CprE / EE / SE 491
Team MAY14-07
3D Printer Software Programming
Design Document

Team Members:

Arielle Czalbowski **ariellec@iastate.edu**

Piriya Kristofer Hall **pkhall@iastate.edu**

Albert Kurniawan **albertk@iastate.edu**

Wanting Zhao **wzhao@iastate.edu**

Advisor/Client:

Dr. Thomas Daniels **daniels@iastate.edu**

I. System Design

i. System Requirements

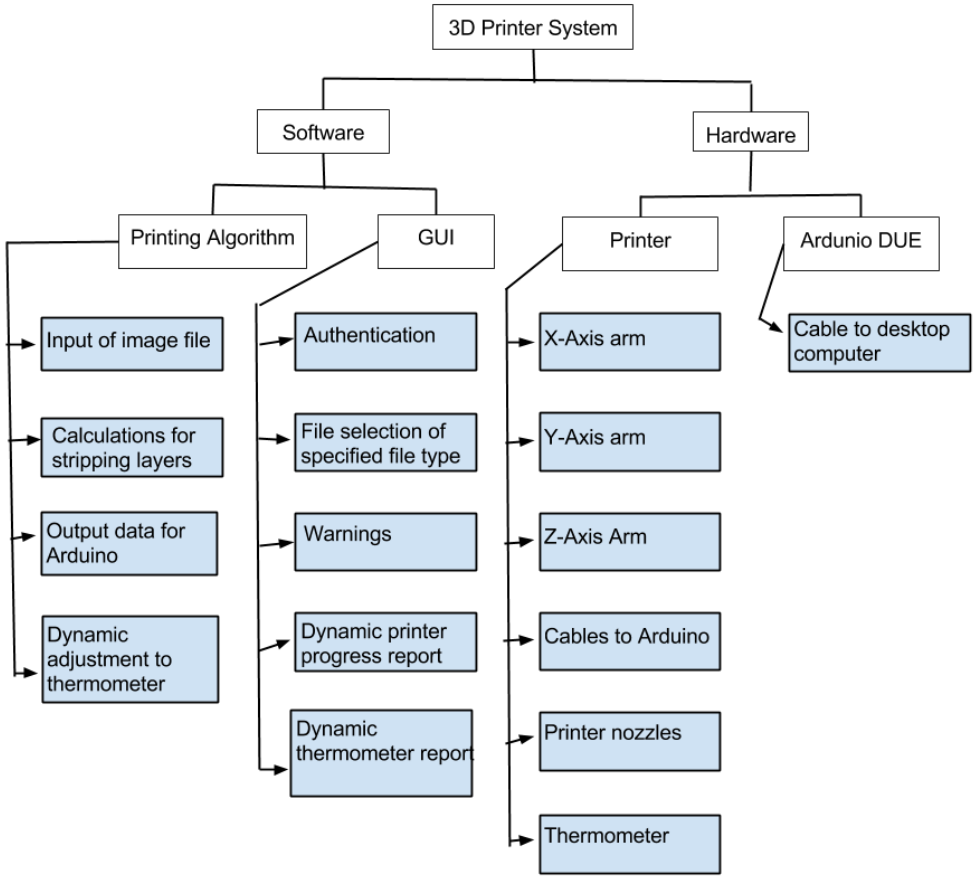
Functional Requirements

- The product shall take as input a model file that is compatible with our program, and will otherwise produce an error.
- The product shall determine a path that the printer nozzle can take in order to print the 3D object.
- The product shall produce a path for the printer head to follow.
- The product shall send the information of the printer head path to the Arduino DUE via USB.
- The product shall only accept one model file at a time.
- The product shall not allow the user to manually move the printer via commands excepting maintenance and calibration.
- The product shall be able to pause printing if there is inadequate filament and allow the user to replace the filament cartridge.
- The product shall allow the user to cancel the print job at any time.
- The product shall print the plastic model file onto a heat plate.
- The product shall only print when the heat plate is hot enough for the print job to be successful and without significant error.
- The product shall regulate the temperature of the heat plate.
- The product shall give power to the heat plate if it is deemed to be under a printable temperature.
- The product shall reduce power to the heat plate if it is deemed to be over a printable temperature.
- The product shall pause printing if the heat plate is too cold or too hot until it is back within an acceptable temperature range.
- The product shall inform the user if a temperature adjustment is being made.

Non-Functional Requirements

- The product shall take the average user no longer than 10 seconds to figure out the interface.
- The product shall produce an error message due to software programming during no more than 10% of usage time over a 3 month period.
- The product shall be easy to use by anyone with at least a high school understanding of English.
- The product shall be easy to use by someone who has had no more than 6 months of exposure to computers.
- The product shall take no more than 10 seconds to complete the path finding algorithm per layer and send it to the Arduino DUE.
- The product shall produce a faulty print during no more than 5% of usage time over a 3 month period.
- The product shall not allow a model file over 1 GB.
- The product shall always clearly warn users of the safety concerns of utilizing the 3D printer.
- The product shall provide clear instructions for preparing the printer hardware.
- The product shall only allow Iowa State University students and faculty access to printing, and will therefore always require authentication.
- The product shall display the current temperature of the heat plate on the user interface screen 90% of the time.
- The product shall take no more than 10 seconds to read input from the thermometer.
- The product shall always clearly warn users of the safety concerns of working with the heat plate.
- The product shall not allow the user to manually adjust the thermometer.

ii. Functional Decomposition Diagram



iii. System Analysis

1. Scope Definition

The scope of this project is to produce software that will allow an ordinary desktop computer to interface with a 3D printer and produce a 3D model. This will be done through software we write, which will utilize G Code conversion so that an Arduino board can communicate with a 3D printer. Within the scope of our work is the actual software that does the communication, as well as a GUI and a calibration program. Also within the scope of our project is control of a heat plate. This interface involves gathering, displaying data from, and dynamically parsing input for the program. Outside the scope of our program are the specifics and technicalities of the hardware that we are are interfacing with, including the setup of the hardware.

2. Problem Analysis

The overall problem that we are facing is our ability to create a piece of software that will allow the user to easily produce a 3D object via a 3D printer from a .STL file.

One portion of this software requires us to deconstruct the .STL file in such a way that the hardware will be able to understand the commands that it is being fed and will respond to them appropriately in order to accurately print a 3D object. To this end, we will be utilizing the model slicing software of the RepRap CAM software stack to create a method for the software to communicate with the hardware that will be printing the object.

Another portion of our software will allow the user to interact with the graphical user interface in order to control the printer's software. The user must be able to select an appropriate image file and control the printer's movement, both for convenience and for safety.

We also need to have a thermometer that can interface with the printer. The program must dynamically send back the current temperature of the heat plate to the user so that, if there are any major problems, the user will know if action needs to be taken. The print job should also be able to be automatically adjusted by the program based on temperature fluctuation.

Finally, the additional program that we are writing, a program to calibrate the hardware for the 3D printer, must be able to provide an accurate and useful method of calibration data so that any print jobs will be successful.

3. Decision Analysis

We have come to the following decisions about the direction our project should take, as well as the rationale of why they should be taken:

- We have chosen to write the calibration program in a method such that it will run a pre-set certain number of times. This is because allowing the program to run a

large set number of times will make it simpler for the hardware team that we are working with to calibrate the printer.

- We have chosen to implement a graphical user interface that works solely with buttons in order to simplify the program and make it easier for people without strong backgrounds in computers to use.
- We have chosen to display the live output of the 3D printer in order to make it easier for users to see the status of the 3D printer and be aware of any errors in an easy-to-read manner.
- We have chosen to display the status and temperature of the thermometer because it will allow the user to see how hot the printer is, and if the temperature of the heat plate is too hot or too cold, or there are any other related problems, the user can stop the print before any significant damage to the hardware is caused or plastic is wasted.
- We have chosen to utilize the RepRap software stack because it has a lot of pre-existing tools for 3D printing that will aid us in creating adequate software for the printer.
- We have chosen to only support .STL files for the 3D printer because they are the most common types of files to print, and we will therefore be supporting a majority of model types that a user may want to print.

II. Detailed Design

i. Input / Output Specification

Our system's input will be a model file in STL format so that it can be sliced properly to be sent to the 3D printer. The software that we write will not take any other input; the printer will be restricted to an STL-format file in order to function properly.

Our system's output will be data that can be sent to the Arduino in order to control the 3D printer. The data will direct the 3D

printer to print a plastic model of the file that was sent in as an STL file.

Our system has an additional, intangible output: Our program will show dynamic data on the user interface of the status of the printer as well as the status of the thermometer. The repeated output will be shown prominently on the user interface to inform the user of the current printer's work and any adjustments that the program is making to the printing.

ii. User Interface Specification

The user interface will be effectively utilized in order to allow the user an easy and effective way to communicate with the 3D printer and accomplish the task of printing a 3D object without extensive prior experience in computer technology.

The user interface will consist of a number of buttons as well as a text area for feedback. The buttons that the user may select from include:

- A button that will allow the user to select a .STL format file from their hard drive in order to print. The user may select a file as many times as they wish, but once printing has begun, the user may not select another file.
- A button that will allow the user the print a file. Once pressed, this will begin the slicing of the model, which produces G Code, then the conversion of G Code to language that the Arduino may process.
- A button that will allow the user to cancel their current print job. This will stop the printer from continuing its print and terminate the job immediately.
- A button that will allow the user to pause their current print. This will suspend the printer's path indefinitely until the user chooses to unpause the print job, in which case the 3D printer will continue its print job exactly where it left off.
- A button that will allow the user to restart the current print job. This will stop the user's print job and go back to the beginning of the code, where it will start the print over again.

- A window that will display dynamic feedback from the printer. As the user prints their file, they will see a live feed of the status of the printer and what the printer has accomplished. The temperature of the heat plate will also be displayed. Any errors that occur during print, as well as confirmation messages, will also appear in this box.

iii. Hardware Specification

Due to the fact that we are doing a largely software-based product, the hardware that we provide is necessary for the peripheral functionality of the software rather than the core functionality:

- A desktop or laptop computer containing:
  - At least one (1) USB port
  - A modern operating system
    - This includes: Windows XP or above, Mac OS X, or an updated Linux distro (Ubuntu, RedHat, etc.)
- A USB cable that will interface between the desktop or laptop computer and the Arduino board. This USB cable must have a male Type A USB port on one end, and a male Type Micro-A USB port on the other end.
- A thermometer that will monitor the heat of the heat plate.

iv. Software Specification

The software specifications for our project are as follows:

- A calibration program that must:
  - Run a number of specified times in a set pattern
  - Run easily as to allow visual calibration of hardware.
- A 3D printing program that must:
  - Prepare the 3D printer to receive a print job.
  - Take a .STL file as input.
  - Allow the user full control over what the printer is doing.
  - Not allow the user to manually move the printer.
  - Not allow the user to interfere with the print job.
  - Convert G Code into script that the Arduino can use to control the 3D printer.

- Output information to the user via the program interface on print status.
- Output a 3D-printed version of a model file.
- Interface with a thermometer and return the value of the temperature to the user.
- Adjust the print based on what the thermometer's temperature is and what is needed for the 3D model to print successfully.

v. Test Specification

i. What we are testing

We are planning to test the two pieces of software that we are planning to deliver:

1. A program to calibrate the 3D printer's hardware.
2. A program that will accept an STL file as input and output a 3D-printed item by slicing the model into layers, saving the layers in G Code, then converting G Code to Arduino-appropriate commands.
    a. This program must successfully interface with a thermometer and must accurately read the temperature of the hot plate and feed the data back to the user in a timely manner.

ii. What we are not testing

While the following items are related to our 3D printer, they are outside of the scope of our software and therefore out of the scope of our test area:

1. The physical circuit boards for the 3D printer.
2. The hardware that the 3D printer itself is comprised of.
3. The physical Arduino board that the 3D printer is connected to.
4. The physical stability of the hot plate and its hardware setup.

iii. Major areas to test

While fully testing all aspects of our software is important, we feel that the following are the most important items to test:

1. Whether the conversion of G Code to Arduino-specific language is accurate.
2. Whether the printer receives all of the commands in a timely manner.
3. Whether the printer produces a viable 3D model.
4. Whether the software we write communicates effectively with the printer hardware.
5. Whether the software we write communicates effectively and accurately with the thermometer.

vi. Simulation / Modeling

We will not be working with any simulation or modelling because we do not have any physical deliverables; the software we write does not require this.

vii. Prototyping / Testing

We plan on initially having two different types of software to prototype.

The first kind of prototype we will deliver is a calibration program for the hardware team, which will allow them to determine the accuracy of the 3D printer. Specifically, with this program, the accuracy of the stepper motor will be testable, which will allow us to determine the types and accuracy of models that we may do.

The second prototype that we will deliver is a prototype of the program that will convert G Code to commands that the Arduino can understand and feed to the 3D printer. We initially plan to test this program by attaching something inexpensive, such as a marker, to the print head so that more expensive plastic is not wasted. Later, we will test with plastic. The tests that we plan to perform on this program include testing how well the 3D printer can handle printing one layer through a simple STL model file. Once the printer has demonstrated its ability to handle single layers, the entire STL file will be used in order to determine how well the 3D printer can produce a full model.

After we have determined that the program will theoretically print a 3D model correctly, we will attach the heat plate and begin working on interfacing the printer with the heat plate, and test whether the heat plate is able to successfully interface with the hardware and heat up and cool down as it should. Once this test has been deemed successful, we will test whether our code dynamically works with the thermometer temperature.

Finally, after all of this has been tested, we will test how real 3D models work with the printer and whether these 3D models have any problems while printing.

III. Design Documents

i. Mechanical CAD

We will not require mechanical CAD because we do not have a hardware component to our project. Our product solely deals in software design and specification.

ii. Electronic CAD

We will not require an electronic CAD since we are not drawing a circuit board. Due to the purely software-oriented nature of our project, we do not have any circuits to design.

iii. PCB

Since we do not deal in any hardware, we do not have a PCB. However, the hardware team who is assisting in implementing our project should have the appropriate documentation.

iv. Software Design

The majority of our software stack, especially in regards to the design and implementation of the model slicer, will be taken from the RepRap CAM stack. The software we produce will be able to convert G Code to commands that the Arduino board can understand, and we will be programming in C++ to this end.

We plan to perform the conversion by using a dictionary-style comparison of G Code to control software commands. The

control software commands that we will be utilizing be sent from the computer that is running our software to the Arduino board via USB; therefore, the libusb library will be a vital component in our design.

We also need to take heating the heat plate appropriately into account. We will need to control the heat going towards the heat plate so that the plastic will be appropriately malleable and easy to print with, and then we will need to control the temperature to make sure that it is not too cold or too hot for the plastic to print well.

v. Timeline

Week of 8/26

- Form teams and decide on project

Responsibility: Arielle, Piriya, Albert, Wanting

Week of 9/2

- Perform research on 3D printers, microcontrollers, and Arduino boards to better understand project scope.

Responsibility: Arielle, Piriya, Albert, Wanting

Week of 9/9

- Continue research on hardware and software

Responsibility: Albert, Wanting

- Begin meeting with client in order to discuss their visions for project.

Responsibility: Arielle, Piriya

Week of 9/16

- Determine how to give both teams necessary access to 3D printer and how to best store the hardware to keep it safe and in good working condition.

Responsibility: Arielle, Piriya, Albert, Wanting

Week of 9/23

- Keep communication with Team 06 (partner team) open and determine anticipated problems and concerns and how to mitigate them

Responsibility: Arielle, Piriya

- Figure out how our teams intersect.

Responsibility: Arielle, Piriya, Albert, Wanting

Week of 9/30

- Receive code from professor in order to begin work on 3D printer

Responsibility: Piriya

- Set up a time to get a full demonstration of how the code works and what the system is meant to do

Responsibility: Arielle

- Determine necessary standards we may need to follow.

Responsibility: Albert, Wanting

Week of 10/7

- Dedicate work to project plan

Responsibility: Piriya, Albert

- Continue requirements gathering process and continue examining code to determine its functionality and features.

Responsibility: Arielle, Wanting

Week of 10/14

- Continue requirements gathering for design process

Responsibility: Arielle, Wanting

- Finish securing access to laboratory so both teams can work on the machine

Responsibility: Piriya

- Compare project plans with Team 06 to eliminate discrepancies

Responsibility: Piriya, Albert

- Work on design document.
    Responsibility: Arielle, Albert

Week of 10/21
- Finish design document
    Responsibility: Arielle, Wanting

- Further examine provided code and gain working understanding of logistics of printer software
    Responsibility: Piriya, Albert, Wanting

- Finalize printer hardware.
    Responsibility: Piriya, Albert

Week of 10/28
- Get administrator access to our lab computer so that we can download the requisite software for interfacing the 3D printer.
    Responsibility: Arielle

- Continue working on prototype
    Responsibility: Piriya, Albert, Wanting

Week of 11/4
- Contact CSG and Jason Boyd to finalize administrative access to computer
    Responsibility: Arielle

- Attempt to interface machine with Ubuntu while administrative access is uncertain.
    Responsibility: Piriya

- Write a prototype program for calibration automation to be immediately put to use once problems are solved with access.
    Responsibility: Albert, Wanting

Week of 11/11
- Work on formulating the website for our project and uploading documents to showcase what we are planning to do and what

we have currently down.
Responsibility: Arielle

- Work on and finalize project plan revisions.
Responsibility: Arielle, Piriya

- Write full pseudocode for calibration program.
Responsibility: Albert

- Understand what we have been doing in full due to being out of the country.
Responsibility: Wanting

<u>Week of 11/18</u>
- Begin revising design document
Responsibility: Arielle, Piriya

- Upload revised documents to website
Responsibility: Arielle

- Begin base implementation of calibration program.
Responsibility: Albert, Wanting

<u>Week of 11/25</u>
- Thanksgiving Break is this week which may make progress difficult since none of our team members will be on campus; continue revising design document and project plan, and think on solutions to any problems that may have occurred.
Responsibility: Arielle, Piriya, Albert, Wanting

<u>Week of 12/2</u>
- Make final revisions to design document and project plan
Responsibility: Arielle, Piriya, Albert, Wanting

- Continue implementation of algorithm into hardware components
Responsibility: Piriya, Albert

- Begin designing user interface

Responsibility: Arielle, Wanting

- Prepare for presentation.
Responsibility: Arielle, Piriya, Albert, Wanting

<u>Week of 12/9</u>
- Continue implementation of algorithm into hardware components
Responsibility: Piriya, Albert

- Show user interface to client to ensure that interface matches vision of product.
Responsibility: Arielle, Wanting

<u>Week of 12/16</u>
- Finish revising design document and project plan
Responsibility: Arielle, Piriya, Albert, Wanting

- Continue implementation of algorithm
Responsibility: Piriya, Albert

- Continue working on user interface
Responsibility: Arielle, Wanting

- Make plan to decide which items may be considered over Winter Break.
Responsibility: Arielle, Piriya, Albert, Wanting

- Make final updates to website for the semester.
Responsibility: Arielle