Team MAY14-07
3D Printer – Software

Piriya Hall

Arielle Czalbowski

Wanting Zhao

Albert Kurniawan

Scope

- Main idea: Build software for a 3D printer
  - User inputs model file (.STL)
  - Printer prints plastic model
- User should be able to print any .STL file
- Printer should be able to communicate with hardware to move arms to print specified file.
- User should be able to easily utilize GUI to communicate with printer

Primary Functional Requirements

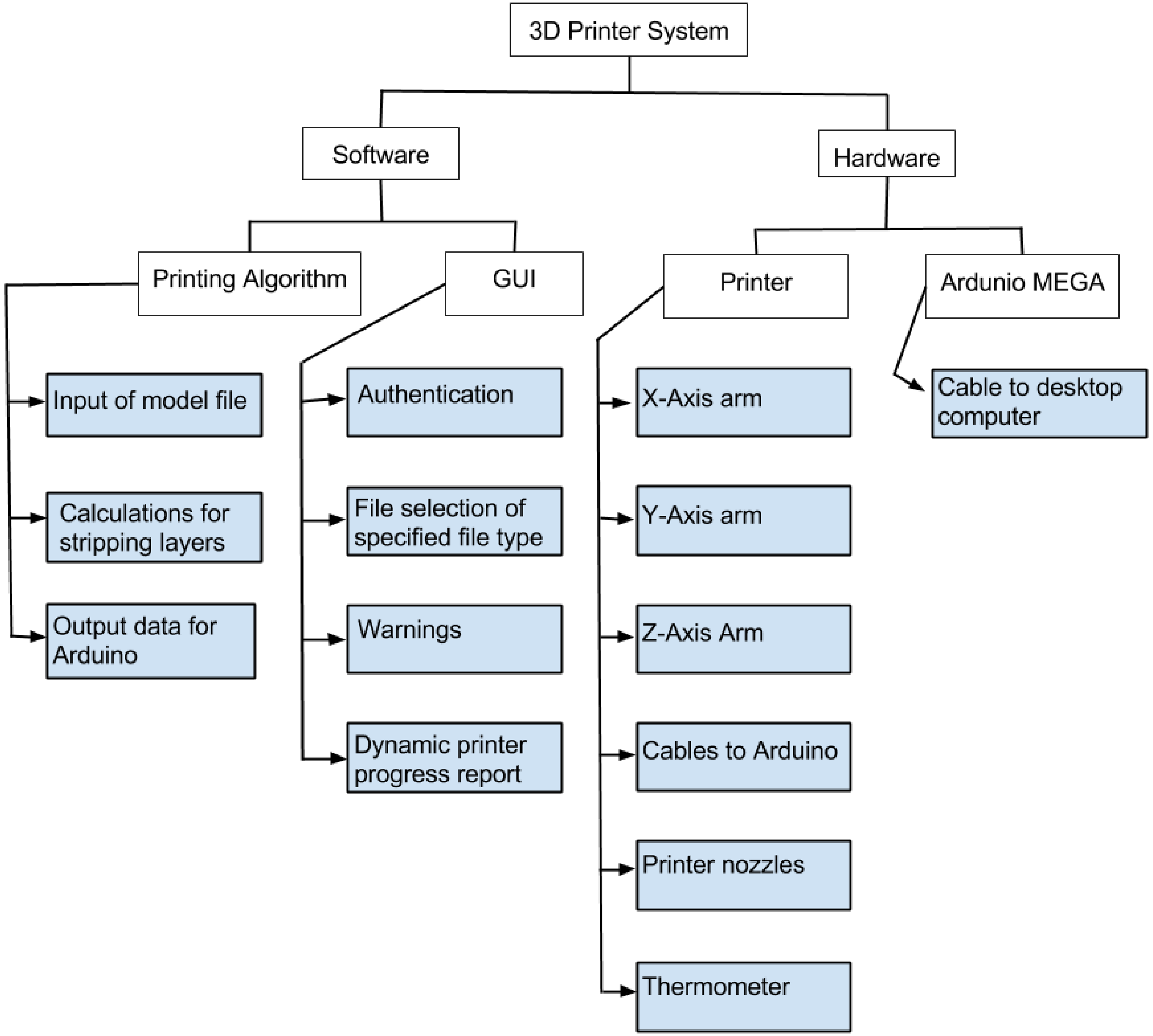
- Product shall take .STL model file as input and output 3D model
- Product shall send information of printer head to Arduino MEGA via USB
- Product shall not allow user to manually move printer via commands
- Product shall only print when heat plate is hot enough for print job to be successful and without significant error

Primary Nonfunctional Requirements

- 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 only allow Iowa State University students and faculty to access to printing
- The product shall require the user to enter contact information before being allowed to utilize the 3D printer

Design

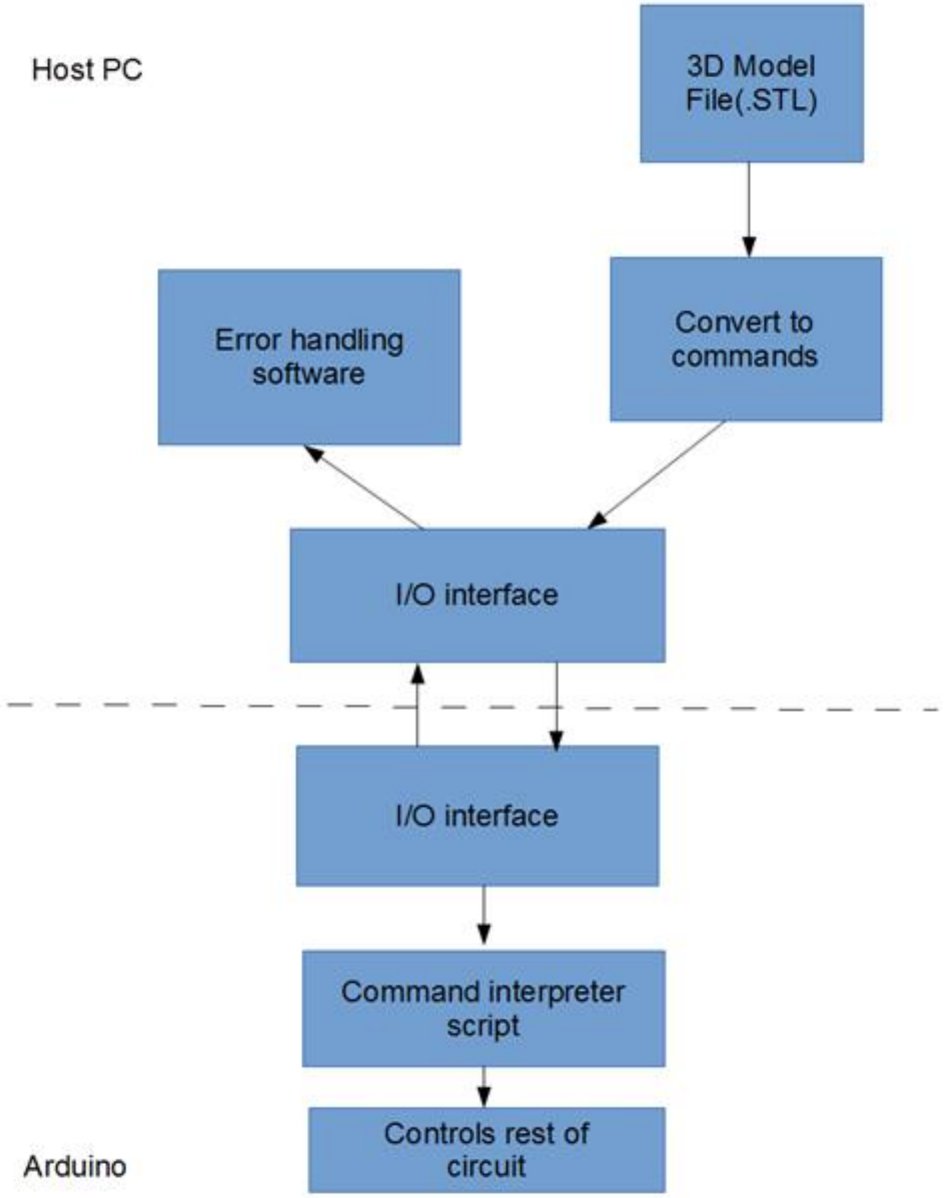
- Hardware and software taken into account
- Firmware design
  - Calculations
  - Adjustment with hardware
  - Heat regulation
  - Coded in C++
- Calibration Program
- GUI design
  - Error checking
  - Information collecting
  - Dynamic report
  - C#



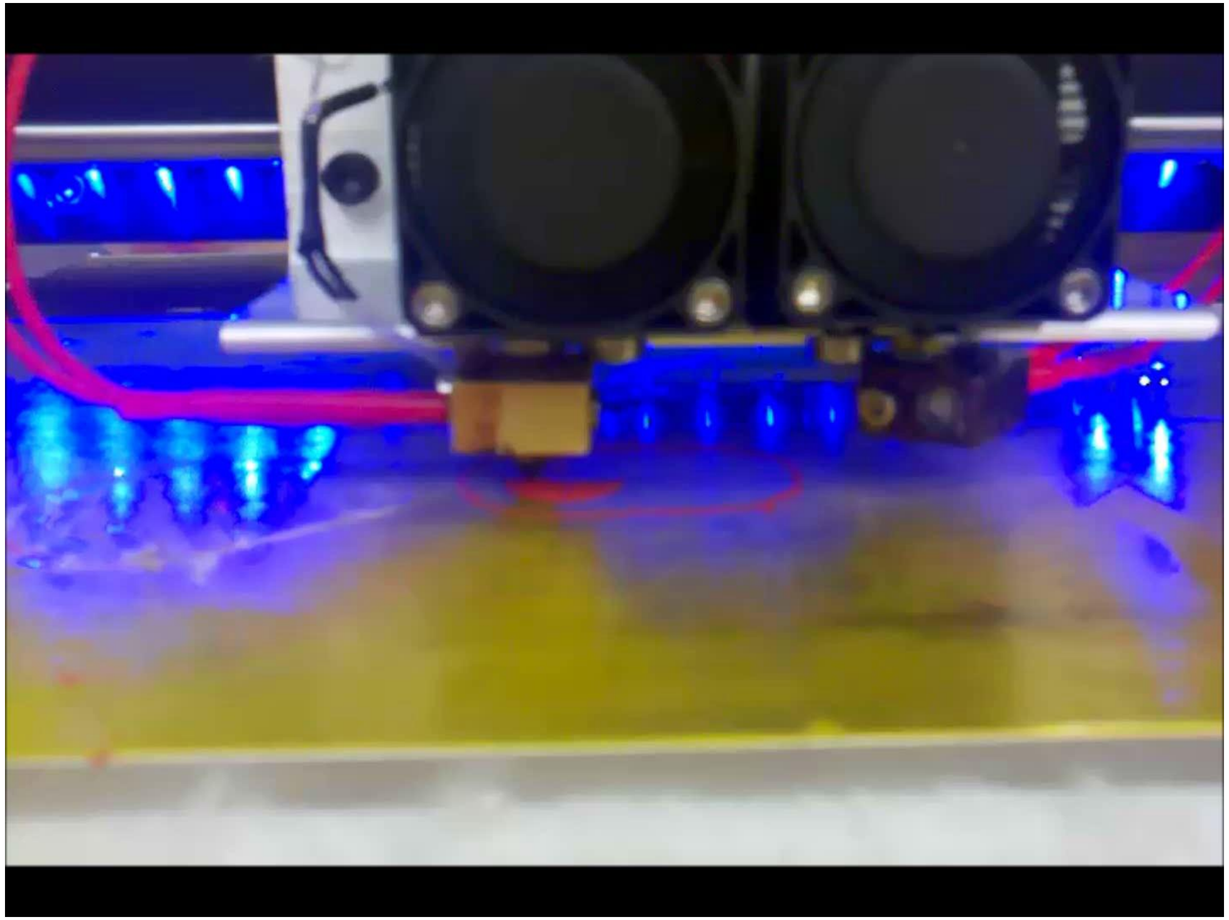
Demo – GUI

Design

- Host PC uses .STL file
- Converts to commands
- Sends through interface
- Arduino interprets script
- Gives script to printer to print



Demo – Printer



Design

- Reached current design:
  - Examining hardware limitations
  - Determining best possible software
  - Meticulous formation of requirements

- Progress
  - GUI completed
  - Configuration for firmware is complete
  - Majority of slicer completed

Alternative Designs

- GUI
  - C++
    - Not intuitive for making GUI

- Firmware
  - SPrinter
    - Didn’t control arms correctly
  - Marlin
    - Didn’t control heat plate
  - Versions of Repetier
    - Combination of above across various versions

Testing

- Calibration program
  - Run small number of times
- Back-end program
  - Test if we can communicate with Arduino
  - Test with GUI
- GUI
  - Look through requirements
  - Have users utilize and try to break

Testing Results

- Calibration program successful
- Back-end COM port opened and connection established
- Repetier works as expected
- GUI has trouble communicating with COM port but is otherwise successful (strict error checking)
- GUI successful; known errors are account for, Client expressed satisfaction
- Produced desired result overall

Challenges & Solutions

- Challenge: Finding a programming language that would allow for both tight communication and ease of programming and debugging.
- Solution: Between C++ and C#, found C# to fulfill most needs the easiest

- Challenge: Finding correct firmware for use with the printer
- Solution: Explored viable solutions and weighed their pros and cons until we decided on Repetier.

Challenges & Solutions

- Challenge: Opening up the COM port for communication with the Arduino for data
- Solution: Exploring options and attempting to follow similar problems people have had in the past with regards to communication

- Challenge: Firmware changed between teams fairly frequently.
- Solution: Kept up with firmware that was decided upon and did rapid research to understand and utilize it.

Scenario Examination

- Thorough examination of requirements
  - Use of VOLERE process
  - Frequent discussion with client

- Thorough testing
  - Based on requirements
  - Based on past experience of code

- Many different options explored
  - Different languages, standards, etc.

Questions

Standards

- ANSI ISO 10005 / 10006 / 1007 (Quality Management Standards)

- Underwriters Laboratories Standards
  - Standard 2785 (Printing cartridges)
  - Standard 796 (Printed-wiring boards)
  - Standard 746F (Polymeric materials)

Decisions

- Calibration program runs pre-set number of times – makes it easy to calibrate printer
- GUI – simplifies for user and helps with security
- Live output – easy for users to see status and be aware of any errors
- RepRap – lots of pre-existing tools for printer

Project Schedule

- Fall 2013 – Setting up printer, solving problems, taking care of errors; calibration program
- January 2014 – Work through firmware; use C++ for GUI
- February 2014 – Working through firmware; switched to C#
- March 2014 – Finalized firmware
- April 2014 – Finished the GUI and back-end components

Project Costs

- Not much due to software not costing money
  - License for Visual Studio (paid for by University)
  - Team May 14-06 utilized our budget for the 3D printer hardware