UNIVERSITY OF WASHINGTON

ME495 – SPRING 2011

DIY INKJET PRINTER

TEAM

Patrick Hannan
Jared Knutzen
Nicholas C Lewis
Joy Markham

PROFESSOR

M.A. Ganter

14 MAY 2011

EXECUTIVE SUMMARY

The goal of this project was to develop a low cost, open source inkjet printer utilizing standard inkjet technology, for personal use. This project was partly in response to the fact that there are no DIY inkjet kits available. There was a kit available from Parallax (96 dpi Serial Inkjet Printer Development Kit (#27949)). This kit is no longer made and the book (Gilliland, 2005) that was written to use that kit is now out of print.

The prototype design used a carriage assembly constructed from steel rods that were assembled using connectors that can be printed on an FDM machine. The entire carriage system is driven along the x-axis by a belt attached to a stepper motor. The print cartridge, taken from an HP point of sale printer, is driven along the y-axis by another stepper motor belt drive. The electronic controls use an Arduino Mega to run all of the printing systems.

The design resulted in a working prototype that fulfills all of the design constraints. The rod frame carriage design is lightweight, easy to assemble and easy to integrate with the other systems. The Arduino used in the electronics has a large library of resources available to perform things like LCD, SD card, and stepper control.

Areas where future work should be focused include making molds and casting printable parts to bring down the overall cost, developing host side software, and optimizing the speed.

TABLE OF CONTENTS

List of Figures iii
List of Tables iv
Introduction 1
Design Problem Definition 1
Overview of Current Technology 2
Design Sub-Systems 2
X/Y Axis Carriage system 3
Concept Generation 3
Final Conceptual Design 8
Future work 12
Electronics/Software 13
Concept Generation 13
Final Conceptual Design 18
Future Work 21
Economic Evaluation/Cost List 22
Bibliography 23

LIST OF FIGURES

Figure 1: Platform Carriage Design (Side View) 3
Figure 2: Carriage Design with Side Platform (Top View) 4
Figure 3: Rod Frame Carriage Design 4
Figure 4: First Iteration Bearing Design 6
Figure 5: Second Iteration Bearing Design 7
Figure 6: Third Iteration Bearing Design 7
Figure 7: Motor Mount 8
Figure 8: Motor Mount Clamp 8
Figure 9: Idler Mount 8
Figure 10: Cartridge Holder 8
Figure 11: Left Angle Connector 9
Figure 12: Right Angle Connector 9
Figure 13: Y-Bar Clamp 9
Figure 14: Cartridge Belt Clamp “t” 9
Figure 15: Cartridge Belt Clamp Bar 10
Figure 16: Bitmap data arrangement 17
Figure 17: Print Data Arrangement 17
Figure 18: Schematic of Prototype Circuit 18
Figure 19: Flow Chart oF Printer Operations 20

LIST OF TABLES

Table 1: Functional Requirements 1
Table 2: Pugh Decision Matrix for Carriage System 4
Table 3: Summary of Bearing Characteristics 6
Table 4: Cost Breakdown of Printed Carriage Parts 10
Table 5: Costs of all Carriage Assembly Components 11
Table 6: Cost of Molds and Polyurethane Parts 12
Table 7: IO Needs of Printer 14
Table 8: Microcontroller Board Comparison 14
Table 9: Binary File Format for Printing 16
Table 10: Cost Breakdown of Electronic Components 19
Table 11: Simplified Code Set 20
Table 12: Cost Breakdown by Sub-System 22
Table 13: Estimated Cost Breakdown For Molded Printer 22

INTRODUCTION

This project was partly in response to the fact that there are no DIY inkjet kits available. There was a kit available from Parallax (96 dpi Serial Inkjet Printer Development Kit (#27949)). This kit is no longer made and the book (Gilliland, 2005) that was made to go with it is now out of print. Copies of the book can occasionally be found on online auction sites.

DESIGN PROBLEM DEFINITION

The goal of this project is to develop a low cost, open source Inkjet printer utilizing standard inkjet technology, for personal use. The following functional requirements were developed to further solidify the project foundation:

TABLE 1: FUNCTIONAL REQUIREMENTS

Primary FR's (Needs):	**Secondary FR's (Wants):**
• must use standard inkjet print cartridges • must be relatively inexpensive • must be made of readily available materials that are easy to source	• must be easy to assemble by a single user • must be simple to operate, requiring no specialized training

The use of standard inkjet print cartridges is critical to many facets of the project, directly influencing cost, ease-of-use and sourceability. This requirement was imposed by Professor Ganter, after initial research into possible print cartridge sources, to help focus this project prior to its official launch. Cost is extremely important as this product is designed for personal use, and therefore must be affordable to the average individual. A low cost is necessary for successful implementation and widespread use, which are the underlying goals of this project.

To allow for use on a global scale, this printer must be made of materials that are easy to obtain and easy to source. In addition, by designing custom parts to be printable by a RepRap printer portions of the machine can be easily manufactured by any individual with a personal rapid prototyping system, which can be found throughout the world.

Secondary requirements include simplicity in design allowing a single user to assemble and operate this printer using only the provided instructions, reducing any indirect costs inherent in purchase and potentially increasing user satisfaction.

The following design constraints were identified by the team at the outset of the project:

- Cost: <$200
- Open source

Cost is a critical constraint imposed on this project, as it will most likely dictate the commercial success or failure of this printer. Taking into consideration the costs of commercially available inkjet printers and the general financial state of likely consumers, namely students and hobbyists makers, a final cost of $200 for manufacture was decided upon. This constraint is flexible for the prototype phase of the project, but is rigid for the marketing and implementation phase: the final product must not exceed $200 in total cost.

As an open source project, an open tool chain and open design that utilizes non-proprietary software for all aspects of the project must be maintained. As with the cost, this constraint is critical for the marketing and implementation phase, but was not rigidly enforced during the prototyping phase, as proof of concept is first necessary to ensure future development.

Additional constraints exist for each sub-section of the design. These are listed at the start of each section, along with the goals for that section.

OVERVIEW OF CURRENT TECHNOLOGY

There are currently no DIY options for inkjet printing. The only comparable technology would involve hacks to commercial inkjet printers. These are very limited in what can be accomplished as the systems are not designed to be modified.

DESIGN SUB-SYSTEMS

A functioning printer is complex machine, with numerous parts. To make the design process easier, the project was divided into 2 main sub-systems, each of which would undergo a separate but concurrent design process. Based on the research into current technology and the functional requirements and constraints given for the entire system, the following sub-systems were decided upon:

- **X/Y Axis Carriage System**
  This sub-system is concerned with the support and placement of the print head as well as the movement across the length and width of the printer.
- **Electronics/Software**
  This sub-system is concerned with all electrical components of the printer, as well as the necessary pre-processing and operational software.

X/Y AXIS CARRIAGE SYSTEM

Constraints:

- Support the weight of the printing system
- <$100 combined cost for all non-electronics systems (1/2 total cost)

Goals:

- Easy to source or print parts
- Easy to assemble

CONCEPT GENERATION

Three basic ideas were generated and evaluated for the main carriage frame. The first idea involved a main platform with a "T" shaped center section for the attachment of necessary components. This design is a proven concept that is similar to the carriage design used on many commercial printers. The platform would need to have a slot in its base underneath the print head so that the ink can reach the print bed. A visual representation of this platform carriage idea is shown below in Figure 1.

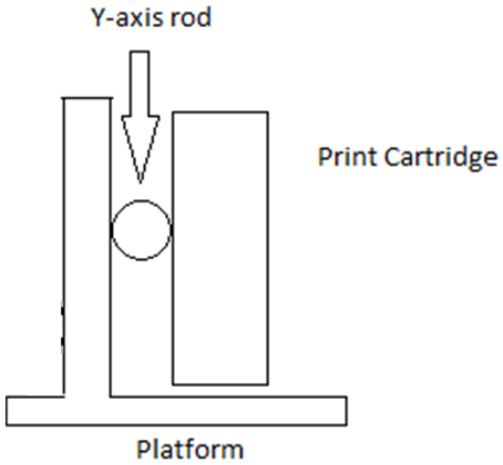

FIGURE 1: PLATFORM CARRIAGE DESIGN (SIDE VIEW)

The second idea was similar to the platform idea discussed above but also had an extension of the platform running parallel to the x-axis along one side of the printer. Rather than mounting the motor used drive the print head on the center "T" shaped piece in the first concept, the motor would be mounted along the side of the printer on this flat platform. A diagram of this second platform idea is shown below in Figure 2.

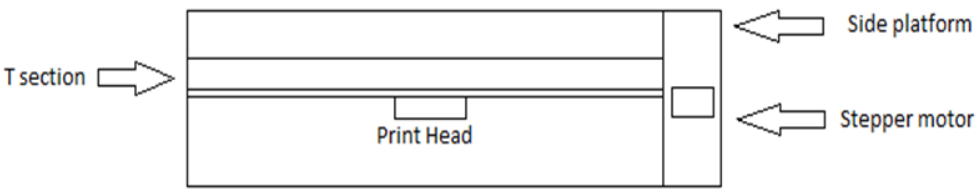

FIGURE 2: CARRIAGE DESIGN WITH SIDE PLATFORM (TOP VIEW)

The third design for the carriage was inspired by the RepRap Mendel carriage design and uses a rod frame construction. This design uses sections of steel rod held together by connectors that can be printed on an FDM machine. Figure 3 below shows a SolidWorks rendering of this carriage design.

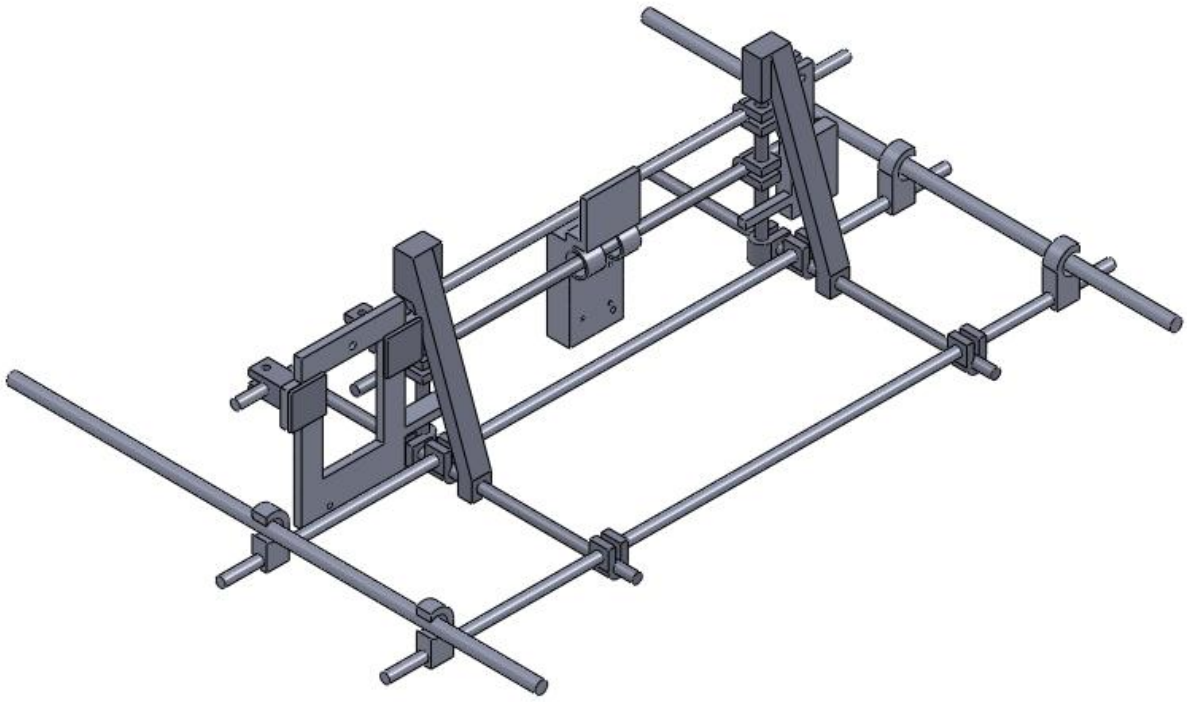

FIGURE 3: ROD FRAME CARRIAGE DESIGN

All three ideas were evaluated using a Pugh decision matrix based on criteria including weight, cost, rigidity and simplicity. The basic platform design was chosen as the datum concept and the other two designs were evaluated against the platform design. The results are presented in Table 2.

TABLE 2: PUGH DECISION MATRIX FOR CARRIAGE SYSTEM

Criteria	Platform with side mounts	Platform (Datum)	Rod Frame
Weight	-	0	+
Rigidity	+	0	0
Ease of Manufacturing	-	0	+
Ease of Assembly	0	0	+

Cost	-	0	+
Integration with other systems	+	0	0
Simplicity	-	0	0
Total	-2	0	+4

After examining the Pugh matrix, both of the platform designs were rejected for several reasons. The first reason was overall weight. Both platform designs incorporated a large central platform structure that would contribute significantly to the weight of the carriage system. By using steel rods for its construction, the rod frame design has much more open space and will weigh less. This reduction in weight becomes important because a lighter carriage is easier for the motors to drive.

The central platform also made the platform designs more difficult to manufacture. The platform would either have to be machined from a piece of metal or printed. However, due to the size of the platform, were it to be printed it would need to be done in multiple builds and would require a large amount of build material. This would also drive up the cost significantly. The Pugh matrix demonstrates that the rod frame design was superior to the platform designs in almost every category and thus it was selected as the final carriage design.

Several options were considered for methods of driving the carriage assembly. The most obvious solution was to use a belt drive similar to the ones used in most inkjet printers currently on the market. This is an effective and proven concept as demonstrated by its widespread use in inkjet printers.

The belt could be driven using either a stepper or DC motor. Stepper motors were selected for two main reasons. The first reason was that stepper motors have an advantage over DC motors in that they are easier to drive. The other reason that stepper motors were selected was that two stepper motors with matching gears, idlers, and belts were readily available. Thus it was decided that in order to have a working prototype at the end of the ten week time constraint, it would be beneficial to use the two motors that would be easiest to drive and that came with matching gears, idlers, and belts. For future design iterations, it would be beneficial to revisit this issue and consider using DC motors as an alternative method for driving the carriage.

As an alternative to a belt drive system, a screw drive system could be used to move the carriage assembly. Although a screw drive system could work for this type of application, the belt drive was selected for several reasons. The first consideration involves speed of a belt drive versus a screw drive. The ten week time constraint imposed on this project also made the screw drive system less desirable. As mentioned above, the belts with matching gears and idlers were available and ready to integrate with the carriage system. Thus it was decided to use belt drives rather than screw drives.

One final problem that needed to be addressed for the carriage system was bearings. Some initial testing was done with bearings found in household inkjet printers to see how the different bearings compared. Bearings and the rods they were designed to ride on were removed from three different printers (a Lexmark and two HP printers). Table 3 below gives a summary of the characteristics of the three bearings.

TABLE 3: SUMMARY OF BEARING CHARACTERISTICS

Criteria	Lexmark	HP (1)	HP (2)
Shape	Circular	Circular	“C” shape
Sleeve Material	Rubber	Felt	Felt
Degree of Lubrication	Well-lubricated	Moderately Lubricated	Minimal Lubrication

It was clear from the results of this test that the “C” shaped bearings performed the best. This is due to the reduction in surface area in contact in the “C” shape design. This reduction in surface area in contact reduces friction which allows the bearings to slide smoothly while still properly constraining the carriage. Thus it was decided to use bearings with a “C” shape in the prototype. In addition, the bearings were designed in such a way so that a nylon sleeve could be inserted into them to further reduce friction.

Even after the “C” shape was decided on, the bearings went through several design iterations. The original bearing design is shown below in Figure 4. It has the required “C” shape into which the nylon sleeve was fitted.

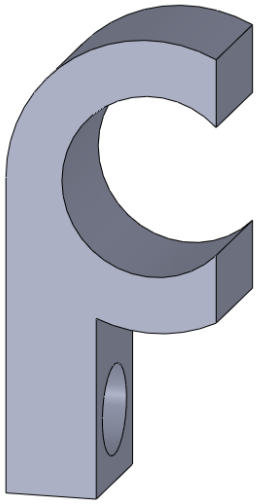

FIGURE 4: FIRST ITERATION BEARING DESIGN

When four of these bearings were installed on the carriage and tested it was discovered that the entire carriage assembly twisted when the motor switched directions along the x-axis. It was determined that the twisting was due to the lack of the support on the bearings in the area directly below where the bearings were riding on the x-axis rails. Support was added and the redesigned bearing is shown below in Figure 5.

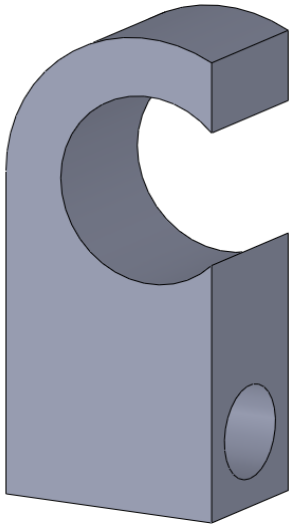

FIGURE 5: SECOND ITERATION BEARING DESIGN

Installing these bearings on the carriage assembly did significantly reduce the twisting problem to a point where the printer was functional. However, it was very difficult to adjust all four bearings so that the carriage rode smoothly without binding. It was then determined that having four "C" shaped bearings was over constraining the carriage assembly. Thus it was decided that it would be beneficial to go to a bearing system involving two circular bearings on the side of the carriage with the motor and the belt drive. On the other side, two sliders were designed to rest on the x-axis rails. The idea was to fully constrain the system with the circular bearings on one side and use the sliders on the opposite side to keep it from being over constrained. Figure 6 below shows the circular bearings and the sliders.

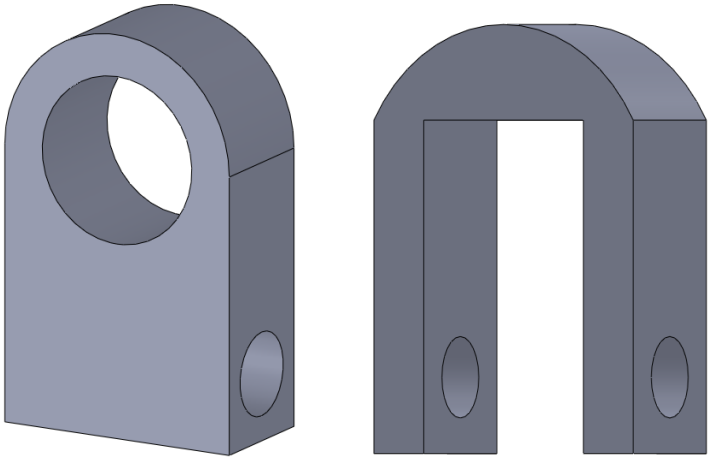

FIGURE 6: THIRD ITERATION BEARING DESIGN

As stated previously in the concept generation section, it was decided that the rod frame design was the best choice for the carriage frame. The carriage is driven by two stepper motor belt drives (one for the x-axis and one for the y-axis). Circular bearings were installed on one side and sliders on the other side to minimize the over-constraint.

All of the connectors used to construct the frame were printed either on the Stratasys machine or on a Mendel. Solidworks renderings of these connectors are shown in Figure 7-Figure 15 below.

FIGURE 7: MOTOR MOUNT

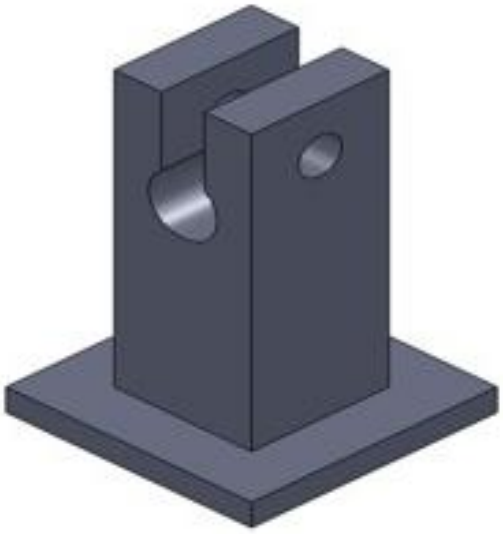

FIGURE 8: MOTOR MOUNT CLAMP

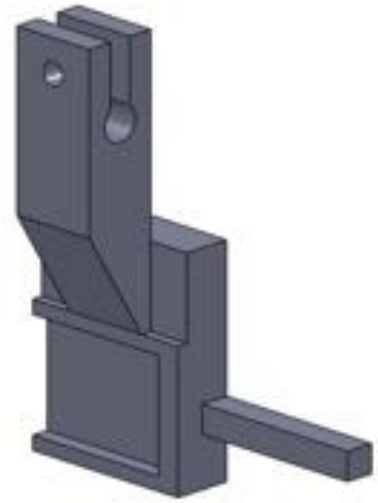

FIGURE 9: IDLER MOUNT

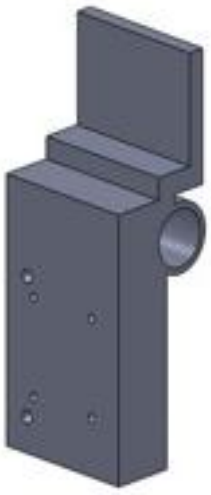

FIGURE 10: CARTRIDGE HOLDER

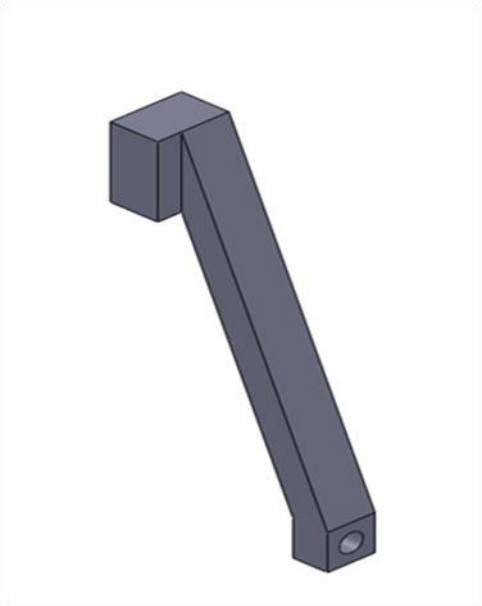

FIGURE 11: LEFT ANGLE CONNECTOR

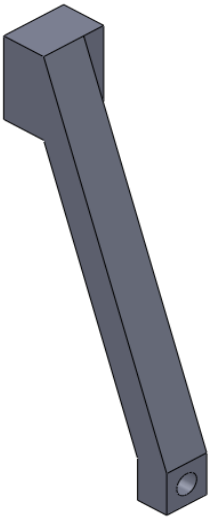

FIGURE 12: RIGHT ANGLE CONNECTOR

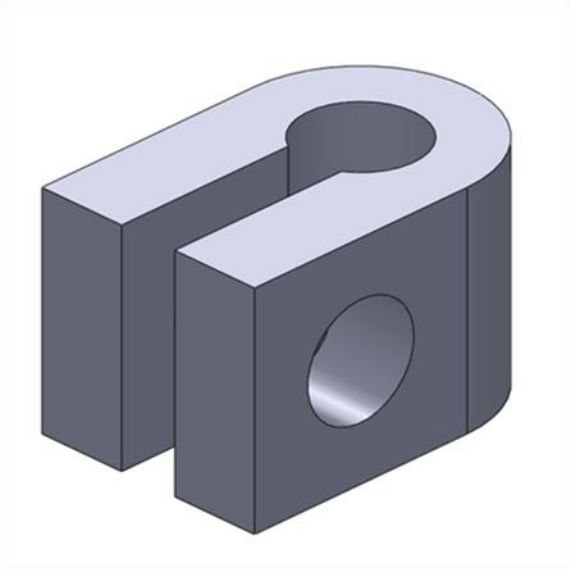

FIGURE 13: Y-BAR CLAMP

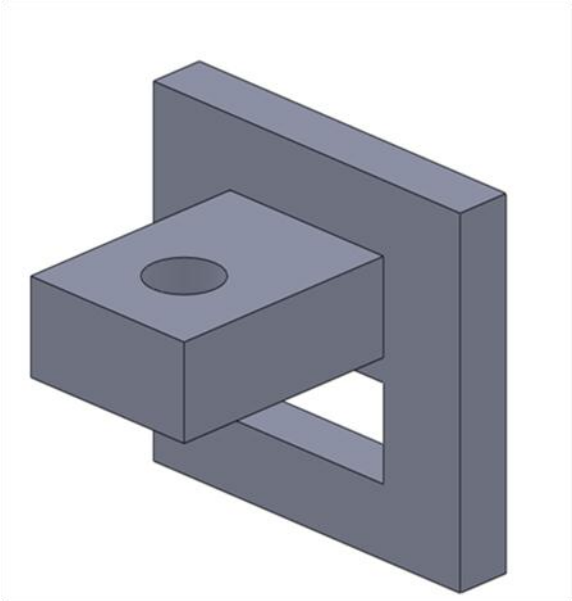

FIGURE 14: CARTRIDGE BELT CLAMP "T"

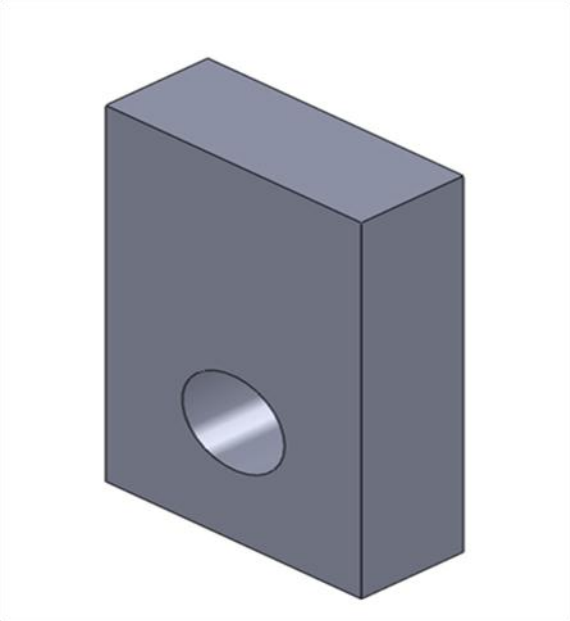

FIGURE 15: CARTRIDGE BELT CLAMP BAR

A breakdown of the cost of printing these parts is given below in Table 4. The costs were estimated based on the fact that it costs approximately $10.00 per hour to print on a Mendel.

TABLE 4: COST BREAKDOWN OF PRINTED CARRIAGE PARTS

Part Name	Printing time (Mendel)	Cost per part	Qty	Total cost
Y-bar clamp	15	$2.50	22	$55.00
Angle connector	50	$8.33	2	$16.66
Bearing	15	$2.50	4	$10.00
Y-axis motor mount	50	$8.33	1	$8.33
Motor mount clamp	25	$2.50	2	$5.00
Y-axis idler mount	45	$7.50	1	$7.50
Belt clamp bar	10	$1.67	1	$1.67
Belt clamp T	10	$1.67	1	$1.67
Print cartridge holder	100	$16.67	1	$16.67
Total				**$105.83**

In addition to the printed parts, various rods, nuts, washers and screws were needed to assemble the carriage. The majority of the rods used to construct the frame are ¼" threaded steel rods. The only exceptions are the two rods that the print head runs on which are ¼" smooth steel rods. Also, the x-axis rails which the entire carriage system runs on, are $^{5}/_{16}$" smooth steel rods. These rods needed to have a larger diameter in order to prevent them from bending under the weight of the carriage assembly. Nuts and washers are used to secure the connectors positioned on the threaded rods. Table 5 below gives the costs for these additional components along with the costs of the printed parts.

TABLE 5: COSTS OF ALL CARRIAGE ASSEMBLY COMPONENTS

Vendor	Part Number	Description	Qty	Cost/ea	Ext Cost
N/A	N/A	Y-bar clamp	22	$2.50	$55.00
N/A	N/A	Angle connector	2	$8.33	$16.66
N/A	N/A	Bearing	4	$2.50	$10.00
N/A	N/A	Y-axis motor mount	1	$8.33	$8.33
N/A	N/A	Motor mount clamp	2	$2.50	$5.00
N/A	N/A	Y-axis idler mount	1	$7.50	$7.50
N/A	N/A	Belt clamp bar	1	$1.67	$1.67
N/A	N/A	Belt clamp T	1	$1.67	$1.67
N/A	N/A	Print cartridge holder	1	$16.67	$16.67
McMaster	98841A029	¼" Threaded steel rod	6 ft	$0.53 per ft	$3.18
McMaster	1327K113	¼" Round steel rod	2 ft	$0.97 per ft	$1.95
McMaster	98841A030	$^5/_{16}$" Threaded steel rod	4 ft	$0.76 per ft	$3.03
McMaster	8893K41	$^5/_{16}$" Steel drill rod	6 ft	$1.43 per ft	$8.60
McMaster	93827A211	¼" - 20 Hex nut	35	$0.06 *	$2.10
McMaster	93827A219	$^5/_{16}$" – 18 Hex nut	24	$0.09 *	$2.16
McMaster	92497A200	3 mm Hex nut	6	$0.25 *	$1.50
McMaster	90279A537	¼" Screw	5	$0.06 *	$0.30
McMaster	92005A120	3 mm Screw	6	$0.03 *	$0.18
McMaster	91083A029	¼" Washer	22	$0.02 *	$0.44
McMaster	91166A210	3 mm Washer	3	$0.02 *	$0.06
Lowes	136987	¼" Nylon sleeve	2	$0.58	$1.16
Lowes	136994	$^5/_{16}$" Nylon sleeve	4	$0.58	$2.32
					$149.48

* Indicates the cost of the component is for bulk orders, ie packs of 100.

As the table above shows, the cost for the carriage assembly exceeded the imposed constraint of $100.00 for all the non-electrical components of the printer. However, this cost could be drastically reduced by making silicon molds of the printed parts and casting them using a polyurethane resin. Several of the parts are not moldable in their current configuration but with slight redesigns they could be made into molds fairly easily. Table 6 below shows the estimated costs involved in making molds and then casting the molded parts.

TABLE 6: COST OF MOLDS AND POLYURETHANE PARTS

Part Name	Qty Required	Cost of silicon mold	Cost of polyurethane	Cost each	Total cost
Y-bar clamp	22	$0.80	$0.20	$1.00	$22.00
Angle connector	2	$4.00	$0.33	$4.33	$8.66
Bearing	4	$0.80	$0.20	$1.00	$4.00
Y-axis motor mount	1	$1.80	$0.30	$2.10	$2.10
Motor mount clamp	2	$1.00	$0.25	$1.25	$2.50
Y-axis idler mount	1	$2.45	$0.45	$2.90	$2.90
Print cartridge holder	1	$3.25	$0.55	$3.80	$3.80
Total					**$45.96**

The above table demonstrates that the cost of the printed parts can be cut in half by making molds and casting in polyurethane. The cost can actually be reduced further because the table does not take into account the fact that one mold can be used to cast multiple sets of parts. Thus multiple printers could be made after the initial investment in one set of molds. After the initial investment in the molds is made, it will only cost about $8.00 to cast a set of parts.

FUTURE WORK

Although the first prototype is functional, there are some significant issues that should be addressed as work proceeds on this project. The first and most important step should be redesigning the printed parts so that they can be molded and then cast. As demonstrated earlier in the report, this would greatly reduce the cost of the carriage system.

ELECTRONICS/SOFTWARE

Constraints:

- <$100 total cost (1/2 total cost)
- Open tool chain, open design (non-proprietary software used for all aspects)

Goals:

- Easy to source materials
- Easy to assemble
- Easy to operate
- Versatile usage (can be used with or without a PC and possibly have multiple ways to connect)

CONCEPT GENERATION

There are several key portions of the electrical system that had to be designed. There are 2 axes that must be controlled, some form of inkjet head, limit switches, control buttons, and a method of print data input. All of this must be controlled by a computer or microcontroller.

Several options were considered for inkjet heads. The first option was to use an HP 51604 cartridge as there was a book (Gilliland, 2005) with information about directly controlling the head with a microcontroller and Darlington arrays. Some research was done into these small 12 nozzle cartridges and the HP C6602 was found. This head has a carriage holder that is available to purchase where the holder for the HP 51604 is not available. A few higher nozzle count cartridges were considered however they are much more complex to drive and there is very little documentation on their driving specifications. The simple 12 nozzle heads require 12 output lines to drive them. With additional control circuitry this could be dropped to 5 outputs (Gilliland, 2005).

There are two main types of stepper motors, unipolar and bipolar. Both types of motors are about the same cost but the driver requirements can be quite different. The unipolars can be controlled with a simple, inexpensive (<$0.50) transistor array while the bipolars require at least an H-bridge (~$1.50) and ideally a board in the $15.00 range. The simplicity of driving a unipolar comes from the fact that each of the four coils is driven only one direction, hence “one polar”, this means that the driver circuit does not have to change polarity of the current. However this also means that you potentially have 3 coils unused at any given time. A bipolar stepper only has 2 coils and they have the current change polarity therefore only one coil is unused at a time. This tends to make unipolar motors heavier for the same torque (Wikipedia). Either type requires 2-4 digital output lines per motor for a maximum of 8 digital lines.

Each axis also needs a minimum of one end stop to allow homing which is 2 input lines.

Options for connectivity are either communication over USB, or serial, or some method of data storage on the printer, like an SD card. Both of these methods have their advantages and we would

like both. This requires that the microcontroller have an SPI bus for the SD card and a serial bus. If the printer can print standalone then we also need some buttons and a display.

The above requirements lead to the following IO needs:

TABLE 7: IO NEEDS OF PRINTER

Task	# of I/O lines
Inkjet	12
Steppers & End stops	10
Display	6
Buttons	1-12
Serial	2 – Serial lines
SPI	4 – SPI lines
Total	29-40* + Serial & SPI **depending on number of buttons*

Comparing some of the options for available microcontroller boards we see that the mBed has a proprietary tool chain and not enough I/O lines (Table 8). The Arduino boards are open and have a very large user base. Of the two the Uno does not have enough I/O lines. The Maple is open and has a faster chip however the user base is not as large and there are fewer resources available for it.

Table 8: Microcontroller Board Comparison

Board	Cost	Chip	I/O	Notes
mbed NXP LPC1768	$59.00	96MHz Cortex-M3 - 32bit	25 GPIO	Proprietary tool chain New board with very little community support yet
Arduino Uno	$29.95	16MHz Atmega328 - 8bit	20 GPIO	Large user base and example code
Arduino Mega	$59.95	16MHz Atmega2560 - 8bit	70 GPIO	
LeafLabs Maple	$49.99	72MHz Cortex-M3 - 32bit	55 GPIO	Arduino-like IDE, but smaller user base

Both the mbed and the Arduino Mega were also tested to confirm the microsecond switching requirement could be met. Both passed this testing. However as the mBed is not open it was no longer an option. The Maple could also be a good choice however with the smaller number of available resources it is not quite as ideal a choice as the Mega.

The design of the firmware started after selecting electronics as it depends on the platform selected. Once the Arduino Mega was selected research was conducted in several areas:

The first tests were designed to ensure that all the nozzles could be addressed and that the Arduino could handle the timing required as defined in (Gilliland, 2005). This is a 5µs pulse with a 0.5µs delay before firing a different nozzle and an 800µs delay between two pulses on a given nozzle.

We managed to get microsecond pulsing to work on the MEGA. The digitalOUT() command is too slow but direct port manipulation works.
This code will output 1.2µs pulses with 2µs delays:

```
PORTB |= 10000000;
delayMicroseconds(1);
PORTB = PORTB & 01111111;
delayMicroseconds(1);
```

This gives 1.2µs ON & 2.7µs OFF:

```
PORTB |= 10000000;
delayMicroseconds(1);
PORTB = PORTB & 01111111;
delayMicroseconds(2);
```

This gives 2µs ON & 2µs OFF:

```
PORTB = PORTB & 01111111;
delayMicroseconds(2);
PORTB |= 10000000;
delayMicroseconds(1);
```

This led to the conclusion that the loop part of the program has about 0.7µs of overhead, as the last command before it starts again has an extra delay. I think we can work with this as the ONLY time critical part is not burning up the head which means we just need to make sure it is ON and back OFF within the loop.

The next tests were to determine voltage and pulse width ranges and see if these variables would affect the drop size or ink volume.

The first test involved holding the voltage at 19V and varying the pulse width from 5µs down until ink stopped spraying and then up until a nozzle failed. It was found that the minimum pulse width that would cause ink to spray was 3µs and the maximum before failure was 20µs. The size of the drops was also examined for each of these pulses and there were no noticeable changes from 3µs - 25µs.

The second test involved holding the pulse width at 5µs and varying the voltage from 19V down until ink stopped spraying and then up until a nozzle failed. It was found that the minimum voltage that would cause ink to spray was 17.5V and the maximum before failure was 27V. The size of the drops was also examined for each of these voltages and there were no noticeable changes from 17.5V – 27V.

This left multiple drops in one location as the only option for ink volume change.

The next tests were to test printing predefined patterns. To do this code was adapted from the Nickel-O-Matic (Davey, 2008) to test printing text. This worked well and was then adapted to read a text file and print it. This involved adapting example circuits and code from Lady Ada (Fried, Logger Shield, 2010).

After successfully printing text from a file on an SD card the next step was to read and print graphics data. This was our own code creation.

The main steps in printing are:

1. Convert BMP layers into 12bit slices for printing
2. Step in Y direction (96steps/inch) and fire the 12 pins to place the 12bit words.
3. Step in X direction (8steps/inch) and repeat the Y steps.
4. When max X reached, reset print head.

The first step to printing is taking a BMP and converting it into rows and strips. Each strip is the 12 nozzles we can fire and each row is the print width worth of strips.

Research was done with respect to the BMP format and methods of decoding them. Some excellent information was found on the format and the offsets (Circuit Ideas Design).

This then led to the determination of a binary file format for printing with 2 bytes per strip as follows:

TABLE 9: BINARY FILE FORMAT FOR PRINTING

bit	Odd bytes (1,3,5,7…)	Even bytes (2,4,6,8…)
0	Nozzle 1	Nozzle 9
1	Nozzle 2	Nozzle 10
2	Nozzle 3	Nozzle 11
3	Nozzle 4	Nozzle 12
4	Nozzle 5	**SPECIAL CODES**
5	Nozzle 6	**SPECIAL CODES**
6	Nozzle 7	**SPECIAL CODES**
7	Nozzle 8	**SPECIAL CODES**

i.e. 00110000 **XXXX**0110 would fire nozzles 5, 6, 10, 11 (XXXX are the special codes)
00000011 **XXXX**1100 would fire nozzles 1, 2, 11, 12 (XXXX are the special codes)

The special codes would allow for data compression and additional printer control:

0000 - ***SUPER SPECIAL CODE***
0001 - Print this slice 1 time - normal operation
0010 - Print this slice 2 times
0011 - Print this slice 3 times
...
1111 - Print this slice 15 times

SUPER SPECIAL CODE:
A super special code would indicate that the other 12 bits are not print data. This gives 4096 additional codes that can be used for things such as:

- Large expanses of nothing (only consumes 256 of our numbers...)
  - If we have even byte **0000**0000 then we could use odd byte to indicate blank space ranging from 1 slice to 256 slices (or 1/96" to 2.667") of course this duplicates **0001**000000000000 - **1111**000000000000 so maybe it means 16 slices to 272 slices (2.833")...
- end of row (reset for next row)
- saturation changes

- This could be a mode change – one code would switch to a grey scale mode where you would then have 8, 16, or ? slices that are all printed on top of each other (giving 8, 16, or ? shades of grey). That could happen in the beginning (setting it for the whole file) or possibly on the fly changing modes anywhere (this is more complex). *I am not sure what that does to the "repeat" special codes.*

- ~3838 more ***super special codes***...

Processing.org seemed to be a natural place to look into writing this pre-processor as it is very similar to Arduino. This lead to a script that reads the BMP file 12 rows at a time, decodes the picture data, and saves it in our new data format. This is also converted from left to right (Figure 16) on every row to a zig zag pattern (Figure 17).

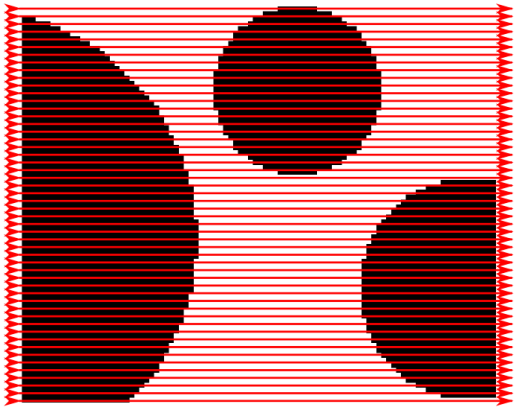

FIGURE 16: BITMAP DATA ARRANGEMENT

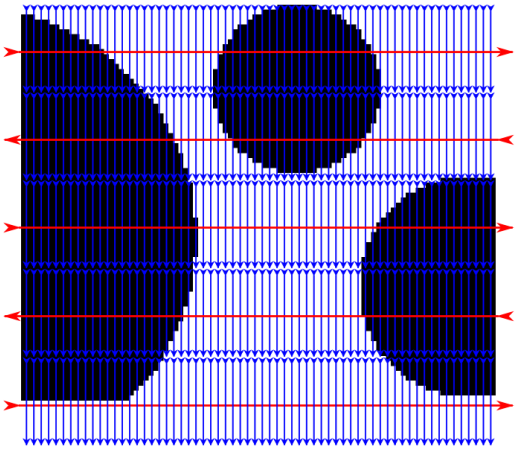

FIGURE 17: PRINT DATA ARRANGEMENT

Then an LCD screen was tested using example circuits and code adapted from Lady Ada (Fried, Character LCDs, 2010).

Steppers were added using example circuits and code adapted from the Arduino website (Igoe, 2007). These worked with minimal additional adjustments.

Initially multiple buttons on a single analog pin were tested (Dougl, 2008) (neillzero, 2006). However there were issues with consistently reading the correct presses and as you add more buttons it becomes very complex to manage them all this way.

After more research we found a routine for managing multiple buttons with de-bouncing (Fried, Blog: Adafruit, 2009). This was setup to allow you to specify the pins you had buttons connected to and the rest was handled for you. This was very scalable and efficient code that was used almost un-modified.

The final design incorporates an Arduino Mega, which was selected do to the large community behind the Arduino, availability of libraries to perform things like LCD, SD card, and stepper control. The HP C6602 ink cartridge was selected to the availability of the cartridge holder and the available driving specifications. Do to the ease of connecting and the low cost of drivers, unipolar motors were selected. However, nothing in the design would preclude the use of bipolar motors in the future. They would just require replacing the Darlington arrays with stepper drivers. PC connectivity via USB (or serial) requires having some addition PC side host software. It was determined that for the initial prototype this was not feasible. Using an SD card, some buttons, and a display would take less time and allow us to prove out the major concepts.

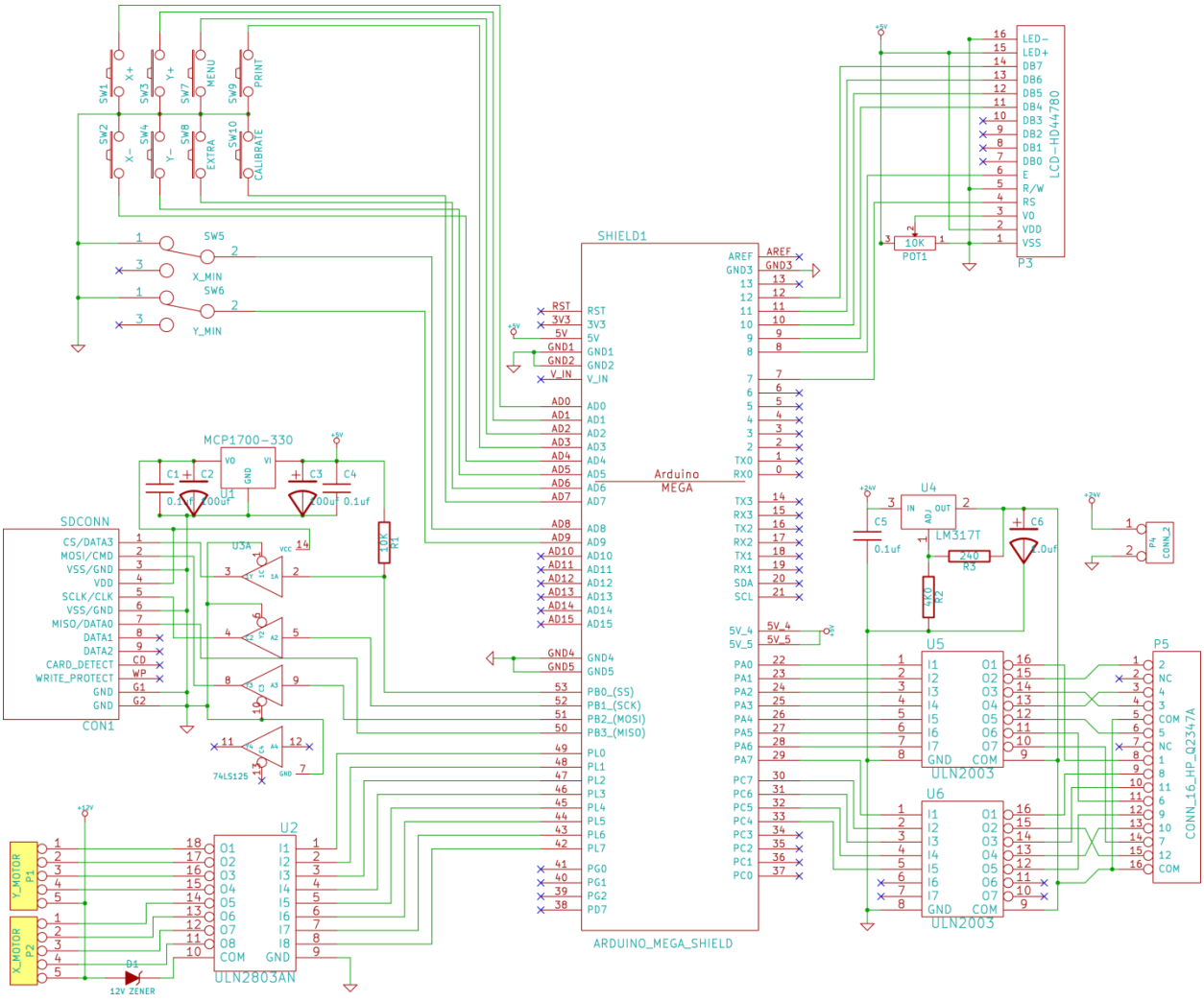

FIGURE 18: SCHEMATIC OF PROTOTYPE CIRCUIT

Overall we were able to keep the cost of the electronics under the budgeted amount with a total of $130.58 (plus shipping) and this cost could be brought lower with some more thrifty choices like an Arduino Clone (~$30.00 savings).

TABLE 10: COST BREAKDOWN OF ELECTRONIC COMPONENTS

Vendor	Part Number	Description	Qty	Cost/ea	Ext Cost
Digikey	490-5380-ND	0.1uf Capacitor	3	$ 0.29	$ 0.87
Digikey	493-1099-ND	1.0uf Capacitor	1	$ 0.17	$ 0.17
Digikey	493-1040-ND	100uf Capacitor	2	$ 0.17	$ 0.34
Digikey	101-00708-64-ND	SDCONN	1	$ 1.33	$ 1.33
Digikey	1N5242BFSCT-ND	12V ZENER	1	$ 0.27	$ 0.27
Digikey	A19432-ND	5pin Connector	2	$ 0.32	$ 0.64
Digikey	67-1781-ND	LCD-HD44780	1	$ 7.79	$ 7.79
Digikey	609-1944-ND	CONN_16_HP_Q2347A	1	$ 0.85	$ 0.85
Digikey	3M9447-ND	2pin Connector	2	$ 0.09	$ 0.18
Digikey	262UR103B-ND	10K POT	1	$ 0.51	$ 0.51
Digikey	CF14JT10K0CT-ND	10K ohm Resistor	1	$ 0.08	$ 0.08
Digikey	CF14JT3K90CT-ND	4K0 ohm Resistor	1	$ 0.08	$ 0.08
Digikey	CF14JT240RCT-ND	240 ohm Resistor	1	$ 0.08	$ 0.08
Adafruit	Arduino Mega 2560	Arduino Mega 2560	1	$ 65.00	$ 65.00
Digikey	P12349S-ND	Push Button Switch (NO)	8	$ 1.13	$ 9.04
Digikey	SW776-ND	Micro Switch (NC)	2	$ 0.92	$ 1.84
Digikey	296-19046-5-ND	ULN2803AN	1	$ 0.96	$ 0.96
Digikey	MCP1700-3302E/TO-ND	MCP1700-330	1	$ 0.44	$ 0.44
Digikey	296-1638-5-ND	74LS125	1	$ 0.79	$ 0.79
Digikey	LM317TFS-ND	LM317T	1	$ 0.68	$ 0.68
Digikey	ULN2003APG-ND	ULN2003	2	$ 0.52	$ 1.04
Transact	200-00299	HP Carriage Assembly (HP Q2347A)	1	$ 8.25	$ 8.25
Transact	98-01570	BLACK Inkjet Cartridge (HP C6602A)	1	$ 15.45	$ 15.45
Alltronics	417-11-48-02	Stepper Motor	2	$ 6.95	$ 13.90
					$ 130.58

The firmware that was settled on for the prototype follows the basic flow shown in Figure 19. This allows manual jogging of the axes, purging or testing of the inkjet, adjusting motor speeds, and starting a print. All of this was implemented in Arduino code combining all the aspects tested in the concept generation phase.

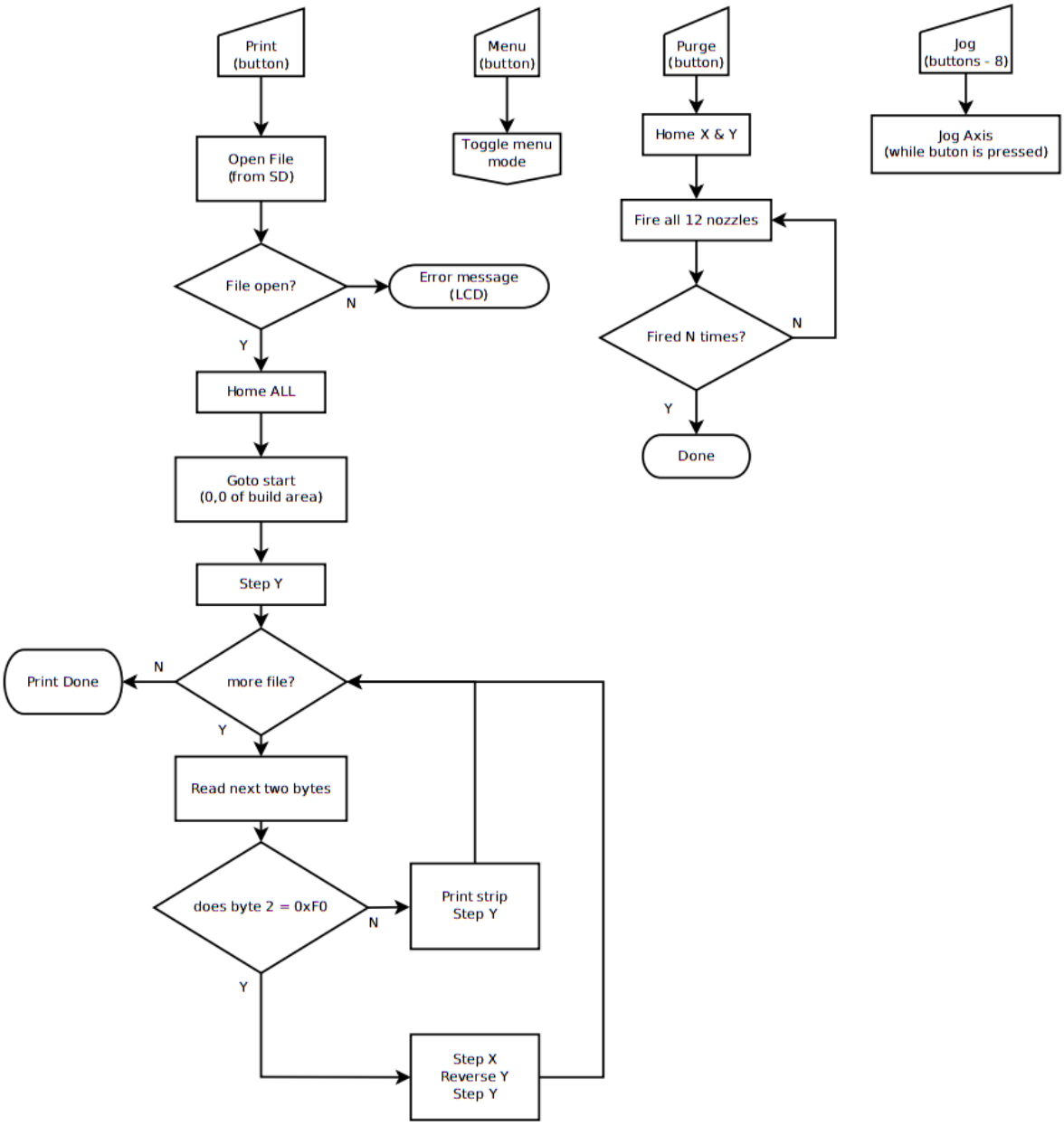

FIGURE 19: FLOW CHART OF PRINTER OPERATIONS

It was determined that variable saturation and file compression using “Special Codes” as outlined in the concept phase was unnecessary at this time and a simplified set was chosen.

TABLE 11: SIMPLIFIED CODE SET

Data	First Byte	Second Byte
Print Data	XXXXXXXXX*	**0000**XXXX*
Step Row	**Anything**	**11110000**

**X’s denote print information*

This was implemented in both the Arduino code and a custom Proccessing.org script that reads in a single BMP and outputs a binary print data file.

One area of concern is the print speed. This is limited by two factors. One is the maximum speed at which the head can be fired. If you fire the nozzles with the minimum 800µs delay then you have:
One "page" is (96dpi)(5in)/(12dot per row) = 40 rows
One row is (96strips per in)(5in)=480 strips
Converting to time:
One row is (800µs/strip)(480 strips)= 384ms
One "page" is (40 rows)(384ms) = 15.36s

This is the theoretical max speed given a full 5"x5" "page" and only a single nozzle firing (i.e. saturation = 1) at each location. For our tests we were firing 3 times in each location to obtain the desired saturation. This would bring the max speed down to about 45s/"page". Additionally from our preliminary testing we have seen that the y-axis motor takes ~16000µs to step. This leads to the following print speed:
One row is (16000µs/ strip)(480 strips)= 7680ms
One "page" is (40 rows)(7680 ms) = 307.2s

From empirical testing of a 2"x2" print area it took approximately 120s. This gives:
(120s)/(4in^2)(25in^2)=150s/"page"*
which is about twice as fast as the initial stepper testing but still shows that the current stepper motor is a limiting factor on print speed.

After replacing the y-axis motor we have obtained about double the testing speed or ~75s/"page".

During the testing of the prototype it was found that the only changes that had to be made were minor logic errors. Overall the firmware worked well and allowed us to print.

FUTURE WORK

Another area that still needs work is the development of host side software and improvements of the pre-processing. This should be combined into a single tool to allow the user to open a BMP and convert it to print data which can then be saved to an SD card or sent to the printer over USB. The software should also allow the user to adjust settings and save them for reuse.

The firmware should be refined to allow connectivity to the host side software and allow additional control and setting adjustment on the printer.

The next area that needs further development is a PCB design. This could either be a shield for the Arduino Mega or a dedicated board.

Finally more advanced inkjet heads should be looked at to improve both the resolution and speed. These could also give the ability to print in color.

ECONOMIC EVALUATION/COST LIST

The total cost of the printer, assuming all printed parts were printed on the Mendel, comes out to be $467.12, as shown in Table 12.

TABLE 12: COST BREAKDOWN BY SUB-SYSTEM

Sub-System	Ext Cost
X/Y Axis Carriage System (hardware)	$ 44.15
(printed parts)	$ 105.83
Electronics/Software	$ 130.58
	$ 280.56

This cost exceeds the design constraint of <$200 for the printer, however this cost is not necessarily representative of a final version of this printer. One of the key production changes would be molded parts. The cost of printing parts, even on the Mendel, is expensive compared to most of the other purchased items. If the printed parts could then be used to make molds, then parts could be made directly from the molds. Mold material, i.e. silicone, is much cheaper than the plastics used in FDM printers as is mold part material, like polyurethane. Another benefit of molding parts is a reduction in production time. Silicone molds could potentially take up to a day to set, but polyurethane parts take mere minutes to set. In all, once molds are made, parts could be produced far quicker and cheaper. An estimated cost for production via molding is shown in Table 13.

TABLE 13: ESTIMATED COST BREAKDOWN FOR MOLDED PRINTER

Sub-System	Ext Cost
X/Y Axis Carriage System (hardware)	$ 44.15
(molded parts)	$ 45.96
Electronics/Software (Arduino clone)	$ 100.58
	$ 190.69

It is clear that molding parts greatly reduces the cost of the system. The electronics' costs, are mostly beyond buyer control. They are a necessary expense and, aside from choosing vendors wisely, are subject to market prices. However, by utilizing an Arduino clone rather the official Arduino Mega, the total cost can be brought below our $200 constraint. Aside from the electronics, the other hardware was fairly inexpensive and easy to purchase at any local hardware store. These hardware costs will be brought down further in future optimization of the design, as excess material was used for the initial prototype.

BIBLIOGRAPHY

96 dpi Serial Inkjet Printer Development Kit (#27949). (n.d.). Retrieved from Parallax Web Site: www.parallax.com/dl/docs/prod/robo/InkjetKitDocs-v1.0.pdf

Circuit Ideas Design. (n.d.). *Build photo frame with AVR and BL-TFT240320PLUS*. Retrieved from Circuit Ideas Design: http://www.circuitidea.com/Article/DIY-photo-frame-with-BL-TFT240320PLUS.html

Davey, M. (2008). *Nickel-O-Matic Robot*. Retrieved from Parallax Inc Web site: http://www.parallax.com/tabid/769/Default.aspx

Dougl. (2008, November 17). *One analog pin to read 5 switches* . Retrieved from Arduino: Forum: http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1226896251

Fried, L. ". (2009, October 20). *Blog: Adafruit*. Retrieved from Adafruit Industries: http://www.adafruit.com/blog/2009/10/20/example-code-for-multi-button-checker-with-debouncing/

Fried, L. ". (2010). *Character LCDs*. Retrieved from Ladyada.net: http://www.ladyada.net/learn/lcd/charlcd.html

Fried, L. ". (2010). *Logger Shield*. Retrieved from Ladyada.net: http://www.ladyada.net/make/logshield/design.html

Gilliland, M. (2005). *Inkjet Applications.* Woodglen Press .

Igoe, T. (2007, May 17). *Stepper Library*. Retrieved from Arduino: http://www.arduino.cc/en/Reference/Stepper

neillzero. (2006, October 16). *LCD4Bit v0.1 (http://abstractplain.net).* Retrieved from R Cube Station: http://www.r3cube.com/Drivers-software/LCD4Bit_mod.zip

Wikipedia. (n.d.). *Stepper Motor*. Retrieved from Wikipedia: http://en.wikipedia.org/wiki/Stepper_motor