

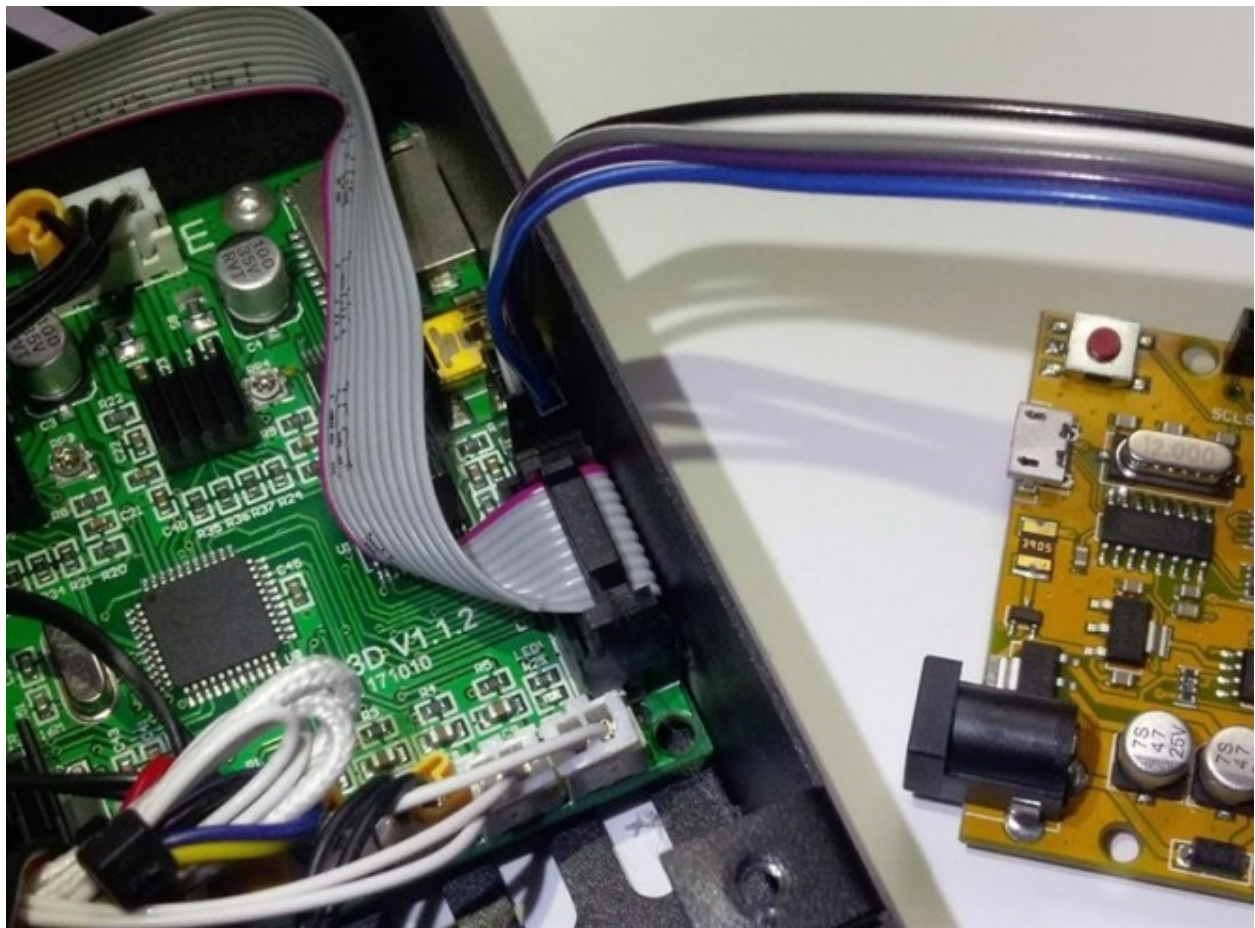
How to Flash a Bootloader on Melzi Boards

Last Updated July 22, 2018 by Brett

If you want to add new features on your 3D Printer, such as Auto Bed Leveling or Power Loss Resume, you will probably need to flash new firmware on the electronics board. The problem is that in order to do so, the board must already have a bootloader pre-installed. Melzi boards, often used in budget 3D printers like the CR-10, Ender-3 and Maker Select are almost exclusively distributed without them. Since this is necessary to upload firmware via USB, we will look at one of the easiest ways to do it.

Most importantly, we will need a device that can interact with and write code to our boards. We can use anything from an AVR programmer to a Raspberry Pi, but to keep things simple and cheap, an Arduino Uno and a few breadboard jumper wires will get the job done.

The best part... a bootloader is permanent. Once it has been burned on your board, you will never have to do it again. From that point, firmware can be flashed over USB as many times as needed. When you are finished, the Arduino is no longer required and can be repurposed for other fun projects instead.



Purchased Parts List

[Arduino Uno R3](#) - \$

[Breadboard Jumper Wires](#) - \$3.99

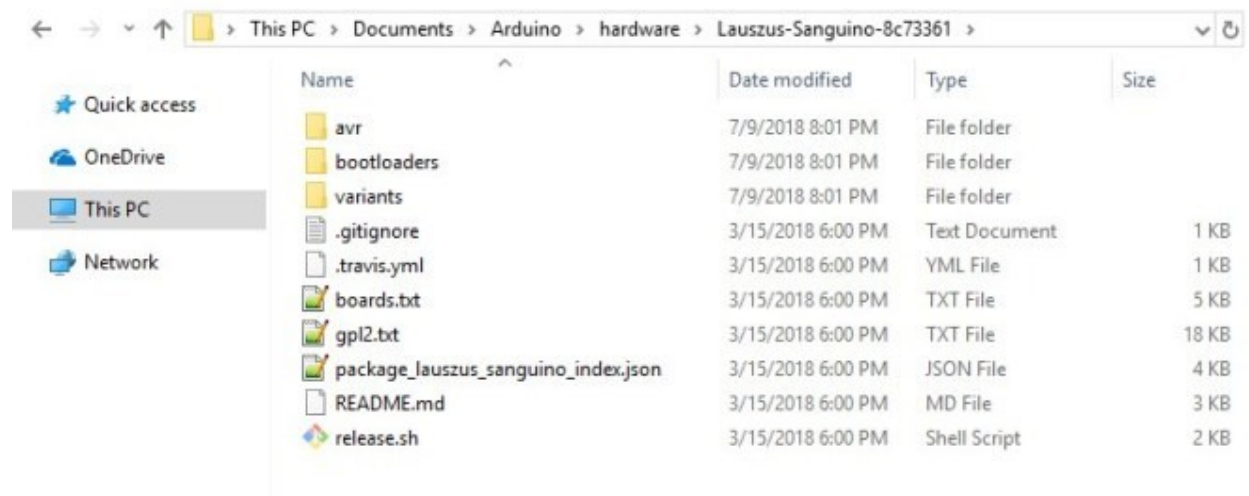
Getting Started

Before we dive in, we first need to download the Arduino IDE software. This is a basic programming tool that can write code to various electronic devices. There are builds available for Windows, Mac and Linux, so just grab the one that works for your computer and run through the setup process.

We also need to download a 3rd party package called Sanguino, which adds support to the Arduino IDE for these particular boards. Download the ZIP file from the following page, and extract it to the hardware folder in the Arduino directory. For Windows, this is located at *Documents > Arduino > hardware*. If the hardware folder doesn't exist, go ahead and create it, then extract the ZIP file as shown in the image below.

[Arduino IDE Downloads](#)

[Sanguino Downloads](#)



If the Arduino IDE is already open, restart the program and check that the Sanguino package is now loaded. To confirm, click the *Tools menu -> Board* and scroll down to the bottom of the list, where it should have Sanguino as the last option. If the Sanguino package is for some reason missing from the Arduino IDE, double check the paths to ensure it was installed in the correct location.

Making an Arduino ISP

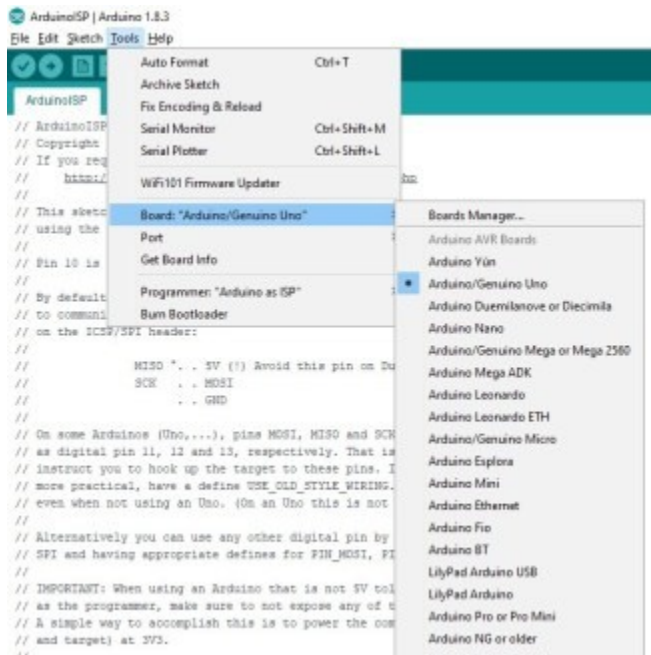
The Arduino IDE makes it simple to turn our Arduino Uno into an ISP (serial programmer) device, which can write a bootloader to the Melzi board. We just need to configure a few quick options in the software, upload the code and voila... we have created a working Arduino ISP to use.

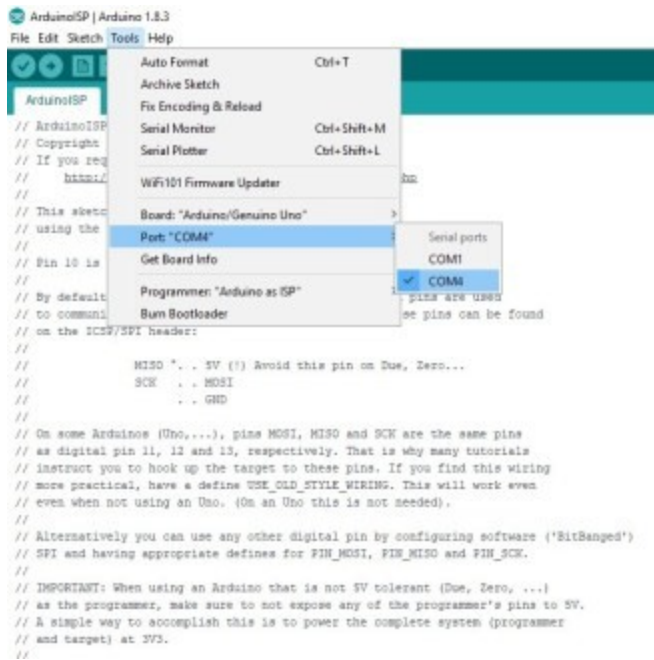
With the Arduino IDE opened on your computer, go to the *File menu -> Examples -> ArduinoISP -> ArduinoISP*. This will open a new sketch window containing the ISP code, which is what we will write to our Arduino Uno board. We will just set the the board, COM port and programmer before uploading it.

Board: To configure the board, go to the *Tools menu -> Board -> Arduino/Genuino Uno*.

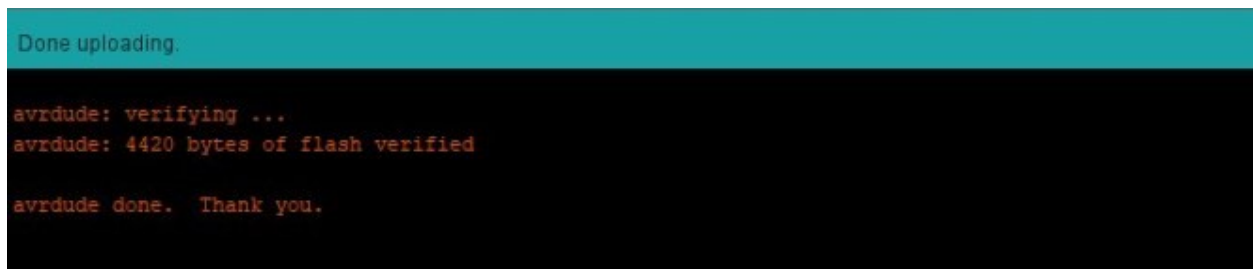
Port: To configure the port, go to the *Tools menu -> Port* and see what ports currently active. Plug the Arduino Uno board into the computer via USB cable and check the list again, which should now have a new COM port listed. This is your Arduino Uno board and you can go ahead and select it.

Programmer: To configure the programmer, go to the *Tools menu -> Programmer -> AVRISP mkII*.





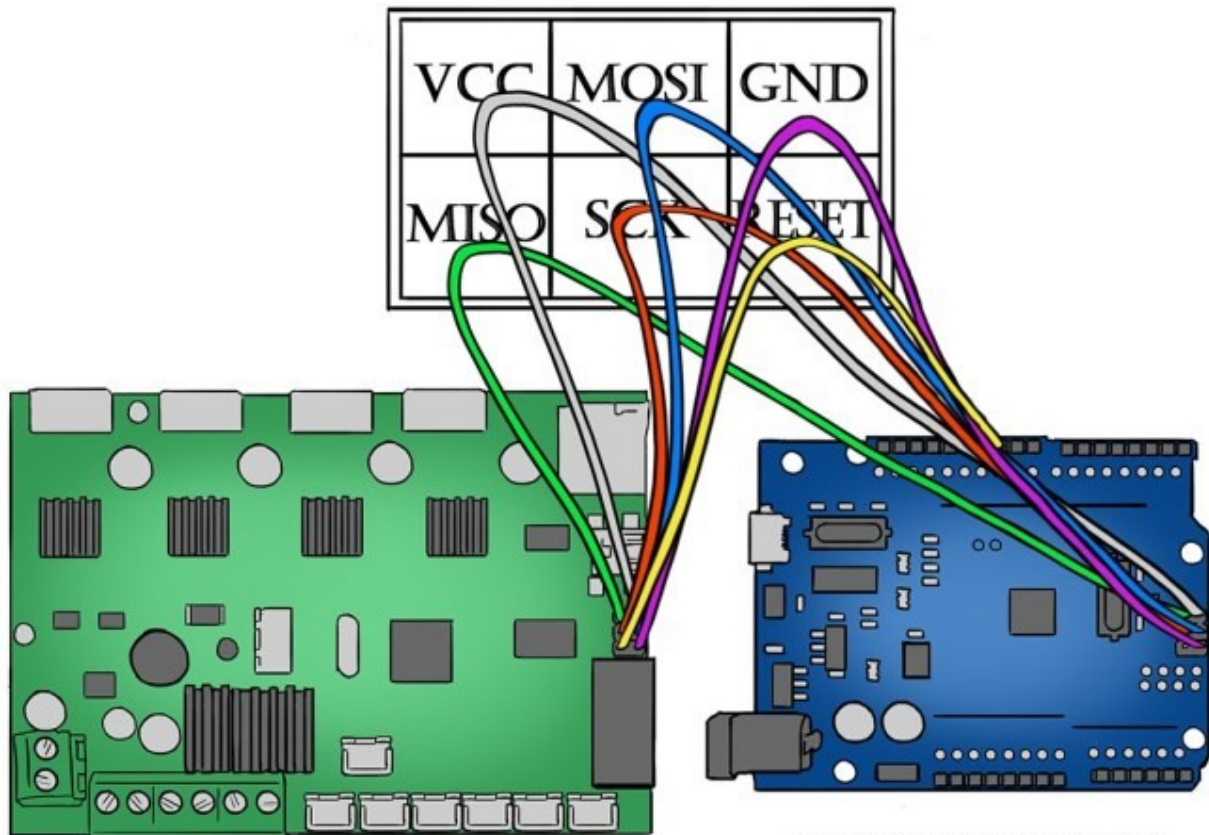
At this point, we are ready to go ahead and write the code to our Arduino Uno board. Go to the *Sketch menu* -> *Upload* (or press Control + U) to initiate the process, which should take around 10 seconds to complete. The upload progress is shown at the bottom of the screen and will confirm when it's done.



Wiring the Boards

Now that our Arduino Uno is a working programmer (ISP), we can just connect it up to the Melzi board and burn a bootloader to it. Before moving forward, make absolutely sure to power off and disconnect the electronics while we install the jumper wires, otherwise there is a risk of causing damage to the components.

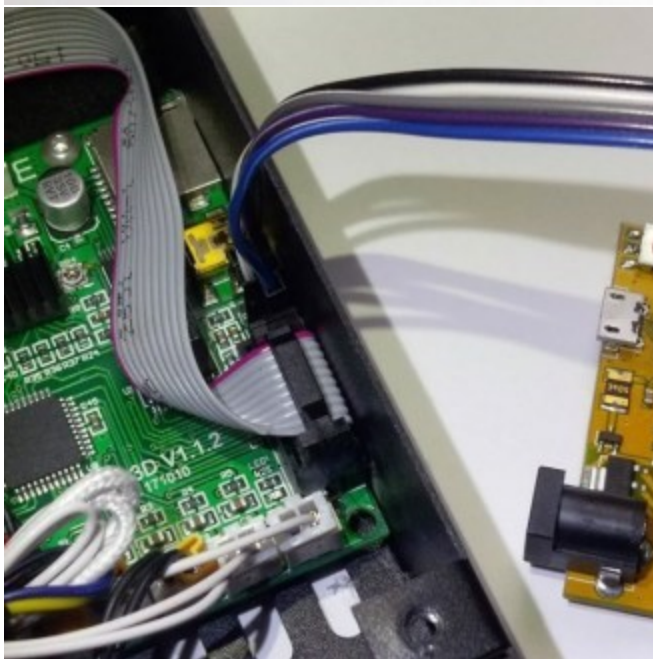
We will need (5) female to female wires and (1) female to male wire. If you purchased an assorted pack of breadboard jumper wires, these often come attached together so just tear off a strip with the amount that you need. It doesn't matter what colors are used but they will make it easier to tell the wires apart, ensuring they are placed on the correct pins.



COPYRIGHT © 2018 LetsPrint3D.net

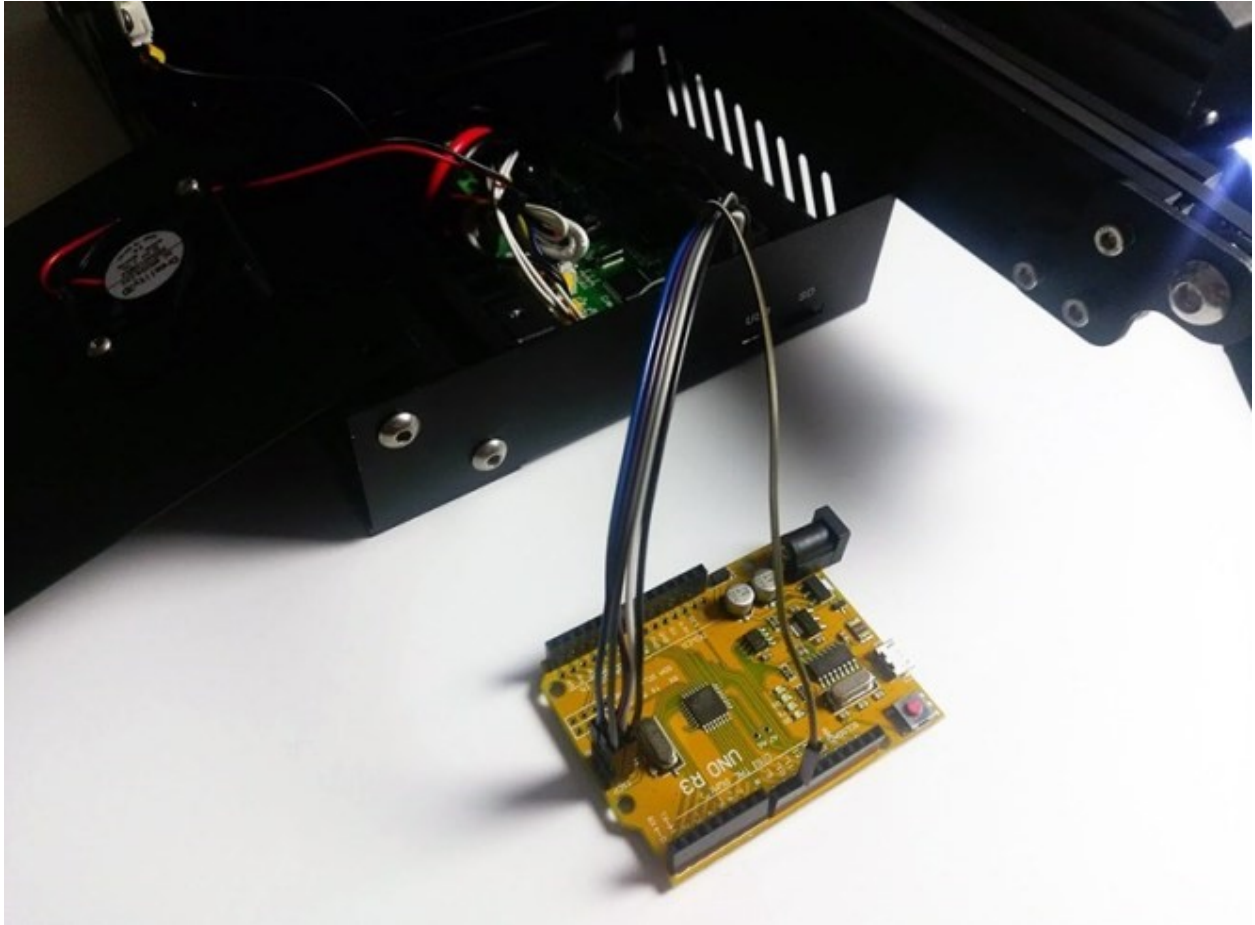
Using the diagram above as reference, locate the 6 pins on both the Melzi board and the Arduino Uno. We will connect 5 of these using (1) female/female jumper wire each so they are the same across boards. The one exception is the RESET pin, indicated by a yellow wire in the diagram.

For the RESET pin, we need to instead use the female/male jumper wire, connecting the female end to the Melzi board and the male end to Digital Pin 10 on the Arduino Uno. This is located in the black connector on the side of the board, with the number 10 written next to it.



It's important to double and triple check the wiring at this point, making sure the connectors are well seated and have been placed on the right pins. Board orientation can make things a bit confusing here, especially if the Melzi is left inside of the case.

I mistakenly switched two jumper wires the first time, missed it once again when taking a second look and still didn't notice until I was about to power up. It's better to be safe than sorry, so do whatever you need to confirm everything is correct.



Flashing the Bootloader

With the hard part out of the way, we are now ready to flash the bootloader on our Melzi board. Go ahead and plug the Arduino Uno into the computer using a USB cable, which should be enough to power both boards. If this is not the case, you can also plug the Melzi board in to USB as well, but keep track of the COM ports assigned to each device.

Open up the Arduino IDE software once more, this time making several adjustments to reflect our new configuration. The Melzi board is our target device, while the Arduino Uno uploads the data to it. Tweak the software settings to match the following...

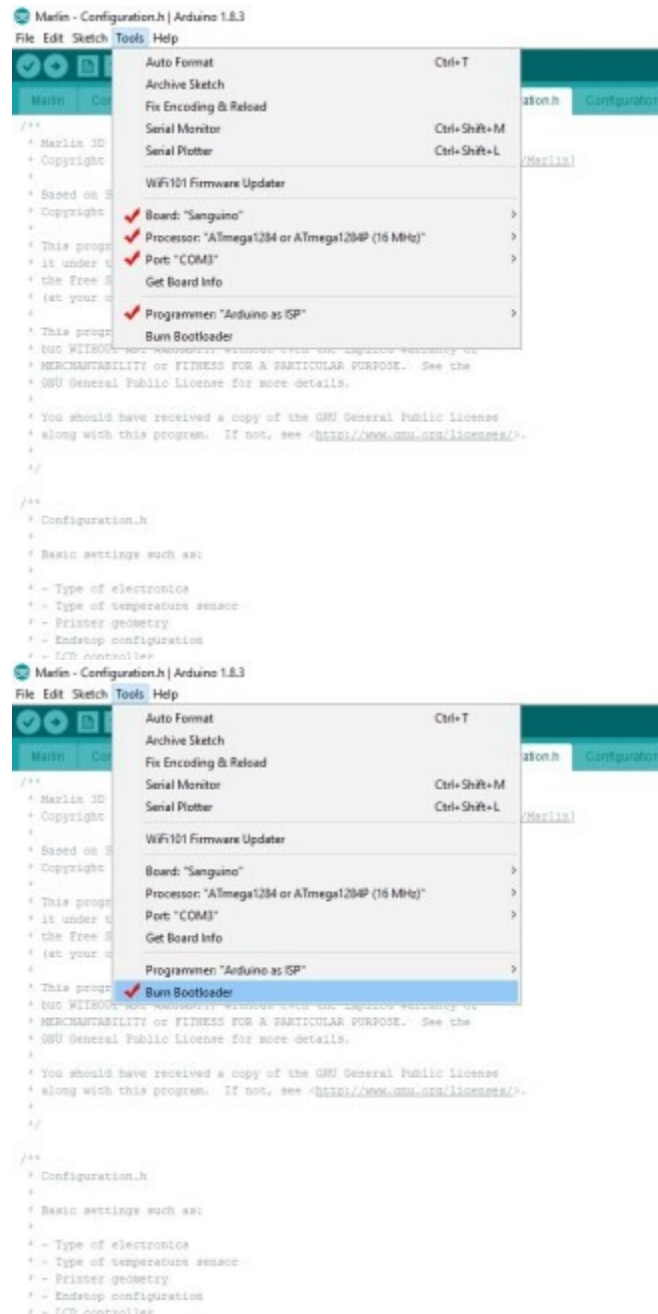
Board: go to the *Tools menu -> Board -> Sanguino*.

Processor: go to the *Tools menu -> Processor -> ATmega1284 or ATmega1284P (16 MHz)*.

Port: go to the *Tools menu -> Port* and set the COM port for the Arduino Uno.

Programmer: go to the *Tools menu -> Programmer -> Arduino as ISP*.

With everything set, all we have left to do is burn the bootloader to our Melzi board. Go to the *Tools menu -> Burn Bootloader* and wait for it to complete. This should only take a few seconds to do, and will report once it has finished at the bottom.



If it was a success, you can power down the electronics and remove the jumper wires. Keep in mind that the existing firmware has been erased, so turning on the 3D printer will result in a blank blue screen for now. This is completely normal so don't panic, you are ready to flash new firmware!

