Setting up the Hardware and Software for a 3D Printer

Caroline Kerbelis

Application Note

March 28th 2014

Design Team 8

ECE 480

Abstract

The purpose of team 8's project is to design, fabricate, simulate, test and demonstrate a specialty printer that will be used to produce tactile graphics and maps for visually impaired students at MSU. Ideally this printer will produce durable images and be more feasible than current specialty printers available commercially. Extensive knowledge of 3D printer hardware and software and how they both interacted with each other is essential to the success of the project.

Table of Contents

Introduction

3D printing, also known as additive manufacturing, is a process consisting of making a solid three-dimensional object from a digital model. This form of printing is achieved by using an additive process, where successive layers of material are added to a surface in different shapes to create a particular object, figure or graphic. The image is created using a Computer-Aided Design program (CAD) and then saving this file as a STereoLithography (STL) file. The file is then transferred to a g-code generator program which converts the file containing the image into a set of instructions/language that the printer understands. The 3D printer then used Computer Aided Manufacturing (CAM) software that reads the file and send electronic signals to control uses an extruder and motors that move along a x, y, and z axis to extrude a filament onto a surface creating the object through sequential layering.

Key Words

Hardware- RepRap Arduino MEGA 2560 Pololu Shield, Stepper Motors, Stepper Drivers, End Stops, and Thermistor

Software- Computer Aided Design, Computer Aided Manufacturing, STereoLithography, Constructive Solid Geometry, and G-code

Electronics

In general, for 3D printing electronics are broken down into 4 different areas, the controller, stepper motors, stepper drivers, and end stops. The brain of the operation, the controller, is a stack of circuit boards that translates computer commands into printing actions. Most commonly that controller is an Arduino microcontroller with an add-on board, such as the RepRap Arduino MEGA 2560 Pololu Shield or RAMPS for short. It is designed to fit the entire electronics needed for a 3D printer in one small package (See figure 1 below).

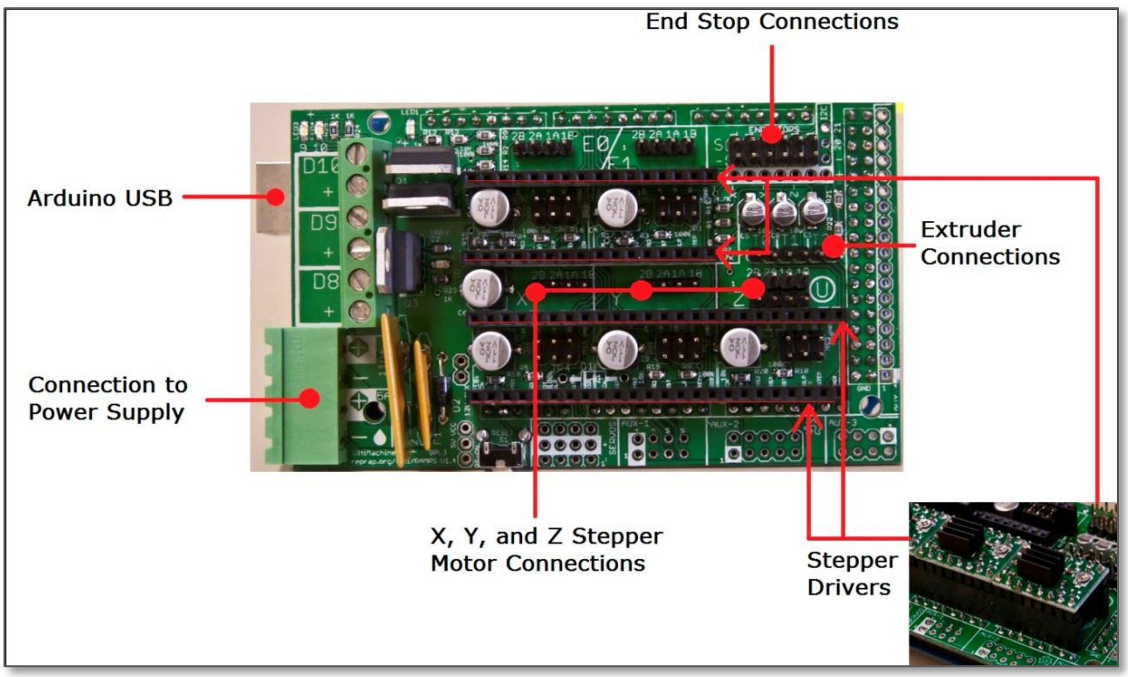

Figure 1: RAMPS Board

The design includes plug in stepper drivers and extruder control electronics on an Arduino MEGA shield for easy service and part replacement. The stepper drivers are chips that act as a middle-man between a stepper motor and the controller. They simplify the signals that need to be sent to the stepper motor in order to get it to move. The stepper drivers are on small circuit boards that plug directly into the controller itself. Each stepper driver is connected to a stepper motor. The stepper motors are a type of electric motor that can be accurately controlled with the controller. Another piece of the hardware is the end stops. These are very small and simple circuit boards with a switch on it that tells the controller when it has moved too far in one direction.

Software

Now that you have a background on the electronic hardware, the steps of 3D printing software can be introduced. 3D printing software allows you to create, view, and alter 3-D images, converts the image into instructions for the printer, and "slices" the file into horizontal pieces that the printer will understand the printer deposits plastic in horizontally. The workflow for turning an idea into a 3D print can be summed up as creating a model, slicing, and printing. At each step, there are multiple software solutions to choose from. In general, for 3D printing software can be broken down into 3 different areas, CAD tools, CAM tools, and firmware for electronics.

CAD Tools

Computer Aided Design, or CAD, tools are used to design 3D parts for printing. CAD tools allow you to easily change and manipulate parts based on parameters. One of the techniques used in solid modeling CAD tools is called Constructive Solid Geometry, or CSG. Using CSG, parts can be represented as a tree of Boolean Operations performed on basic shapes such as cubes, spheres, cylinders, and pyramids to create complex surfaces. For example, a hollow ball can be modeled by drawing two overlapping spheres, one lightly smaller than the other and subtracting the smaller from the larger. Simply CSG presents a model or surface to appear visually complex, but it is merely a combination of objects. Open Source Software applications for CAD are OpenSCAD, FreeCAD, SketchUp, and HeeksCAD. Examples of proprietary CAD tools are Solidworks and Autodesk Inventor.

The next step in turning a model from CAD is generating it into an STL file. Most 3D software applications save their files in an application-specific format, but there are very few interchangeable CAD file formats. The two most widely used interchangeable CSG file formats that should not be used are STEP and IGES, because both strip the geometries from data and create flat solids. The ideal file used to export a 3D model is an STL file. STL files can be generated from CAD, these STL files can be sliced and printing unlike STEP and IGES files. One of the most common mistakes for beginner users is not using the correct file type so it's a good idea to design it using a CSG CAD application and save the original parametric file along with generated STL files.

CAM Tools

The next step in the software process is using Computer Aided Manufacturing, or CAM, tools to translate CAD files into a machine-friendly format used by the 3D printer's electronics. In order to turn a 3D part into a machine friendly format, CAM software needs an STL file. The machine friendly format that is used for printing is called G-code. G-code tells the printer where to move the print head and when to extrude plastic, by creating a list of commands that will adjust the acceleration of the motors. This is one of the most critical phases because of its careful balance between quality, speed, and amount of filament used. To Convert STL files to G-code, you use a slicing program. Some examples of slicing programs include Slic3r, Kisslicer, RepSnapper, and RepRap Host Software. The process of converting STL to G-code slices the model, then looks at the cross section of each slice and figures out the path that the print head must travel in order to extrude out plastic, and calculates the amount of filament to feed through the extruder for the distance covered.

After you have your G-code file, the next step in the software process is running the file through a G-code interpreter. This reads each line of the file and sends the actual electronic signals to the motors to tell the 3D printer how to move. To send the G-code files to an integrated hardware interpreter, you need a program to feed the G-codes over a USB connection. For example, RepSnapper, RepRap Host Software, send.py, and Printrun can be used. Most programs that slice the STL files and turn them into G-code can also send G-code files to the hardware.

Firmware

The firmware process can be described in the following four steps.

1. Downloading Arduino IDE- This software is available for WIndows, Linux and Mac. The IDE contains only the officially supported board driver so it is essential that you understand your Arduino and RAMPS electronic hardware before proceeding.

2. Download firmware source code- Firmware and source code needs to be downloaded in a ZIP file. Popular firmware includes Marlin, Teacup, and Sprinter. Once downloaded the contents of the ZIP should be unpacked and then the ".pde" file should be selected and opened, but before you can compile and upload the firmware, you need to select the board and port. This was done by using the Tools menu, plugging the board into the computer, and selecting the port it's connected to.

3. Modify code- Once the .pde file is opened in Arduino changes needed to be made in the code to customize it to the printer. First, "Configuration.h" is opened, and then the motherboard and baud rate needs to be defined as seen in figure 2 below.

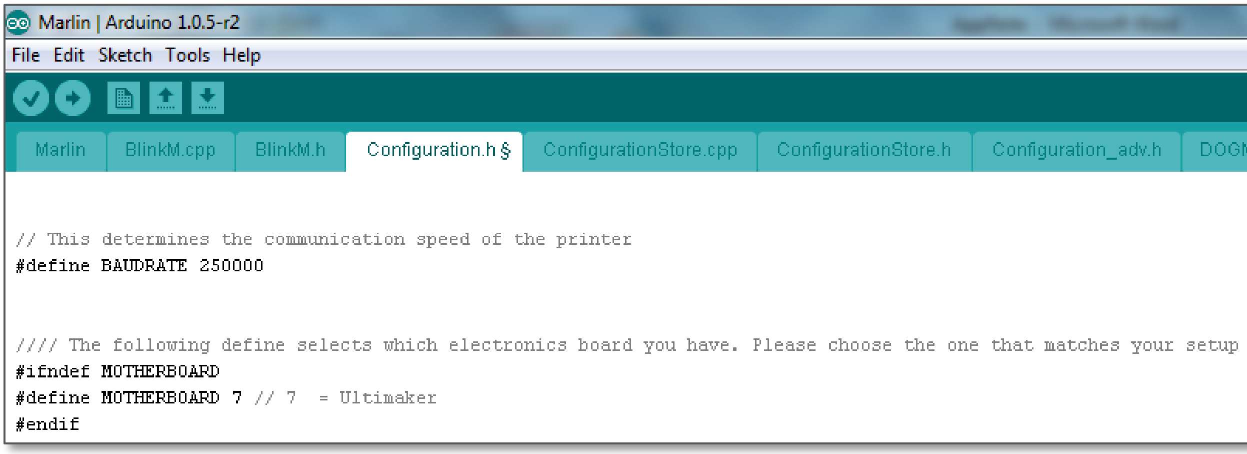

Figure 2: Define Motherboard and Baud Rate

4. Compile and upload the firmware to the controller- Once Configuration.h's code is customized to the printer it can then be compiles and uploaded to the Arduino. After the led on the board stops flickering, the upload is verified and then you see the message "Upload done" in the IDE.

Finally after your microcontroller has its firmware loaded, it is ready to accept G-codes via the COM port. The four steps above are the minimum steps to take to ensure your printer will work. There are numerous modifications to the code to control accuracy. Some of these modifications include steps per distance, temperature measurement, jerk control, path planning, and display etc.

Of these extra modifications, temperature measurement is crucial for accuracy of filament extruded. Before you can control the temperature of the filament, you need to measure it. In our case a NTC thermistor was installed. Below in figure 3 is the schematic for our thermistor. Resistance is changed by the NTC, this means with increased temperature the resistance drops. This causes the measured voltage to change, then the voltage is converted into 0V (ground) and up to 5V. With this knowledge of temperature, the heated bed can also be controlled.

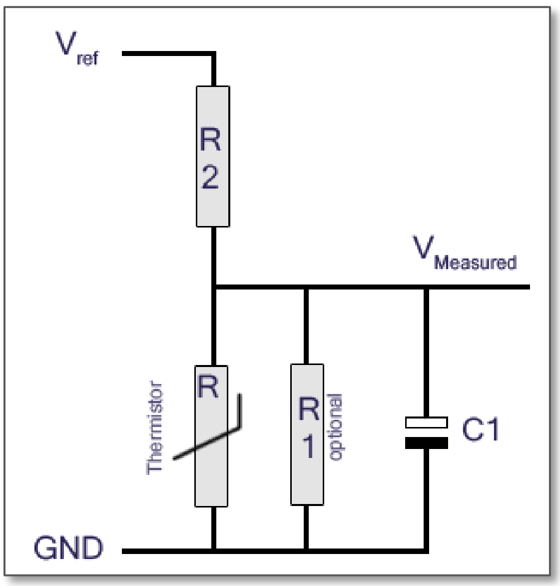

Figure 3: Thermistor Schematic

Conclusion

In conclusion, this application note explains how to set up hardware and software for a 3D printer. A model is created using a CAD program and then saved, this file as a STL file. The file is then transferred to a g-code generator program which converts the file containing the image into a set of instructions/language that the printer understands. The 3D printer then used CAM software that reads the file and send electronic signals to control uses an extruder and motors that move along a x, y, and z axis to extrude a filament onto a surface creating the object through sequential layering.

References

Devijver, Steven. "Building Your Own 3D Printer." *Building Your Own 3D Printer*. RepRap, 2011. Web. 27 Mar. 2014. <http://reprapbook.appspot.com/#d0e436>.

Kirsch, Florian. "OpenCSG." *The CSG Rendering Library*. Hasso-Plattner-Institute Potsdam, 12 Feb. 2011. Web. 25 Mar. 2014. <http://opencsg.org/>.

"List of Firmware." *RepRapWiki*. RepRap, 10 Mar. 2014. Web. 28 Mar. 2014. <http://reprap.org/wiki/Firmware>.

Makezine.com. "The Ultimate Guide to 3D Printing." *Make* Jan. 2013: Print.

Nop Head. "HydraRaptor." *Measuring Temperature the Easy Way*. HydraRaptor, 9 Oct. 2007. Web. 28 Mar. 2014. <http://hydraraptor.blogspot.com/2007/10/measuring-temperature-easy-way.html>.