

# RepRapPro Mendel maintenance

From RepRapWiki

[Introduction](#) | [Frame assembly](#) | [Y axis assembly](#) | [X axis assembly](#) | [Z axis assembly](#) | [Heated bed assembly](#) | [Extruder drive assembly](#) | [Hot end assembly](#) | [Power supply](#) | [Wiring](#) | [Commissioning](#) | [Printing](#) | [Multi-colour/multi-materials](#) | [Colour Printing - Slic3r](#) | [Colour Printing - RepRapPro Slicer](#) | **Maintenance** | [Troubleshooting](#) | [Improvements](#)

## Contents

[\[hide\]](#)

- [1 Uploading new firmware](#)
  - [1.1 Required software](#)
    - [1.1.1 Arduino IDE](#)
    - [1.1.2 Sanguino hardware](#)
    - [1.1.3 The 'Bootloader'](#)
  - [1.2 Git repository](#)
    - [1.2.1 Downloading from Github](#)
  - [1.3 Uploading firmware to your Melzi](#)
    - [1.3.1 Huxley / Mendel / Mono Mendel / Tricolour Mendel Master - Marlin Firmware Configuration](#)
    - [1.3.2 Marlin Firmware Upload](#)
    - [1.3.3 Multimaterials / Tricolour Mendel Slave - Marlin Firmware Configuration](#)
  - [1.4 Simple Melzi Test](#)
- [2 Dismantling the hot end](#)

## Uploading new firmware

The firmware is the computer program that resides in the microcontroller chip on the controller printed circuit board. This section tells you how to upload a new copy of that program to your RepRapPro Melzi or Sanguinololu controller.

**Important:** If you have a multi-colour/multi-material RepRapPro RepRap, then you must first disconnect one end of the 3-way communications wire between the two controllers before you reload firmware into either of them. Reconnect it when you have finished reloading the firmware.

## Required software

## Arduino IDE

Arduino is the architecture of the main processor/microcontroller (see <http://en.wikipedia.org/wiki/Arduino> )

The Arduino IDE (Integrated Development Environment) allows you to upload the firmware to the board. Get it from [the Arduino website here](#).

**The latest version of the firmware, Marlin V1.0.4, is proven with Arduino v1.04.**

You may also need to install Java.

## Sanguino hardware

What is Sanguino?! Sanguino is the basic architecture of the board (see <http://sanguino.cc/> ). Sanguinololu is Sanguino, with the addition of stepper drivers. Melzi is based on Sanguinololu, so needs some extra files so Arduino IDE can talk to the Melzi.

The Sanguino drivers are located within the Marlin folder of the firmware. Copy the 'Sanguino' folder to your /arduino-1.0.4/hardware/ folder.

## The 'Bootloader'

At times, you may see references to the 'bootloader'. This is a piece of code that is flashed to the Arduino chip before firmware, to allow basic communications. **If you have a RepRapPro machine, this has been done for you.** It is part of the Sanguino code, and it tells the chip how to talk to the USB port; without it, it doesn't know how to! Flashing firmware doesn't replace this or require that you flash the bootloader as well; it stays there all the time.

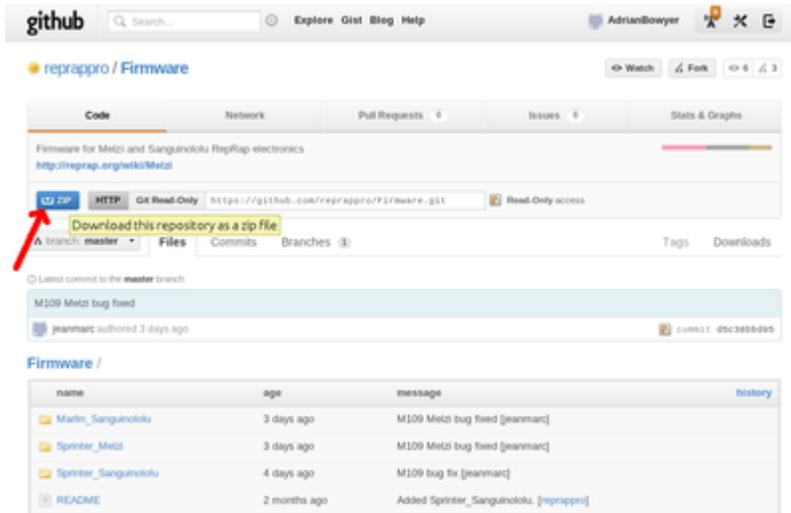
The only reason to reflash the bootloader is if the EEPROM on the Melzi has been completely erased somehow, or if you have replaced the chip on the board.

## Git repository

Most of the data you need for RepRapPro hardware and software is in our Github repository. Firmware source code is stored in the [RepRapPro git repository for Huxley and monochrome Mendel](#) or [RepRapPro git repository for multi-material/multi-colour Mendel](#).

## Downloading from Github

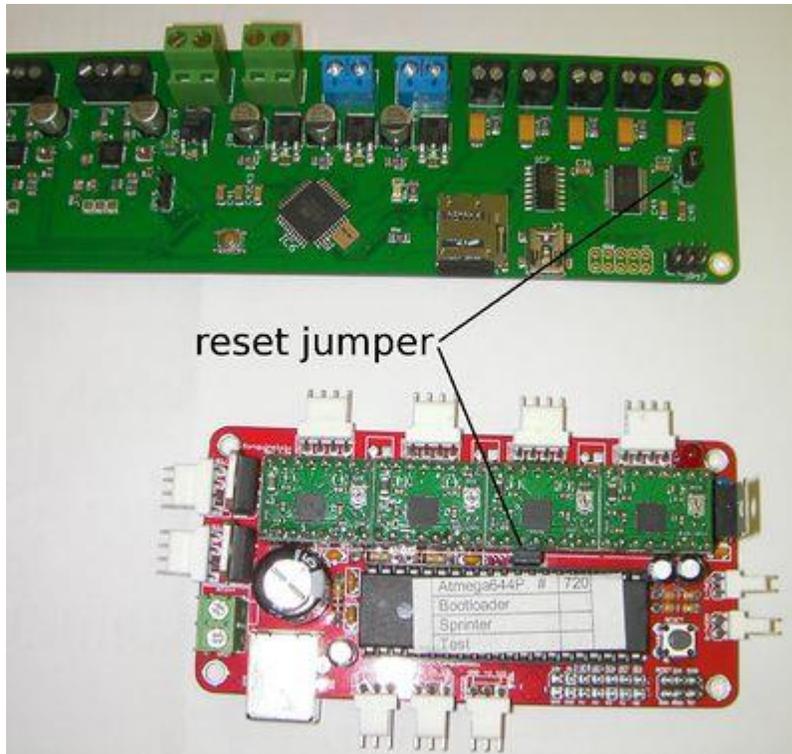
Navigate to the appropriate page for the firmware you need (see above). Then download your software with the button on the top-left to get it as a .ZIP file:



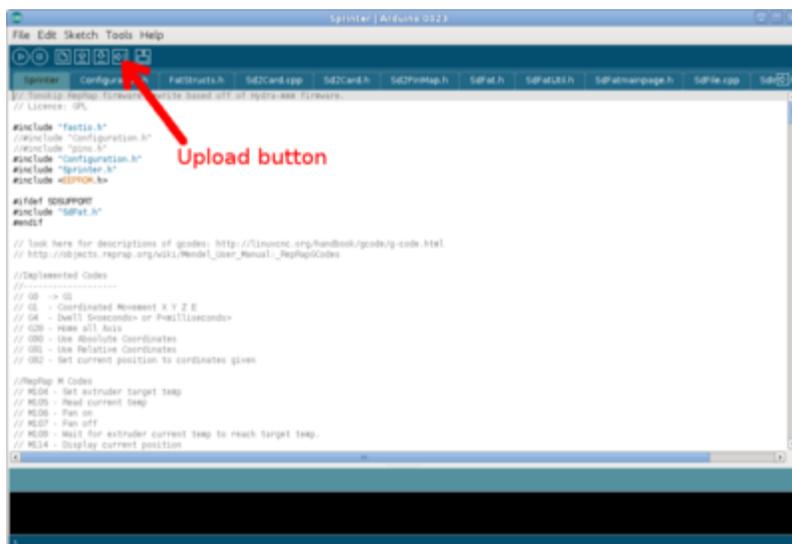
Download the zip file appropriate to your machine as shown above (the little picture of a cloud with a down-arrow labelled ZIP) and then extract the ZIP to your drive. It will create a folder called 'Marlin'. You can move this into your ~/sketchbook folder to make it easily visible within the Arduino IDE. The Arduino software is much easier to use when it knows where you keep the RepRapPro files, so use File/Preferences to set your preferred location.

## Uploading firmware to your Melzi

In order to upload firmware into the controller you need to make sure that the board's reset jumper is installed:



You can take it off when you have finished uploading. Hang it on one of the pins so it doesn't get lost.



Select the Sanguino board from the Tools | Board menu. You will need to select "Sanguino W/ATmega1284p 16 mhz". (If you can't see the Sanguino boards you will need to check that you have downloaded the right files and moved them to the correct folder - see above).

Ensure the serial port is ticked from the Tools | Serial port menu (the RepRap controller board must be physically connected to your computer with the USB cable at this stage).

## **Huxley / Mendel / Mono Mendel / Tricolour Mendel Master - Marlin Firmware Configuration**

Select the tab for the file **Configuration.h**. At the top are the following lines:

```
// Uncomment ONE of the next three lines - the one for your RepRap machine
//#define REPRAPPRO_HUXLEY
//#define REPRAPPRO_MENDEL
//#define REPRAPPRO_WALLACE

// Uncomment ONE of the next two lines - the one for your master controller
electronics
//#define REPRAPPRO_MELZI
//#define REPRAPPRO_SANGUINOLOLU

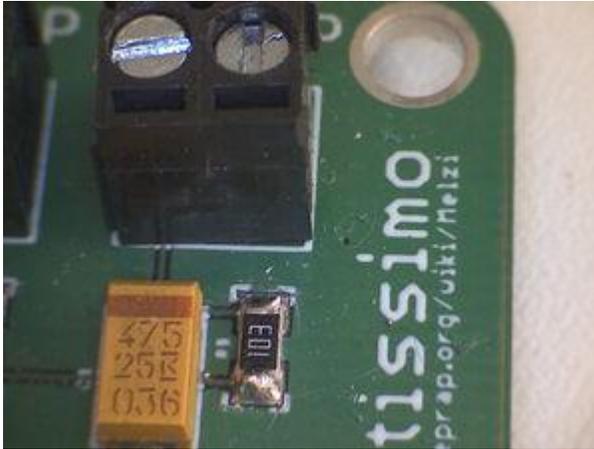
// Uncomment ONE of the next two lines - the one for the series resistors on
your controller
//#define SERIAL_R 4700
//#define SERIAL_R 10000

// Uncomment the next line if your machine has more than one extruder
//#define REPRAPPRO_MULTIMATERIALS
```

Uncomment (*i.e.* remove the two // characters) from the line corresponding to your machine and your control electronics.

If you have a Sanguinololu controller uncomment the  
#define SERIAL\_R 4700  
line.

If you have a Melzi controller you need to look at the resistor value on the printed circuit board:



The screw connector at the top of the picture is the one for the hot-end thermistor. Just behind it you can see a small black resistor with 103 written on it. That is a 10K resistor (the last digit is the number of zeros - sometimes it will say 1002, which is obviously also 10K). If your board has that, uncomment the

```
#define SERIAL_R 10000
```

line. If it has one labelled 472, uncomment the

```
#define SERIAL_R 4700
```

line.

If you have a multi-material/multi-colour or mixer-extruder Mendel uncomment the

```
//#define REPRAPPRO_MULTIMATERIALS
```

line.

Further on you will see the following section:

```
// X, Y, Z, E steps per mm
```

```
#define DEFAULT_AXIS_STEPS_PER_UNIT {91.4286, 91.4286, 4000, 875}
```

These four values allow you to store accurate settings for your printer. Once you have calibrated your printer, particularly the fourth value for Extruder steps, you can edit this line.

The values for the X and Y axis are calculated using the following formula:

$DEFAULT\_AXIS\_STEPS\_PER\_UNIT = (Steps\ per\ rotation * Microsteps\ per\ step) / (Pulley\ teeth * Belt\ pitch)$ . The stepper motors supplied with the kit are 1.8 degree/step motors, which is 200 steps per revolution, and 1/16th microsteps. The pulleys are 14 teeth and the belt pitch is 2.5mm. Thus this figure  $(200 * 16) / (14 * 2.5) = 91.4286$ . If you wish to use 16 teeth pulleys, for example, the figure becomes  $(200 * 16) / (16 * 2.5) = 80$ .

## Marlin Firmware Upload

To upload your firmware, click the Play button (the first button) to verify that the code compiles correctly. It should respond with:

```
Done compiling.  
Binary sketch size: [XXXX] bytes (of a [XXXX] byte maximum)
```

If you don't get this, you'll get an error message. Check you installed all the necessary software parts, that you made no errors in the firmware editing, and that you selected the right option under the Tools | Boards and Tools | Serial Ports menus.

Click Upload (the second with 1.0) to send the firmware to your Melzi. A message will show that the firmware is compiling as above, then uploading. Once it has finished, it should say "Done uploading" and the LED will come on on the Melzi.

If you get an error message, try uploading again. Check that you installed the autoreset jumper - it is needed for firmware uploading. Remove it after the firmware has uploaded.

## **Multimaterials / Tricolour Mendel Slave - Marlin Firmware Configuration**

The slave board on a Multimaterials / Tricolour Mendel uses different firmware. In Arduino, open the firmware 'Slave.ino' in the 'Marlin | Slave' folder. No configuration is necessary. Remove the master to slave communication cable, and install the autoreset jumper. Plug the USB cable directly into the slave Melzi, and upload as above.

## **Simple Melzi Test**

Having trouble getting uploads to work? Follow these steps...

Here is a short program that should cause the Melzi LED to blink and nothing else:

```
const int ledPin = 27;  
int ledState = LOW;  
long previousMillis = 0;  
long interval = 1000;  
  
void setup() {  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop()  
{  
  unsigned long currentMillis = millis();  
  
  if(currentMillis - previousMillis > interval) {  
    previousMillis = currentMillis;  
    if (ledState == LOW)  
      ledState = HIGH;  
    else  
      ledState = LOW;  
    digitalWrite(ledPin, ledState);  
  }
```

```
}  
}
```

Make sure you are running Arduino version 1.04(<http://arduino.cc/en/Main/Software>) and that you have installed the Sanguino files.

Leave the main RepRap power supply disconnected.

Connect the power jumper on your Melzi (the one near the middle of the board) for USB power (jumper the two pins of the three on the reset button side), and connect the reset jumper (the one at the **etemp** connector end of the board). Plug the Melzi USB into your computer and run the Arduino development environment.

Under Tools->Board select "Sanguino W/ATmega 1284p 16 mhz".

Copy and paste the above program into the window and select upload (the button with the right-pointing arrow). If it fails, unplug the USB lead from the computer, wait a few seconds, plug it back in, and try again.

The program should compile then upload, and the on-board LED should then blink at 1 Hz.

Of course, you now have to upload the proper firmware, as described above...