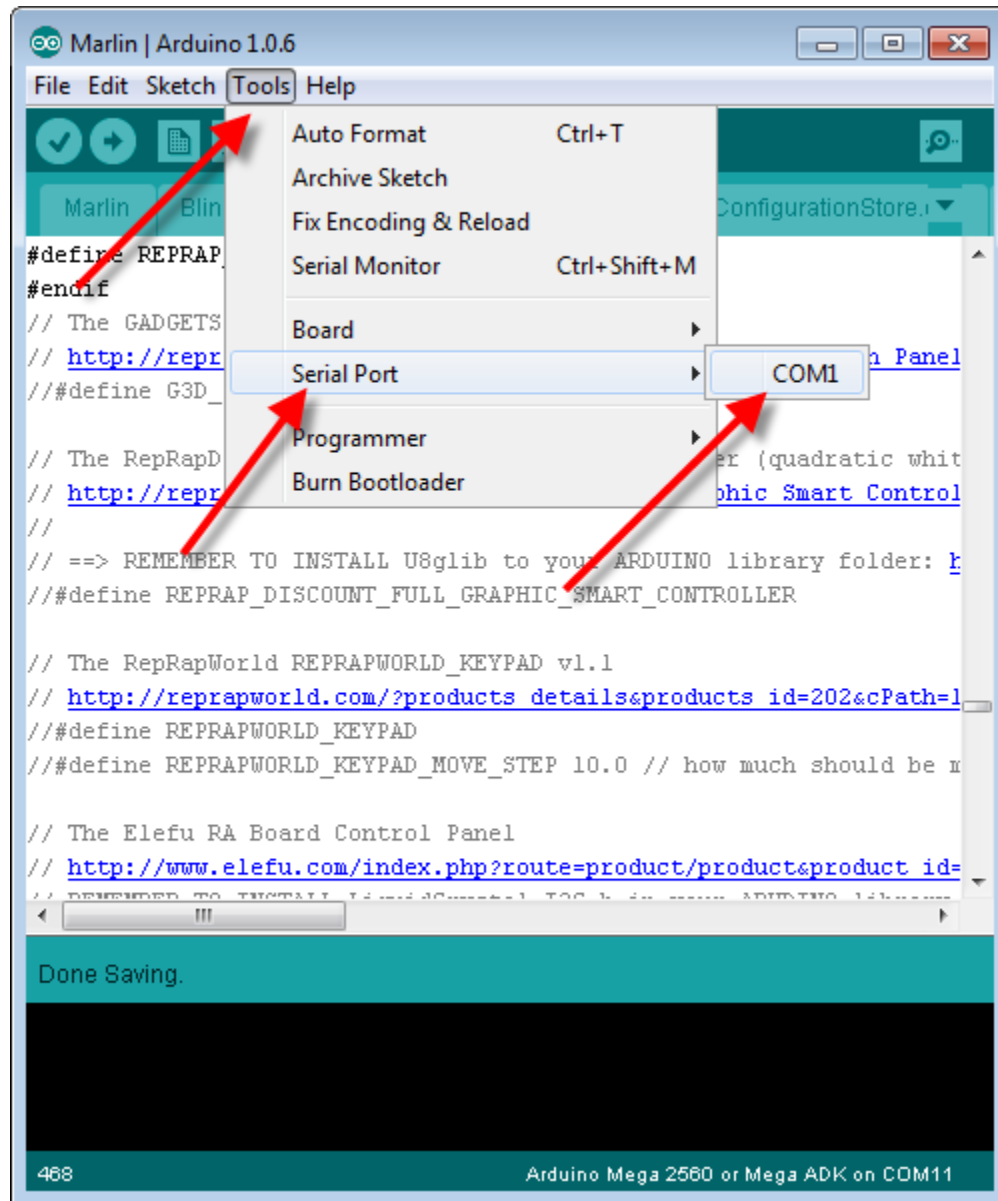
Before the firmware can be flashed, Arduino needs to know the type of Arduino board being flashed and the serial port being used.

- Click on Tools
- Click on Board
- Click on the line that indicates Arduino Mega 2560 or Mega ADK



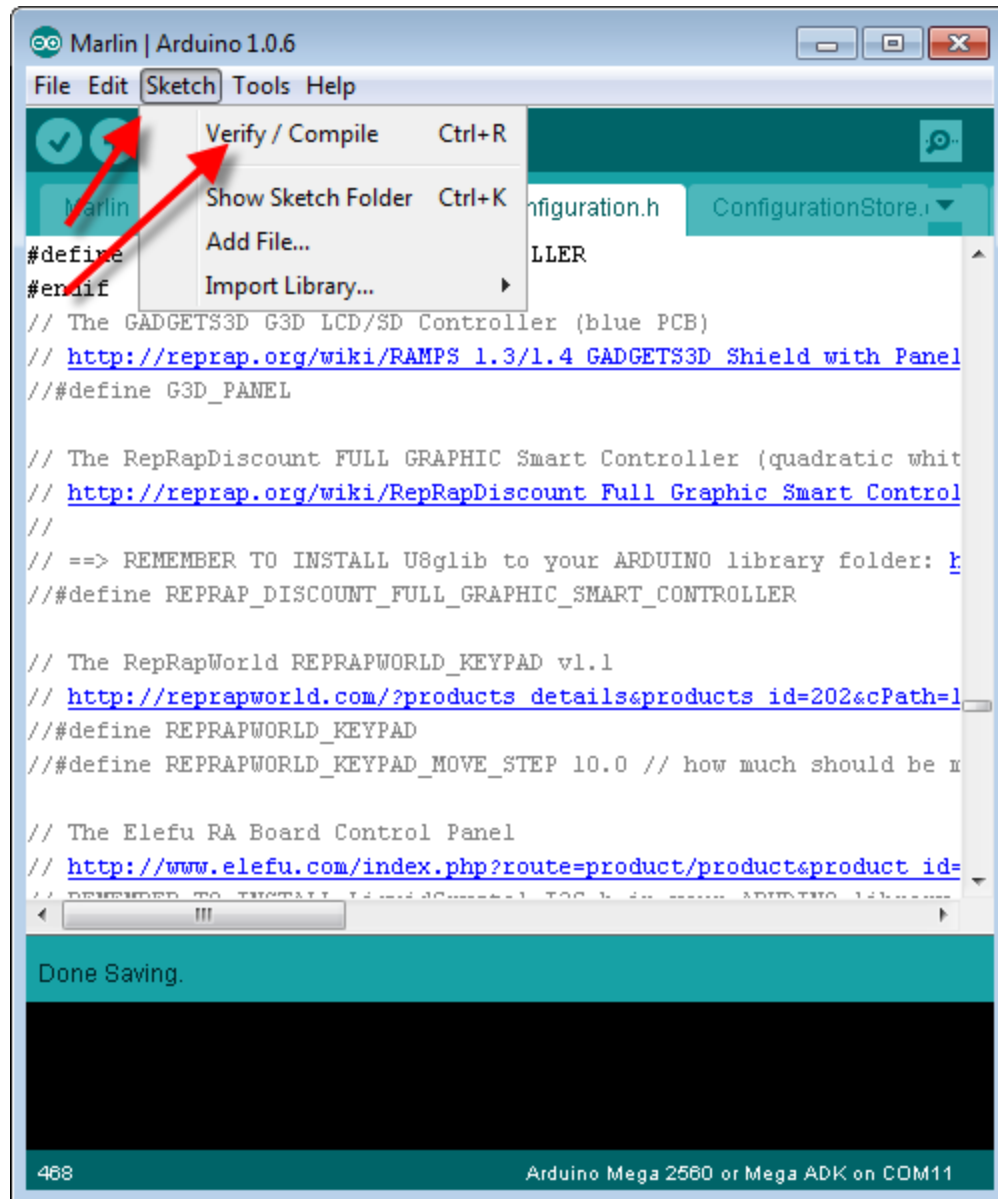
BASIC MARLIN FIRMWARE CONFIGURATION USER GUIDE FOR THE SMARTRAP 3D PRINTER WITH LCD

- Ensure that the Arduino Mega 2560 board is connected to your computer
- Ensure that your computer recognized the board and has assigned a COM port
- Click on Tools
- Click on Serial Port
- Click on the Serial port assigned by your computer

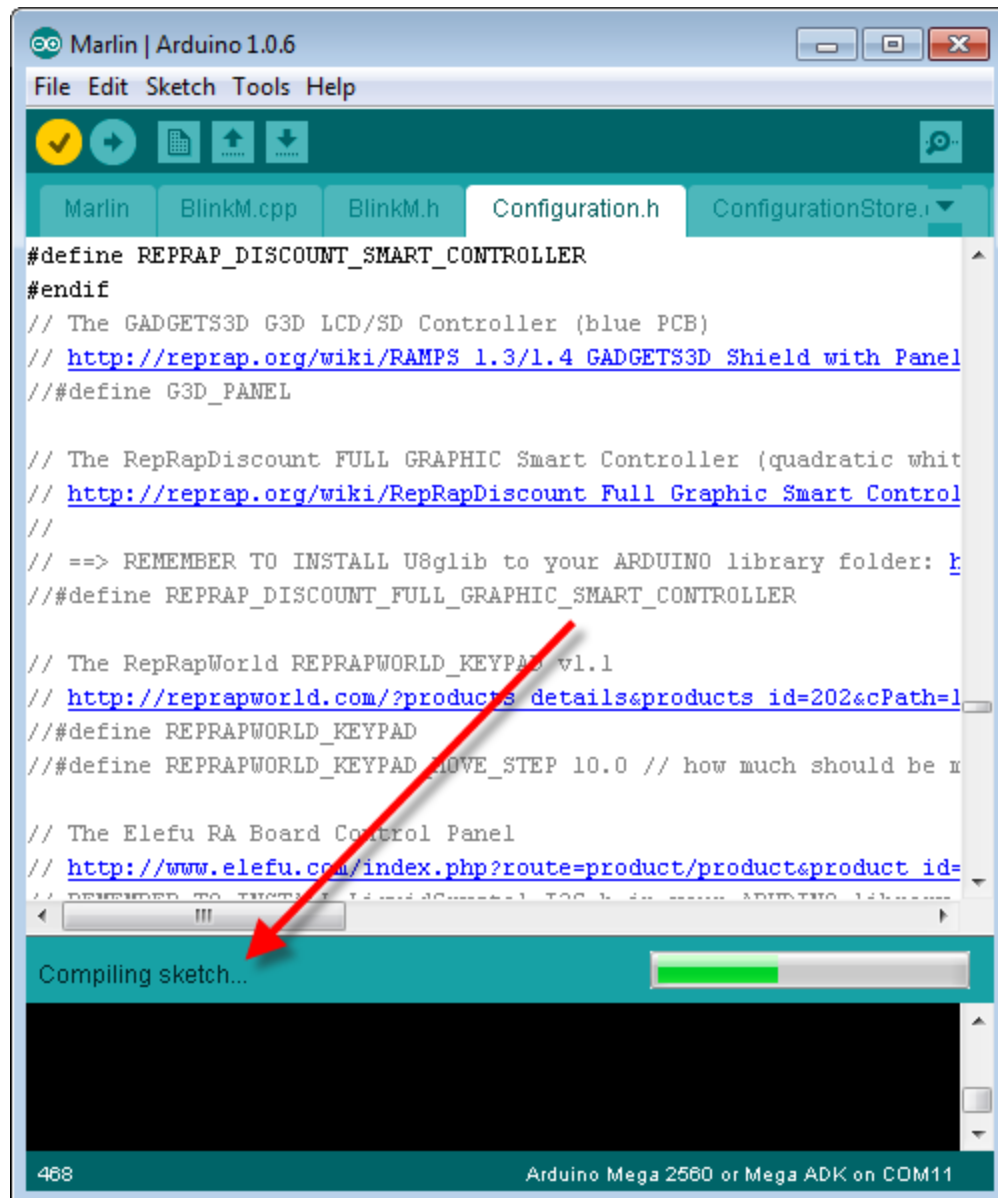


It's a good idea to verify and compile the firmware before flashing. In this manner you can identify any errors before flashing.

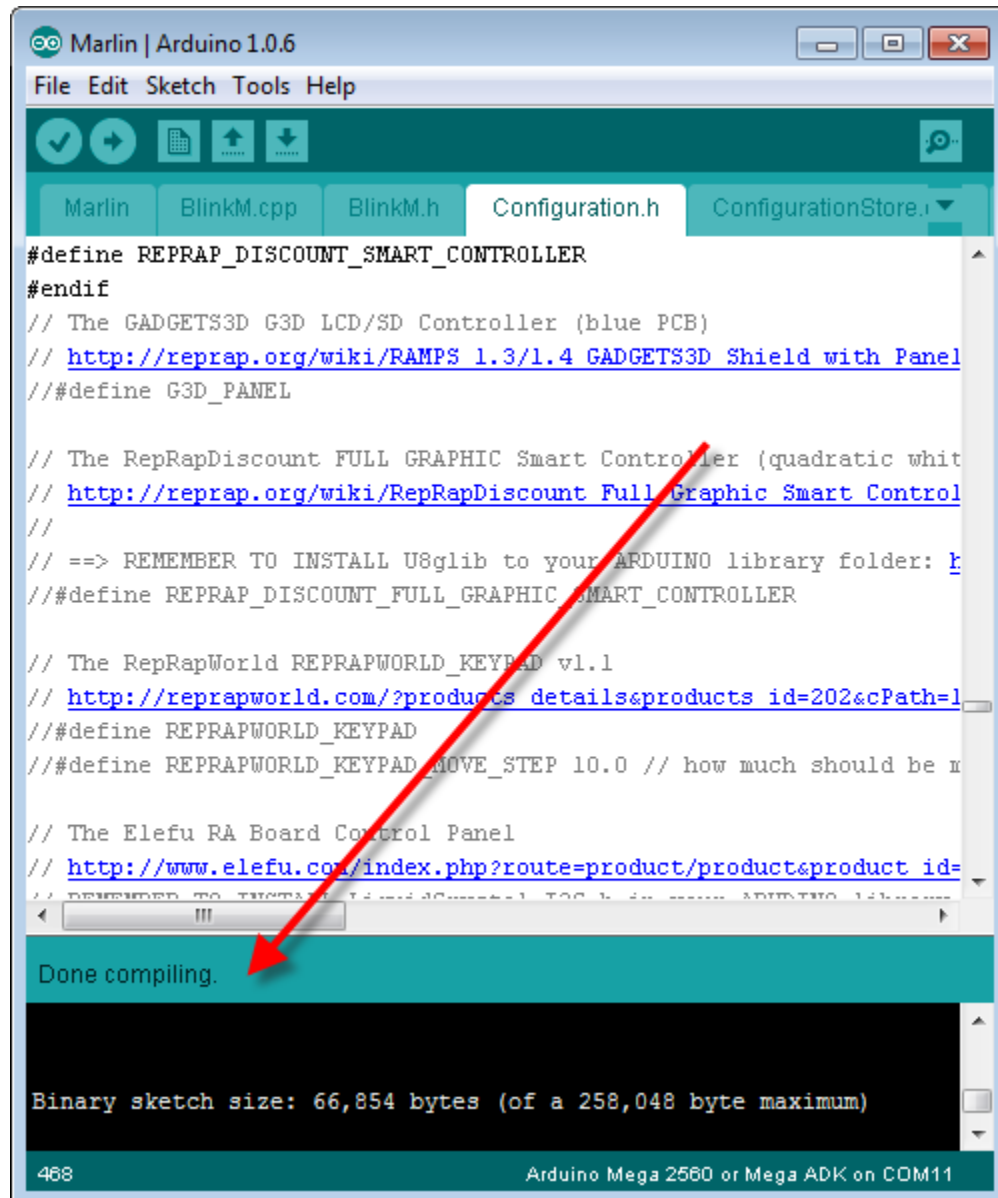
- Click on Sketch
- Click on Verify/Compile



- A status message is displayed during the compiling
- Do not interrupt the process



- The completed status is displayed



If there are no errors:

- Click on File
- Click on Upload

