3D Printer Program Guide

Prusa-Firmware-MK3-2

By: Prashant Nayak

Indian Institute of Information Technology Allahabad
mail: ibm2013014@iiita.ac.in
9696620781

Downloads - software and firmware

Step 1

After completing the build of your printer, it's time to breathe some life into it.

- Start by downloading the Marlin firmware from this link:

  reprap.pt/downloads/Marlin_Prusai3_reprap_pt.zip

  [Marlin firmware used for Prusa i3 Hephestos - modified for Prusa i3 Rework]

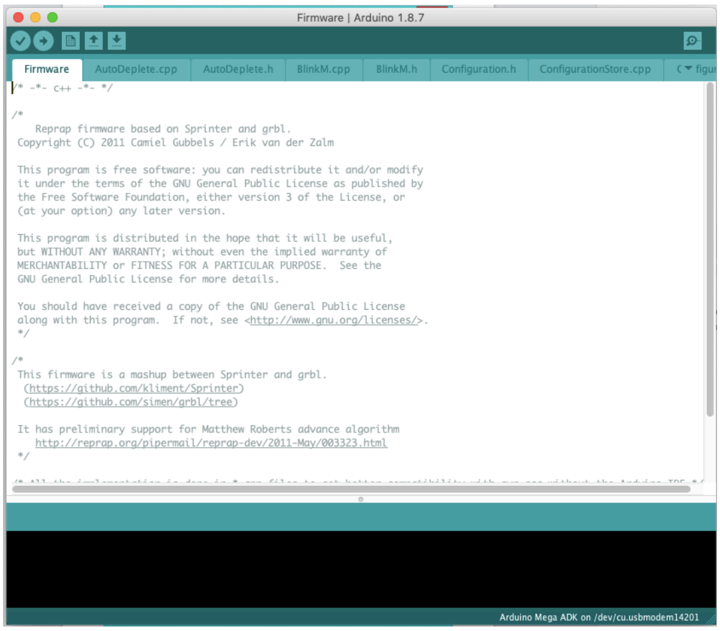
  Or choose for a Geeetech i3 X: [Firmware for Geeetech i3 X]
  Original version can be found on GitHub: Marlin Firmware Repository

- Next download the Arduino IDE from here : www.arduino.cc/downloads
-
- Download the Printrun, Cura Engine, repetier application which will be used to control your printer via USB:

STL and OBJ 3D model files for testing can be obtained from [Thingiverse]. If you want to get into making your own stuff and learn about 3D modeling take a look at [Blender] - it's free and open source (GPL).

Step 2

After Downloading and installing all aforementioned applications start by unzipping the Marlin_Prusai3_reprap_pt.zip file to your desktop or another place where you can easily find it.Doubleclick the file Marlin.ino. This should open a window similar to this one:



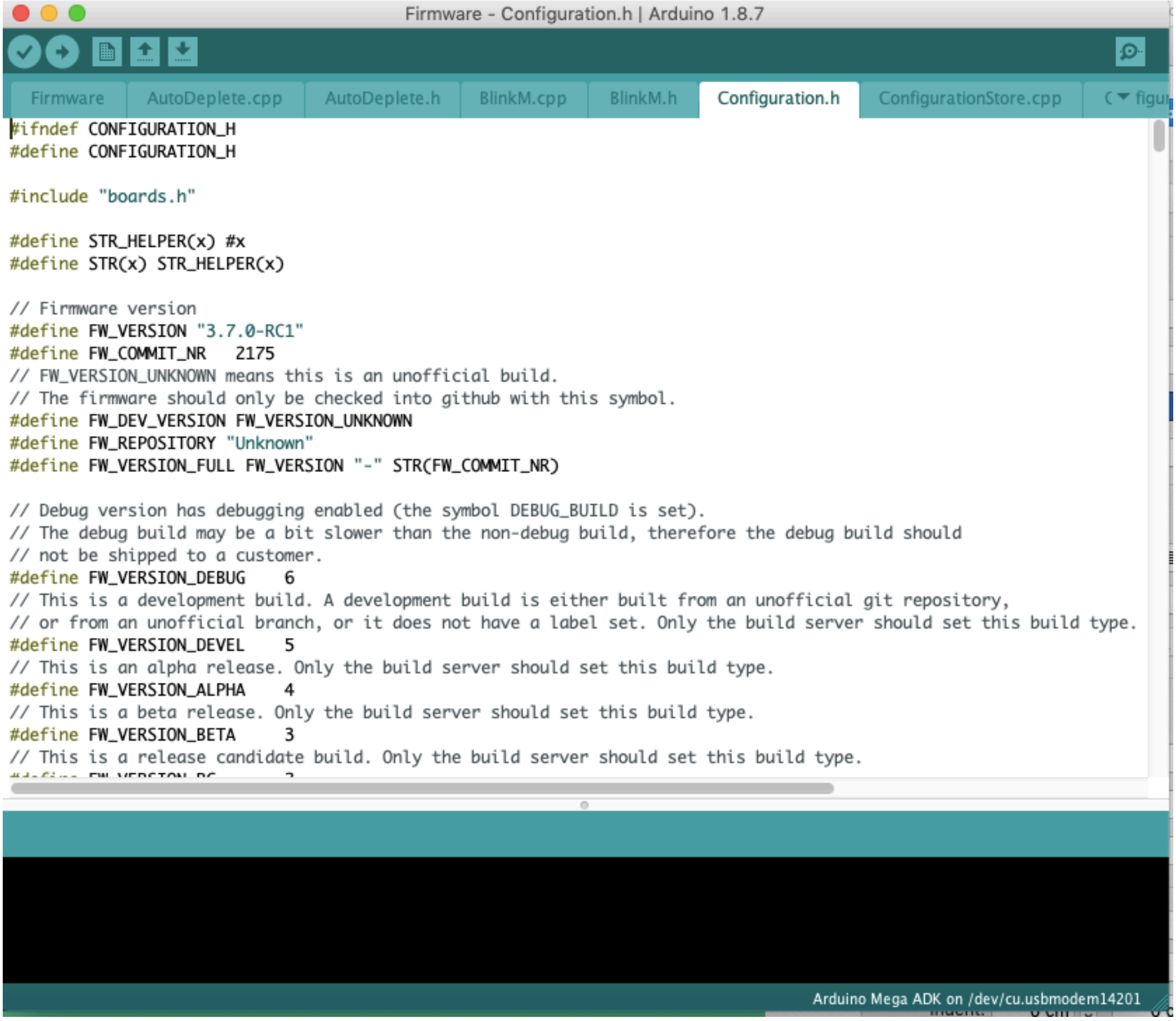
```
/* -*- c++ -*- */

/*
   Reprap firmware based on Sprinter and grbl.
 Copyright (C) 2011 Camiel Gubbels / Erik van der Zalm

 This program is free software: you can redistribute it and/or modify
 it under the terms of the GNU General Public License as published by
 the Free Software Foundation, either version 3 of the License, or
 (at your option) any later version.

 This program is distributed in the hope that it will be useful,
 but WITHOUT ANY WARRANTY; without even the implied warranty of
 MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 GNU General Public License for more details.

 You should have received a copy of the GNU General Public License
 along with this program.  If not, see <http://www.gnu.org/licenses/>.
 */

/*
 This firmware is a mashup between Sprinter and grbl.
  (https://github.com/kliment/Sprinter)
  (https://github.com/simen/grbl/tree)

 It has preliminary support for Matthew Roberts advance algorithm
    http://reprap.org/pipermail/reprap-dev/2011-May/003323.html
 */

/* All the implementation is done in *.cpp files to get better compatibility with avr-gcc without the Arduino IDE */
```

Arduino IDE with Marlin Firmware opened

Change the firmware

Step 1: Open the Settings

A large part of the code has been changed for me, but still can change some parts of the code or confirm in case something be wrong. If you do not change you can skip the following steps up to 10. If you opted for the current version, you should not ignore these steps. Still with the latest version of the variables differ.
To change the code Click on the tab that says Configuration.h.



```
#ifndef CONFIGURATION_H
#define CONFIGURATION_H

#include "boards.h"

#define STR_HELPER(x) #x
#define STR(x) STR_HELPER(x)

// Firmware version
#define FW_VERSION "3.7.0-RC1"
#define FW_COMMIT_NR   2175
// FW_VERSION_UNKNOWN means this is an unofficial build.
// The firmware should only be checked into github with this symbol.
#define FW_DEV_VERSION FW_VERSION_UNKNOWN
#define FW_REPOSITORY "Unknown"
#define FW_VERSION_FULL FW_VERSION "-" STR(FW_COMMIT_NR)

// Debug version has debugging enabled (the symbol DEBUG_BUILD is set).
// The debug build may be a bit slower than the non-debug build, therefore the debug build should
// not be shipped to a customer.
#define FW_VERSION_DEBUG    6
// This is a development build. A development build is either built from an unofficial git repository,
// or from an unofficial branch, or it does not have a label set. Only the build server should set this build type.
#define FW_VERSION_DEVEL    5
// This is an alpha release. Only the build server should set this build type.
#define FW_VERSION_ALPHA    4
// This is a beta release. Only the build server should set this build type.
#define FW_VERSION_BETA     3
// This is a release candidate build. Only the build server should set this build type.
#define FW_VERSION_RC       2
```

Tab Configuration.h

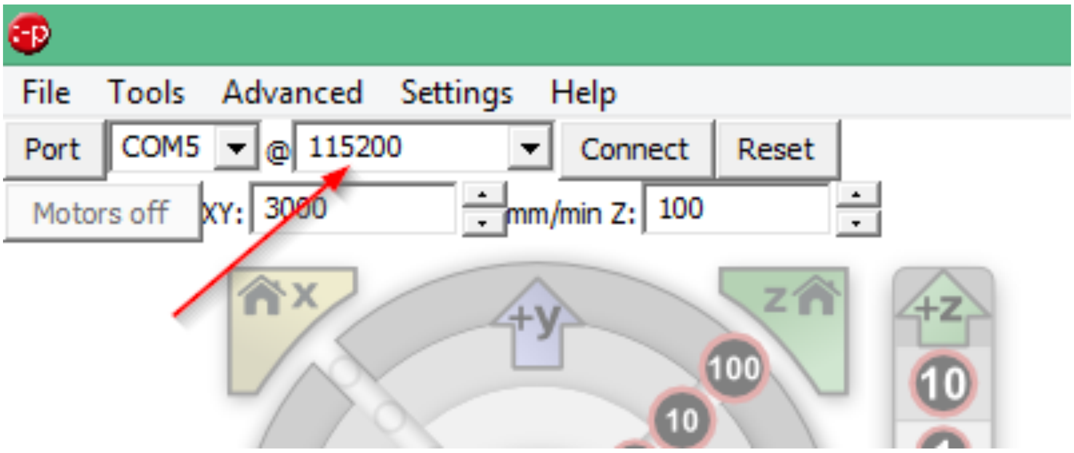
Step 2: Date and name of the author of the changes

To make changes to the code you must indicate the name and the date of the person who initiated the changes.
#define STRING_VERSION_CONFIG_H __DATE__ "27/03/2015 1126h" __TIME__ // build date and time
#define STRING_CONFIG_H_AUTHOR "(Pedro Emanuel, reprap.pt)" // Who made the changes.

In this case replace 27 / 03/2015 1126h' 'by the current date on your computer, and' '(Pedro Emanuel, reprap.pt)' 'by its name and other references.

Step 3: BAUDRATE

When you connect your printer to your computer via USB, this is with a BAUDRATE that determines the printer connection speed, often this value is 115200.
#define BAUDRATE 115200



Port COM5 @ 115200

XY: 3000 mm/min Z: 100

.

Step 4: RAMPS Motherboard

The next step will set the motherboard. In our case the chosen motherboard was' '33 = RAMPS 1.3 / 1.4 (Power outputs: Extruder, Fan, Bed) .
So we chose the value 33:
#define MOTHERBOARD 33
'Note:' If we had two print heads it would be 34.
'Note on latest version of Marlin:' If you use a newer version of the plates are in boards.h and writing The option will be #define MOTHERBOARD BOARD_RAMPS_13_EFB instead of #define MOTHERBOARD BOARD_ULTIMAKER' '.

Step 5: Thermistor

This step is to set the thermistor, as we are using 2 thermistors 100k, one for the heated bed and the other to the print head. The thermistor is who will get the temperature information.
So we chose a value of 1 for both:
#define TEMP_SENSOR_0 1
#define TEMP_SENSOR_1 0
#define TEMP_SENSOR_2 0
#define TEMP_SENSOR_BED 1
Note: 'TEMP_SENSOR_0' 'refers to the print head thermistor and' TEMP_SENSOR_BED for the heated platen. If using a thermistor 10k (not recommended for heated table) would have to choose another option.

Step 6: Temperatures Min and Max

In this case you can set the minimum and maximum temperatures that the thermistor can detect .
minimum :

```
#define HEATER_0_MINTEMP 5
#define HEATER_1_MINTEMP 5
#define HEATER_2_MINTEMP 5
#define BED_MINTEMP 5
```

'Note' HEATER_0_MINTEMP refers to the print head thermistor and 'BED_MINTEMP' 'for the heated platen. Maximum:

```
#define HEATER_0_MAXTEMP 275
#define HEATER_1_MAXTEMP 275
#define HEATER_2_MAXTEMP 275
#define BED_MAXTEMP 150
```

'Note' HEATER_0_MAXTEMP refers to the print head thermistor and 'BED_MAXTEMP' 'for the heated platen.

Step 7: Reverse movement of the motors

In these settings intentionally reverses some directions of the engines. When you have to make the first movement tests and verify that the motor is moving in the opposite direction can alter the value of true to false here.

```
#define INVERT_X_DIR true    // for Mendel set to false, for Orca set to true
#define INVERT_Y_DIR true    // for Mendel set to true, for Orca set to false
#define INVERT_Z_DIR true     // for Mendel set to false, for Orca set to true
#define INVERT_E0_DIR true   // for direct drive extruder v9 set to true, for geared extruder set to false
#define INVERT_E1_DIR false    // for direct drive extruder v9 set to true, for geared extruder set to false
#define INVERT_E2_DIR false   // for direct drive extruder v9 set to true, for geared extruder set to false
```

Step 8: Position of endstops

If your limit switches (endstops) are the root cause when sends the printer to Home which in this case will be: X = 215mm (MAX), Y = 0 mm (MIN) and Z = 0 mm (MIN).
So the code to be applied is:

```
#define X_HOME_DIR 1
#define Y_HOME_DIR -1
#define Z_HOME_DIR -1
```

Where 1 is the maximum position of the X -1 and the minimum Y and Z.

Step 9: Set size of the print area

By default the Prusa i3 is approximately a dimension of 215 mm X, Y, Z of 210 mm and 180 mm. If you have a larger print area you can always change the following values:

```
#define X_MAX_POS 215
#define X_MIN_POS 0
#define Y_MAX_POS 210
#define Y_MIN_POS 0
#define Z_MAX_POS 180
#define Z_MIN_POS 0
```

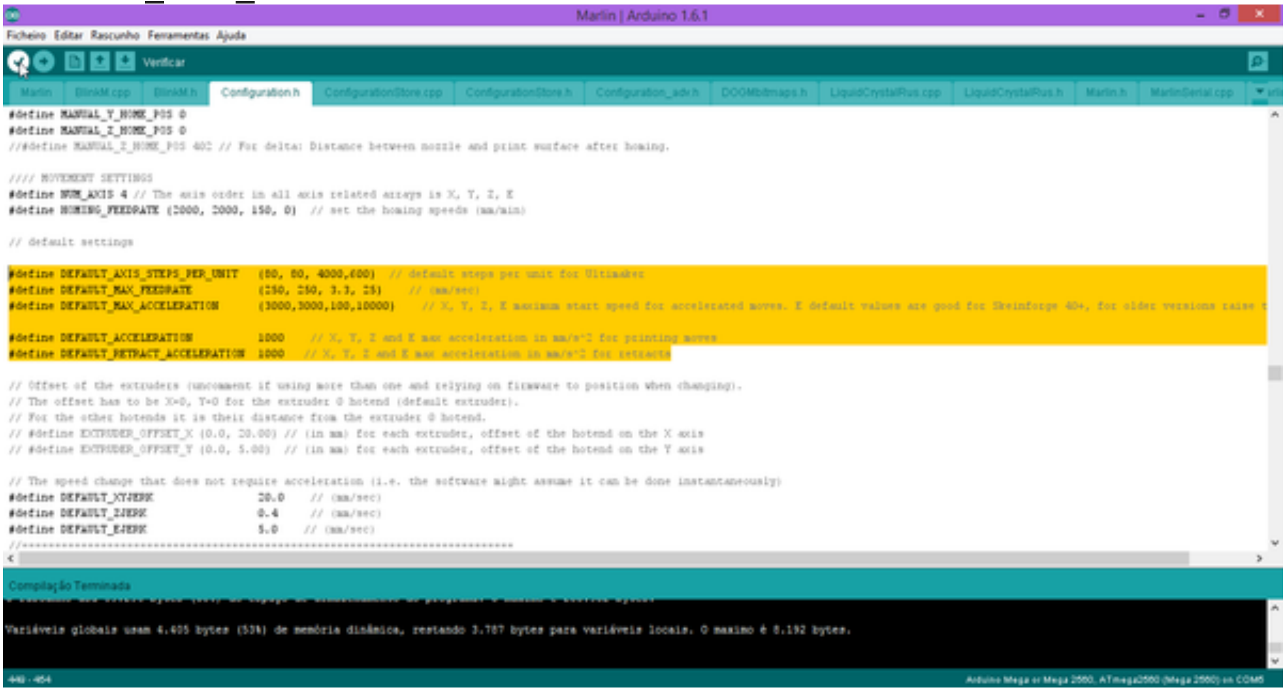
Step 10: Speed of the motor

You can set the speed of the motors, these settings can be improved at the time of calibration, but most of these values are defined in the following calculator [[http://prusaprinters.org/calculator/ Josef Prusa]] .

// prusa i3 settings
#define DEFAULT_AXIS_STEPS_PER_UNIT   {80,80,4000,600} // default steps per unit for prusa i3 rework
#define DEFAULT_MAX_FEEDRATE   {500, 500, 2, 25}   // (mm/sec)
#define DEFAULT_MAX_ACCELERATION   {2000,2000,20,1000}   // X, Y, Z, E maximum start speed for accelerated moves. E default values are good for skeinforge 40+, for older versions raise them a lot.

Note : these values work for the stepper drivers A4988. If you use DRV8825 drivers, the STEPS_PER_UNIT values must be changed to double.

At the time of calibration recommend to draw a mark on the desired axis (X , Y or Z) and moving 10 cm and see how mm it moves, then we can get the correct value DEFAULT_AXIS_STEPS_PER_UNIT comparing with the result.



```
#define MANUAL_Y_HOME_POS 0
#define MANUAL_Z_HOME_POS 0
//#define MANUAL_Z_HOME_POS 402 // For delta: Distance between nozzle and print surface after homing.

//// MOVEMENT SETTINGS
#define NUM_AXIS 4 // The axis order in all axis related arrays is X, Y, Z, E
#define HOMING_FEEDRATE {3000, 3000, 150, 0}  // set the homing speeds (mm/min)

// default settings

#define DEFAULT_AXIS_STEPS_PER_UNIT   {80, 80, 4000,600}  // default steps per unit for Ultimaker
#define DEFAULT_MAX_FEEDRATE          {250, 250, 3.3, 25}    // (mm/sec)
#define DEFAULT_MAX_ACCELERATION      {3000,3000,100,10000}    // X, Y, Z, E maximum start speed for accelerated moves. E default values are good for Skeinforge 40+, for older versions raise t

#define DEFAULT_ACCELERATION          1000    // X, Y, Z and E max acceleration in mm/s^2 for printing moves
#define DEFAULT_RETRACT_ACCELERATION  1000   // X, Y, Z and E max acceleration in mm/s^2 for retracts

// Offset of the extruders (uncomment if using more than one and relying on firmware to position when changing).
// The offset has to be X=0, Y=0 for the extruder 0 hotend (default extruder).
// For the other hotends it is their distance from the extruder 0 hotend.
// #define EXTRUDER_OFFSET_X {0.0, 20.00} // (in mm) for each extruder, offset of the hotend on the X axis
// #define EXTRUDER_OFFSET_Y {0.0, 5.00}  // (in mm) for each extruder, offset of the hotend on the Y axis

// The speed change that does not require acceleration (i.e. the software might assume it can be done instantaneously)
#define DEFAULT_XYJERK                20.0    // (mm/sec)
#define DEFAULT_ZJERK                 0.4     // (mm/sec)
#define DEFAULT_EJERK                 5.0    // (mm/sec)
//===========================================================================
```

```
Variáveis globais usam 4.405 bytes (53%) de memória dinâmica, restando 3.787 bytes para variáveis locais. O maximo é 8.192 bytes.
```

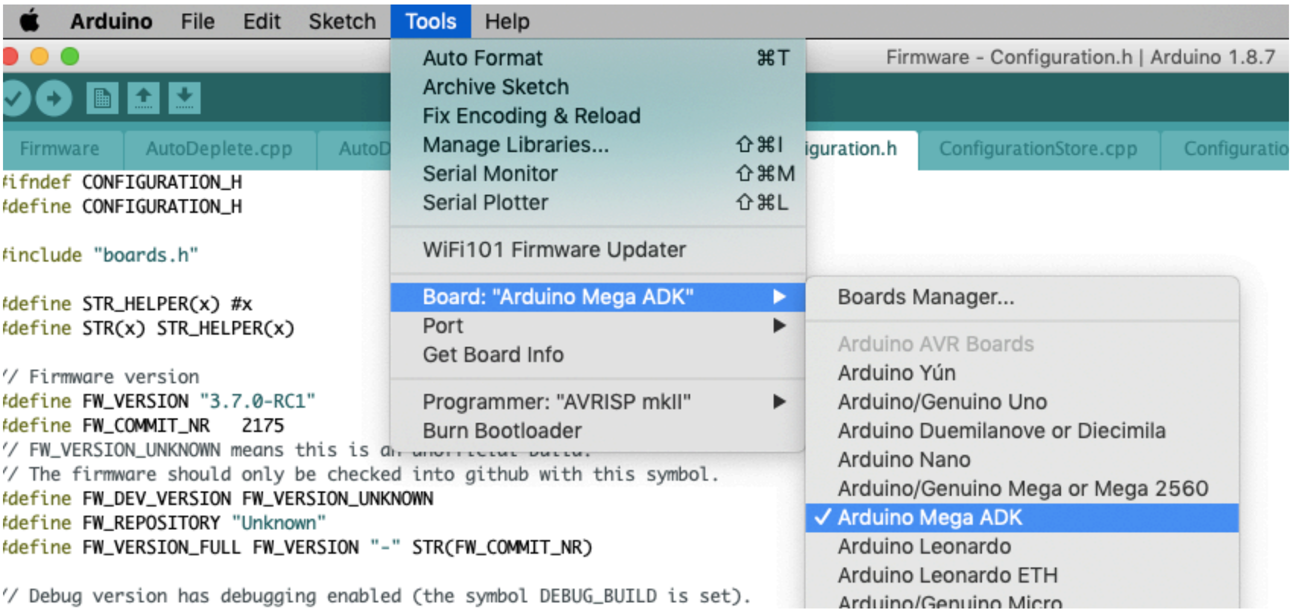
Install Marlin Firmware on the board

Step 1: Choose Board

Connect USB cable to the electronic assembly to the computer (this is connected to a USB port on the Arduino Mega). Wait for the system to install the drivers.

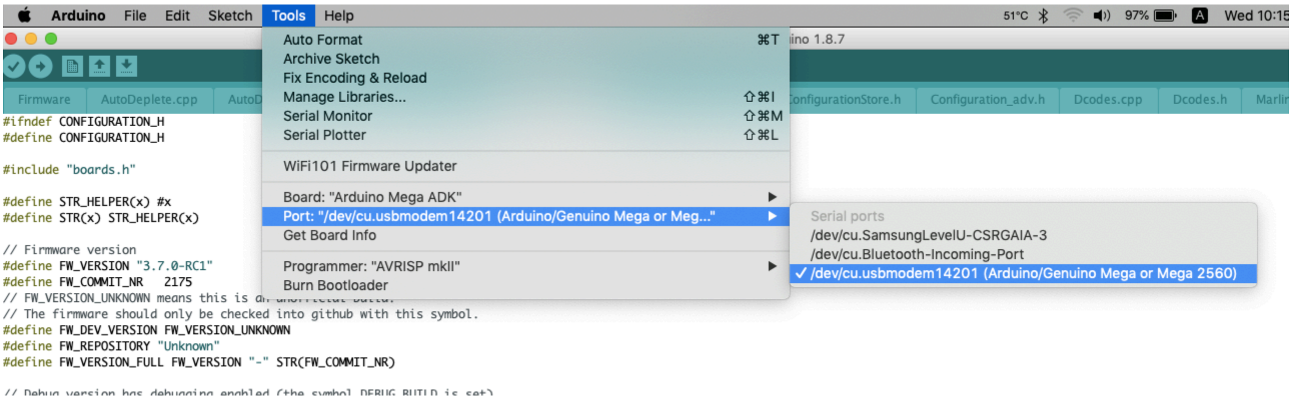
In the Arduino software choose 'Tools> Board> Arduino Mega or Mega 2560' '

Step 2: Choose Serial COM Port



```
#ifndef CONFIGURATION_H
#define CONFIGURATION_H

#include "boards.h"

#define STR_HELPER(x) #x
#define STR(x) STR_HELPER(x)

// Firmware version
#define FW_VERSION "3.7.0-RC1"
#define FW_COMMIT_NR   2175
// FW_VERSION_UNKNOWN means this is an unofficial build.
// The firmware should only be checked into github with this symbol.
#define FW_DEV_VERSION FW_VERSION_UNKNOWN
#define FW_REPOSITORY "Unknown"
#define FW_VERSION_FULL FW_VERSION "-" STR(FW_COMMIT_NR)

// Debug version has debugging enabled (the symbol DEBUG_BUILD is set).
```

When connecting my computer has assigned the number COM5 to the Serial Port of my Arduino, in this case it may have assigned a different number. Avoid having other peripherals connected in order to easily find the right port of your Arduino.
If you have difficulties finding the COM Port, use the Windows Device Manager and connect your Arduino to the COM Port that you find.
In the Arduino software choose 'Tools> Port> COM5' '

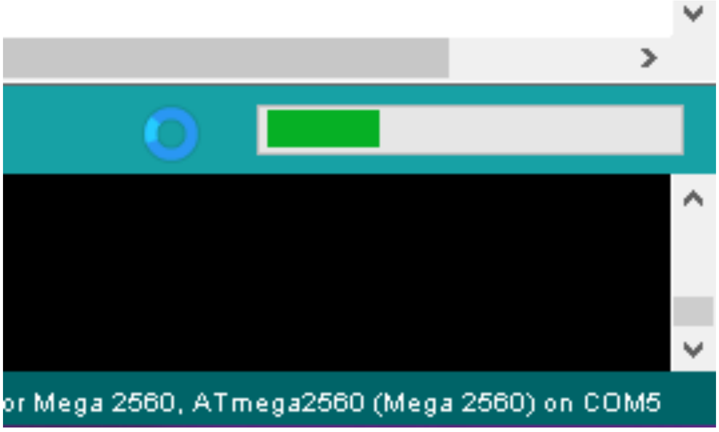


```
#ifndef CONFIGURATION_H
#define CONFIGURATION_H

#include "boards.h"

#define STR_HELPER(x) #x
#define STR(x) STR_HELPER(x)

// Firmware version
#define FW_VERSION "3.7.0-RC1"
#define FW_COMMIT_NR   2175
// FW_VERSION_UNKNOWN means this is an unofficial build.
// The firmware should only be checked into github with this symbol.
#define FW_DEV_VERSION FW_VERSION_UNKNOWN
#define FW_REPOSITORY "Unknown"
#define FW_VERSION_FULL FW_VERSION "-" STR(FW_COMMIT_NR)

// Debug version has debugging enabled (the symbol DEBUG_BUILD is set).
```

Step 3: Compile and verify the firmware

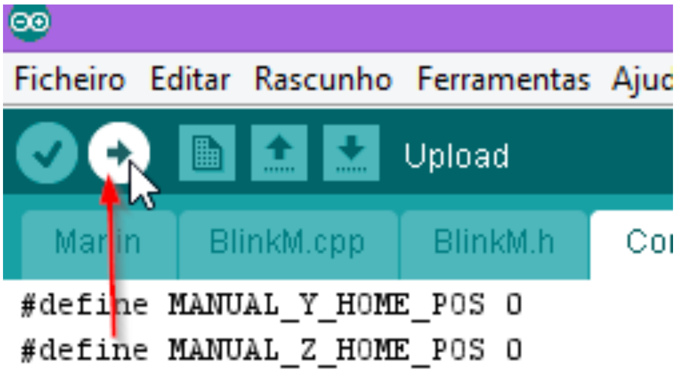
Before making the Upload be sure to test to see if everything is OK by pressing the round button in the upper left corner in the shape of a checkmark inside:

When checking a progress bar shows the results below. If it is an error, and if it is not the first time you are following this process, make sure there is Pronterface connected to the printer or other application.



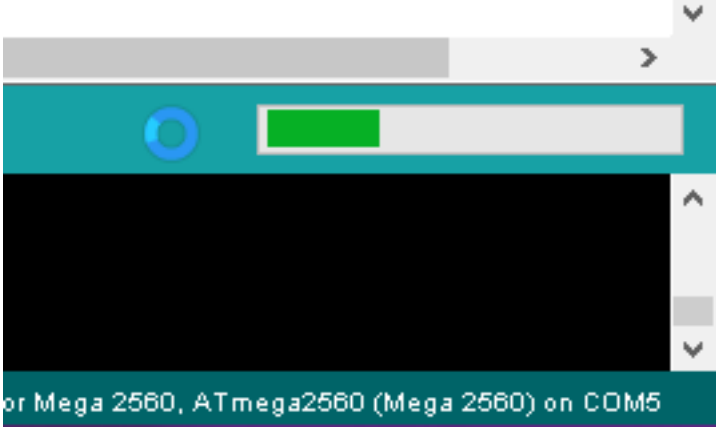
Progress Bar

Step 4: Upload to Arduino



```
#define MANUAL_Y_HOME_POS 0
#define MANUAL_Z_HOME_POS 0
```

Upload



Progress Bar

Your firmware is now uploaded. No need to make this process more than necessary, unless you want to change the permanent settings to the Printer or components.

(Note from user c8888:)

If you encounter any problems while uploading firmware (for example: avrdude: stk500v2_ReceiveMessage(): timeout), the first thing to do is to change your (long - 2m) USB cable for a shorter (50cm) one.