

University of Puerto Rico
Mayagüez Campus
Department of Electrical and Computer Engineering



PyroBot

by

Jonathan Gonzalez
Alex Suarez
Angel Viera
Jose Hernandez

For: Professor Manuel A. Jimenez
Course: ICOM 5217, Section 88
Date: May 31, 2012

INDEX

| | |
|--|----|
| Abstract | 7 |
| 1 Introduction..... | 8 |
| 1.1 Pyrography | 8 |
| 1.2 Motivation..... | 9 |
| 1.3 Unique Features..... | 9 |
| 2 Theoretical Background..... | 10 |
| 2.1 Laser & Grayscale..... | 10 |
| 2.2 CNC Market..... | 11 |
| 2.2.1 Market Analysis..... | 11 |
| 2.2.2 CNC Structure..... | 11 |
| 3 System overview | 13 |
| 3.1 Structural Design..... | 13 |
| 3.1.1 Linear Bearing System..... | 14 |
| 3.1.2 Structure | 16 |
| 3.1.3 Parts..... | 20 |
| 3.2 System Block Diagram | 21 |
| 3.3 Architectural Components | 22 |
| 3.4 Software Overview..... | 23 |
| 3.5 Core Design decisions..... | 23 |
| 3.5.1 MCU Selection..... | 23 |
| 3.5.2 Laser Choice | 25 |
| 3.5.3 Stepper Choice..... | 26 |
| 3.5.4 Stepper Torque..... | 26 |
| 3.5.5 Optical Sensor, Temperature Sensor, and Buzzer | 26 |

| | | |
|-------|---|----|
| 3.5.6 | Power Supply | 27 |
| 3.5.7 | Why did we use the TI DRV8825? | 27 |
| 3.5.8 | Schmitt Triggers | 28 |
| 3.6 | TI Chips Used | 28 |
| 4 | Hardware Design | 30 |
| 4.1 | Laser and Laser Driver | 30 |
| 4.1.1 | Laser Driver Design design considerations..... | 31 |
| 4.1.2 | Regulator Power and Thermal Calculations: | 33 |
| 4.2 | Stepper and Stepper Driver Design..... | 34 |
| 4.2.1 | Type..... | 34 |
| 4.2.2 | Stepping | 35 |
| 4.2.3 | Microstepping | 35 |
| 4.2.4 | Stepper Drivers | 36 |
| 4.2.5 | Driving Steppers Using H-Bridges | 36 |
| 4.2.6 | PowerPad | 43 |
| 4.2.7 | Protection circuits | 43 |
| 4.3 | Temperature and Optical Sensor | 44 |
| 4.4 | LCD / Touchscreen | 46 |
| 4.4.1 | Touch Response | 47 |
| 4.5 | Driving Analysis | 48 |
| 4.5.1 | GPIO Supply Current Compatibility | 48 |
| 4.5.2 | Voltage Logic Levels Compatibility..... | 49 |
| 4.5.3 | Laser driver calculations: | 50 |
| 4.5.4 | Buzzer Driver Calculations | 50 |
| 4.5.5 | Laser's Heatsink Fan Driver calculations | 51 |
| 4.5.6 | Laser's Smoke Fan Driver calculations | 52 |

| | | |
|--------|---|----|
| 4.6 | Power Supply Analysis | 54 |
| 4.6.1 | Laser's Driver | 54 |
| 4.6.2 | Stepper Driver Circuit | 54 |
| 4.6.3 | Maple Native | 54 |
| 4.6.4 | 254-35C1-ROX Buzzer | 55 |
| 4.6.5 | RPR-220 Optical Sensor | 55 |
| 4.6.6 | ULCD-32PT LCD-TFT Screen | 55 |
| 4.6.7 | LM34CZ Temperature Sensor's Circuit | 56 |
| 4.6.8 | MB40100V2 Laser's Heatsink Fan | 56 |
| 4.6.9 | MB20100V2 Laser's Smoke Fan | 56 |
| 4.6.10 | MB50100V2 Rear Fan x3 | 56 |
| 4.7 | Port Loading | 57 |
| 4.7.1 | IO Port Current Loading | 57 |
| 4.7.2 | STM32F103ZET6 Package Power & Temperature | 59 |
| 4.8 | Timing Analysis | 59 |
| 4.8.1 | Timing Bases | 59 |
| 4.8.2 | Laser's PWM Timer | 59 |
| 4.8.3 | Buzzer's Timer | 60 |
| 4.8.4 | Progress Bar's Timer | 60 |
| 4.8.5 | ULCD-32PT LCD-TFT Screen Timing | 60 |
| 4.8.6 | DRV8825 Stepper Driver Timing | 60 |
| 4.8.7 | Serial Communication between Host & System | 61 |
| 4.9 | MCU Memory Usage | 62 |
| 4.10 | Hardware Reliability & Professional Component | 63 |
| 5 | Software Design | 64 |
| 5.1 | Firmware | 64 |

| | | |
|-------|--|-----|
| 5.1.1 | Initialization State | 65 |
| 5.1.2 | Connecting State..... | 66 |
| 5.1.3 | Uploading State..... | 67 |
| 5.1.4 | Ready State..... | 72 |
| 5.1.5 | Burning State | 74 |
| 5.1.6 | Shutdown State | 76 |
| 5.1.7 | Emergency Lock State | 77 |
| 5.2 | Host application | 77 |
| 5.2.1 | Connecting with The System..... | 79 |
| 5.2.2 | Selecting an Image..... | 80 |
| 5.2.3 | Uploading an Image to the System..... | 81 |
| 5.2.4 | Asynchronous Behavior | 84 |
| 5.3 | Serial Communication between Host & System..... | 85 |
| 5.3.1 | Command Packet Structure | 85 |
| 5.3.2 | Data Packet Structure..... | 86 |
| 5.3.3 | Image Upload Packet Structure | 86 |
| 5.4 | Reliability and Professional Considerations..... | 87 |
| 6 | Results..... | 89 |
| 6.1 | System Photos..... | 90 |
| 6.2 | Art created by PyroBot..... | 94 |
| 7 | Conclusions..... | 100 |
| 8 | Future Work..... | 101 |
| 9 | Works Cited | 102 |
| 10 | Appendix..... | 106 |
| 10.1 | Full Schematic..... | 106 |
| 10.2 | Part List and Cost Analysis..... | 108 |

| | | |
|--------|-------------------|-----|
| 10.3 | Users Guide | 111 |
| 10.3.1 | Host | 111 |
| 10.3.2 | System | 124 |

ABSTRACT

Systems like PyroBot allow us to emulate the skilled hands of an artist with hardware and software. The art form PyroBot focuses on is Pyrography. PyroBot can create an art form analog to pyrography several times faster than if done by hand. Our system does this by combining a laser with open hardware CNC solutions. The art is created by first sending a grayscale image from a host application to an embedded system controlled by an ARM STM32 series microcontroller. The image sent can then be burned on a piece of wood without the need of a host being present. PyroBot differentiates itself from its competition mainly because of its cost, which is about 5 times less than a normal CNC machine, and its ability to burn images in different shades of grey, instead of just monochrome. Through PyroBot, users can achieve results as beautiful and artful as with a human hand.

1 INTRODUCTION

PyroBot was created under the belief that technology can be inspired and designed with the purpose of creating art. We believed that artist should not be limited to the things they can create with their hands, but should be allowed to explore the things they can create with their mind. PyroBot is a machine that allows anybody to become a pyrography artist and create beautiful works of art on wood.

1.1 PYROGRAPHY

Pyrography is an art form in which an image is burned onto a surface using a heating device.^[1] Wood is a common host surface for this art form. Contrast and different shades of color are created by varying the intensity of heat on a device similar to a soldering iron with varied tips. The artist chooses the tips that best fit a particular design. This art form is known for its beautiful and natural high contrast colors. Below is an example of a pyrographed wooden canvas:



Fig. 1. Pyrography example created by StefyMante. [Creative Commons]

Our project, known as PyroBot, will aim to mimic the art of pyrography with the use of a Computer Numerical Control, also known as a CNC.^[2] However, unlike standard pyrography, the image will be

transferred onto the wood with the use of a laser. A software application on a PC will allow users to specify the image to transfer.

1.2 MOTIVATION

There were various motivations behind the creation of PyroBot. Our main motivation was to push the limits of art and technology with the purpose of creating a low cost solution that anyone could obtain. However, we also felt passionate about building a system out of mostly open hardware solutions, allowing hobbyist and students to build on top of our initial design. We believe that not only does that improve the overall advancement in the field, but also motivates future generations to enter the field of embedded systems. Lastly, we were motivated to help the unconventional artist. With PyroBot, any artist, photographer, or person with imagination can create beautiful wooden art. This can potentially create a new market for natural art forms where everyone can be involved. It also raises opportunities for aspiring artists that lack the skills to create art with their hands. For these and many more reasons we were motivated to create PyroBot.

1.3 UNIQUE FEATURES

When we set out to design PyroBot we intended it to be a unique and innovative CNC system. We achieved this by integrating many unique and useful features into our design. PyroBot is unique from conventional laser CNCs for the following key reasons:

- PyroBot has a very low cost compared to conventional CNCs. This cost is as low as 1/5 of the cost of most CNCs.
- PyroBot is a standalone machine that can be run with minimum support from a host. This is distinctive among CNCs.
- PyroBot has a “soft” laser. Most conventional laser CNCs are too powerful to correctly burn a grayscale onto the surface of wood. These CNCs tend to penetrate the surface and damage the wood. PyroBot is unique in this because its power is regulated so as to only lightly toast the surface, keeping the woods natural features intact.
- PyroBot increases operator security by containing the following security components:
 - Optical sensor used for wood detection, preventing the system from burning without proper wood being placed.
 - Temperature sensor that monitors the laser is working within temperature ratings.

- Emergency stop button for instant stop of the system regardless of what it's doing.
- Emergency backup that stores the systems state on power loss and informs the operator of a bad shutdown.
- PyroBot has an LCD Touchscreen that supplies a simple and easy user interface.
- PyroBot can manage 8-bit images up to 1MB and can print on any wooden size up to 9" x 7.2". However, the true working area of PyroBot is 10" x 8".
- PyroBot has a standard dpi of 141 but can be calibrated to work up to 282dpi and more.
- PyroBot is an open hardware and low cost solution that can be replicated and assembled by hand.

2 THEORETICAL BACKGROUND

Many design decisions were based heavily on the theoretical background of its components. For this reason we will cover a few basics before entering into the detailed designs. More theory will be discussed once we enter the detailed design of each component. For now we will focus on two main areas: how we achieve grayscale using a laser, and CNC basics.

2.1 LASER & GRAYSCALE

In order to accurately replicate the art of pyrography, we needed an instrument that was both precise and had the capability of burning wood without damaging it. Research brought laser engravers to our attention. These machines use a high power laser in order to etch patterns onto a surface with a very high degree of precision. Further investigation revealed that it was possible to burn an image onto a surface without etching it using a laser. This could be done as long as you kept the surface temperature around 700F for a short period of time. It was clear then that a laser would fit our requirements precisely.

Achieving a grayscale gradient using a laser required us to control the method in which the beam irradiates the wooden surface. Two methods were immediately available to us: either variation of the time of exposure, or variation of the beam's intensity. We determined experimentally that the former produced better results than the latter. Pulse width modulation was used to vary the average energy irradiated onto the wooden surface. The grayscale that resulted from this method proved too inaccurate for our needs. Modulating time resulted in a 60 value gradient that, while not linear, seemed good enough for our purposes. From this scale, we extracted the 32 most representative values and used those for the system's grayscale. Figure 2 shows a few of our experimental results with laser variation.

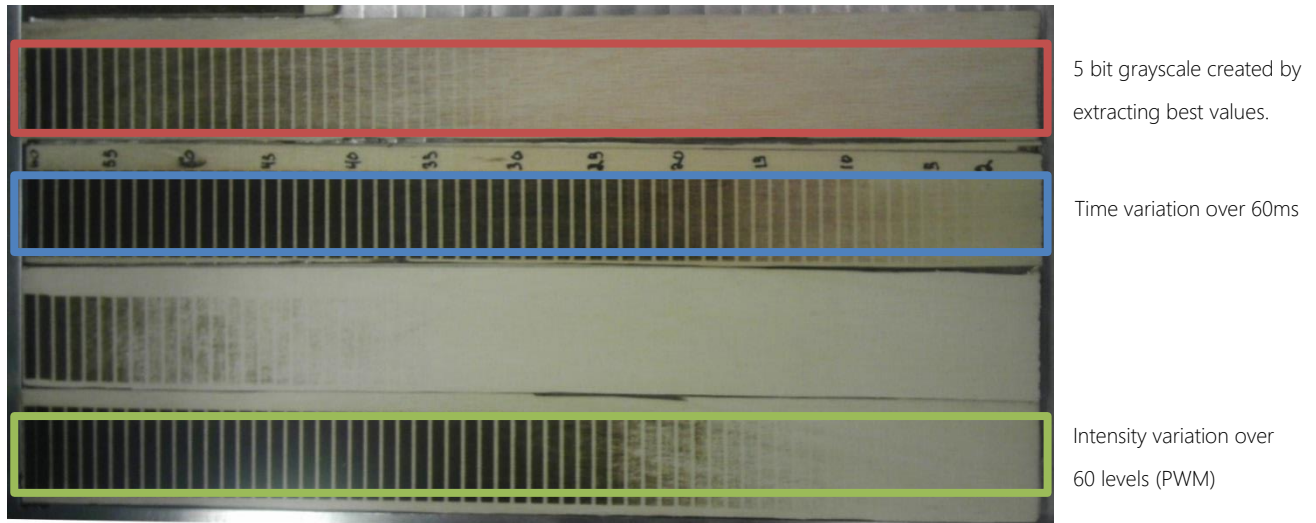


Fig. 2. Some of our experimental results. As can be seen, varying time proved to be much more effective in producing grayscale.

2.2 CNC MARKET

The CNC component of our project proved to be one of the most important and key components in our design. It was this component that would define the overall cost of PyroBot.

2.2.1 MARKET ANALYSIS

Commercial laser based CNC's are used for two main purposes: cutting and engraving (or marking) on different kinds of materials, depending on the intensity of the laser. Powerful laser CNC's, such as the XY-YAG-500 [3], which uses a YAG laser, are primarily used for cutting and scribing metals and ceramics. However, The Epilog Legend Helix 24 works well on wood, acrylic, and plastics. [4] The price of the Epilog Legend Helix was \$10,000.

It became apparent that if we wanted to build PyroBot, at a low cost, we would have to go with custom built and open hardware designs. However, most of these designs were intended for other uses, so modifications had to be made.

2.2.2 CNC STRUCTURE

Overall, you can divide most CNC structures into two components; a base component, and a laser/tool mount component. Which component moves and to what degree defines the rest of the structure. The size and complexity of the structure is inversely proportional to how much the base moves. Figure 3 shows this relationship.

Increase in base
movement and size
of overall structure



- Laser Mount moves in X and Y axis.
- Laser and Base move in one axis.
- Base moves in X and Y axis.



Increase in complexity

Fig. 3. The relationship between base movement and complexity.

3 SYSTEM OVERVIEW

Overall, PyroBot's design consists of a CNC design with an embedded system at its heart. For simplicities sake, it is best to see the CNC and embedded components separately.

3.1 STRUCTURAL DESIGN

The CNC structure of PyroBot proved to be one of the most difficult components to design. Initially, PyroBot was intended to be a happy medium between the base and laser mount when it came to movement. Figure 4 shows one of the initial designs.

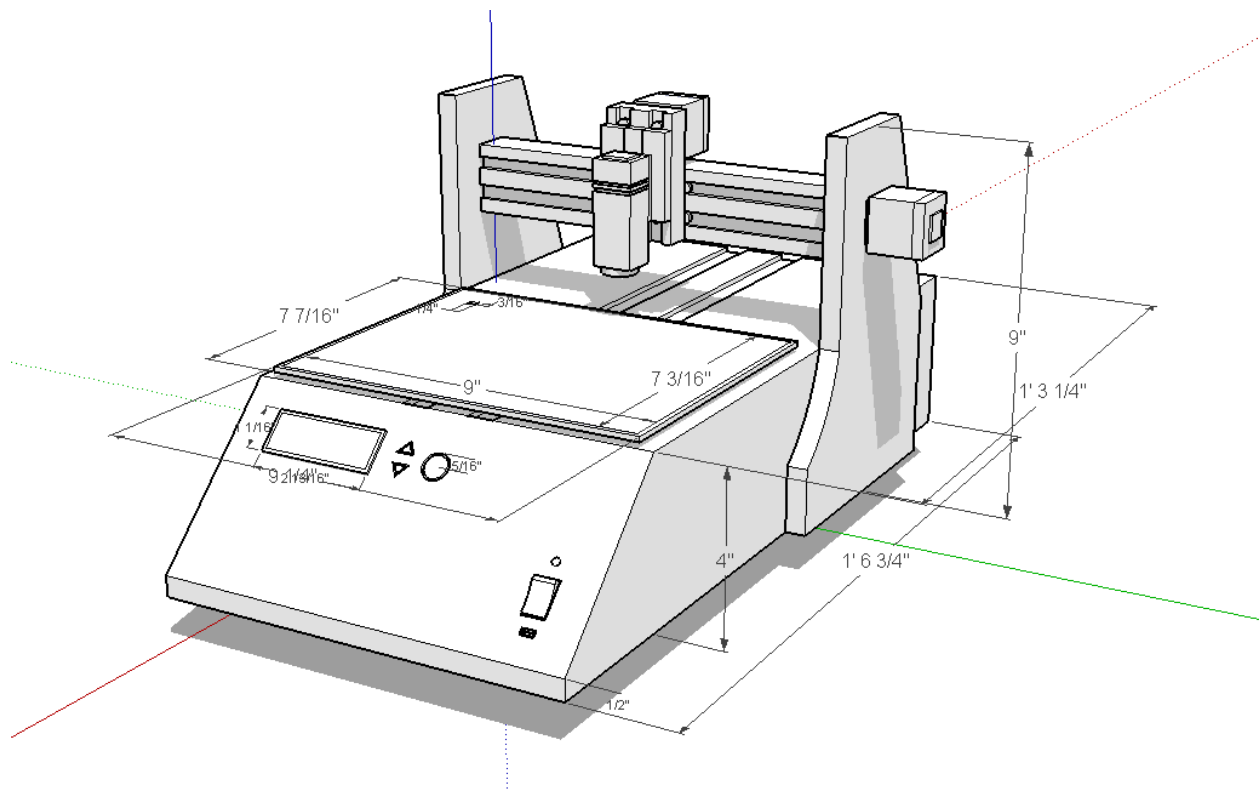


Fig. 4. An initial PyroBot design.

However, in the end, it actually became less complex to allow the laser to move on all axis. In addition, we were able to find an open hardware design called the ShapeOko that integrated a lot of our needs. For this reason we modified our PyroBot design to match the ShapeOko linear bearing system. The ShapeOko design was created by Edward Ford [5].

3.1.1 LINEAR BEARING SYSTEM

One of the biggest design, cost and fabrication hurdles in a new CNC design are the linear bearings. [6]
The linear bearing system is what allows the system to move on its axis.

3.1.1.1 MAKERSLIDE

Most linear bearing systems are expensive or require accurate fabrication to assemble. PyroBot uses an open hardware solution that is inexpensive and fits with standard aluminum extrusion systems. Our linear bearing system is called the MakerSlide and was created by Barton Dring. [6]

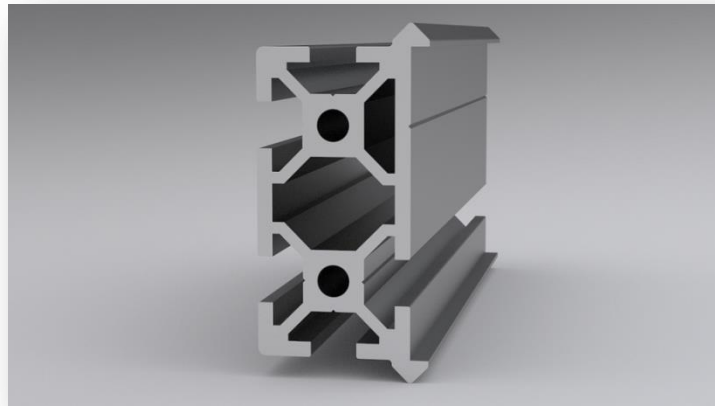


Fig. 5. The MakerSlide Extrusion with built in V rail. [6]

The maker slide consists of a V rail integrated into a standard extrusion profile. [6] This V rail allows for a V type wheel to “slide” on the extrusion. The following figure illustrates the concept.

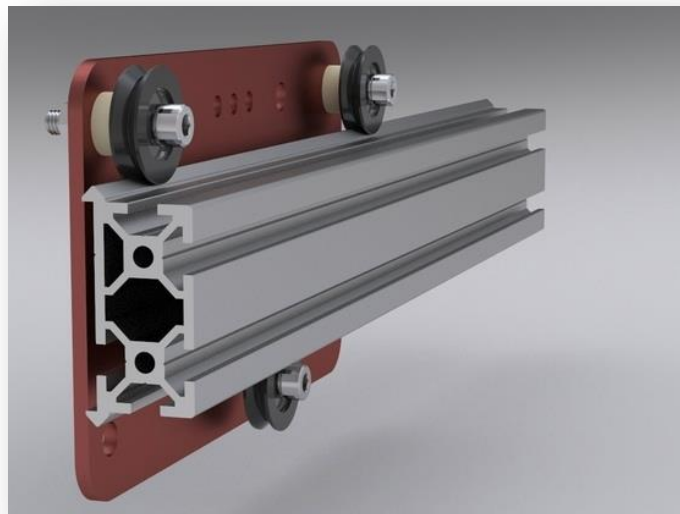


Fig. 6. V Wheel on MakerSlide. [6]

3.1.1.2 LINEAR MOVEMENT

The movement on the MakerSlide is created by using a belt drive system. This is necessary to guarantee consistent movement across the rail.

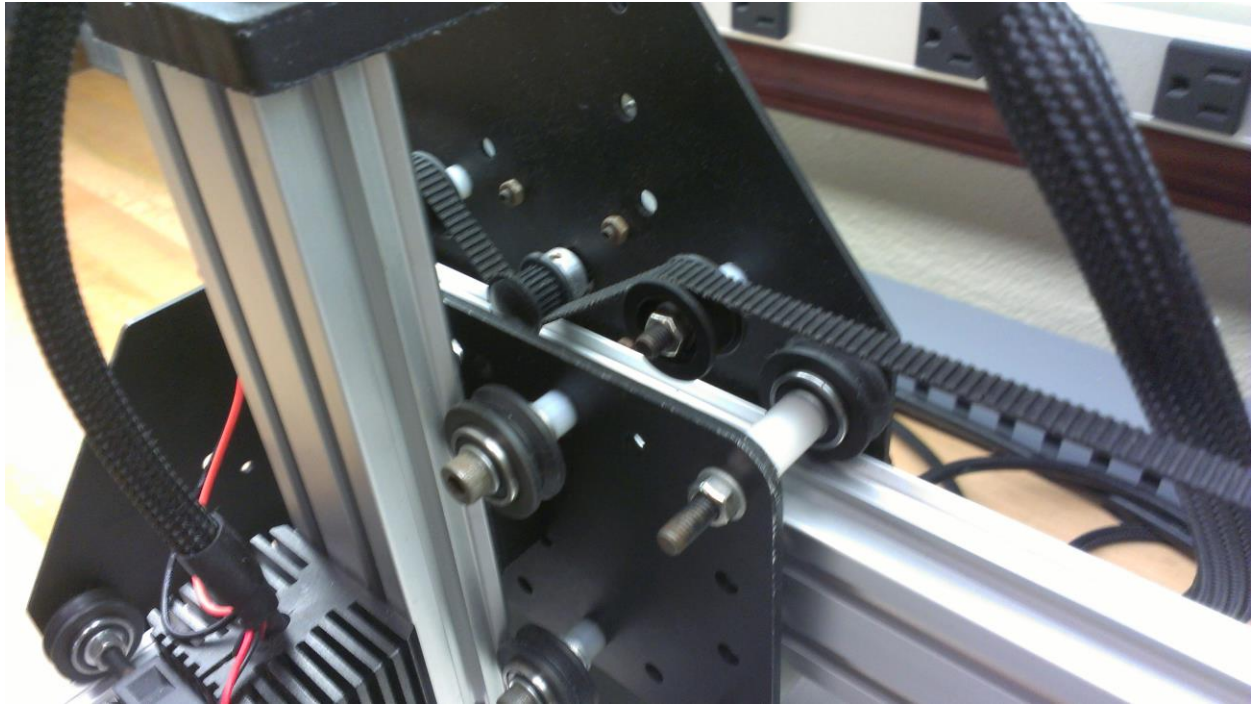


Fig. 7. The belt drive system used in PyroBot.

The size of the teeth's on the belt and the type of pulley used is defined by our dpi. PyroBot uses an MXL type belt that has a 0.08 inch pitch. It runs this belt by an 18 tooth pulley that is attached to a 400 step stepper. We can calculate the precision with the following equation:

$$\text{resolution} = \text{stepperSteps} / \text{pulleyTeeth} / \text{beltpitch}$$

$$\text{stepperSteps} = 400 \text{ steps / rev}$$

$$\text{pulleyTeeth} = 18$$

$$\text{beltpitch} = 0.08 \text{ inches}$$

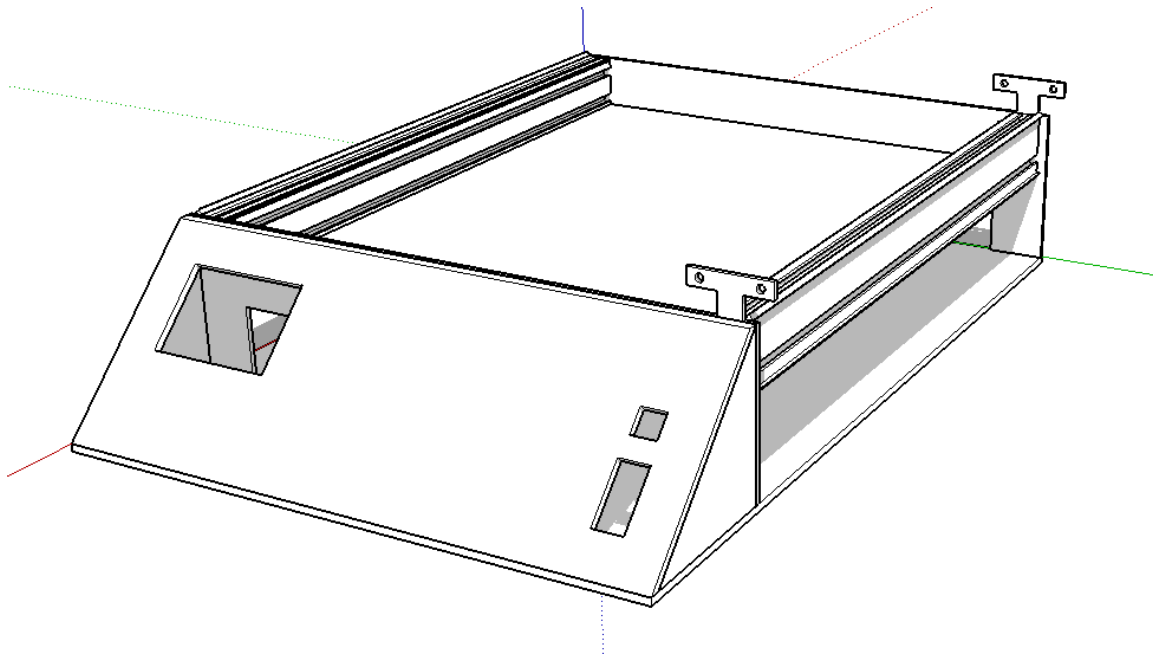
$$400 / 18 / 0.08 = 277.778$$

This gives us a resolution of 277.778 steps per inch which is roughly 0.09mm per steps. With two steps we reach the resolution we desire of 1.8mm.

3.1.2 STRUCTURE

The PyroBot main support structure is comprised of various plates that are either welded or screwed together. The plates can be divided up into the following:

- The front face plate.
- The side triangles.
- The back and center support plates. These hold the MakerSlides in place.
- The wood support plate that is lodged between the MakerSlides.
- The system base.



1.1 The main structure of PyroBot.

3.1.2.1 FRONT FACE PLATE

The front plate holds the LCD Touchscreen, emergency stop button, and on / off switch. It is the main point of contact between the system and the user.

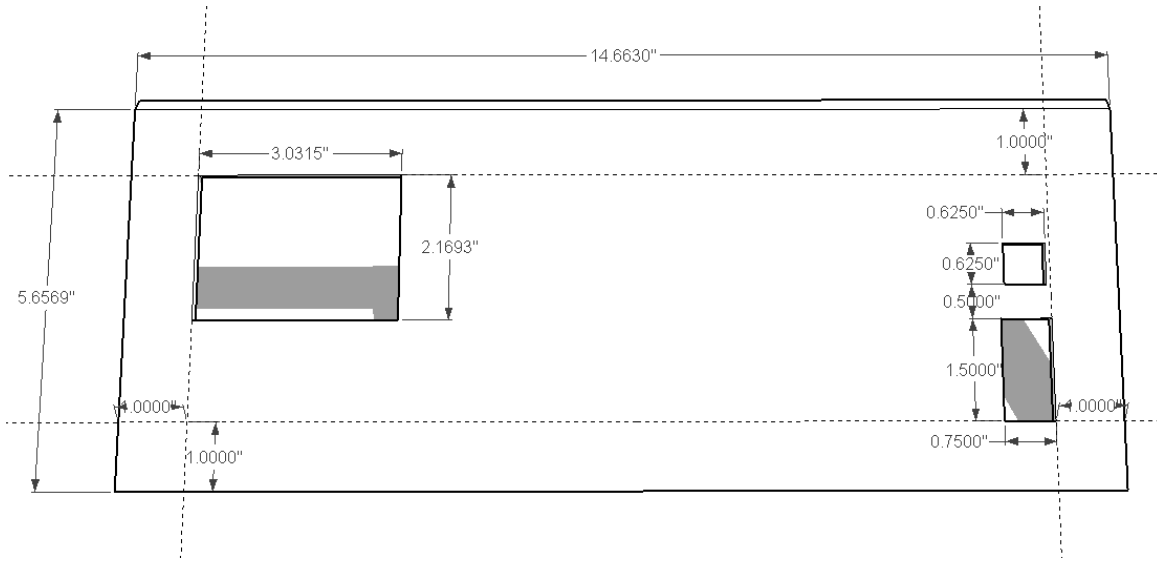


Fig. 8. Front view of the front plate.

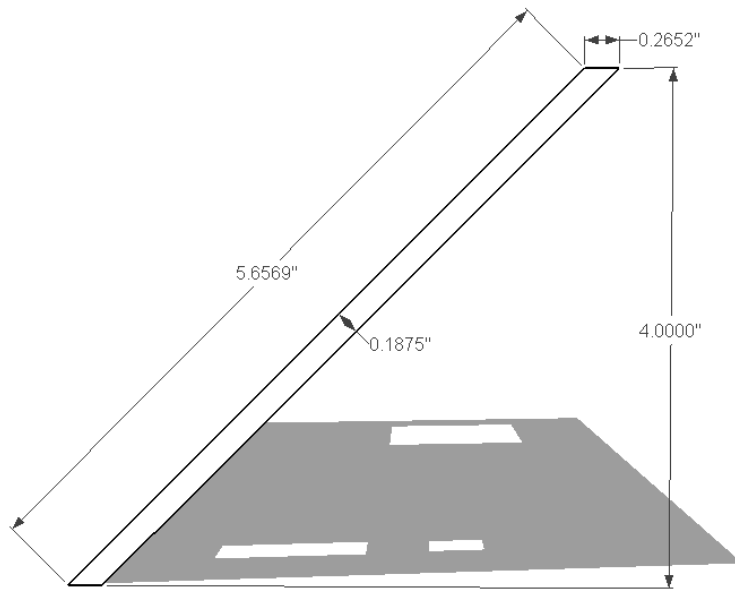


Fig. 9. Side view of the front plate.

3.1.2.2 SIDE TRIANGLES

The side triangles keep the front plate components safe from side damage.

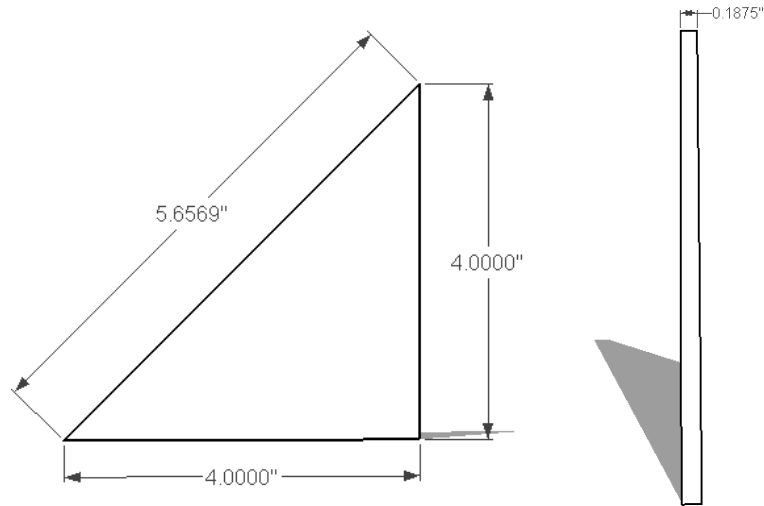


Fig. 10. Side and front view of side triangles.

3.1.2.3 BACK AND CENTER SUPPORT PLATES

The back and center support plates hold the MakerSlide in place and serve as the main support pillars for PyroBot.

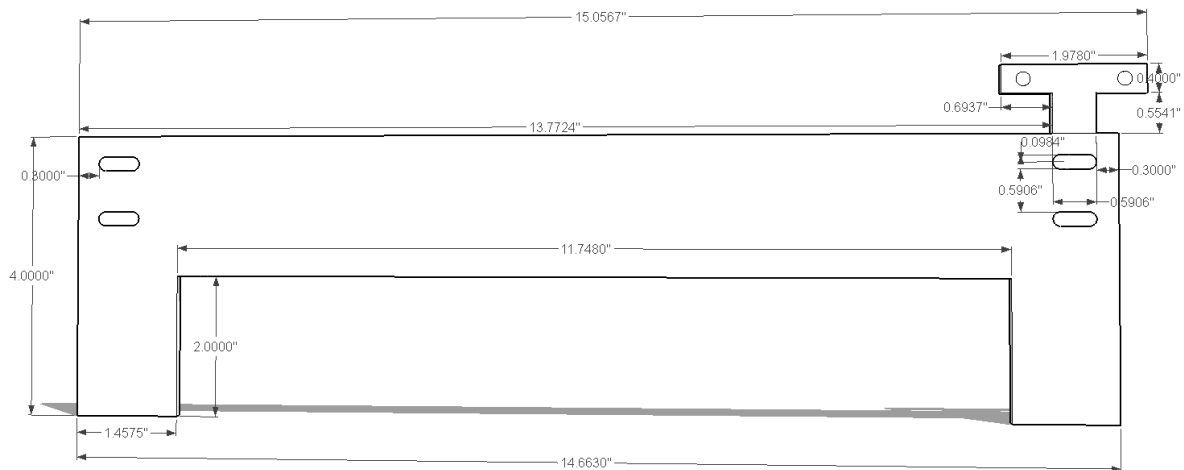


Fig. 11. Front view of the back and center support plate.

3.1.2.4 WOOD SUPPORT PLATE

The wood support plate holds the wood in place and contains the optical sensor for wood detection. Note how the wood support plate is dipped inside of the overall structure. This allows us to use the natural walls built by the MakerSlide to protect people from the lasers reflection.

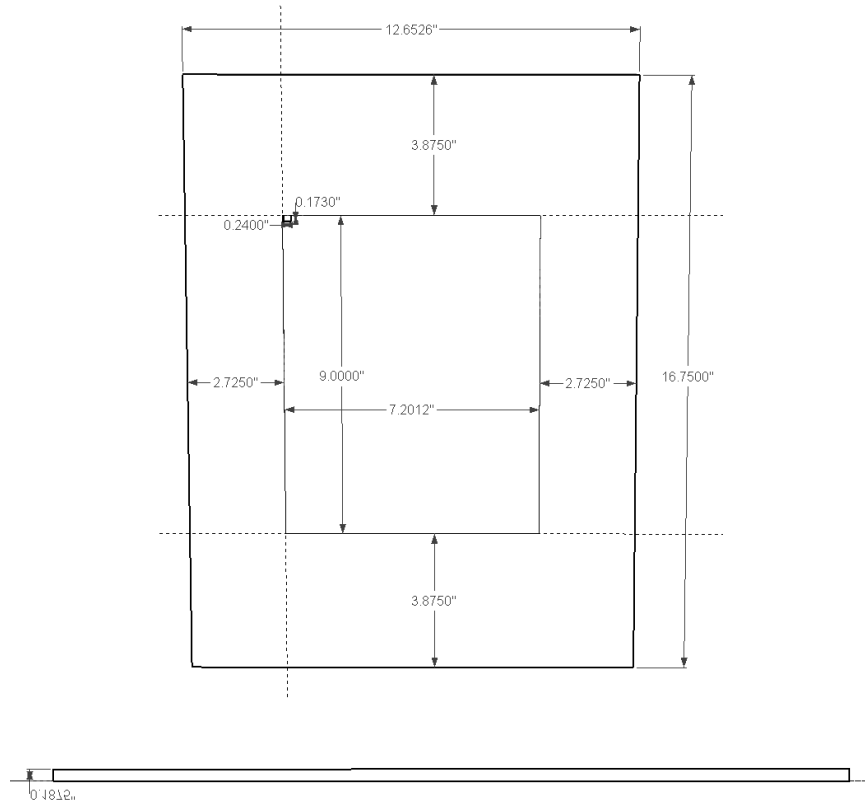


Fig. 12. Top and front view of the wood support plate.

3.1.2.5 SYSTEM FLOOR

The system floor serves as the floor where all the electronics reside on.

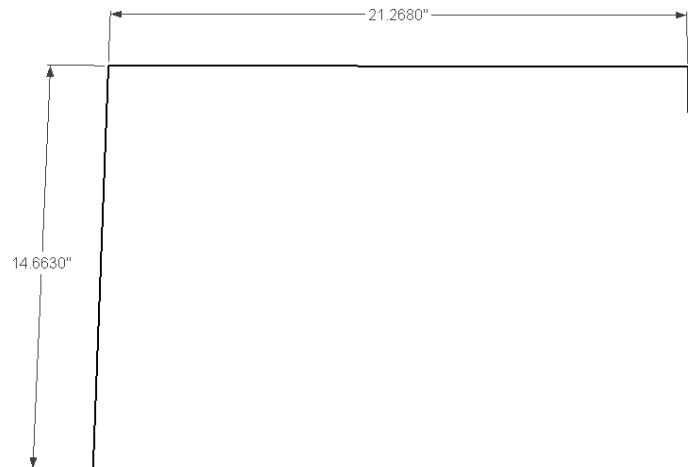


Fig. 13. Top view of system floor.

3.1.3 PARTS

The rest of the PyroBot structure is built from standard hardware parts. During our research for CNC parts we were lucky enough to find a kit that included a lot of the parts we needed. The kit was based on the previously mentioned ShapeOko design. The specific parts in the kit can be found in our part list. We decided to base our system on the ShapeOko because of its simplicity and low cost.

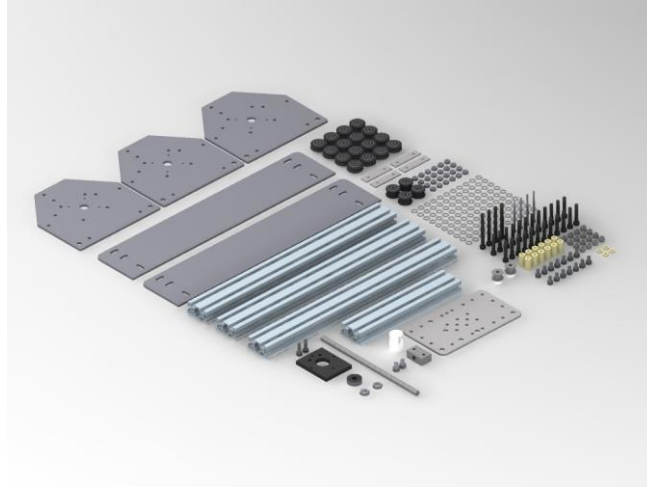


Fig. 14. The contents of the ShapeOko kit. Only some of these parts were used in the construction of PyroBot.

When you combine the custom structure, the ShapeOko kit, and the MakerSlide, you get the final PyroBot design, which was ultimately built.

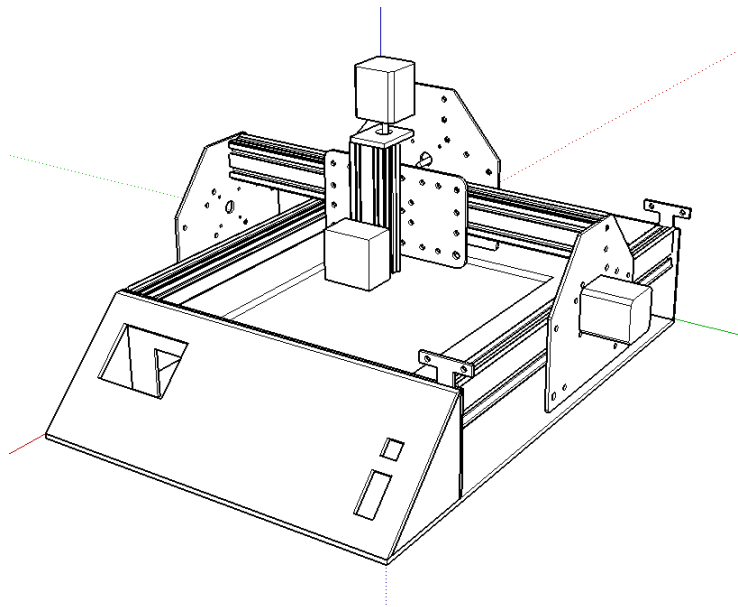


Fig. 15. Final version of the PyroBot design.

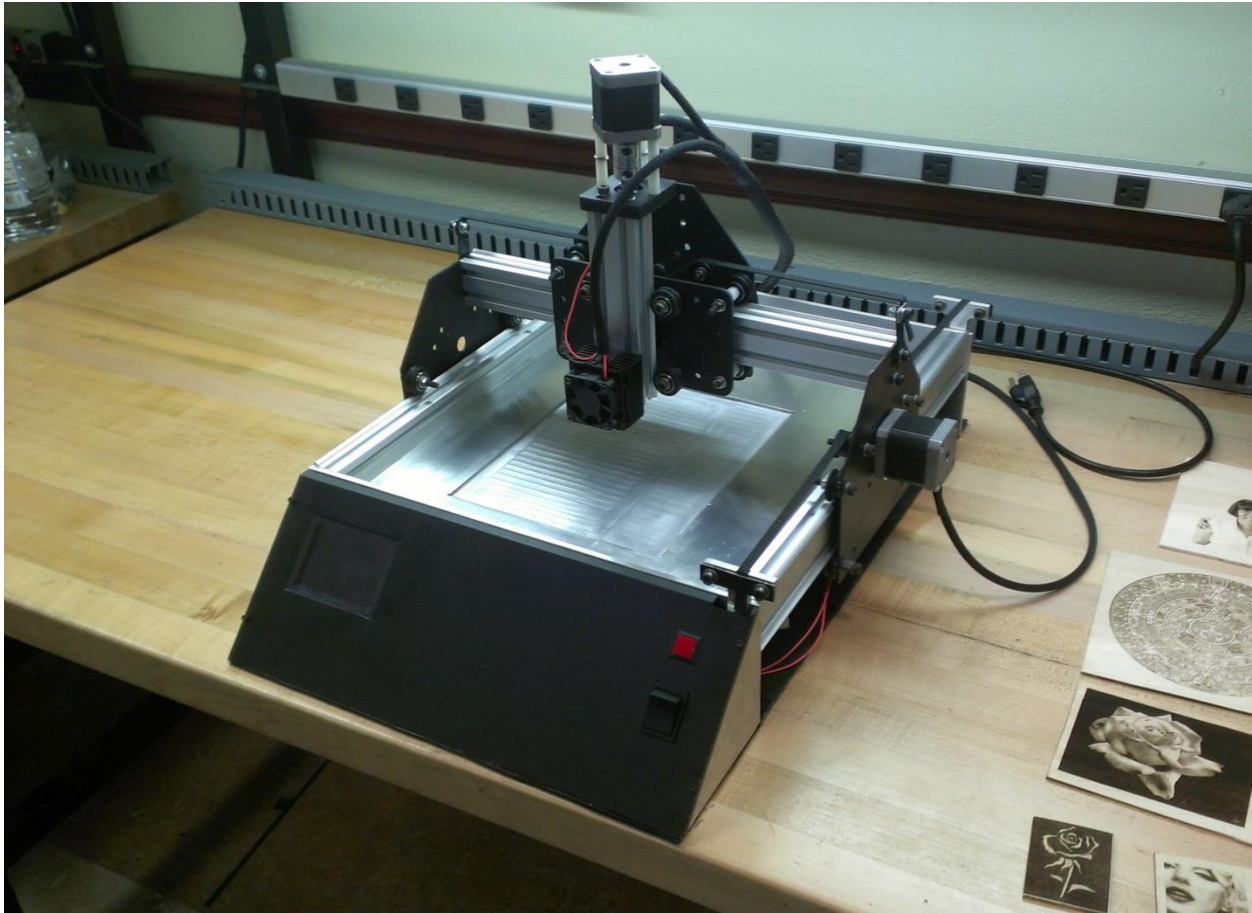


Fig. 16. The version of PyroBot that was built.

3.2 SYSTEM BLOCK DIAGRAM

The overall system block diagram can be seen in figure 18 and gives you a decent overview of the system.

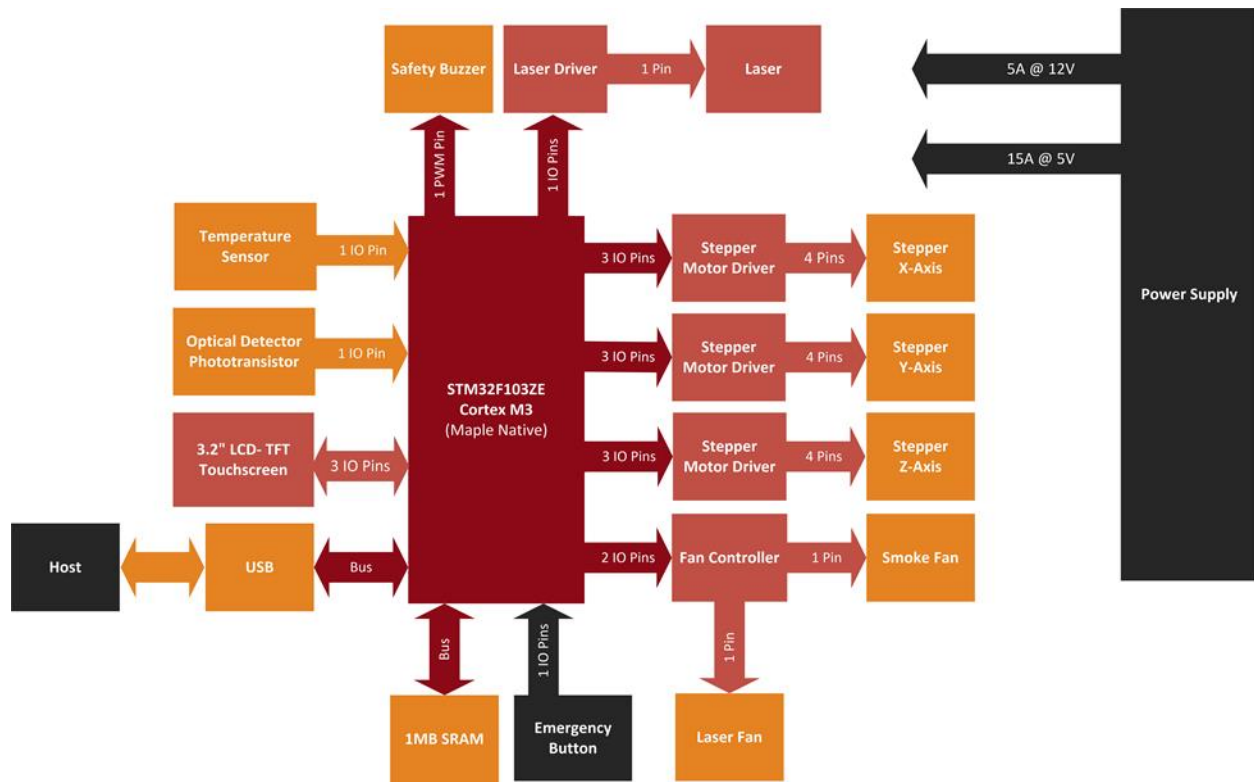


Fig. 17. System block diagram

3.3 ARCHITECTURAL COMPONENTS

The architectural components in the block diagram can be summed up as the following:

- 1 USB- The USB connection interfaces PyroBot with the host.
- 1 3.2" LCD-TFT Touchscreen- Used to accept the upload request sent by the host and start the burning process, shows relevant information to the user.
- 1 Optical Detector- Used to detect if a piece of wood is placed in PyroBot, if it is not placed, the burn process will not start.
- 1 Temperature Sensor- Monitors the temperature of the laser diode, if it rises to 50 degrees Celsius or higher, it sends a signal to the MCU.
- 1 Safety Buzzer- Used to emit an audible signal before the laser turns on.
- 1 Laser Driver- Used to control the current supplied to the laser diode.
- Fan Controller- Used for turning ON/OFF the Smoke fan and Laser's heatsink fan.
- 1 Laser- Used to burn the image on the wood.
- 1 Power Supply- Supplies power to the whole system.
- 1 1MB SRAM- Used to store the image while it's being burned.

- Emergency Power 3 Stepper Motor Driver- Used to control the stepper motors, one for X-axis, one for Y-axis and one for Z-axis
- 3 Stepper Motors- Used to move the laser on the X-axis, Y-axis and Z-axis.

3.4 SOFTWARE OVERVIEW

The software of PyroBot can be divided into host software and system software. On the system side, the user interacts almost exclusively through the onboard LCD touch screen. When the system is initially turned on, a welcome screen is presented to the user, followed by a prompt for him/her to connect the system onto a host computer. After connection, the user can commence the uploading of an image via the host machine. While data transfer is occurring, the system's screen is modified so that the user has feedback on the upload process. On completion of image upload, a menu screen is presented to the user. This would be the main system hub. From here, the user can perform actions such as the burning of the uploaded image or the re-upload of another image. Feedback is provided to the user through the screen throughout the whole process. The screen is also present to provide information to the user in case of error conditions, as is the case of problems with the laser and/or power failure.

Since images are to be transferred through a host computer, a complementary computer application is used. This application would be in charge of communicating with PyroBot for data transmission. It provides an interface for the selection of the desired image in the form of a menu where users can browse their directories in order to find the desired file. Once the file is chosen, the user simply uploads the image and continues the process from the system side. The user is also free to disconnect the host from PyroBot once the image is transferred.

3.5 CORE DESIGN DECISIONS

Throughout the development of the project various core design decisions were made. Some of the design decisions were simple but others were less obvious. This section will cover some of our most important design decisions.

3.5.1 MCU SELECTION

We took many considerations for the selection of our MCU. From a specifications point of view, the MCU had to meet the following requirements:

- At least 61 I/O pins

- 9 pins for the stepper motor drivers
- 1 pin for laser driver
- 1 analog pin for the optical sensor
- 3 pins for the LCD
- 1 pin for the Emergency Stop Button
- 1 pin for the battery manager (relay)
- 45 pins for 1MB of Parallel RAM
- Fairly fast clock speed
- Support for PWM
- At least 4 internal timers
- 1 MB of SRAM (or enough pins to accommodate one)

The secondary factors that were considered were the following:

- Good documentation
- Familiar platform and /or relatively easy learning curve
- Scalability
- Active community

The MCU that was ultimately chosen was the Maple Native.

3.5.1.1 MAPLE NATIVE DEVELOPMENT BOARD - STM32F103ZE MCU

Clock: 72Mhz **Flash:** 512KB **GPIO:** 106 (48 for RAM) **RAM:** 64KB on-chip, 1MB external

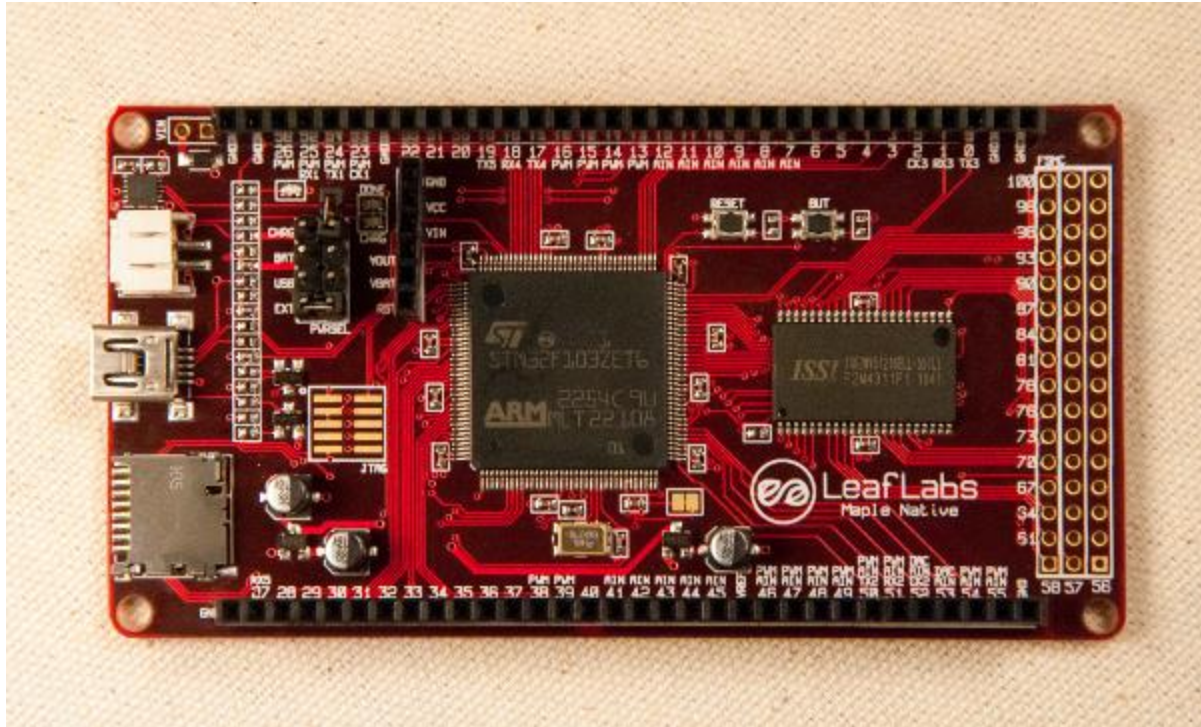


Fig. 18. The maple native.

The Maple Native development board uses an STM32F103 32 bit microcontroller with an ARM architecture. Its flash, RAM, and SD card slot more than accommodated any storage problem we might have encountered. The most notable difference, and the main reason we chose it, to other platforms is the 1MB of parallel SRAM, addressable by virtue of the microcontroller's 32 bit address space. This allows for fast read/write times of images, and, once the image is uploaded, the opportunity to work independently of the host machine.

While we could've selected other, similarly capable platforms for the development of PyroBot, our choice was based on three factors. First, scalability and the benefit of having an SRAM peripheral already onboard; second, with the Maple board being open source, it presented the opportunity of working open source hardware. Finally, the board gives us the advantage of working with the ARM architecture which is rapidly taking over a significant amount of the embedded processor market.

3.5.2 LASER CHOICE

The laser choice proved to be one of the first core and important decisions made for the project. The laser burns or cuts through objects when the object absorbs the energy of the laser. Not all wavelengths are absorbed by a particular object. For example, YAG lasers with wavelengths in the infrared spectrum

(800nm- 2.5 μm) react better with metals. Organic materials like wood, absorb best wavelengths between 400-500nm (blue-violet visible light). After researching laser and engraving techniques, it was determined that a laser capable of emitting between 0.7W to 2W was suited for PyroBot. Lasers with the required wavelength and power output were widely available for purchase.

At the moment of designing the laser driver we also made sure that our simulations allowed for the capability of turning the laser on and off with a signal from the MCU at least at a frequency of 1000hz.

3.5.3 STEPPER CHOICE

Based on structural calculations we needed at least $1.8^\circ/\text{step}$ steppers to properly create the movement we wanted in our system. For this, a 200 step motor, which rotates $1.8^\circ/\text{step}$ would have been enough. However, we wanted to design PyroBot in such a way that an increase of dpi would be easy to implement. For this reason we opted for a much higher resolution of $0.9^\circ/\text{step}$. Interestingly enough, this decision did not raise the overall price of our system. With the inclusion of a micro stepping driver we could design for much higher resolutions in future iterations.

3.5.4 STEPPER TORQUE

Based on empirical data and external expertise we concluded that we needed at least 60oz*in steppers. This was backed up by users of the ShapeOko kit that also recommended the same value. Although our system has much less friction than standard milling CNC's we decided to go with this value for safety.

3.5.5 OPTICAL SENSOR, TEMPERATURE SENSOR, AND BUZZER

In order to prevent the laser from being damaged from overheating, a temperature sensor, along with various fans and heat sinks, were placed beside it. This sensor ensures that the laser is only being used within its operating temperature range. The laser datasheet specifies this range to be within 0 to 50°C . The LM34 was chosen because its operating range can easily accommodate the laser's temperature range.

As a precaution, PyroBot was required to check for the presence of a burning surface on its base. It would be hazardous to allow the system to operate freely, without consent, due to the dangers of specular reflections. To handle this, an optical sensor was added to PyroBot's base plate. The sensor asserts an interrupt whenever a plate is placed on top of it. The system can then keep track of whether or not a surface has been placed on top of it. The QRD1114 was chosen because its form factor allowed for easy placing on PyroBot's base plate.

Another precaution was informing the user and the people nearby of when the laser was in use. This was considered, again, to mitigate the dangers of specular reflections. The 254-35C1-ROX piezoelectric buzzer was chosen because its decibel magnitude was deemed loud enough to be heard within the confines of a regular sized room.

3.5.6 POWER SUPPLY

PyroBot uses a lot of current and is very sensitive to noise. Noise in the circuit has a visual impact on the quality of the prints. We needed a very powerful supply with a sophisticated method of noise filtering. Power supplies have a lot of intricate details for them to work properly with high power and low noise. We could have, of course, spent time designing this. However, there were 3 main reasons why we opted to purchase one instead:

1. Designing a good power supply may have taken too much time.
2. It' was a critical part of our design, so if corners were cut because of time or money, it would have affected the rest of the system in a negative way.
3. This was out of the scope for our project. Time was better spent working on more critical parts of the system, that had to be designed by ourselves.

3.5.7 WHY DID WE USE THE TI DRV8825?

In general, we wanted to use a stepper motor driver, as opposed to a dual H-Bridge, to free up the microcontroller from doing the step sequences, and any error in the step sequence would have been very problematic. Also, this results in less parts, less money, and a cleaner schematic. These are the specific features that led us to choose the DRV8825:

3.5.7.1 MAXIMUM CURRENT

Each of our stepper motors are rated at 1.7A/Phase and the DRV8825, which can run at a maximum of 2.5A/Phase, gives us enough headroom for us to feel comfortable with.

3.5.7.2 MICROSTEPPING

It take PyroBot two steps to move one pixel. Without microstepping we could move one step per pixel, but that's about it. Since we want to have the ability to increase PyroBots printing resolution, we need a way to decrease the step size of our motor, and this is why microstepping is relevant to our design.

3.5.7.3 OVER-VOLTAGE, OVER-CURRENT

During the development of any system, problems appear at the worst of times. This feature let us develop with less risk of damaging the drivers, and the rest of our circuit. At one point, it actually prevented us from damaging some of the chips during a testing mistake, saving us time and money.

3.5.8 SCHMITT TRIGGERS

We used Schmitt Triggers in two parts of the system, in the laser and stepper drivers. There were two reasons for this. First, in order for PyroBot to print all uploaded images with the desired fidelity, it needed to print them accurately. Even with bypass capacitors in place, all the external noise was still a noticeable problem within the system; Schmitt Triggers mitigate this by helping to filter out this noise.

Also, whenever the system started up, the microcontroller pins had undefined voltage levels until it finished executing the setup code. These values could lead to unexpected behavior in both the motors and the laser, neither of which were safe.

Therefore Schmitt triggers were used to keep these voltages at a defined state (off) during this initialization state.

3.6 TI CHIPS USED

Although we used various other TI chips in our project, we will focus on the following three, since they had the most profound impact.

LM34CZ – As mentioned previously, the system needed a method to ensure that the laser's temperature stayed within its operating limits. The LM34CZ temperature sensor's specified operating range was large enough to fit our needs. Also, its linear voltage response to temperature was ideal since it meant that the tracking of temperature could be done with minimal overhead. These reasons were the deciding factor in choosing the LM34CZ for our system.

DRV8825 – We used the DRV8825 Stepper Motor Driver. Incorporating this IC meant that we could delegate all stepper timings, sequencing and driving to the DRV8825. This in turn frees up our MCU to carry out other relevant tasks. Our stepper motors required a biasing of 3V at 1.7A. The DRV8825 is capable of outputting a max current of 2.5 from an 8 to 45V input voltage. As a bonus, the IC comes with over-voltage and over-current protection circuitry, along with the choice of up to 32 microsteps per step. The DRV8825 then seemed to be the perfect chip to fit our needs.

LM338 - The LM338 regulator controls the amount of voltage across the laser and a 1 ohm resistor connected in series with it. As part of the calibration phase, we needed a method of controlling the current supplied to the laser. Since the forward voltage of the laser is approximately constant, controlling the voltage on the other end of the resistor meant controlling the current going through the network. This implied that we needed an adjustable voltage source capable of supplying the 1.6A sinked by the laser. The LM338 is capable of this and more, being able to supply an adjustable voltage ranging from 1.2 to 32V at a max current in excess of 5 A. These specifications make the LM338 an ideal candidate for use within PyroBot.

4 HARDWARE DESIGN

The following section contains explanations on the implementation of all major hardware components. These include the laser, the stepper motors, all needed drivers, the LCD screen and the temperature and optical sensors. They cover background information discussing each component's basic functionality. Following that, we go through the driving analysis; here we verify port compatibility and, when necessary, detail the calculations behind the interfacing of peripherals. The power analysis section breaks down the system's power dissipation and verifies power supply compatibility. The port loading and timing analysis sections details the overall I/O current draw and all system time bases, respectively. The next two topics finish off the section. The first presents a system memory map graphing the overall MCU memory usage, while the second covers the professional considerations taken into account while designing the system's hardware.

4.1 LASER AND LASER DRIVER

After evaluating the different options for burning wood in a controllable, precise and relatively fast way, a laser was selected as the burning element of our system. Some advantages of using a laser are its precision (0.2mm beam), it does not make contact with the wood and it requires less maintenance than for example, iron tips. To select the appropriate laser, wavelength and power output had to be considered.

The Nichia 2.5 NDB7875 [7] laser diode was chosen. Its wavelength is 445nm and it has a maximum rated power output of 1.6W. The diode is enclosed in a case that has an adjustable lens attached. The lens is used to adjust the distance between the laser beam and the optimal focus point, where the beam is focused and has maximum effect on the wood. This is the focal point. It is very important to adjust the focal point to be on the surface of the wood that will be burned; otherwise, the burning time will increase. The focal point can be adjusted by turning the lens clockwise to increase the distance between the lens and the focal point, or counter clockwise to decrease the distance between the lens and the focal point. The main reason that PyroBot has an adjustable Z-axis is to be able to make small adjustments to the distance between the wood and the laser, thus keeping the laser at an optimal burning distance. Figure 19 illustrates the concept of the focal point.

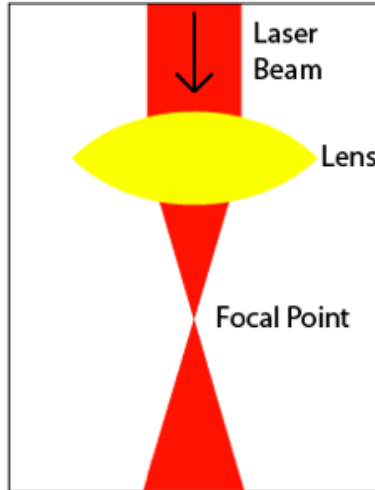


Fig. 19. Laser's Focal Point

A laser diode is a very power hungry device. A current limiting driving circuit is required to prevent the laser diode from absorbing more current than it can handle. For an in-depth explanation of the laser driving circuit, refer to section 4.7.1.

4.1.1 LASER DRIVER DESIGN DESIGN CONSIDERATIONS

The laser is a high power component that requires driving and the ability of being turned ON and OFF by the MCU at frequencies up to 1 kHz. The laser driver design was based on a design for laser controllability using TTL [8]. When V_{in} is low the laser is OFF, when V_{in} is high the laser is ON. To supply more current than the original design, the regulator was swapped for a LM338 [9] capable of supplying up to 5A. The current is never greater than 1.6A, since that is the maximum rated current of the laser. The driver circuit uses two operational amplifiers [10], with the output of the first operational amplifier connected to a potentiometer between it and the inverting input of the second operational amplifier. The potentiometer is used to set the laser current. By varying its resistance value we can lower or increase the current that is sourced to the laser. The second operational amplifier inputs are connected to the potentiometer and to the laser positive terminal, so it takes into account variations of the laser voltage and compensates for them. Its output is connected to the adjust terminal of the regulator. The higher the voltage in the regulator's adjust input, the higher the current it will supply. Figure 20 shows the complete laser circuit.

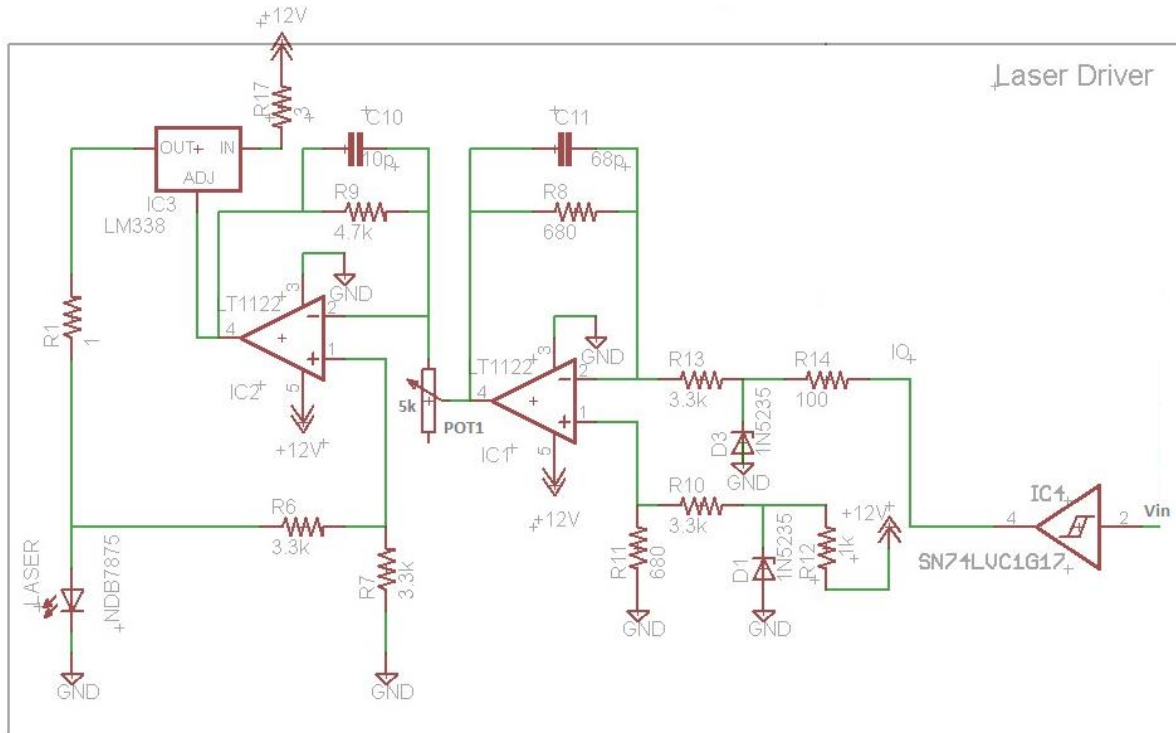


Fig. 20. Laser Driver

To simulate the circuit, a test load had to be used to avoid damaging it with higher currents than what it can handle. The test load used was five 1N4001 diodes connected in series. These simulate the voltage drop of the laser and were utterly helpful in the testing and calibration process of the circuit. The circuit was tested at different frequencies, and it proved reliable to up to 10kHz, 10 times faster switching than what the system uses. Figure 21 shows the simulation output when the input voltage varies from high to low at 10kHz.

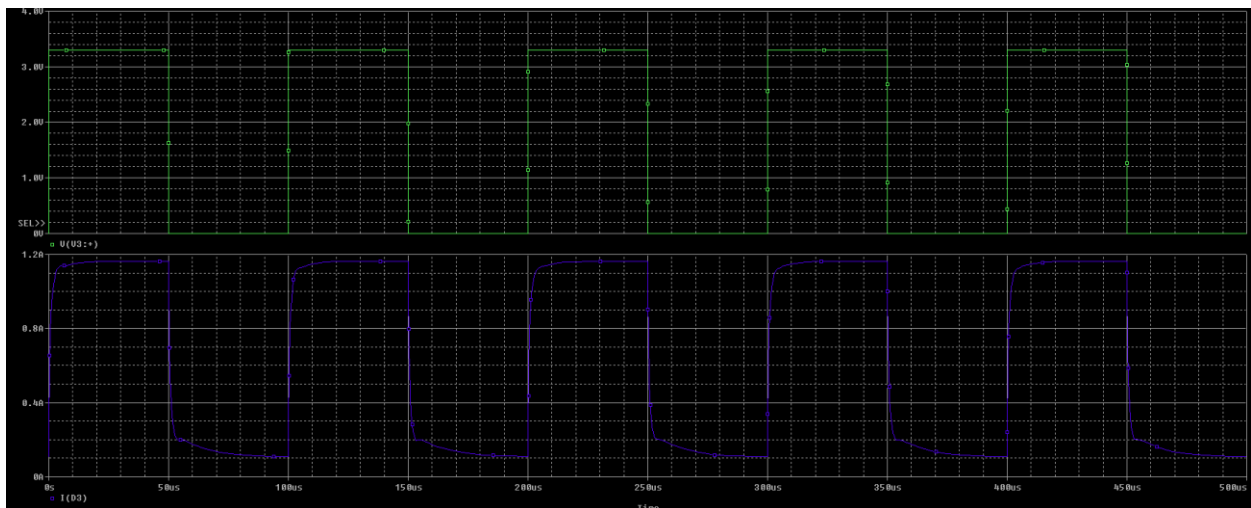


Fig. 21. Test run of the laser driver at a frequency of 10kHz, V_{in} = green line, Laser current = purple line

While testing the system, it was noticed that while the MCU was booting up or resetting, the laser turned ON uncontrollably. When the MCU is resetting or turning ON, its I/O pins are on a high impedance state, so V_{in} it's neither low nor high. If V_{in} is not low, the laser can turn ON. The measured voltage at the input was about $V_{in} \sim 1.2V$, enough to turn the laser at partial power. To solve this design problem, a Schmitt Trigger [11] was connected to the input where the I/O pin from the MCU would normally be connected. See Figure Fig. 20. If the input voltage of the Schmitt Trigger is below a threshold, it connects the output to ground, else it connects it to $V_{CC} = 5V$.

4.1.2 REGULATOR POWER AND THERMAL CALCULATIONS:

| Parameter | Value [9] |
|----------------|--------------------------|
| I_{MAX} | 5A |
| $V_{IN (MAX)}$ | 35V |
| P_{MAX} | 50W |
| θ_{ja} | $50 \frac{^{\circ}C}{W}$ |
| θ_{jc} | $4 \frac{^{\circ}C}{W}$ |
| T_{MAX} | $125^{\circ}C$ |

Table. 1. LM338 Electrical and Thermal Characteristics

The voltage regulator used has a maximum current rating of 5A. The regulator is adjustable, so the amount of current it supplies is controllable by the driving circuit. However the maximum current is fixed at 1.6A using the potentiometer. To lower the $V_{NR} - V_R$ difference, thus reducing the power dissipation of the regulator, a 3Ω high power resistor capable of dissipating up to $13W$ is connected between the 12V source and the input of the regulator. Another option is to use a DC-DC converter to lower the input voltage of the regulator, thus reducing the power dissipated.

$$I_{Laser(MAX)} = 1.6A$$

$$P_{3\Omega} = 3\Omega(1.6A)^2 = \boxed{7.68W < 13W}$$

$$V_{NR} = 12V - 3\Omega(I_{Laser}) = 12V - 3\Omega(1.6A) = \boxed{7.2V}$$

$$V_R = V_{Laser} + R(I_{Laser}) = 4.5V + 1\Omega(1.6A) = \boxed{6.1V}$$

$$P_{Reg} = (V_{NR} - V_R)I_R$$

$$P_{Reg} = (7.2V - 6.1)1.6A = \boxed{1.76W}$$

$$T_{Reg} = P_{Reg} \times (\theta_{jc} + \theta_{ja}) + TA = 1.6W \left(54 \frac{^{\circ}C}{W} \right) + 27^{\circ}C = \boxed{113.4^{\circ}C}$$

$$\boxed{T_{Reg} < T_{MAX}}$$

Heatsink and/or a fan are not required, however as a precautionary measure, a heatsink [12] was installed to the regulator capable of dissipating up to 7W, lowering the temperature even more below the maximum value. This aids the reliability and durability of the regulator.

4.2 STEPPER AND STEPPER DRIVER DESIGN

Stepper motors, also called “steppers”, are a type of DC motor that divides its full rotation into a fixed number of steps, providing precise positioning and speed control without the use of feedback. It has a series of small “teeth” around a gear shaped electromagnet that, when applied current, moves the motor to a fixed position and actively keeps it there.

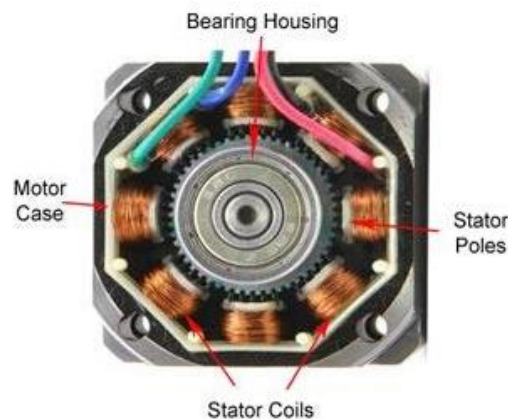


Fig. 22. Inside a Stepper Motor [13]

4.2.1 TYPE

PyroBot uses three **bipolar** steppers, which have a single winding per phase. In order to perform a full step, the winding current needs to be reversed, requiring more complicated driving techniques than unipolar steppers. Regardless, this is sometimes preferred because of the lower pin count, as modern microcontrollers usually have more than enough processing power to handle it, and there are even dedicated IC's that do the job of driving them for you, such as the DRV8825.

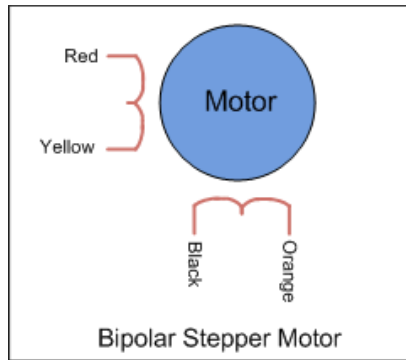


Fig. 23. Bipolar Stepper Motor with Two Motor Coils [14]

4.2.2 STEPPING

Each step causes the shaft to rotate a certain number of degrees. A **step angle** represents the rotation in degrees of the output shaft during each step.

In order to make our bipolar stepper do one full step, it needs to be driven in a very particular way, explained in the “Stepper Drivers” section.

4.2.3 MICROSTEPPING

A driving circuit that makes the motor current of each individual phase increase and decrease like a sine wave and is always kept out of time with the other phase will allow the rotor to reach hundreds of intermediate steps. In fact it is possible for the controller to reach as many as 500 micro steps for a full-step sequence, which will provide 200,000 steps for each revolution to our 400 step/revolution stepper. However this adds greater inaccuracy between steps and requires more complicated driving. [15]

In our project, micro stepping is not currently used but is possible with our current setup in case we want to increase the resolution up to 32 times of what it currently is. [15]

For our project we chose a 3V, 400 step/revolution Bipolar Stepper Motor that runs at 1.7A/Phase. We chose 400 steps because every 1.8° of rotation is the recommended angle to use with the belt system we are using. Since we have 0.9° , this gives us the ability to duplicate the resolution if needed. These are the steppers we will use: [14]



Fig. 24. SparkFun Stepper Motor [14]

This stepper has a holding torque of 48N.cm when stationary. The recommended holding torque in our system is a little over this value, but it's designed for a CNC mill. Since we only use a laser, something that will never have any physical force against it, only its own weight, it will be more than enough.

4.2.4 STEPPER DRIVERS

Stepper motors need a very specific set of instructions to properly operate, usually managed by a microcontroller. The amount of current a microcontroller is able to source is usually much less than a stepper motor needs to properly function so a driver circuit is built to provide the current while the microcontroller provides the logic. The most basic way to properly run a stepper motor is with a dual h-bridge circuit. [16]

4.2.5 DRIVING STEPPERS USING H-BRIDGES

Driving a stepper motor is functionally the same as driving two brushless DC motors. Here we have each coil of a bipolar stepper motor connected an h-bridge:

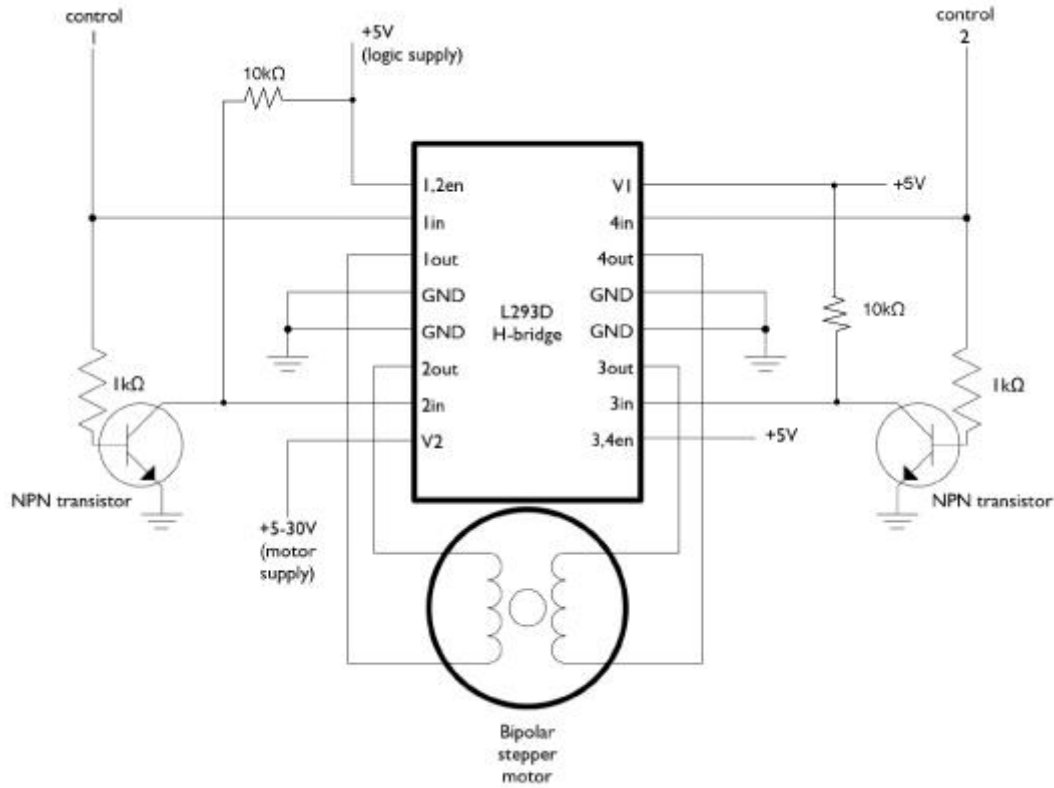


Fig. 25. Stepper Motor Connected to two H-Bridges [17]

Applying the knowledge of motor stepping and running h-bridges one can arrive to a way of making a motor move with two h-bridges. The table of values to achieve one step with a stepper motor is as follows: [17]

| Step | Wire 1 | Wire 2 | Wire 3 | Wire 4 |
|------|--------|--------|--------|--------|
| 1 | High | Low | High | Low |
| 2 | Low | High | High | Low |
| 3 | Low | High | Low | High |
| 4 | High | Low | Low | High |

Table. 2. Values to achieve one step with a stepper motor.

For PyroBot, there are many time sensitive software components, so handling the stepping with the microcontroller would have been a poor design decision. Therefore, we decided to use a dedicated stepper motor controller. We needed the driver to handle at *least* 1.7A/Phase at 12V (it's the only power rail that could handle more components).

For this we chose the DRV8825 from Texas Instruments. It can run a bipolar stepper motor at 2.5A/Phase and run from 8V to 45V, which is more than enough for our motors.



Fig. 26. Texas Instruments DRV8825 in HTSSOP28 Package [18]

The DRV8825 has Over-Current, Under-Voltage, and Over-Temperature protection. All extremely valuable features that have helped during development and can prevent future issues. The driver is essentially two H-Bridges that generate the correct current flow on the coils on every pulse of its "STEP" pin based on the internal indexer that has the values needed to generate steps.

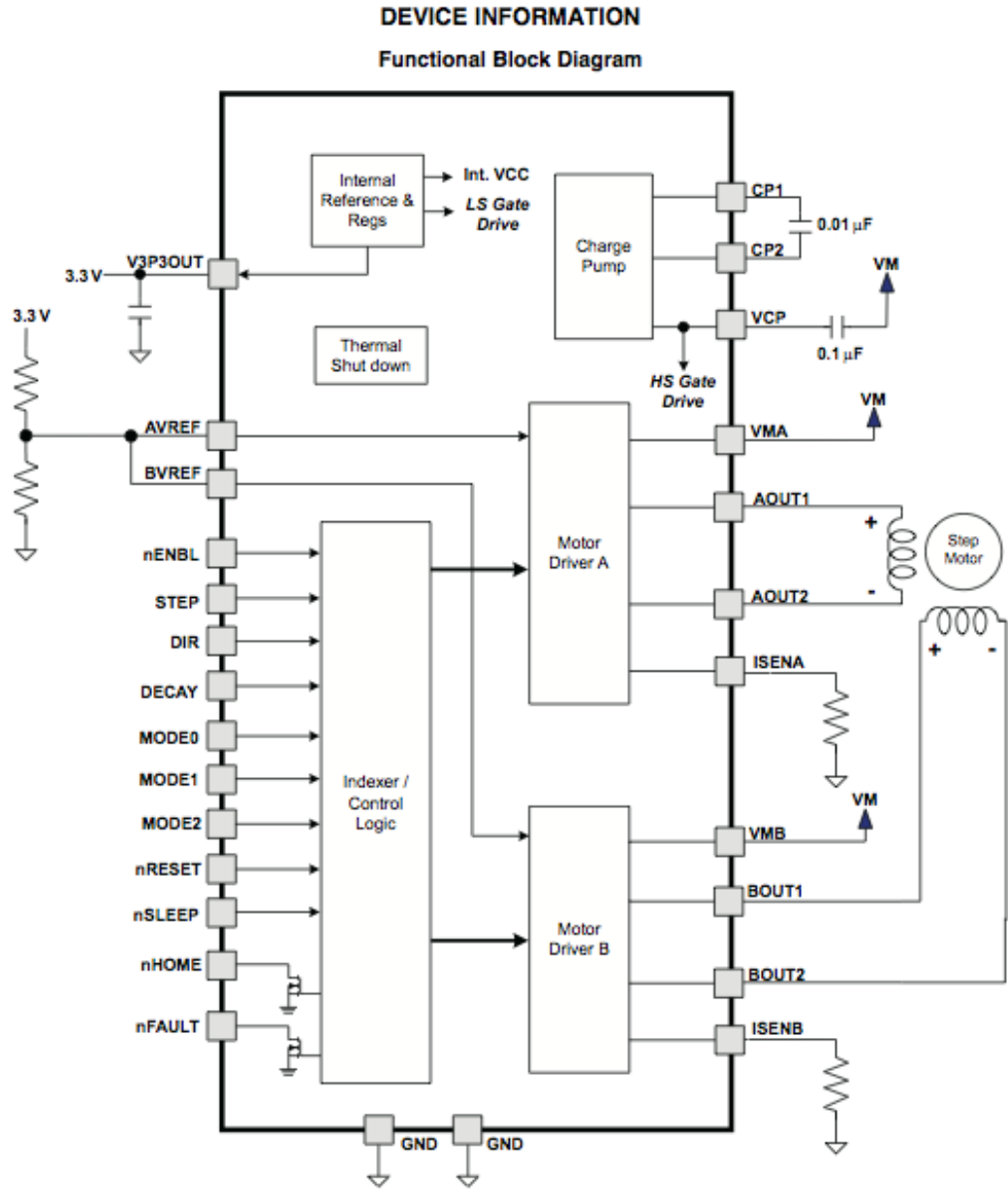


Fig. 27. DRV8825 Block Diagram [18]

The DRV8825 also has an external “FAULT” pin that indicates when it’s in any of its fault conditions.

The current going through the motor coils is controlled by the VREF pins and the ISENX resistors. These values can be calculated using the following formula:

$$I_{CHOP} = \frac{V_{REFX}}{5 \cdot R_{ISENSE}}$$

An important note is that the current sense resistors are in series with the motor coils, so all the motor current will pass through those resistors. That means that those resistors will need to be able to dissipate a fair amount of power.

The current that needs to go through each of our motor coils is 1.7A so, using the formula:

$$1.7 = \frac{V_{ref}}{5 * R_{ISENSE}}$$

There is a potentiometer in V_{ref} to finely calibrate the reference voltage, and we used 300mΩ on each ISENSE pin. With these resistor values we would have to dial the voltage divider in V_{ref} to 2.55 volts to have the correct current flowing through the coils. So now that we know the current sense resistor values we can calculate how much power they need to dissipate: [18]

$$P = I^2R = 1.7^2A * 300m\Omega = .867W$$

For this reason, the sense resistors need to be rated for at least 1W.

To properly design the DRV8825 we had to design our own PCB to interface with it. We based ourselves on the guidelines mentioned for the PCB design and the schematic on TI's Evaluation Module. This is the schematic with all of its components:

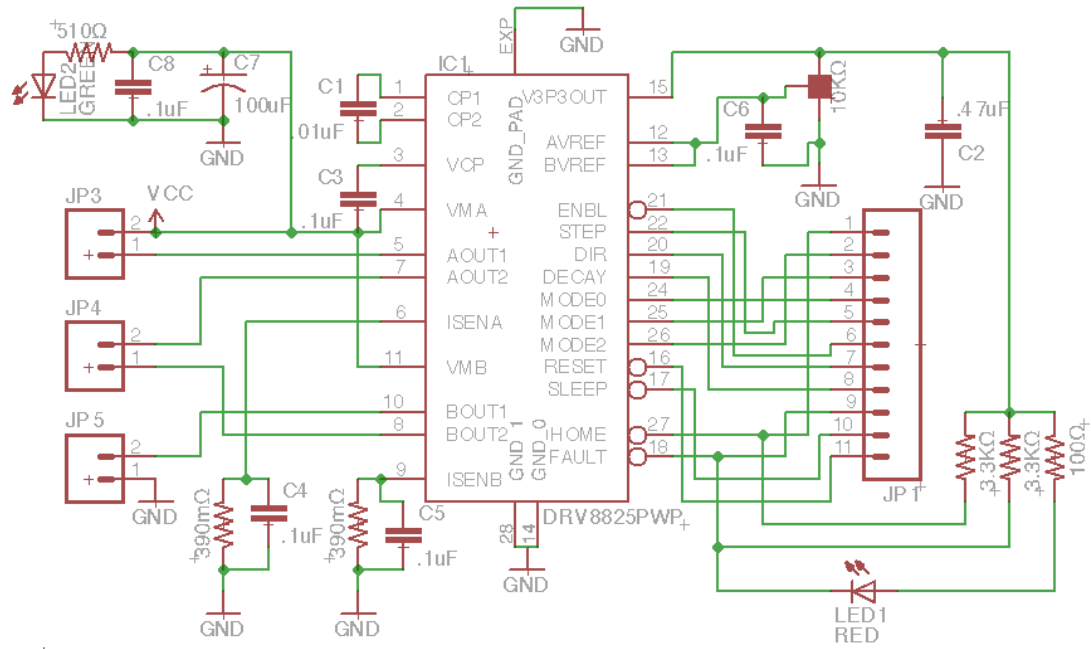


Fig. 28. PyroBot's DRV8825 Board Schematic

Here is what the final PCB design looked like:

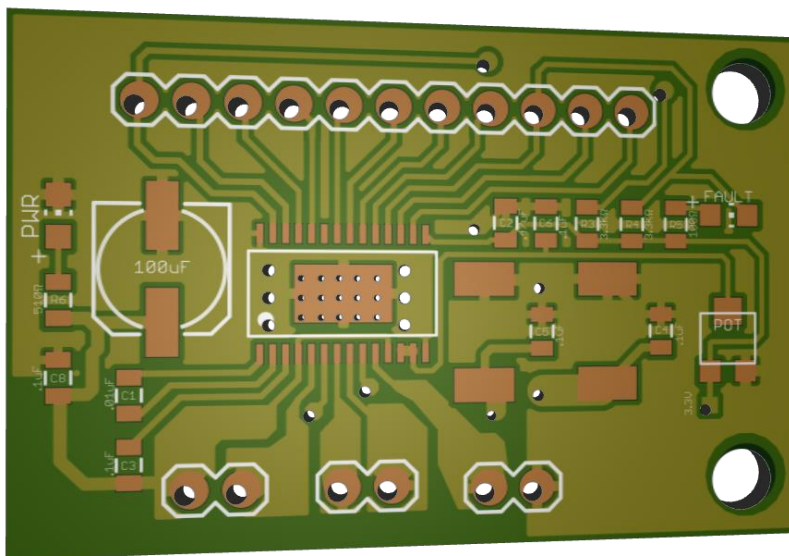


Fig. 29. Top View

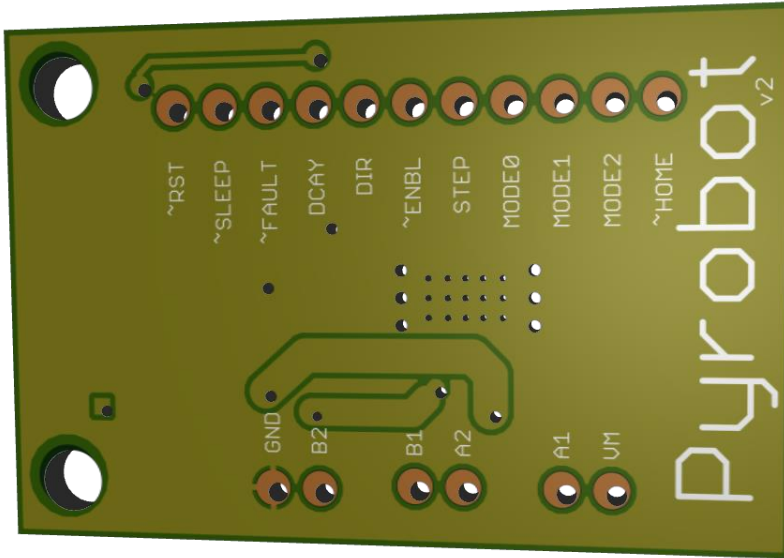


Fig. 30. Bottom View

Note: Misaligned drill holes are an error of the 3D rendering.

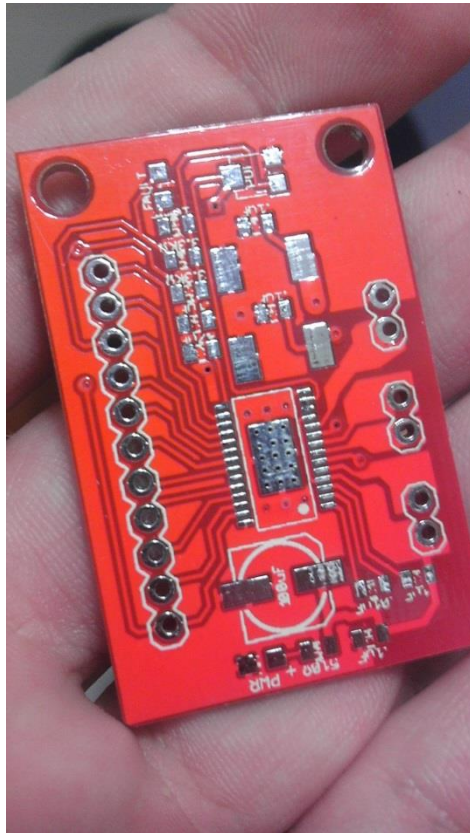


Fig. 31. PCB fully created.

4.2.6 POWERPAD

The DRV8825 comes in a HTSSOP28 package that has what Texas Instruments calls a “PowerPad” under the IC. [19]

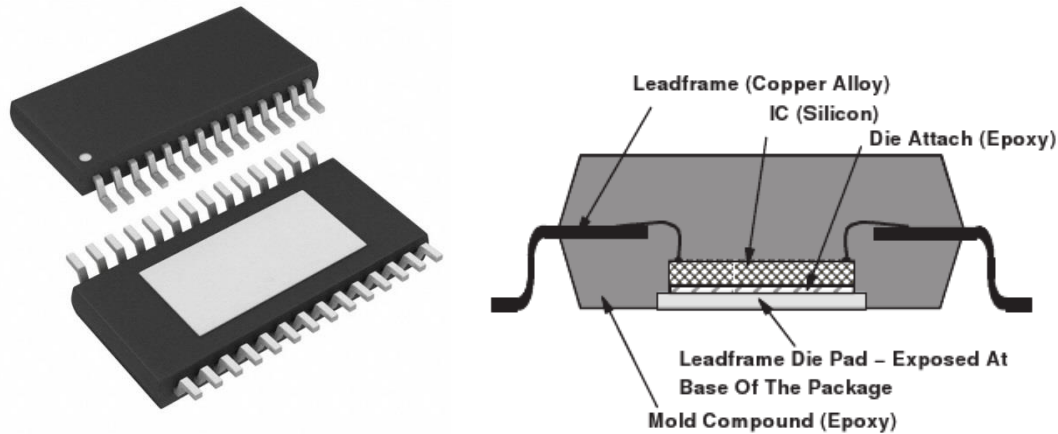


Fig. 32. HTSSOP28 “PowerPad” Overview [19]

For the PowerPad to effectively dissipate heat it needs to be properly soldered onto the copper plane on the PCB. Furthermore, the copper pad on the PCB where the PowerPad will be soldered onto needs to be designed according to the PowerPad this specification. It needs to have proper thermal vias unto a large copper plane and at least 50% of the pad needs to be soldered. In our case, heatsinks and fans will also be used to further dissipate heat. [19]

4.2.7 PROTECTION CIRCUITS

4.2.7.1 OVERCURRENT PROTECTION (OCP)

An analog current limit circuit on each FET limits the current through the FET by removing the gate drive. If this analog current limit persists for longer than the OCP time, all FETs in the H-bridge will be disabled and the nFAULT pin will be driven low (and the red LED should turn on). The device will remain disabled until either nRESET pin is applied, or VM is removed and re-applied.

Overcurrent conditions on both high and low side devices; i.e., a short to ground, supply, or across the motor winding will all result in an overcurrent shutdown. Note that overcurrent protection does not use the current sense circuitry used for PWM current control, and is independent of the ISENSE resistor value

or VREF voltage.

4.2.7.2 THERMAL SHUTDOWN (TSD)

If the die temperature exceeds safe limits, all FETs in the H-bridge will be disabled and the nFAULT pin will be driven low. Once the die temperature has fallen to a safe level operation will automatically resume.

4.2.7.3 UNDERVOLTAGE LOCKOUT (UVLO)

If at any time the voltage on the VM pins falls below the undervoltage lockout threshold voltage, all circuitry in the device will be disabled and internal logic will be reset. Operation will resume when VM rises above the UVLO threshold.

4.2.7.4 DECOUPLING CAPACITORS

The capacitors that don't appear in the datasheets guideline were added based on TI's DRV8825 Evaluation Module's schematic diagram. [20]

4.2.7.5 INDICATOR LED'S

Indicator LED's for power (green) and for the integrated fault condition (red) were added for further safety and usability. [20]

4.3 TEMPERATURE AND OPTICAL SENSOR

In order to ensure the laser operates within its rated temperature values, an LM34CZ temperature sensor was incorporated into the system. This sensor is a transducer capable of outputting a linear voltage response with respect to temperature. It's rated to output 10mV per °F temperature increase. The temperature sensor was mounted on top of the laser in order to ensure accurate readings. Whenever the laser's temperature exceeds its operating range, an interrupt is fired to prevent the system from reusing the laser until it cools off.

A modified version of the LM34CZ temperature controller provided in the transducer's datasheet was used to drive the sensor. The circuit, presented next, uses a positive feedback loop to add hysteresis to the controller.

$$\frac{R_1}{R_3} = 0.07965 \quad (6)$$

From equations (5) and (6), we find $R_3 = 840.33 \text{ k}\Omega$. So choose $R_3 = 1 \text{ M}\Omega$. For R_1 , use a $100 \text{ k}\Omega$ potentiometer and calibrate its value until you obtain the required voltages. This resistance values was found to be around $62.3 \text{ k}\Omega$ experimentally.

The optical sensor is another type of transducer capable of converting the light irradiated on its surface into a voltage signal. Here, it is used to detect whether or not a plate was place on the system's base. Its diagram is shown next.

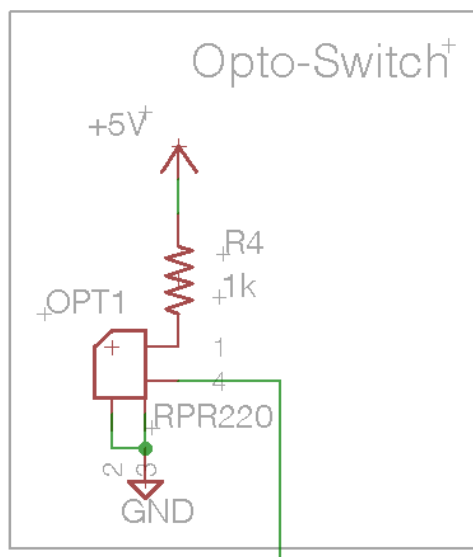


Fig. 34. Optical Sensor Schematic.

4.4 LCD / TOUCHSCREEN

Maintaining and updating an LCD can be difficult for an MCU if an LCD Controller is not present. For this reason PyroBot utilizes the μ LCD-32PT smart display module. The μ LCD includes its own graphic, touch, and usd memory controller named the PICASO-GFX2. The PICASO-GFX2 can be programmed as a stand-alone graphics controller or as a slave graphics controller. [21]

When the μ LCD-32PT is in slave mode all communication and events are initiated by the host. This is done by sending a series of bytes to the display and waiting for a response. The response command lets the host MCU know whether or not the command was executed correctly. If the command succeeded an ACK (06 hex) is sent back. If the command failed a NAK (15 hex) is sent back.

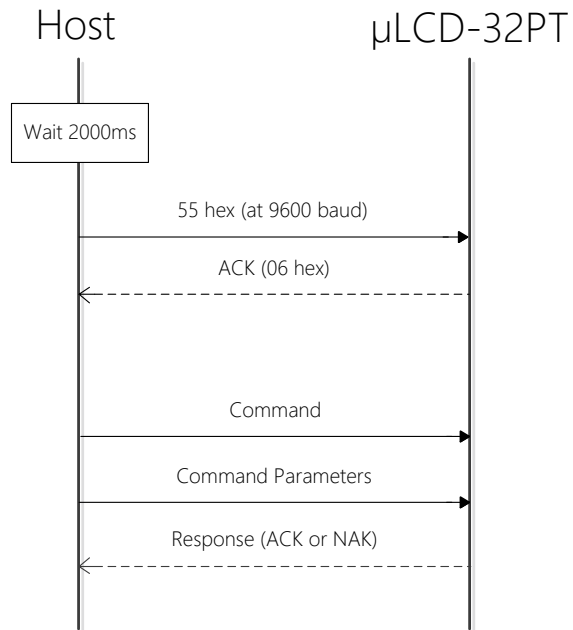


Fig. 35. A typical sequence of events for a command execution after a cold start.

When the display is first turned on it requires at least 500ms to set up before it is ready to receive commands. If a usd card is present it requires more time to set up while it initializes the card. In our case we found 2000ms to be sufficient.

If in 5000ms from the moment the display boots up a 55hex command is not received, the screen goes into a default splash screen.

4.4.1 TOUCH RESPONSE

Touch screen commands work different than normal display commands. All touch commands block the μLCD from taking any further action till a touch event happens. Once a touch event occurs the response is the X and Y coordinates of said event.

4.5 DRIVING ANALYSIS

The following sections describe the driving analysis we carried out to insure all components could be driven properly by our MCU.

4.5.1 GPIO SUPPLY CURRENT COMPATIBILITY

The maximum current that an I/O port of the Maple Native can supply is 25mA and the maximum voltage it supplies is 3.3V. Every component connected to the MCU, must draw less than 25mA from that port, otherwise an interface is needed to supply the required current to the component. Most of the components have dedicated power lines which are connected directly to the power supply, so even though they require current levels higher than those provided by the I/O ports, they do not draw the current from the MCU.

| Part | Required Supply Current | Interface needed? |
|---------------------------------------|-------------------------|-------------------|
| NDB7875 Laser Diode [7] | 1.6A | Yes |
| 254-35C1-ROX Buzzer [22] | 8mA | Yes ¹ |
| RPR-220 Optical Sensor [23] | 0.8mA | No |
| DRV8825 Stepper Motor Controller [24] | 100 μ A | No |
| ULCD-32PT LCD-TFT Screen [25] | 150mA | No ² |
| MB40100V2 Laser's Heatsink Fan [26] | 180mA | Yes |
| MC20100V2 Laser's Smoke Fan [27] | 156mA | Yes |
| G3VM-352C Relay [28] | 7.5mA | No |
| LM34CZ Temperature Sensor [29] | 10mA | No |

Table 3. GPIO Current Compatibility List
1: Requires 5V and has no dedicated Vcc input.
2: Has dedicated Vcc line to draw current from.

4.5.2 VOLTAGE LOGIC LEVELS COMPATIBILITY

Another consideration that has to be taken into account when interfacing components to a MCU is the logic voltage levels to ensure that the lowest high value that a component can send to the MCU is higher than $V_{IH(MCU)}$, and the highest low value than a component sends to the MCU is lower than $V_{IL(MCU)}$. A similar approach applies when verifying the V_{OL} and V_{OH} of the MCU and the components. Table 4 shows the result of this analysis. [30]

| Parameter | Formula (If applicable) [30] | Range of Values [30] |
|--------------|--|--------------------------|
| V_{DD} | -- | 3.3V |
| V_{OL} | -- | 0.4V |
| V_{OH} | $V_{DD} - 0.4V$ | 2.9V |
| V_{IL} | $0.28 \times (V_{DD} - 2V) + 0.8V$ | 1.16V |
| $V_{IL(FT)}$ | $0.32 \times (V_{DD} - 2V) + 0.75V$ | 1.16V |
| V_{IH} | $0.41 \times (V_{DD} - 2V) + 1.3V \rightarrow V_{DD} + 0.3V$ | 1.83V \rightarrow 3.6V |
| $V_{IH(FT)}$ | $0.42 \times (V_{DD} - 2V) + 1V \rightarrow 5.5V$ | 1.55V \rightarrow 5.5V |

Table 4. Maple Native's Logic Voltage Levels

| Part | V_{IL} | V_{IH} | V_{OL} | V_{OH} | Interface Needed? |
|---------------------------------------|----------|----------------------------|----------|----------|-------------------|
| RPR-220 Optical Sensor [23] | -- | -- | 0.4V | 3.5V | No |
| DRV8825 Stepper Motor Controller [24] | 0.6V | 2.2V \rightarrow 5.5V | 0.5V | 3.3V | No |
| ULCD-32PT LCD-TFT Screen [25] | 0.66V | 2.64V | 0.4V | 2.4V | No |
| LM34CZ Temperature Sensor [29] | -- | -- | 0.4V | 5V | No (FT) |

Table 5. Voltage Compatibility, FT= Five Volt Tolerant GPIO

4.5.3 LASER DRIVER CALCULATIONS:

| Parameter | Value [9] |
|----------------|--------------------------|
| I_{MAX} | 5A |
| $V_{IN (MAX)}$ | 35V |
| P_{MAX} | 50W |
| ϕ_{ja} | $50 \frac{^{\circ}C}{W}$ |
| ϕ_{jc} | $4 \frac{^{\circ}C}{W}$ |
| T_{MAX} | 125 °C |

Table. 6. LM338 Electrical and Thermal Characteristics

$$I_{Laser(MAX)} = 1.6A$$

$$P_{3\Omega} = 3\Omega(1.6A)^2 = \boxed{7.68W < 13W}$$

$$V_{NR} = 12V - 3\Omega(I_{Laser}) = 12V - 3\Omega(1.6A) = \boxed{7.2V}$$

$$V_R = V_{Laser} + R(I_{Laser}) = 4.5V + 1\Omega(1.6A) = \boxed{6.1V}$$

$$P_{Reg} = (V_{NR} - V_R)I_R$$

$$P_{Reg} = (7.2V - 6.1)1.6A = \boxed{1.76W}$$

$$T_{Reg} = P_{Reg} \times (\phi_{jc} + \phi_{ja}) + T_A = 1.6W \left(54 \frac{^{\circ}C}{W} \right) + 27^{\circ}C = \boxed{113.4^{\circ}C}$$

$$\boxed{T_{Reg} < T_{MAX}}$$

4.5.4 BUZZER DRIVER CALCULATIONS

Even though the 254-35C1-ROX Buzzer current requirements that an I/O pin can provide (8mA), a driving circuit is needed because its operating voltage is 5V, and the I/O pins can only supply up to 3.3V.

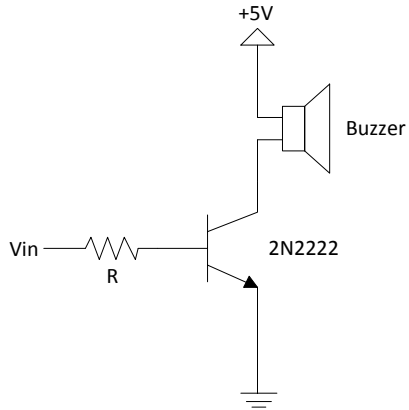


Fig. 36. Buzzer Driver

$$\beta_{Sat} = 100$$

$$I_B = \frac{I_C}{\beta_{Sat}} = \frac{8mA}{100} = \boxed{80\mu A}$$

$$R = \frac{V_{OH} - 0.7V}{I_B} = \frac{(3.3 - 0.4)V - 0.7V}{80\mu A} = \boxed{24k\Omega}$$

4.5.5 LASER'S HEATSINK FAN DRIVER CALCULATIONS

The heatsink fan MB40100V2 requires a driving circuit because it is controlled by the MCU, unlike the MB50100V2 rear fans that are connected directly to the power supply and are always ON.

$$\beta_{Sat} = 100$$

$$P_{Fan} = 0.9 W$$

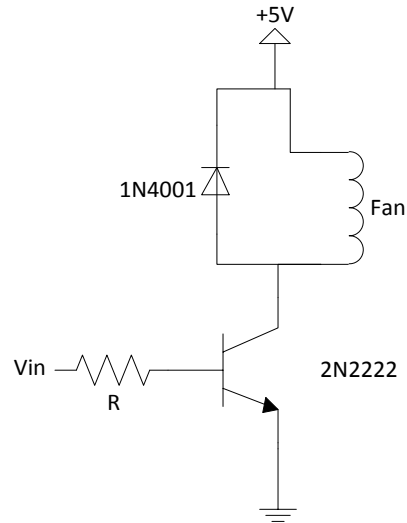


Fig. 37. Laser's Heatsink Fan Driver

$$I_B = \frac{I_C}{\beta_{Sat}} = \frac{180mA}{100} = \boxed{1.8mA}$$

$$R = \frac{V_{OH} - 0.7V}{I_B} = \frac{(3.3 - 0.4)V - 0.7V}{1.8mA} = \boxed{1.22k\Omega}$$

4.5.6 LASER'S SMOKE FAN DRIVER CALCULATIONS

The smoke fan MB20100V2 is used to remove smoke from the wood burning process from the system. It requires a driving circuit because it is controlled by the MCU, unlike the MB50100V2 rear fans that are connected directly to the power supply and are always ON.

$$\beta_{Sat} = 100$$

$$P_{Fan} = 0.78 W$$

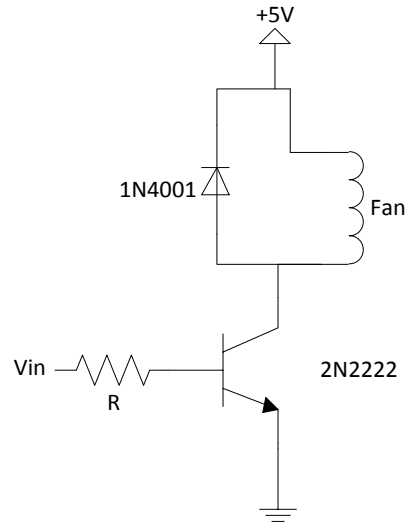


Fig. 38. Laser's Smoke Fan Driver

$$I_B = \frac{I_C}{\beta_{Sat}} = \frac{156mA}{100} = \boxed{1.56mA}$$

$$R = \frac{V_{OH} - 0.7V}{I_B} = \frac{(3.3 - 0.4)V - 0.7V}{1.56mA} = \boxed{1.41k\Omega}$$

4.6 POWER SUPPLY ANALYSIS

The following section discusses the power supply analysis. We chose the XL125-1 as our power supply because it supplied the power we needed

| Rail | Maximum Current | Maximum Power |
|------|-----------------|---------------|
| 5V | 15A | 75W |
| 12V | 5A | 60W |

Table. 7. XL125-1 Power Supply Electrical Characteristics

4.6.1 LASER'S DRIVER

The power supply supplies 1.3A at 12V to the laser, plus 10mA to each operational amplifier [10].

| Parameter | Value |
|-----------|---------------------------------|
| Voltage | 12V |
| Current | $1.3A + 2 \times 10mA = 1.320A$ |
| Power | $12V \times 1.320A = 15.84W$ |

Table. 8. Power Consumption of Laser's Driver

4.6.2 STEPPER DRIVER CIRCUIT

Each motor is fed with 1.7A of current, plus 5mA for the DRV8825 to operate. Never the three stepper motors are ON at the same time, only two are ON simultaneously. [24]

| Parameter | Value |
|-----------|------------------------------|
| Voltage | 12V |
| Current | $1.7A + 5mA = 1.705A$ |
| Power | $12V \times 1.705A = 20.46W$ |

Table. 9. Power Consumption of Stepper Motor's Driver [24]

4.6.3 MAPLE NATIVE

The regulators in the Maple Native consume a maximum of 500mA. Assuming worst case scenario: [31]

| Parameter | Value |
|-----------|--------------------------|
| Voltage | 5V |
| Current | 500mA |
| Power | $5V \times 500mA = 2.5W$ |

Table. 10. Power Consumption of Maple Native Development Board [31]

4.6.4 254-35C1-ROX BUZZER

| Parameter | Value |
|-----------|------------------------|
| Voltage | 5V |
| Current | 8mA |
| Power | $5V \times 8mA = 40mW$ |

Table. 11. Power Consumption of Buzzer [22]

4.6.5 RPR-220 OPTICAL SENSOR

| Parameter | Value |
|-----------|--------------------------|
| Voltage | 5V |
| Current | 50mA |
| Power | $5V \times 50mA = 250mW$ |

Table. 12. Power Consumption of Optical Sensor [23]

4.6.6 ULCD-32PT LCD-TFT SCREEN

| Parameter | Value |
|-----------|---------------------------|
| Voltage | 5V |
| Current | 150mA |
| Power | $5V \times 150mA = 750mW$ |

Table. 13. Power Consumption of the TFT-LCD Screen [25]

4.6.7 LM34CZ TEMPERATURE SENSOR'S CIRCUIT

The circuit used to interface the temperature sensor with the system can be observed on page 10 of the LM34 schematic. [29] The interface uses two active components, the LM34CZ (10mA) sensor and a LM10 [32] operational amplifier (0.27mA). [29]

| Parameter | Value |
|-----------|------------------------|
| Voltage | 5V |
| Current | 10mA + 0.27mA |
| Power | 5V x 10.27mA = 51.35mW |

Table. 14. Power Consumption of Temperature Sensor's Circuit [29]

4.6.8 MB40100V2 LASER'S HEATSINK FAN

| Parameter | Value |
|-----------|--------------------|
| Voltage | 5V |
| Current | 180mA |
| Power | 5V x 180mA = 900mW |

Table. 15. Power Consumption of the Laser's Heatsink Fan [26]

4.6.9 MB20100V2 LASER'S SMOKE FAN

| Parameters | Values |
|------------|--------------------|
| Voltage | 5V |
| Current | 156mA |
| Power | 5V x 156mA = 780mW |

Table. 16. Power Consumption of the Laser's Smoke Fan [27]

4.6.10 MB50100V2 REAR FAN X3

| Parameters | Values |
|------------|--------|
|------------|--------|

| | |
|---------|--------------------|
| Voltage | 5V |
| Current | 236mA |
| Power | 5V x 236mA = 1.18W |

Table. 17. Power Consumption of one Rear Fan [33]

The power supply (XL125-1) [34] can supply up to 125W of power. Calculations for total power supplied and headroom below:

$$P_{Supplied(12V)} = P_{Laser\ Driver} + 2xP_{Stepper}$$

$$P_{Supplied(12V)} = \boxed{56.76W @ 12V}$$

$$Headroom @ 12V Rail = \frac{P_{Max} - P_{Supplied}}{P_{Max}} = \frac{60W - 56.76W}{60W} \times 100 = \boxed{5.4\%}$$

$$P_{Supplied(5V)} = P_{Maple\ Native} + P_{Buzzer} + P_{Optical\ Sensor} + P_{TFT\ Screen} + P_{Temp\ Sensor} + P_{Heatsink\ Fan} + P_{Smoke\ Fan} + 3xP_{Rear\ Fan}$$

$$P_{Supplied(5V)} = \boxed{8.81W @ 5V}$$

$$Headroom @ 5V Rail = \frac{P_{Max} - P_{Supplied}}{P_{Max}} = \frac{75W - 8.81W}{75W} \times 100 = \boxed{88\%}$$

4.7 PORT LOADING

To guarantee the proper functioning of our microprocessor we also made a dedicated port loading analysis.

4.7.1 IO PORT CURRENT LOADING

The Maple Native's integrated regulators can supply up to 250mA for digital connections and 250mA for analog connections. The calculations for MCU port loading are done assuming worst case scenario; that is that the STM32F103ZET6 has to supply current to every pin. This will assure that even at worst-case conditions that will unlikely happen in our system, the STM32F103ZET6 will not overheat or supply more power than what the Native's regulators are capable.

| Parameter | Value |
|-----------|-------|
| | |

| | |
|-------------------------|--------------------------|
| V_{DD} | 3.3V |
| I_{DD} | 70mA |
| $I_{IO\ Digital_{Max}}$ | 250mA |
| $I_{IO\ Analog_{Max}}$ | 250mA |
| $P_{PKG_{Max}}$ | 666mW |
| Φ_{JA} | $30 \frac{^{\circ}C}{W}$ |
| Φ_{JC} | |
| $T_{J_{Max}}$ | 105 °C |

Table. 18. STM32F103ZET6 Electrical and Thermal Characteristics [31] [30]

| Part | Number of Connections | IO Required Current | Total |
|----------------------------------|------------------------------|----------------------------|--------------|
| DRV8825 Stepper Driver (x3) [24] | 33 | 20μA | 660μA |
| Laser's Driver | 1 | 5μA | 5μA |
| 254-35C1-ROX Buzzer [22] | 1 | 80μA | 80μA |
| RPR-220 Optical Sensor [23] | 1 | 300μA | 300μA |
| ULCD-32PT LCD-TFT Screen [25] | 3 | 3.4mA | 10.2mA |
| Laser's Heatsink Fan [26] | 1 | 1.8mA | 1.8mA |
| Laser's Smoke Fan [27] | 1 | 1.56mA | 1.56mA |
| LM34CZ Temperature Sensor [29] | 1 | 10mA | 10mA |
| JZC-11F-05VDC-1Z Relay [35] | 1 | 7.5mA | 7.5mA |

Table. 19. Current Sourced to each component by the I/O ports.

$$I_{IO\ ports} = \boxed{32.11mA \ll 250mA}$$

All the I/O ports are being used as digital connections; they draw the current from the same regulator.

4.7.2 STM32F103ZET6 PACKAGE POWER & TEMPERATURE

The junction temperature T_J cannot exceed the maximum value T_{JMax} from Table Table. 18. The formula for calculating the junction temperature does not depends on θ_{jc} , so the datasheet does not provide a value for it: [30]

$$T_J = P_{PKG} \times \theta_{JA} + T_A$$

$$I_{PKG} = I_{DD} + I_{IO\ ports} = 70mA + 32.11mA = \boxed{102.11mA}$$

$$P_{PKG} = V_{DD} \times I_{PKG} = 3.3V \times 102.11 = \boxed{336.96mW < 666mW}$$

$$T_J = P_{PKG} \times \theta_{JA} + T_A = 336.96mW \left(30 \frac{^{\circ}C}{W}\right) + 27^{\circ}C = \boxed{37.11^{\circ}C < T_{JMax}}$$

As the previous calculations show, the microprocessor operating temperature is well below the maximum value, enhancing the reliability of the system.

4.8 TIMING ANALYSIS

Timing was very important in our application so various timing calculations were done before hand.

4.8.1 TIMING BASES

$$f_{CLK} = 72MHz$$

$$prescaler_{MAX} = 2^{16} - 1 = 65535$$

$$offset_{MAX} = 2^{16} - 1 = 65535$$

4.8.2 LASER'S PWM TIMER

$$f = 1000Hz$$

$$\frac{f_{CLK}}{f} = 72000 \quad \text{A divider is required}$$

$$prescaler = 2, \quad offset = 36000$$

4.8.3 BUZZER'S TIMER

$$f = 2800\text{Hz}$$

$$\frac{f_{CLK}}{f} = 25714 \text{ No divider required}$$

$$prescaler = 1, offset = 25714$$

4.8.4 PROGRESS BAR'S TIMER

$$f = 0.588\text{Hz}$$

$$\frac{f_{CLK}}{f} = 122400000 \text{ A divider required}$$

$$prescaler = 2500, offset = 48960$$

4.8.5 ULCD-32PT LCD-TFT SCREEN TIMING

The touch screen uses a Picaso GFX2 in slave mode as its controller. A delay of 2000ms is required after powering up or resetting the Picaso GFX2 before sending any command. Then the host sends the Auto-Baud command which sets the baud rate at the default 9600 baud. The baud rate of the Picaso GFX2 and the host must be the same. Since the Maple Native is capable of much higher baud rates, the baud rate was set at 282353 baud, the highest value supported by the Picaso GFX2. Note that even though the Picaso GFX2's datasheet shows 250k as the fixed highest baud rate, that value is actually 282353 baud. After the host (Maple Native) receives acknowledgment of the baud rate change from the Picaso GFX2 (06h), the screen is ready.

To communicate with the host, the Picaso GFX2 in slave mode uses asynchronous bidirectional serial communication through its UART. The data format is 8 Bits with no parity and 1 stop bit. The serial data is true and not inverted. [36]

4.8.6 DRV8825 STEPPER DRIVER TIMING

| | | | MIN [24] | MAX [24] | UNIT |
|---|----------------|---------------------------|--------------------|--------------------|-------------|
| 1 | f_{STEP} | Step frequency | | 250 | <i>kHz</i> |
| 2 | $t_{WH(STEP)}$ | Pulse duration, STEP high | 1.9 | | μs |

| | | | | | |
|---|----------------|--------------------------------------|-----|--|---------|
| 3 | $t_{WL(STEP)}$ | Pulse duration, STEP Low | 1.9 | | μs |
| 4 | $t_{SU(STEP)}$ | Setup time, command to STEP rising | 650 | | ns |
| 5 | $t_{H(STEP)}$ | Hold time, command to STEP rising | 650 | | ns |
| 6 | t_{ENBL} | Enable time, nENBL active to STEP | 650 | | ns |
| 7 | t_{WAKE} | Wakeup time, nSLEEP inactive to STEP | 1.7 | | ms |

Table. 20. DRV8825 Timing Requirements

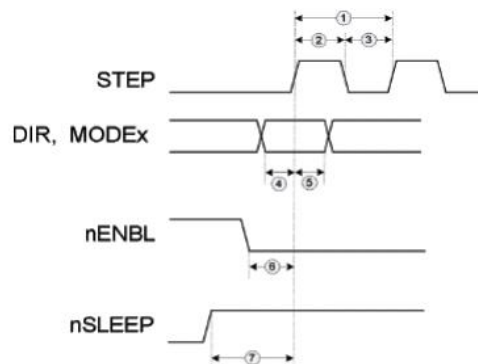


Fig. 39. DRV8825 Timing Diagram [24]

The DRV8825 stepper driver has 5 signals that need to be controlled in order to rotate a motor one step. To move a motor one step the nSleep signal has to be on high at least 1.7ms before the step. The enable signal has to be on low at least 650ns before the step. The DIR (direction of the step) has to be on the value corresponding to the desired direction at least 650ns before the step. MODEx is not used since Pyrobot does not require micro-stepping. After these signals are set, the STEP signal is set to high for at least 1.7 μ s. A step will occur when the signal goes from low to high (rising edge).

4.8.7 SERIAL COMMUNICATION BETWEEN HOST & SYSTEM

The host and system applications engage in asynchronous serial communication. The channel is configured to 57600 Baud rate with 8 data bits, 1 stop bit & no parity with the host acting as master and the system as slave device. Communication is handled as follows: the host initiates data transmission by sending a packet to the system application. It will then wait up to two seconds for a response. If the

response is received, the host may proceed to communicate with the system. If not, then the host will abort the request and try again later.

Packets can be one of two types: command packets and data packets. Command packets are fixed to a length of two bytes. The first byte identifies the packet as a command packet; its value is zero. The second byte represents the command. The system currently recognizes two command bytes: STATUS ('s') and TRANSMIT ('t'). Data packets can be of variable length, up to a max size of 50 bytes. The maximum size buffer of the system defaults to 64 bytes. Sending a packet of this size makes the system hang however, which is the reason the host uses a 50 byte packet instead. The first byte in the packet identifies it as a data packet. The next byte represents the size of the packet, including these first two bytes. The subsequent bytes, sent in little endian format, represent raw data and can be interpreted freely.

Currently, the only time the system receives a data packet is when image data is being transmitted. For this exchange, the system expects the packet to have a particular structure: the first four bytes (after skipping the identifying byte and the size byte) represent the image width. The next four bytes represent its height. All subsequent bytes are interpreted as pixel values, starting from the first row, then the second, and so on.

4.9 MCU MEMORY USAGE

The Maple Native β has a bootloader on the STM32F103ZET6 designed to make it possible for developers to upload code through USB without any external tools. The bootloader is the first thing that runs when the device boots, but only for a few seconds while it waits for the upload command. After it times out, it starts running the user code.

About 12KB of the flash are dedicated to the bootloader but the Native has 512KB of flash memory, which is more than enough to store much more code than what we actually have. [37]

| Block | Base addresses | Size (bytes) |
|----------------|---------------------------|--------------|
| Flash Start | 0x0800 0000 – 0800 2F10 | 12K |
| User Code | 0x0800 2F11 – 0800 62E0 | 13K |
| Emergency Data | 0x0807 F800 - 0x0807 FFFF | 2K* |

| | | |
|---------------------------|---------------------------|------------|
| RAM for Variables | 0x2000 0000 – 0x2000 FFFF | 64K |
| External Image-RAM | 0x60000000 - 0x60100000 | 1MB |

Table. 21. PyroBot's STM32F103ZET6 memory map

* The STM32F103ZET6's flash memory can only be written 1 page at a time. Each individual page in memory is 2KB in size. This last page in the flash (255) is where all the variables will be stored if PyroBot shuts down incorrectly. This only happens when the emergency stop button is pressed or when PyroBot unexpectedly loses power.

When PyroBot reboots, it will check the incorrect shutdown flag. If it's true, then it will save its current X,Y,Z coordinates stored in non-volatile flash to return the motor to its original position. If the incorrect shutdown flag is false, then it will continue with its regular operation.

Images will be mostly stored in the 1MB external RAM module. The microcontroller cannot physically distinguish between its internal RAM and the external module so it will store all of the needed variables in our code inside its 64KB of internal RAM. When it's done, it will start saving the image and continue without interruption when it ends the internal ram and keeps saving into the external module. [37]

4.10 HARDWARE RELIABILITY & PROFESSIONAL COMPONENT

There were three main professional components in the design of PyroBot. The components were safety, reliability, and precision. Safety was mitigated and handled by adding the opto-switch, temperature sensor, lowered base, and fan system. Precision was created by using the proper tools and measurements including the belting, pulley system, laser, and MakerSlide. Reliability has been tested and verified by running long burn jobs. We have run the system for a full 24 hours and it has kept its stability.

5 SOFTWARE DESIGN

In this section we begin by explaining our implementation of the system firmware and the host application. Flowcharts and diagrams are provided for all core software components. Following that we describe how these two applications communicate with each other via serial data transmission. We explain the structure of the transmitted packets and give an example of the transmittal of an image. We finish off with discussing overall professional considerations that dictated the design of the software architecture and its algorithms.

5.1 FIRMWARE

The system firmware can be broken up into two logical blocks: the initialization block and the main program block. The initialization block sets up all peripherals and I/O ports, clears all system variables and initializes the system state before going into the main program block.

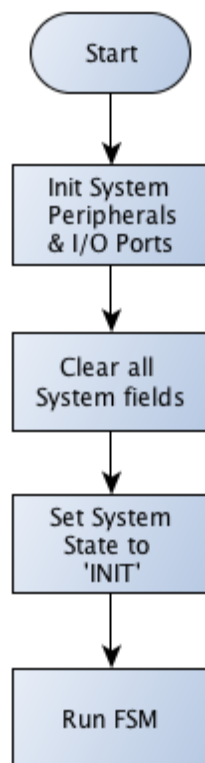


Fig. 40. Initialization Block Flowchart

The main program block was implemented by following a finite state machine design pattern. The code is broken up into 7 logical states as represented in the next diagram.

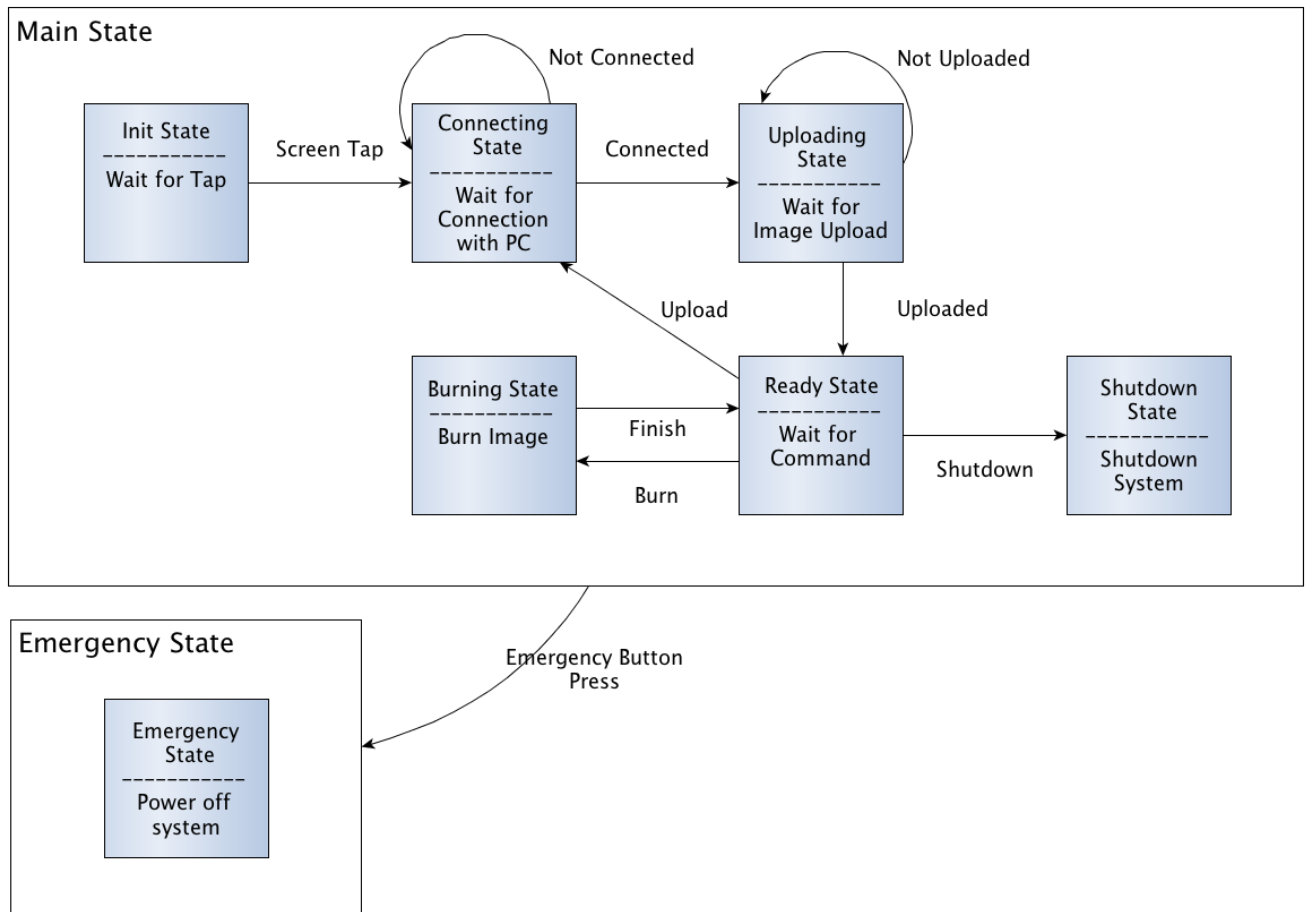


Fig. 41. Firmware Main State Diagram

5.1.1 INITIALIZATION STATE

The state begins by displaying an introduction screen. It then locks until a tap is registered on the LCD screen. Once registered, the system moves to the Recovering State.

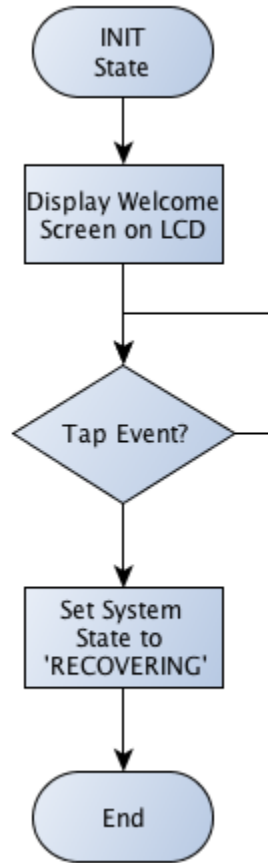


Fig. 42. Initialization State Flowchart

5.1.2 CONNECTING STATE

Here, the system checks for connection to a host device. If no connection is found, it informs the user by updating the LCD. Once a connection is detected, it moves on to the Uploading State.

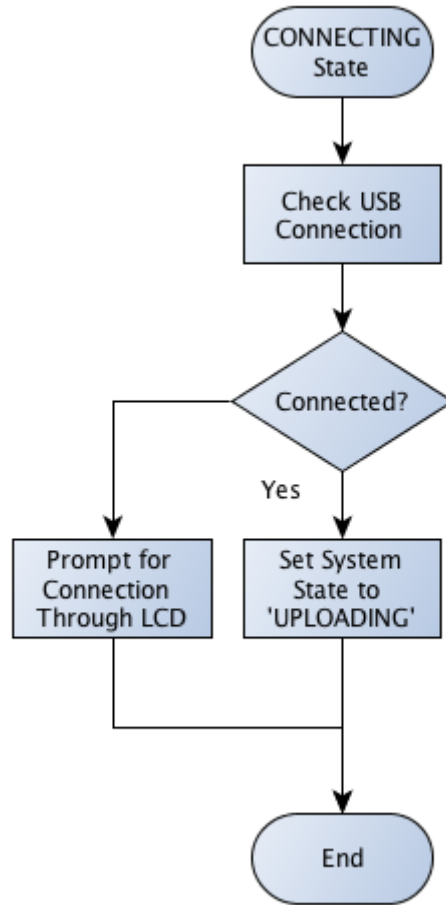


Fig. 43. Connecting State Flowchart

5.1.3 UPLOADING STATE

The Uploading State works by continuously monitoring the USB port for incoming data packets. If one is found, it further inspects it to determine its contents. If a status command is parsed from the packet, the system returns a byte indicating it is ready to accept commands through USB. If a transmit command is parsed instead, the system will flush the USB buffer and prepare to save the incoming image data.

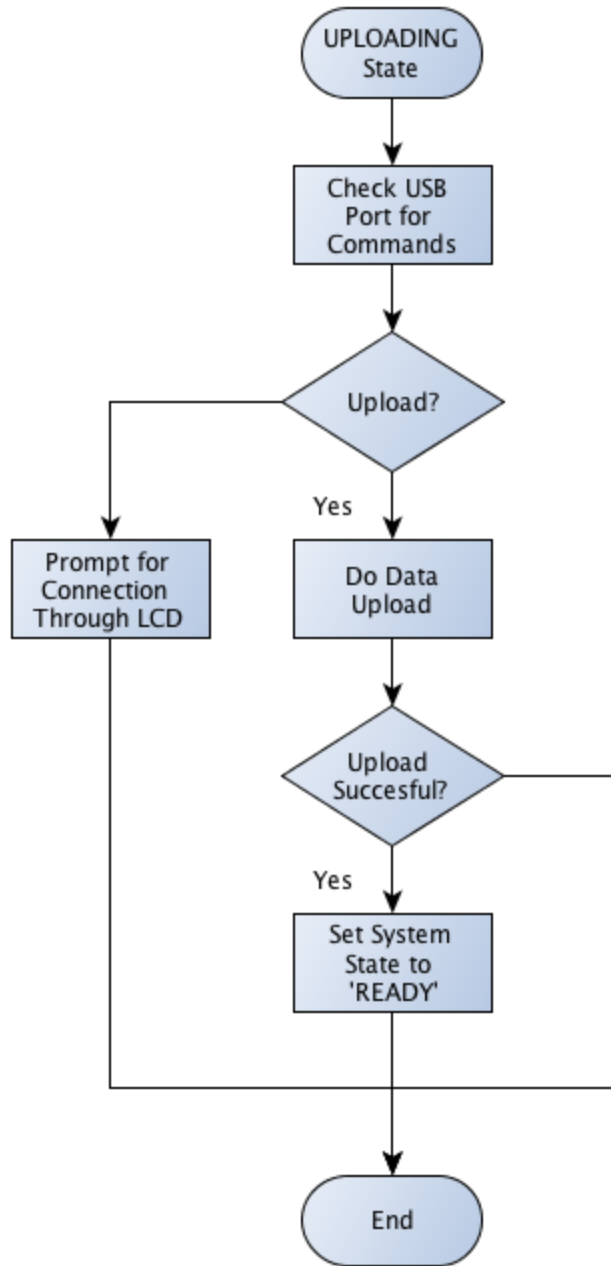


Fig. 44. Uploading State Flowchart

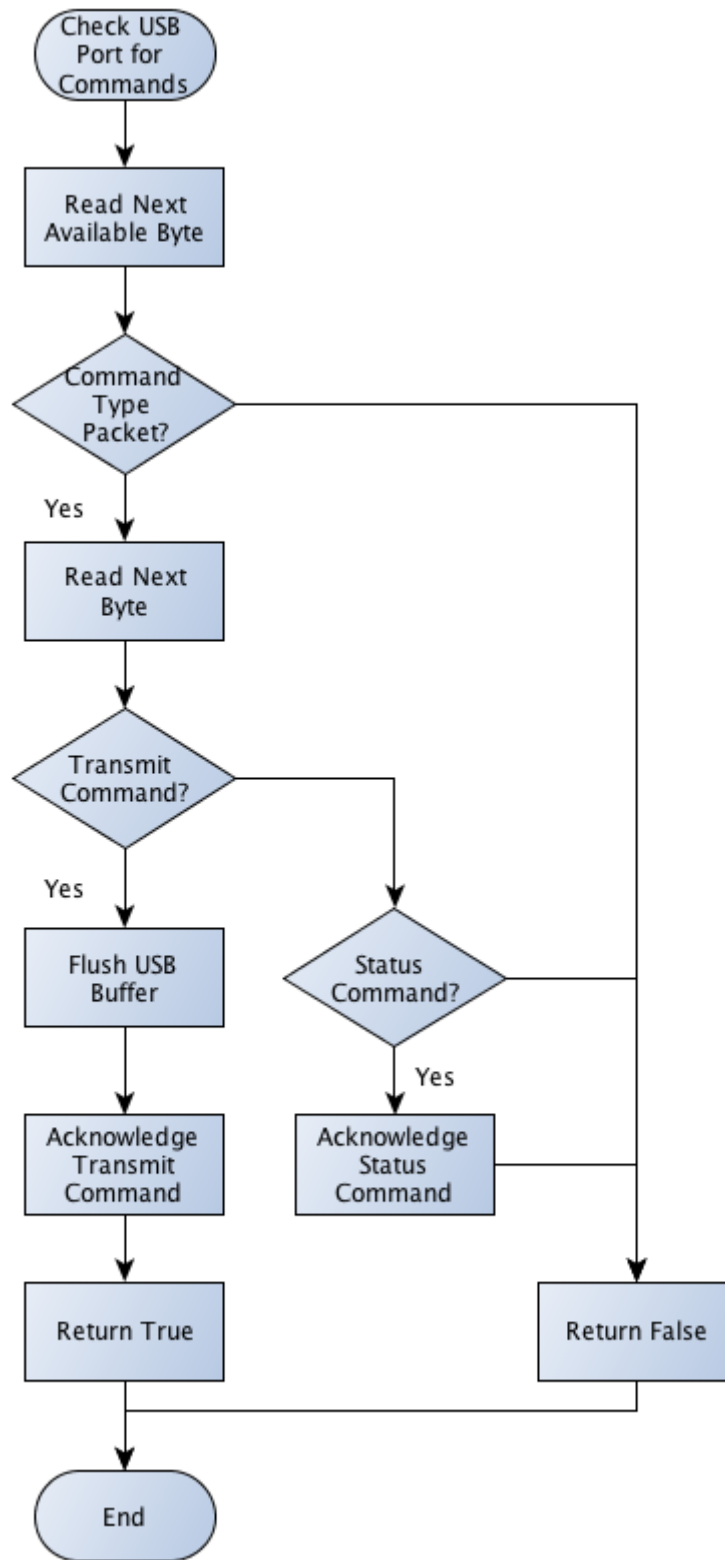


Fig. 45. Check USB Port for Commands Flowchart

When saving image data, the system expects the first 8 bytes of data to represent the image's dimensions. It uses this information to determine the image's size and allocate space in memory for it. If allocation succeeds, then the system will continue to read in data until the size number of bytes are received. Requests for more data are sent when the input buffer is empty. If allocation fails, then the system aborts the upload process and returns to monitoring the USB port.

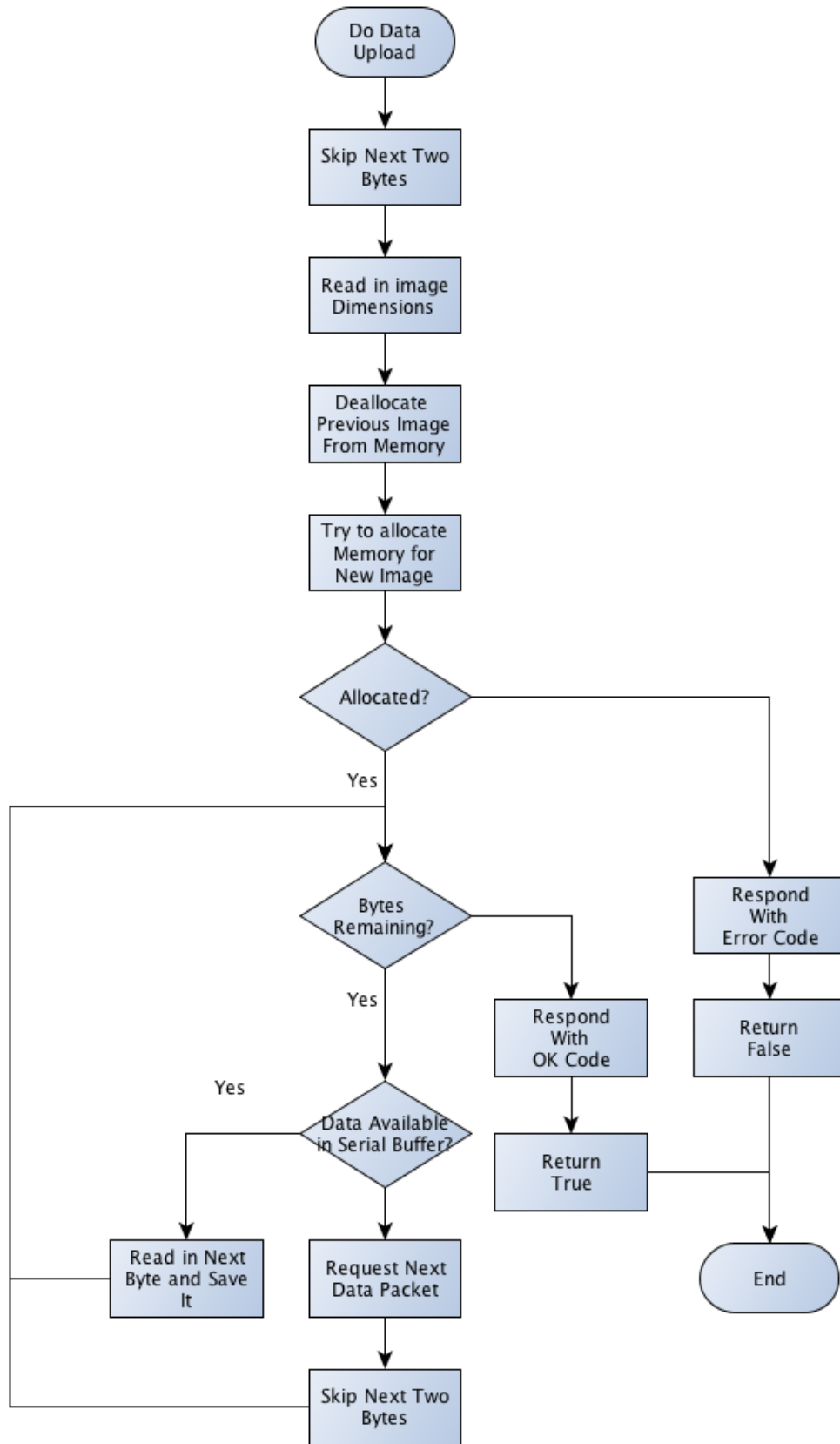


Fig. 46. Do Data Upload Flowchart

5.1.4 **READY STATE**

This state exposes controls for the user to interact with. Burn, Upload, and Shutdown buttons are presented on the LCD screen. The system will move to the Burning, Connecting and Shutdown States, respectively, whenever a particular button is pressed.

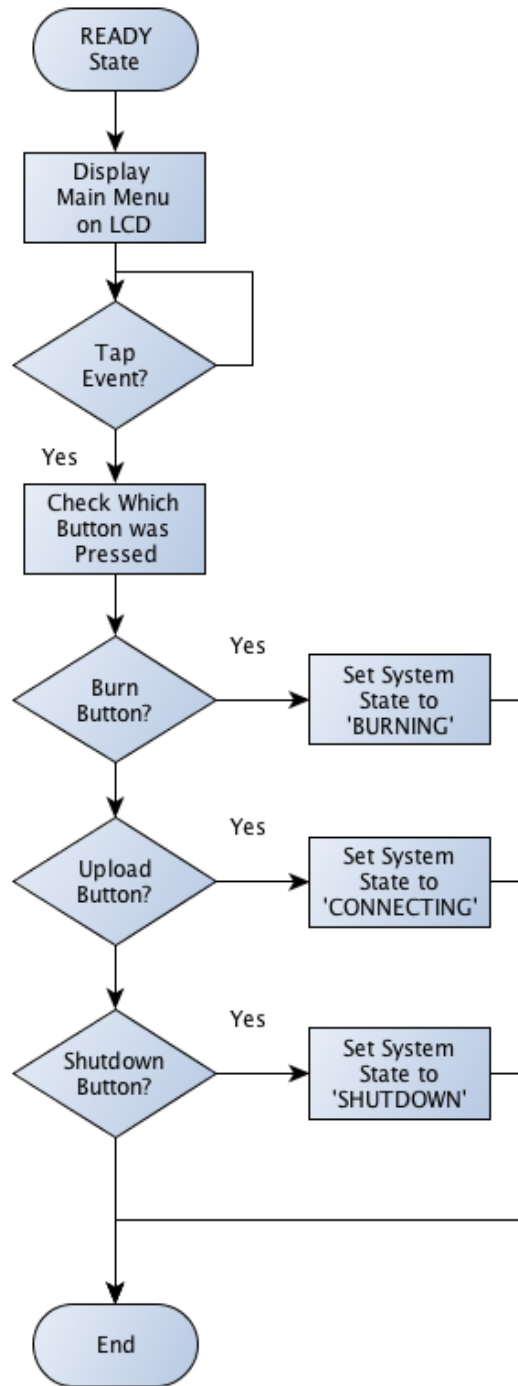


Fig. 47. Ready State Flowchart

5.1.5 BURNING STATE

The system begins by reading from the optical sensor input pin to determine if a wooden plate has been placed on the system's base. If no plate is detected, the system informs the user of the error and returns to the Ready State. If a plate is found to be in place then the system will proceed to burn the uploaded image as expressed in the following diagrams.

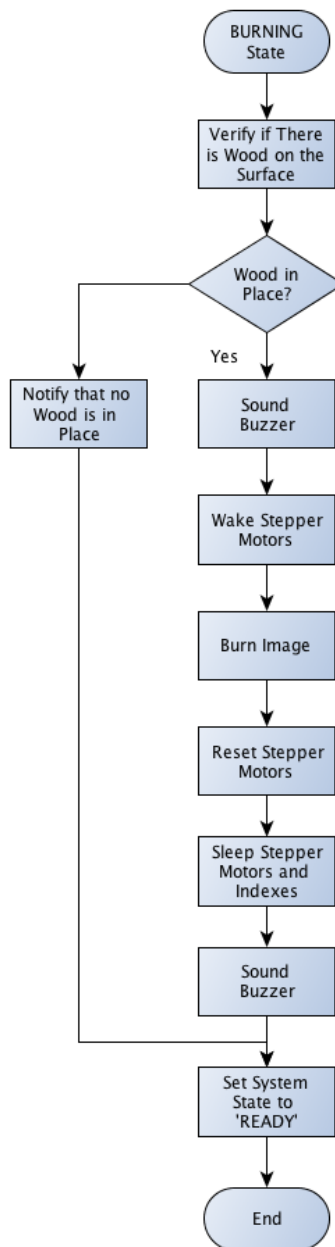


Fig. 48. Burning State Flowchart

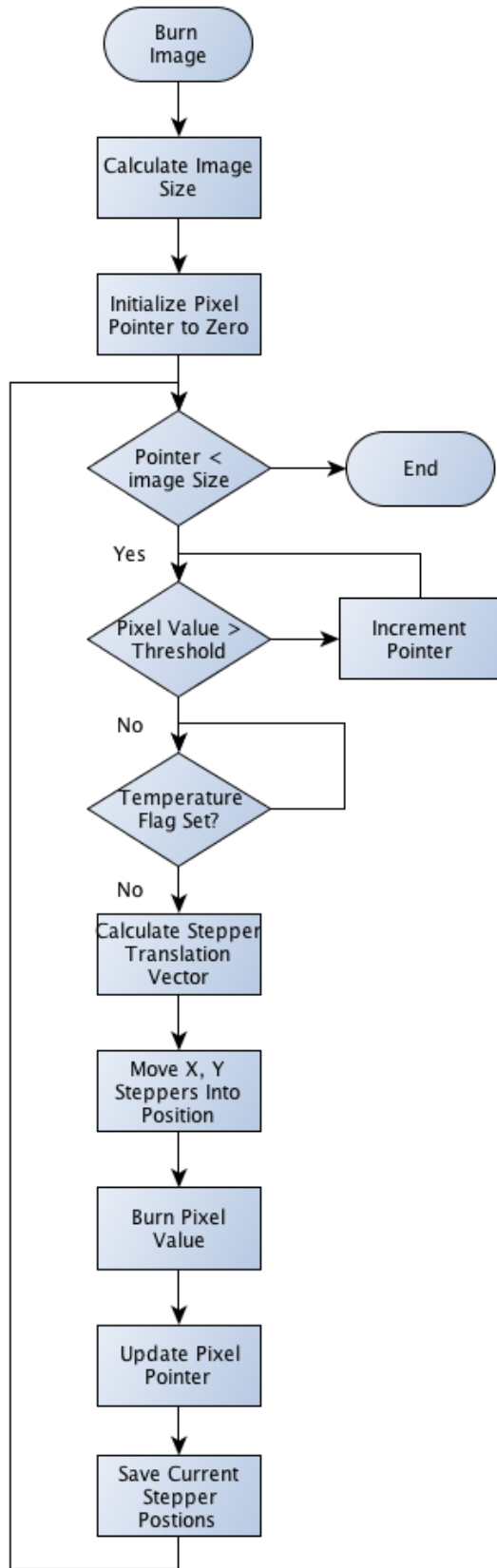


Fig. 49. Burn Image Flowchart

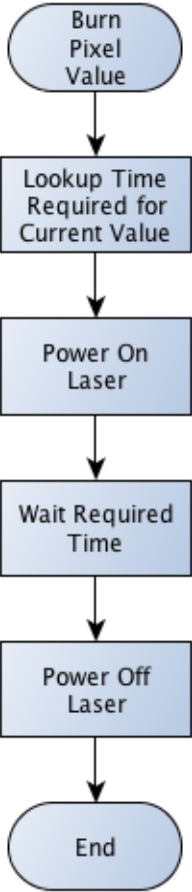


Fig. 50. Burn Pixel Image Flowchart

5.1.6 SHUTDOWN STATE

For the Shutdown State, the system will save in flash memory that it powered off appropriately. The state ends by informing the user of the shutdown event through the LCD.

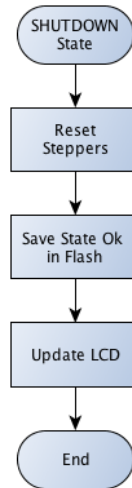


Fig. 51. Shutdown State Flowchart

5.1.7 EMERGENCY LOCK STATE

This state is reached whenever the emergency button located on the front panel is pressed. The system powers off the laser and locks. Power needs to be cut from the system to remove it from this state.

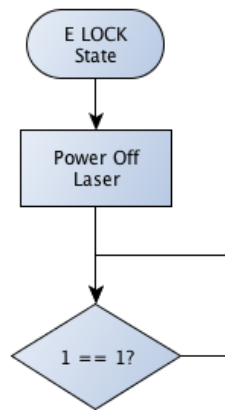


Fig. 52. Emergency Lock State Flowchart

5.2 HOST APPLICATION

A client application allows users to interface the system. It presents a graphical interface through which users can select images and upload them onto the system. The following diagram describes the general logic of the application

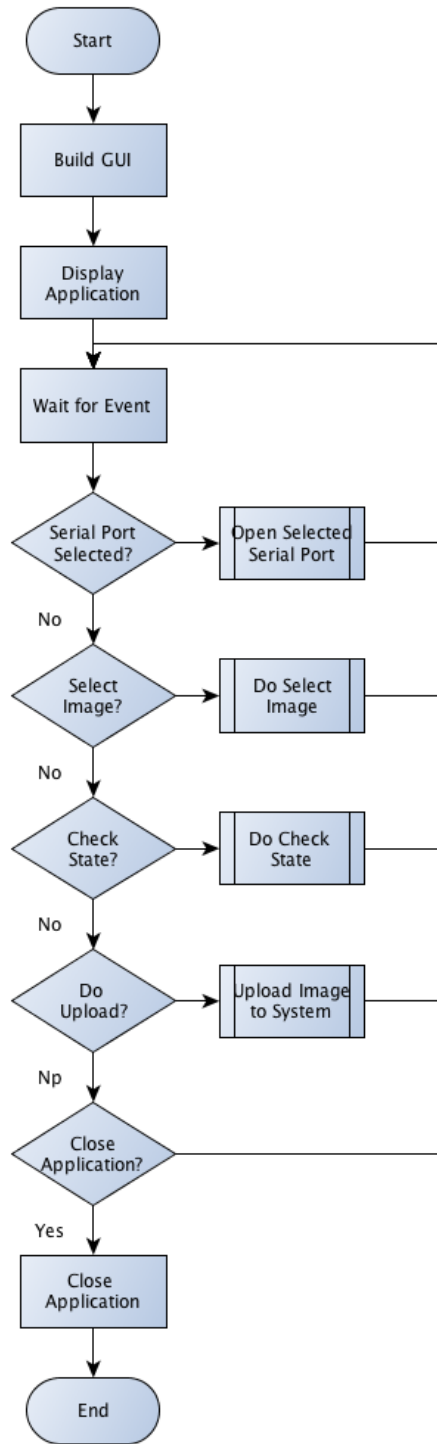


Fig. 53. Host Application Flowchart

5.2.1 CONNECTING WITH THE SYSTEM

Users need to select from one of the available serial ports displayed on the application. On selection, the application will try to establish communication with the port by opening it. If successful, it will launch a background thread to monitor the readiness of the system.

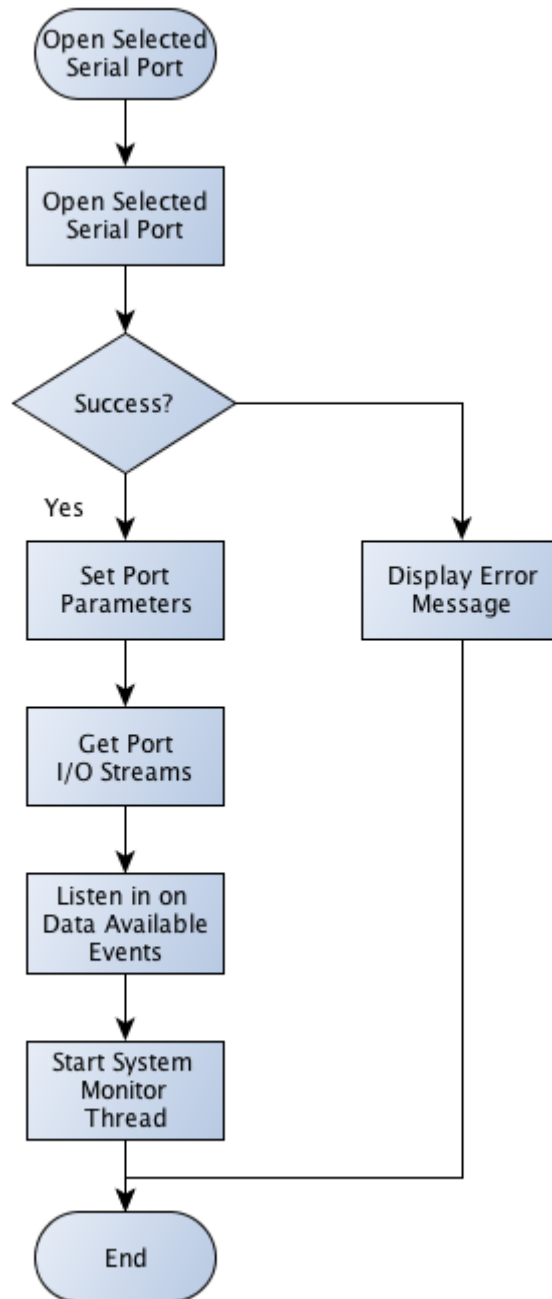


Fig. 54. Connecting with the System Flowchart

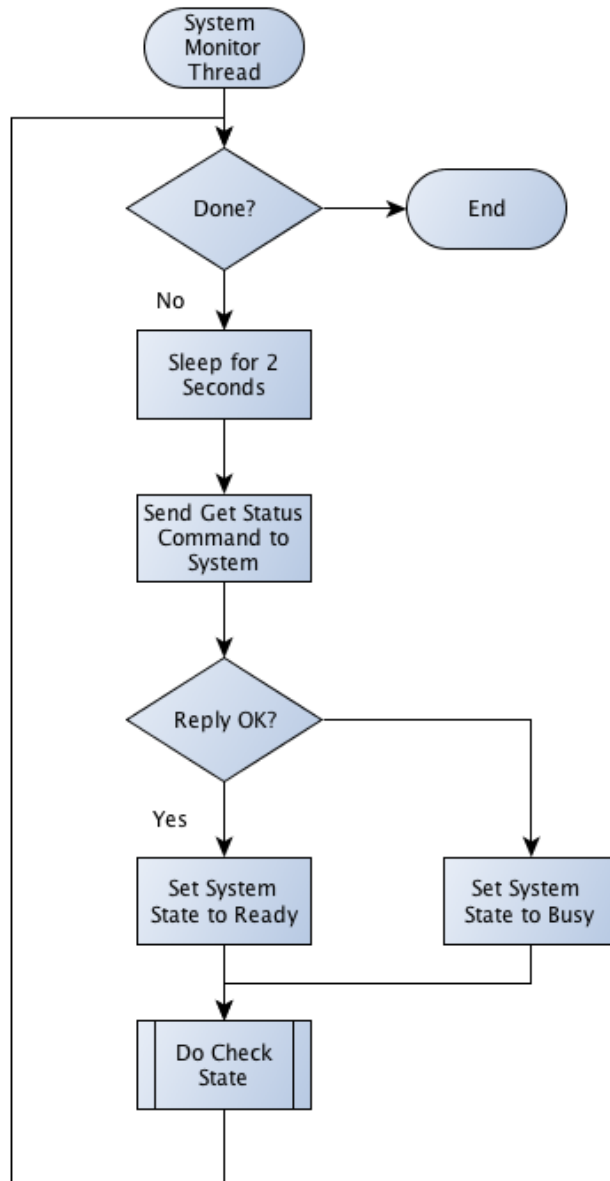


Fig. 55. System Monitor Flowchart

5.2.2 SELECTING AN IMAGE

The application provides a graphical interface through which users can select the image they would like to burn. The selected file will be treated as an image and automatically converted into grayscale. If the image exceeds the maximum permissible dimensions allowed by the system, then it will also be automatically resized.

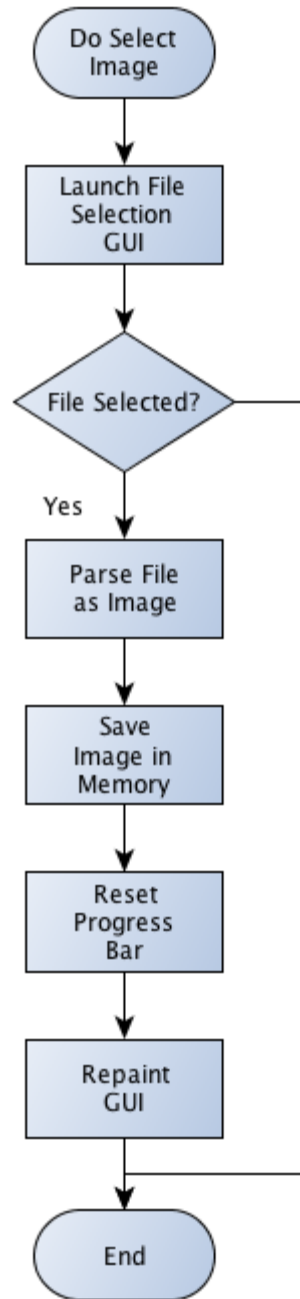


Fig. 56. Selecting an Image Flowchart

5.2.3 UPLOADING AN IMAGE TO THE SYSTEM

Uploading is allowed only when a serial port has been successfully opened, an image has been selected and the monitor thread reports that the system is ready to accept data (this is monitored asynchronously, which is why it isn't presented in the flowchart). On upload, the system will kill the monitor thread and launch another thread to handle the image upload. This thread will communicate with the system as

described in the Serial Communication section of this document. On completion, it will launch a new monitor thread.

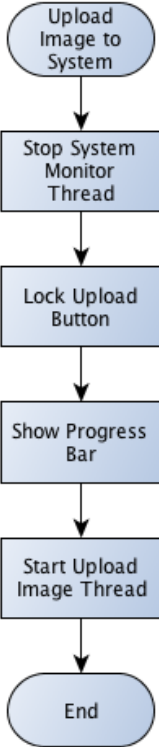


Fig. 57. Upload an Image to System Flowchart

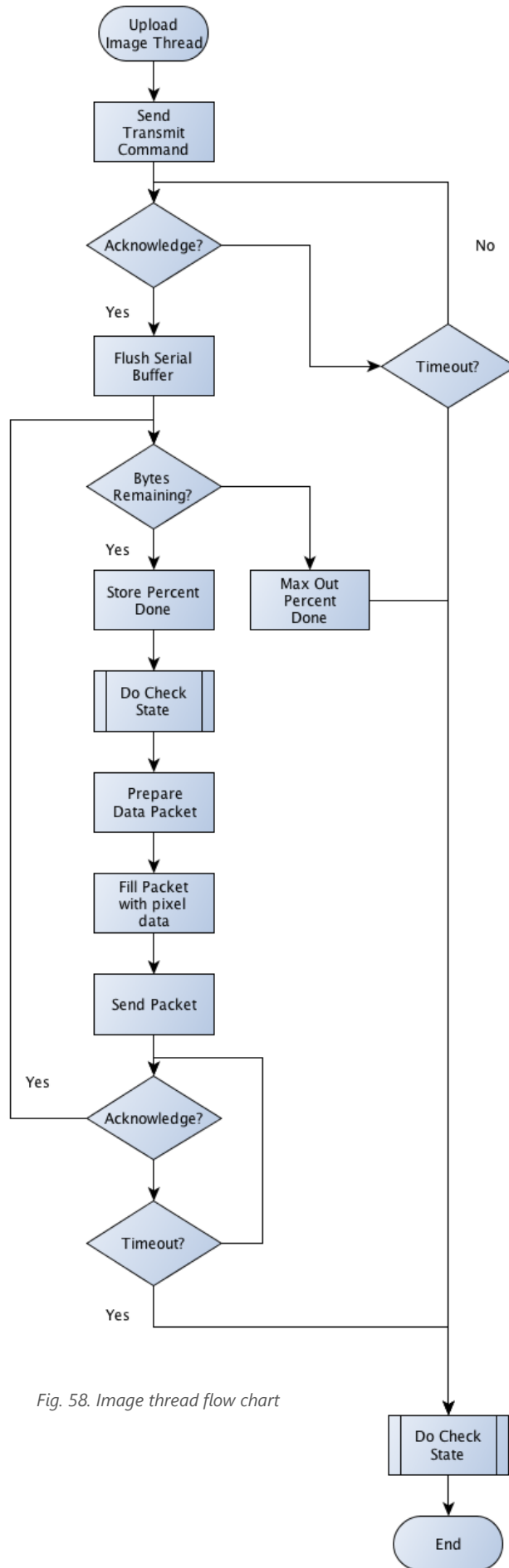
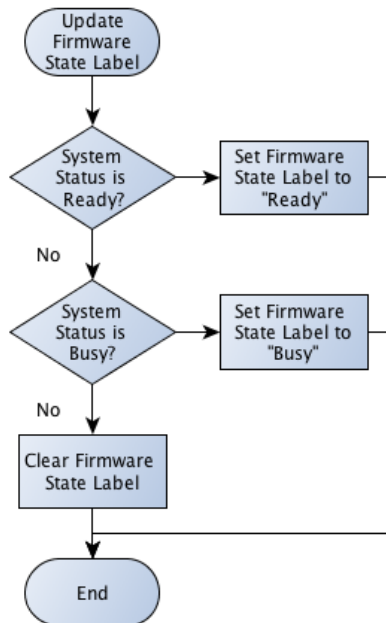
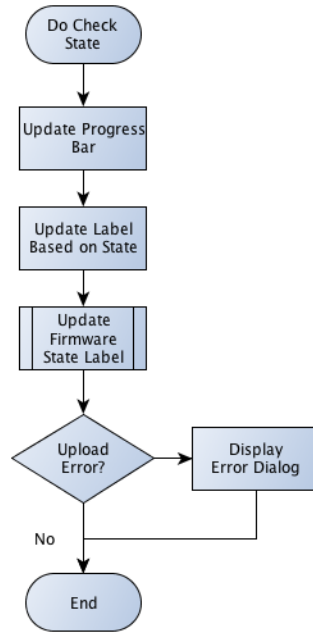


Fig. 58. Image thread flow chart

5.2.4 ASYNCHRONOUS BEHAVIOR

Two functions are handled asynchronously while the application is running: monitoring the system's state and uploading images onto the system. Monitoring occurs every 2 seconds after a serial port has been successfully opened. Uploading happens outside the main execution thread once the upload button has been pressed. Both threads report back to the main application periodically and force it to redraw itself. The UI will update itself according to the following diagrams.



5.3 SERIAL COMMUNICATION BETWEEN HOST & SYSTEM

The host and system applications engage in asynchronous serial communication. The channel is configured to 57600 Baud rate with 8 data bits, 1 stop bit & no parity with the host acting as master and the system as slave device. Communication is handled as follows: the host initiates data transmission by sending a packet to the system application. It will then wait up to two seconds for a response. If the response is received, the host may proceed to communicate with the system. If not, then the host will abort the request and try again later.

Packets can be one of two types: command packets and data packets. Command packets are fixed to a length of two bytes. The first byte identifies the packet as a command packet; its value is zero. The second byte represents the command. The system currently recognizes two command bytes: STATUS ('s') and TRANSMIT ('t'). Data packets can be of variable length, up to a max size of 50 bytes. The maximum size buffer of the system defaults to 64 bytes. Sending a packet of this size makes the system hang however, which is the reason the host uses a 50 byte packet instead. The first byte in the packet identifies it as a data packet. The next byte represents the size of the packet, including these first two bytes. The subsequent bytes, sent in little endian format, represent raw data and can be interpreted freely.

Currently the only time the system receives a data packet is when image data is being transmitted. For this exchange, the system expects the packet to have a particular structure: the first four bytes (after skipping the identifying byte and the size byte) represent the image width. The next four bytes represent its height. All subsequent bytes are interpreted as pixel values, starting from the first row, then the second, and so on.

5.3.1 COMMAND PACKET STRUCTURE

Command packets are two byte long blocks of data constructed as described in the next two figures.

| | | | | | | | | | | | | | | | |
|------------------|----------|----------|----------|----------|----------|----------|----------|--------------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| PACKET_TYPE_BYTE | | | | | | | | COMMAND_BYTE | | | | | | | |

Table. 22. Command packet structure.

| Name | Description | Value(s) |
|------------------|---|----------|
| PACKET_TYPE_BYTE | Identifies the packet as a command packet | 0x00 |

| | | |
|--------------|-------------------------|----------------------------------|
| COMMAND_BYTE | Command to be performed | 0x73 - Status 0x74 - Transmit |
|--------------|-------------------------|----------------------------------|

Table 23. Command packet byte description

5.3.2 DATA PACKET STRUCTURE

Data packets can be of variable length, up to 50 bytes. The packet is constructed as described next.

| | | | | | | | | | | | | | | | | | | | | |
|------------------|----------|----------|----------|----------|----------|----------|----------|-----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|-----------|-----|------------|------------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | ... | 398 | 399 |
| PACKET_TYPE_BYTE | | | | | | | | SIZE_BYTE | | | | | | | | DATA_BYTES | | | | |

Table 24. Data packet structure.

| Name | Description | Value(s) |
|------------------|--|--------------|
| PACKET_TYPE_BYTE | Identifies the packet as a data packet | 0x01 |
| SIZE_BYTE | Total packet size | 0x03 to 0x32 |
| DATA_BYTES | Data to be sent | |

Table 25. Data packet byte description.

5.3.3 IMAGE UPLOAD PACKET STRUCTURE

Tables 26, 27, and 28 describe the first data packet transmitted when uploading an image onto the system. Unique to this packet are bytes 2 through 5 and 6 through 9, which represent the image's width and height, respectively.

| | | | | | | | | | | | | | | | |
|----------|----------|----------|----------|----------|----------|----------|----------|-----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | SIZE_BYTE | | | | | | | |

Table 26. Image upload first block.

| | | | | | | | | | |
|-------------|-----------|-----|-----------|-----------|-----------|--------------|-----|-----------|-----------|
| 16 | 17 | ... | 46 | 47 | 48 | 49 | ... | 78 | 79 |
| IMAGE_WIDTH | | | | | | IMAGE_HEIGHT | | | |

Table. 27. Image upload packet second block.

| | | | | | | | | |
|------------|-----------|-----------|-----------|-----|------------|------------|------------|------------|
| 80 | 81 | 82 | 83 | ... | 396 | 397 | 398 | 399 |
| PIXEL_DATA | | | | | | | | |

Table. 28. Image upload packet third block.

| Name | Description | Value(s) |
|--------------|--|--------------|
| SIZE_BYTE | Total packet size | 0x03 to 0x32 |
| IMAGE_WIDTH | Number of columns in image array | |
| IMAGE_HEIGHT | Number of rows in image array | |
| PIXEL_DATA | Image pixel values for a given <row, column> element | |

Table. 29. Image packet byte description.

5.4 RELIABILITY AND PROFESSIONAL CONSIDERATIONS

Safety considerations had a large influence in all aspects of PyroBot’s design. With regards to software, various mechanisms were put into place to protect both the system and its users. The system can receive notifications from the various sensors equipped on it in order to prevent invalid operation. For instance, the software will not allow the user to burn an image if it detects that no plate was placed on the base. The system will also lock if it detects that the laser is operating beyond its operating temperature or if the emergency lock button is pressed. Power failure scenarios were also taken into consideration and are handled with the help of software. The system will store the laser position if power is cut off so that it may automatically return to the origin during the next power on sequence.

Also, software mechanisms were placed to improve the system’s color spectrum and operation time. Experimental tests showed that, when using time modulation, the system’s color gradient was shifted towards black. To compensate for this, grayscale values were grouped into blocks and nonlinearly

mapped to particular time intervals in order to obtain a more accurate color gradient. This helped ensure the fidelity of the printed image. The operation time was improved by optimizing two aspects of the burning process. Firstly, grayscale values were threshold so that colors that are too light to be printed be skipped. Secondly, the system exploited the x-axis' return motion and prints pixels in both the forward and reverse directions. These two changes resulted in a noticeable reduction of the total time it took to burn an image.

6 RESULTS

Overall the construction of PyroBot was a complete success. Not only is the system stable and robust, but it is simple and easy to run. Here are some of the final specifications for PyroBot:

Laser

- Type: Blue Laser Diode
- Power: 1.6 W
- Wavelength: 445nm

System

- Dimensions: 14.663" x 2.280" x 12.07"
- Working Area: 9" x 7.2"
- Resolution: 141 - 282 dpi
- Power Supply: 100~240 VAC @ 1.8 A
- Total Cost (in materials) : \$721.31

Image

- Size: Up to 1MB
- Dimensions: Up to 1269px x 1015px
- Color: 5 bit grayscale
- Supported Formats: JPEG, PNG, GIF, BMP, WBMP

Display

- Type: 3.2" TFT Screen with 4-Wire Resistive Touch Panel

Connectivity

- USB Standard Type A Plug

6.1 SYSTEM PHOTOS

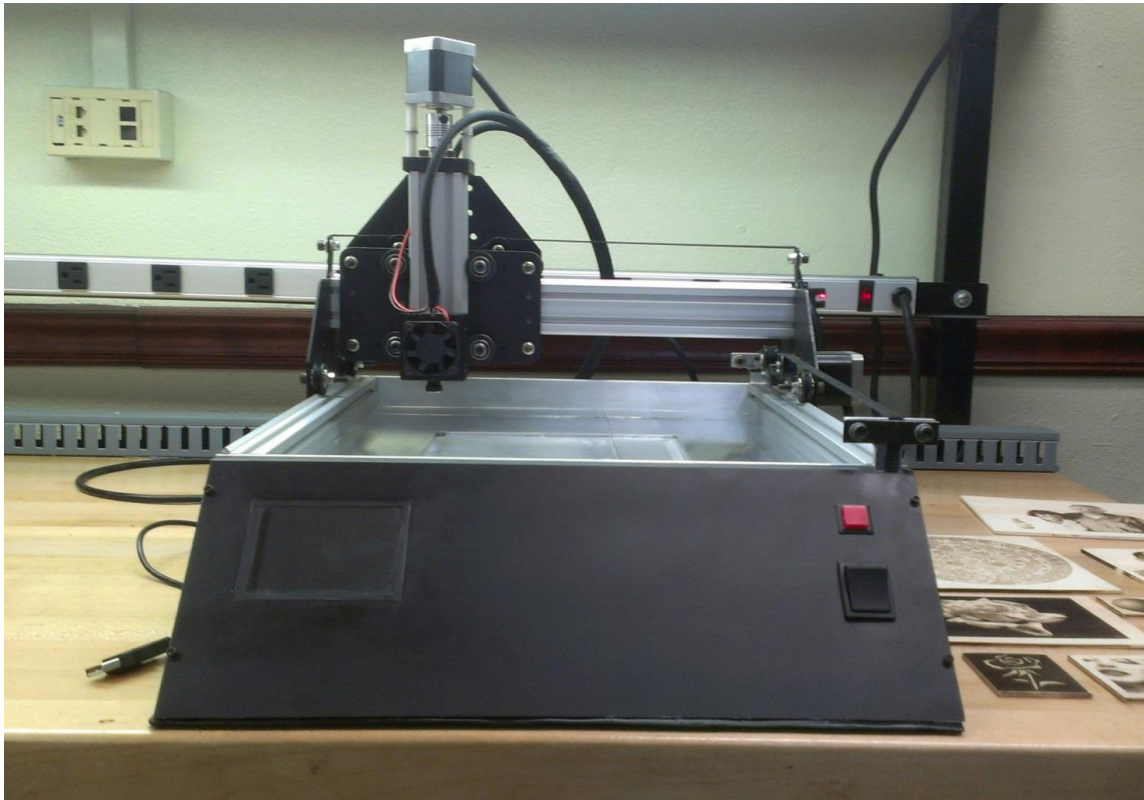


Fig. 59. Front View

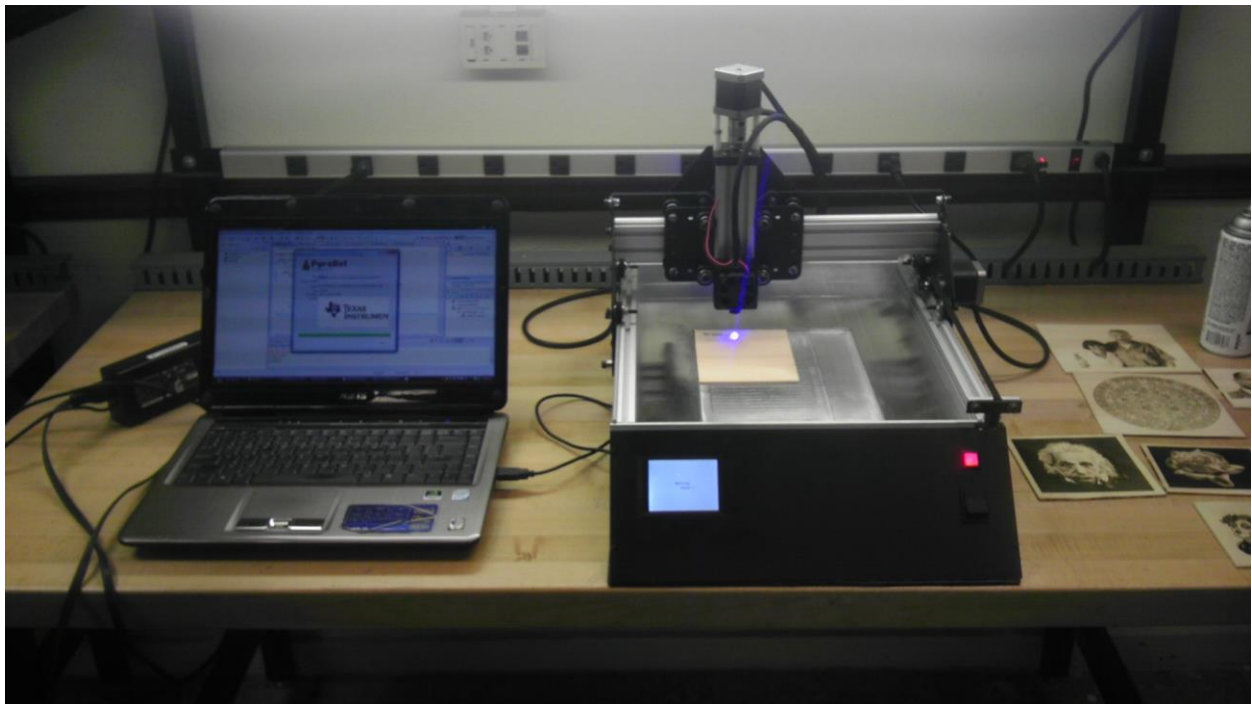


Fig. 60. PyroBot burning the TI logo.

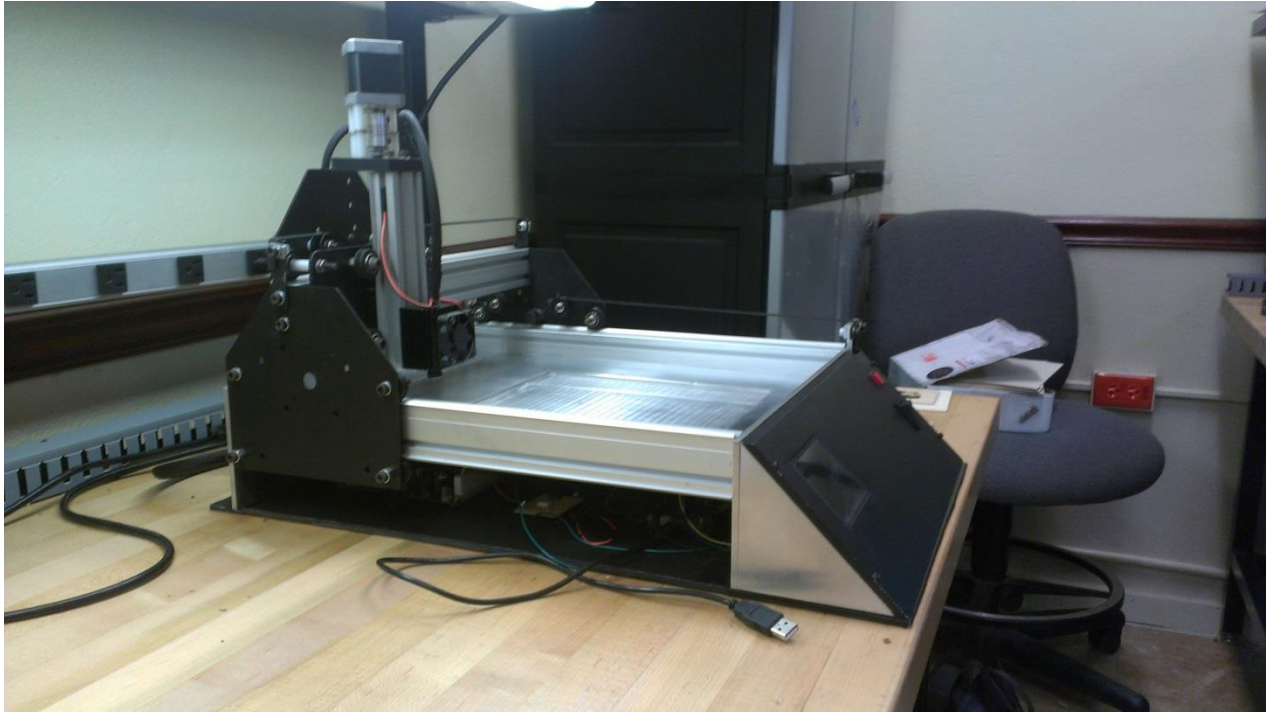


Fig. 61. Side View

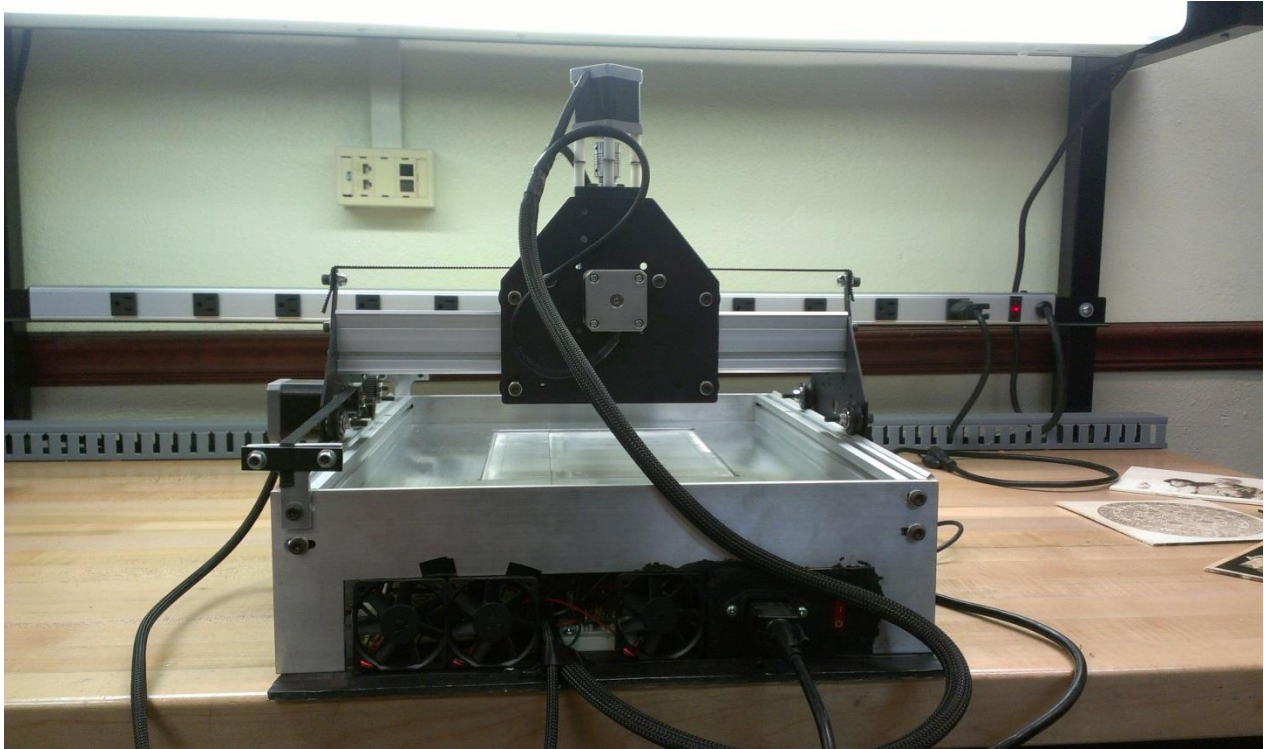


Fig. 62. Back View

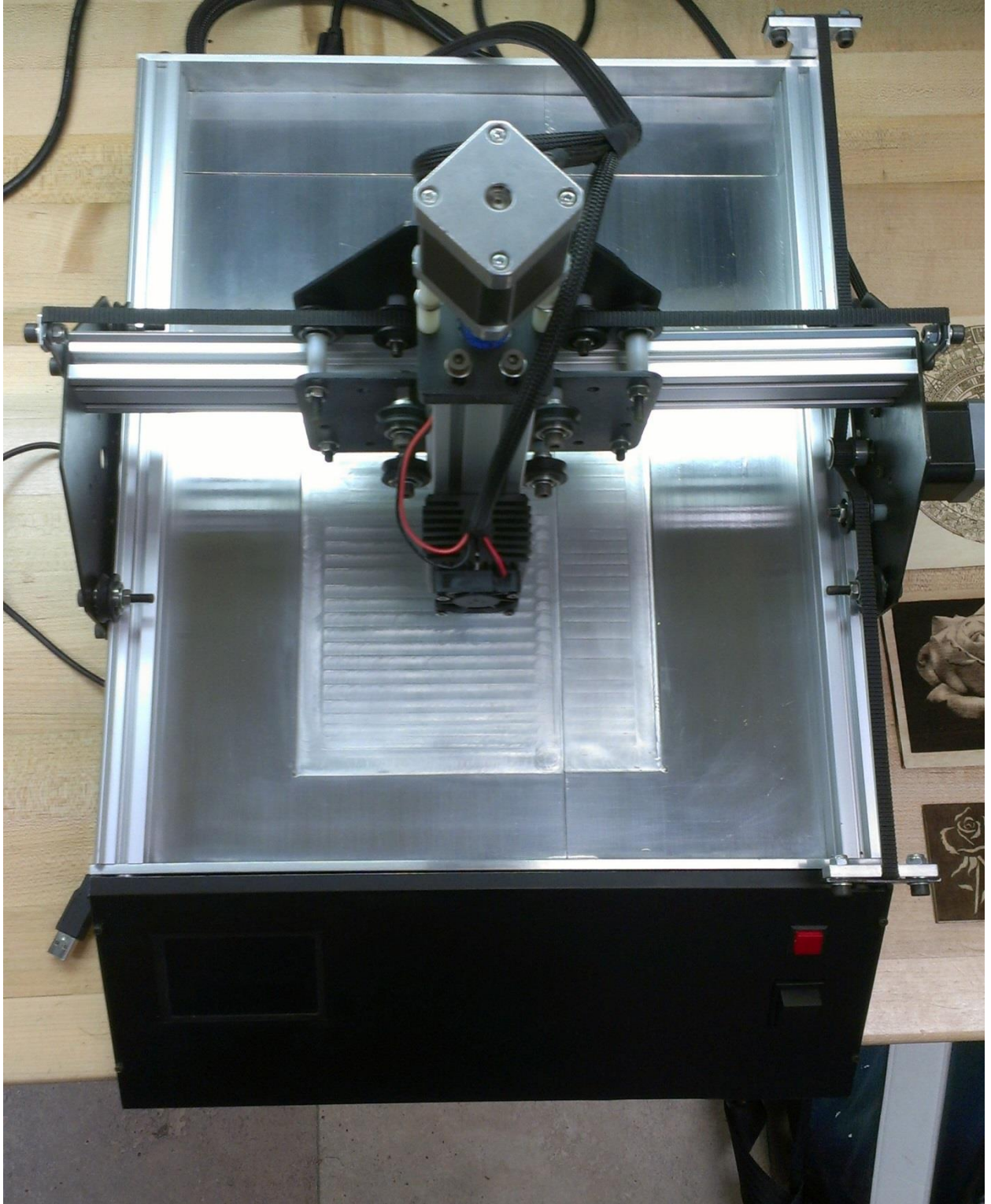


Fig. 63. Top View

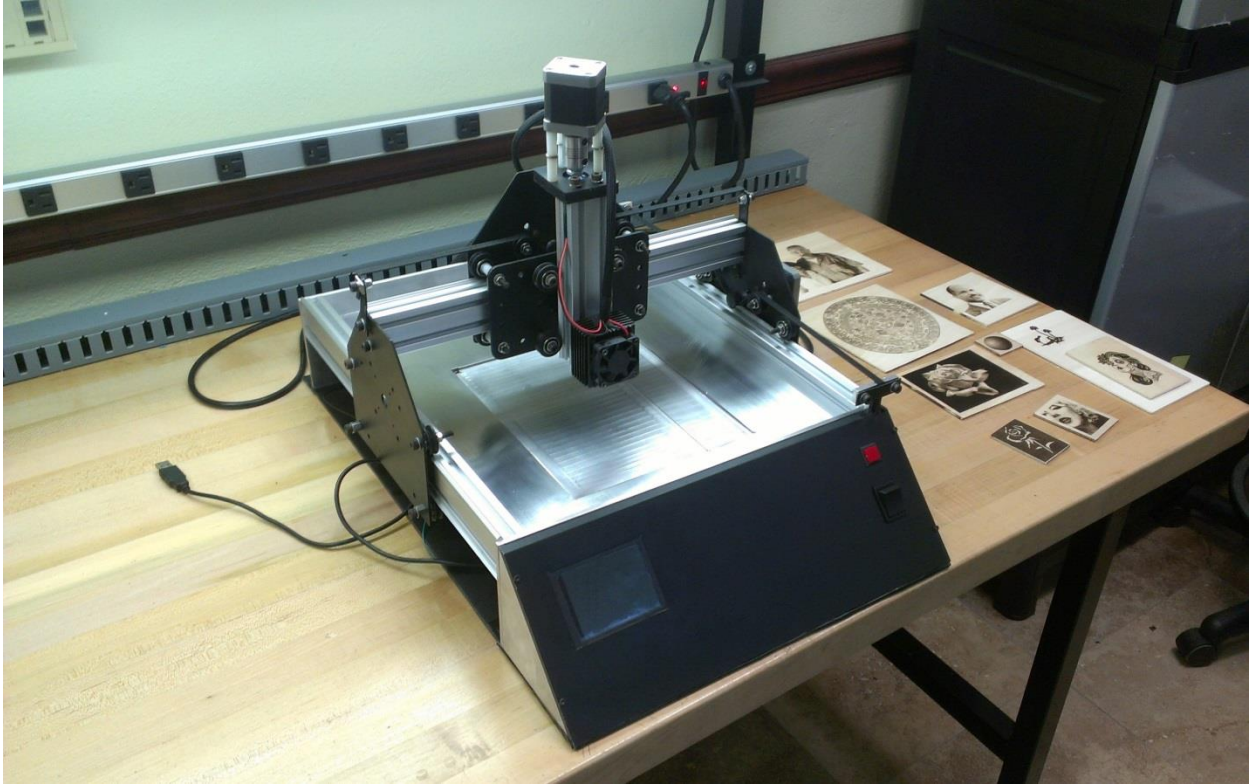


Fig. 64. Isometric view

6.2 ART CREATED BY PYROBOT



Fig. 65. Rose and sphere.

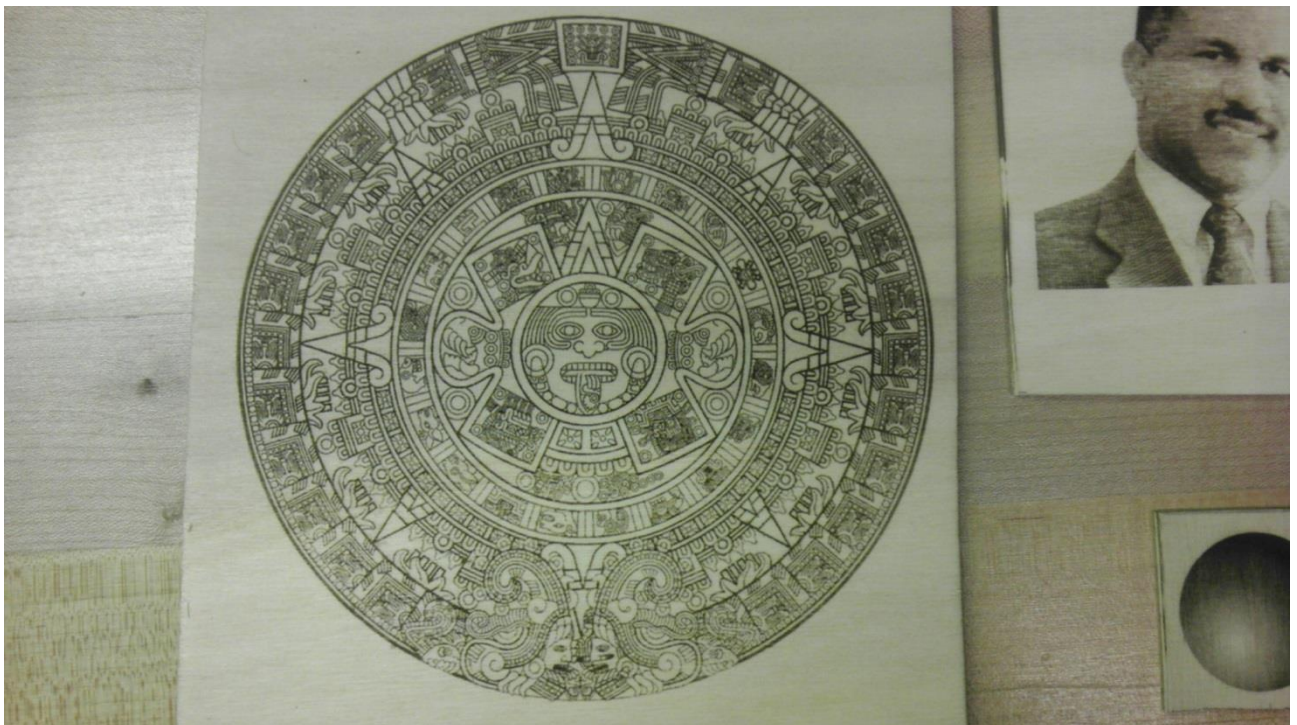


Fig. 66. Mayan calendar and our professor.



Fig. 67. Parents of one of the members.



Fig. 68. Marilyn Monroe and inverted rose.



Fig. 69. High contrast drawing of a woman.

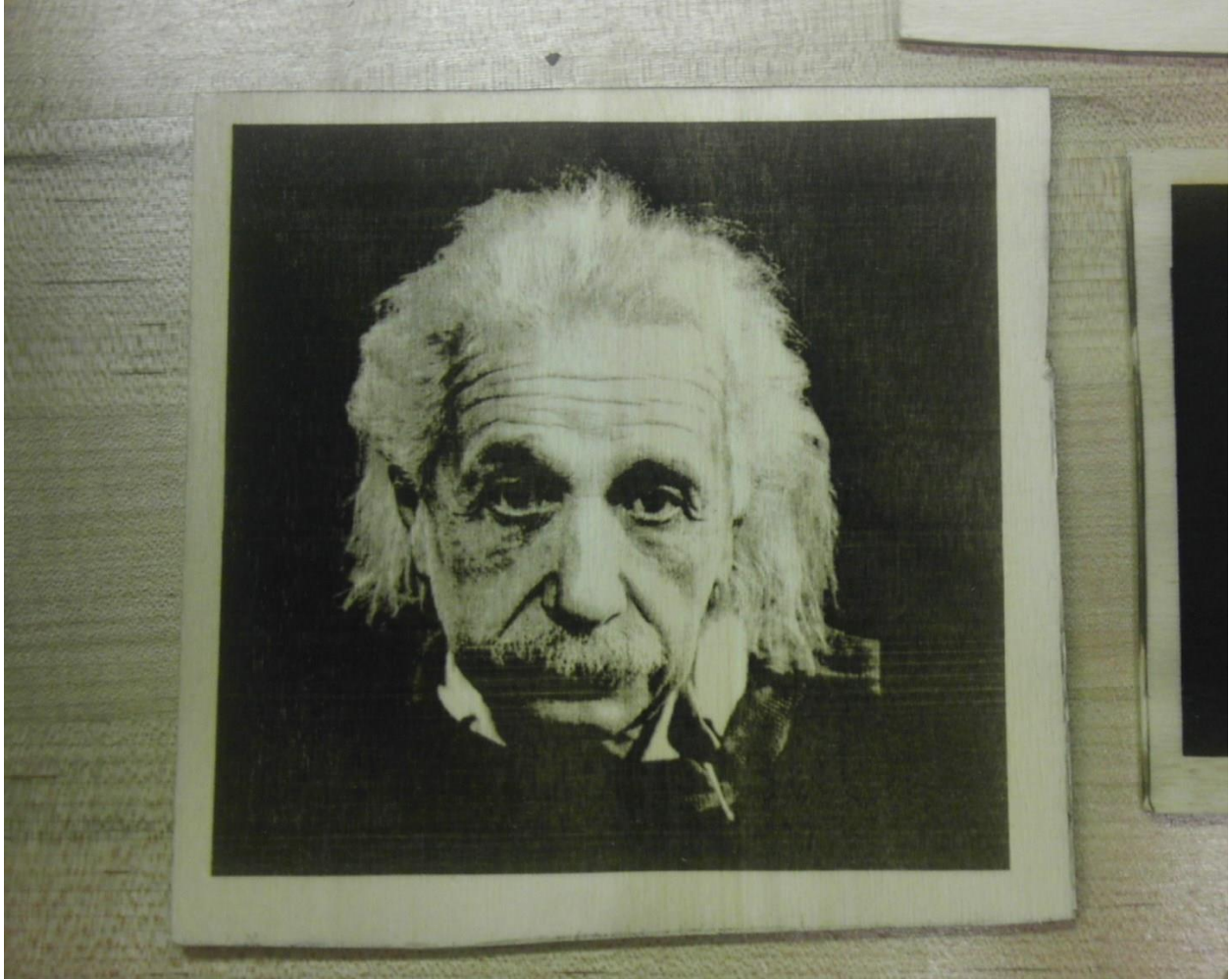


Fig. 70. Einstein.

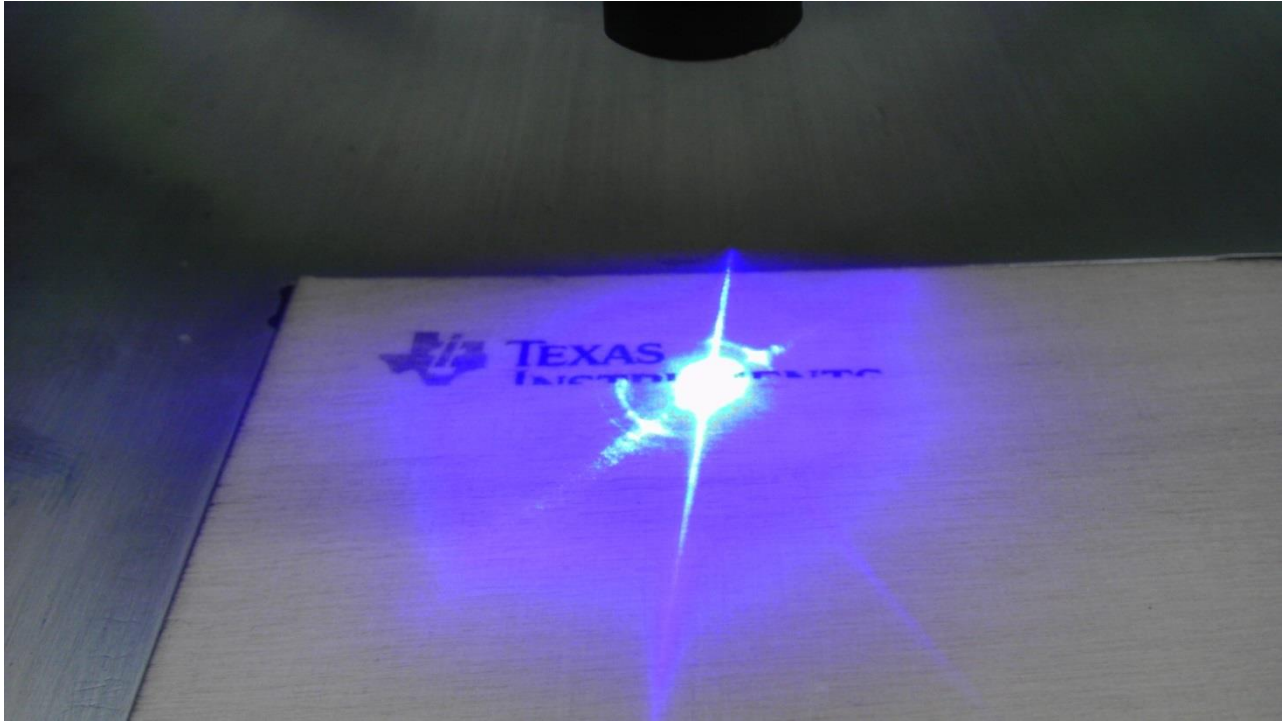


Fig. 71. PyroBot burning the TI logo.



Fig. 72. Result of TI Logo next to a quarter for comparison.



PyroBot art next to each other for comparison.



7 CONCLUSIONS

PyroBot is a machine that attempts to emulate the art of pyrography. It does this by burning an image onto a wooden surface using a laser. For this project, we had initially set out to create a machine capable of burning images in 8 bit color resolution, with a dpi of at least 100 and an a working area of 8" by 6". What was achieved was a machine with a working dpi of 141 and a theoretical maximum dpi of 282. Its working area can reach a maximum size of 9" by 7.2". This value is limited only by the system's on board SRAM; the structure itself can cover areas of greater size. Even though image resolution is currently capped at 5 bits, it can be seen that this color gradient is sufficient to achieve highly detailed images. Further calibrating would allow the resolution to reach a maximum of 7 bits. The end result is a fully fledged laser burning machine, at a price 5 times smaller than its nearest competitor.

8 FUTURE WORK

PyroBot has room for improvement both software and hardware wise. Image uploading could be improved through the use of an SD card. Instead of depending on a host computer, it could be possible find the image on the card via PyroBot's LCD touch screen. This enhancement would turn PyroBot into a completely standalone system. Enhancements can also be made to the precision of the grayscale, increasing it from 32 shades of grey to 256. With finer tuning of the laser timing and laser duty cycle it could be accomplishable. Better laser calibration could lead to doubling the resolution from 141dpi to 282dpi. Increasing the resolution would also increase the maximum image size that can be burned. An auto calibration method for the laser that constantly adjusts the Z-axis while burning an image would greatly increase the precision. It would ensure that the distance between the wood and the laser is always optimal, regardless of the wood thickness.

The main hardware component that can be improved is the laser. By upgrading the laser to a more powerful one (>2W) the burning time could potentially be reduced. This would require small modifications to the laser driver for it to power the larger laser. If an auto calibration method for the Z-axis is implemented, the three stepper motors would be ON at the same time, requiring a power supply with a more powerful 12V rail.

Finally, the base could be extended to accept pieces of wood larger than 9" x 7.2". If the system's resolution is increased as well, PyroBot could then burn larger images that look sharper than the original ones.

9 WORKS CITED

- [1] S. Walters, *Pyrography Workbook. A Complete Guide to the Art of Woodburning*, East Petersburg, EA: Fox Chapel Publishing, 2005.
- [2] "computer numerical control," in *McGraw-Hill Dictionary of Scientific & Technical Terms, 6E*, The McGraw-Hill Companies, Inc, 2003.
- [3] Focus Technology Co., Ltd., "Wuhan Aohua Photoelectric Parts Co., Ltd.," 2012. [Online]. Available: <http://whahlaser.en.made-in-china.com/product/fqAxlZrOUYs/China-Metal-Laser-Cutting-Machine-XY-YAG-500-.html>. [Accessed 30 April 2012].
- [4] Epilog Laser, Inc., "Epilog Laser," 2012. [Online]. Available: http://www.epiloglaser.com/legend_helix.htm. [Accessed 30 April 2012].
- [5] E. Ford, "About," 2012. [Online]. Available: www.shapeoko.com. [Accessed 26 04 2012].
- [6] B. Dring. [Online]. Available: <http://www.makerslide.com/>. [Accessed 29 April 2012].
- [7] Nichia Co, "NDB7875 Blue Laser Diode," Nichia Co, 12 09 2011. [Online]. Available: <http://www.nichia.co.jp/specification/en/product/ld/NDB7875-E.pdf>. [Accessed 30 4 2012].
- [8] RepairFaq, "A 1 MHz Analog Laser Modulator Based On LM317 Regulator," [Online]. Available: <http://www.repairfaq.org/sam/laserdps.htm#dpsldd317>. [Accessed 30 4 2012].
- [9] Texas Instruments, "LM138/LM338 5-Amp Adjustable Regulators Data Sheet," Texas Instruments, 5 1998. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm338.pdf>. [Accessed 30 4 2012].
- [10] Linear Technology, "LT1122 Fast Settling, JFET Input Operational Amplifier Data Sheet," Linear Technology, 1994. [Online]. Available: <http://cds.linear.com/docs/Datasheet/lt1122.pdf>. [Accessed 30 4 2012].
- [11] Texas Instruments, "SN74LVC1G17 SINGLE SCHMITT-TRIGGER BUFFER Data Sheet," Texas Instruments, 6 2011. [Online]. Available: <http://www.ti.com/lit/ds/symlink/sn74lvc1g17.pdf>. [Accessed

30 4 2012].

- [12] Aavid Thermalloy, "Aavid Board Level Heatsinks Catalog," Aavid Thermalloy, [Online]. Available: <https://www.aavidthermalloy.com/sites/default/files/literature/Aavid-Board-Level-Heatsinks-Catalog.pdf#page=59>. [Accessed 2012 30 4].
- [13] "Stepper Motors," [Online]. Available: <http://www.engineersgarage.com/articles/stepper-motors>.
- [14] Sparkfun, "Sparkfun," [Online]. Available: <http://www.sparkfun.com/products/10846>.
- [15] G. Frangel, "Stepper Motor Theory of Operation," [Online]. Available: gyan.fragnel.ac.in/~dileep/stepperworking.pdf.
- [16] Modular Circuits, "H-Bridge Secrets," [Online]. Available: <http://modularcircuits.tantosonline.com/blog/articles/old-h-bridge-secrets/part-1/> .
- [17] T. Igoe. [Online]. Available: <http://www.tigoe.com/pcomp/code/circuits/motors/stepper-motors/>.
- [18] Texas Instruments, [Online]. Available: <http://www.ti.com/product/drv8825> .
- [19] Texas Instruments, "PowerPad Made Easy," [Online]. Available: www.ti.com/lit/an/slma004b/slma004b.pdf .
- [20] Texas Instruments, "DRV8825 Evaluation Module," [Online]. Available: <http://www.ti.com/litv/pdf/slvu361a>.
- [21] "µLCD-32PT(SGC)," 17 Novemeber 2011. [Online]. Available: <http://www.4dsystems.com.au/prod.php?id=113>. [Accessed 30 April 2012].
- [22] Kobitone, "254-35C1-ROX Piezoelectric Buzzer," Kobitone Audio Company, 12 6 2006. [Online]. Available: <http://www.mouser.com/catalog/specsheets/KT-400257.pdf>. [Accessed 30 4 2012].
- [23] Rohm Semi Conductor, "RPR-220 Reflective photosensor," Rohm Semi Conductor, 11 06 2010. [Online]. Available: <http://www.rohm.com/products/databook/s/pdf/rpr-220uc30n.pdf>. [Accessed 30 4 2012].

- [24] Texas Instruments, "DRV8825 Stepper Motor Controller IC," Texas Instruments, 09 2011. [Online]. Available: <http://www.ti.com/lit/ds/symlink/drv8825.pdf>. [Accessed 30 4 2012].
- [25] "µLCD-32PT(SGC) 3.2" Serial LCD Display Module Data Sheet," 4d Systems, 16 12 2011. [Online]. Available: [http://www.4dsystems.com.au/downloads/Serial-Display-Modules/uLCD-32PT\(SGC\)/Docs/uLCD-32PT-SGC-DS-rev5.pdf](http://www.4dsystems.com.au/downloads/Serial-Display-Modules/uLCD-32PT(SGC)/Docs/uLCD-32PT-SGC-DS-rev5.pdf). [Accessed 30 4 2012].
- [26] Sunon, "Sunon MB40100V2 Data Sheet," Sunon Co, [Online]. Available: <http://www.sunon.com.tw/products/pdf/DCFAN/ME4010.pdf>. [Accessed 30 4 2012].
- [27] Sunon, "Sunon MC20100V2 Data Sheet," Sunon Co, [Online]. Available: <http://www.sunon.com.tw/products/pdf/DCFAN/MC2010.pdf>. [Accessed 30 4 2012].
- [28] OMRON, "MOS FET Relays G3VM-352C/F," OMRON, 12 2010. [Online]. Available: [http://www.components.omron.com/components/web/pdflib.nsf/0/C50324DF15AB179585257201007DD63F/\\$file/G3VM_352C_F_1210.pdf](http://www.components.omron.com/components/web/pdflib.nsf/0/C50324DF15AB179585257201007DD63F/$file/G3VM_352C_F_1210.pdf). [Accessed 25 5 2012].
- [29] Texas Instruments, "LM34 Precision Fahrenheit Temperature Sensors," Texas Instruments, 9 2000. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm34.pdf>. [Accessed 30 4 2012].
- [30] STMicroelectronics, "STM32F103xE Data Sheet," STMicroelectronics, 4 2011. [Online]. Available: http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00191185.pdf. [Accessed 30 4 2012].
- [31] LeafLabs, "Maple Native β Docs," LeafLabs, 7 10 2011. [Online]. Available: <http://leafflabs.com/docs/hardware/maple-native-beta.html>. [Accessed 30 4 2012].
- [32] Texas Instruments, "LM10 Operational Amplifier and Voltage Reference," Texas Instruments, 8 2000. [Online]. Available: <http://www.ti.com/lit/ds/symlink/lm10.pdf>. [Accessed 30 4 2012].
- [33] Sunon, "MB50100V2 Data Sheet," Sunon , [Online]. Available: <http://www.sunon.com.tw/products/pdf/DCFAN/ME5010.pdf>. [Accessed 30 4 2012].
- [34] N2Power, N2Power A Qualstar Company, 2011. [Online]. Available: <http://www.n2power.com/700723.pdf>. [Accessed 30 4 2012].

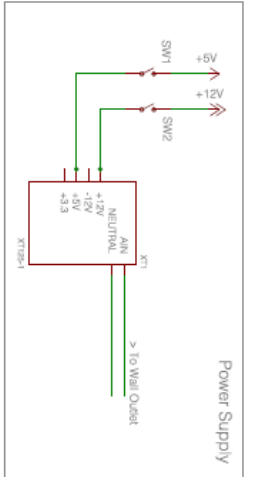
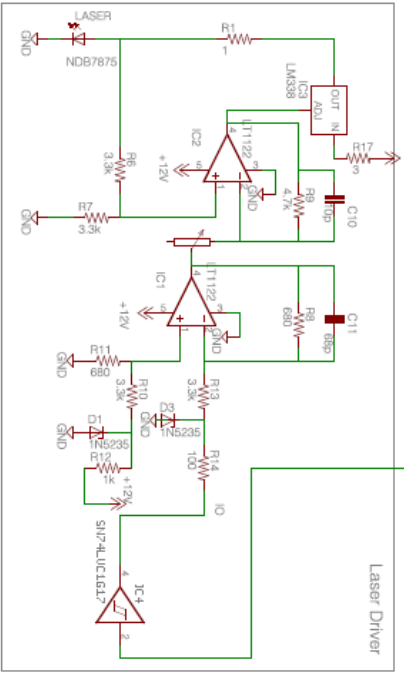
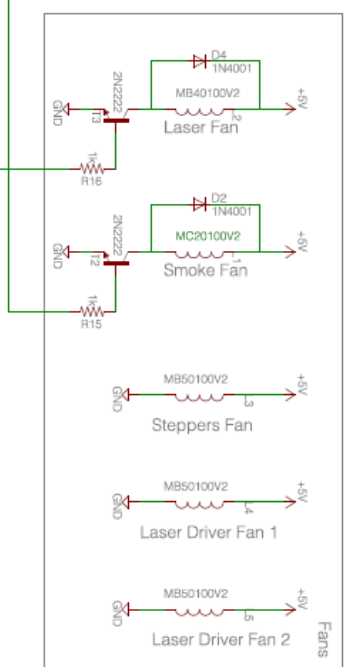
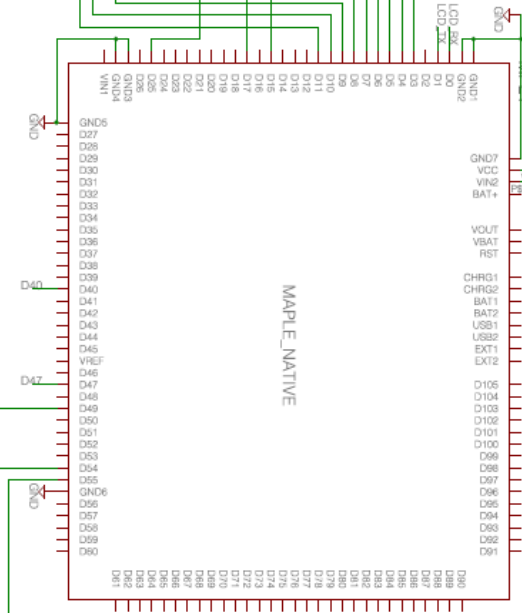
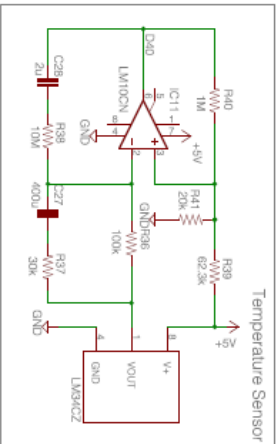
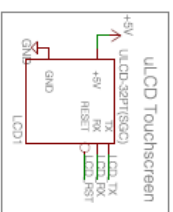
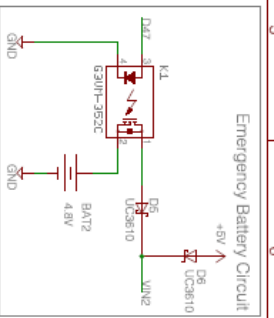
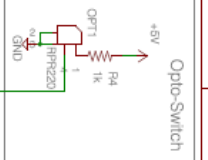
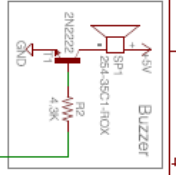
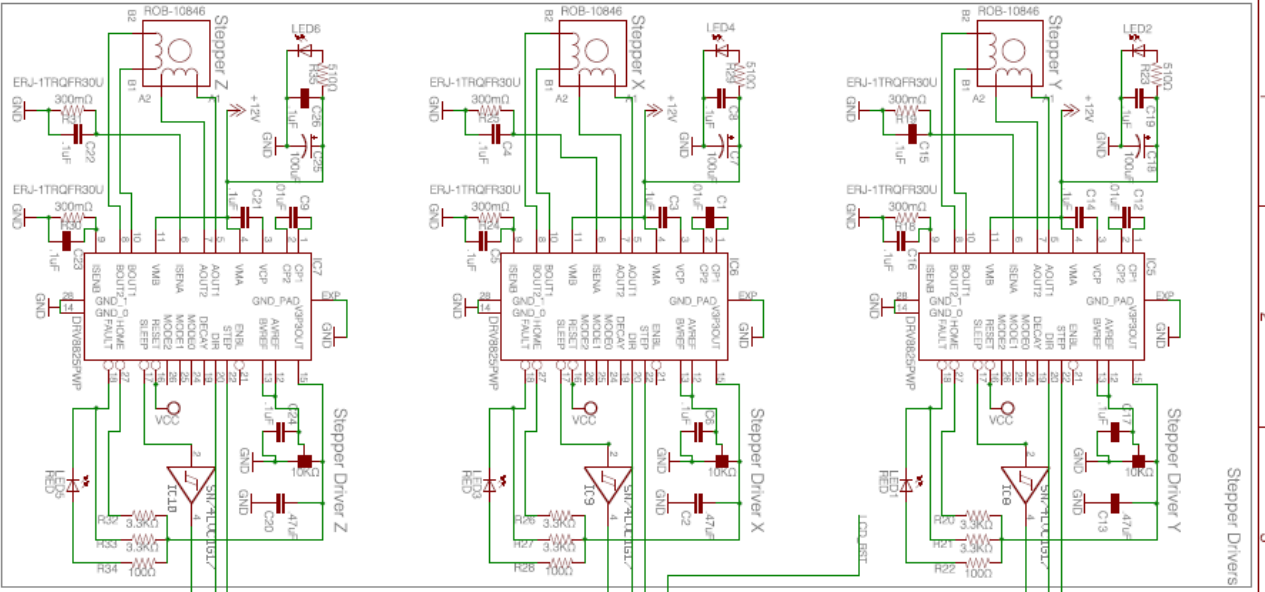
- [35] ZHEJIANG DONGYA ELECTRONIC CO., LTD., "JZC-11F-05VDC – 1Z Relay Data Sheet," ZHEJIANG DONGYA ELECTRONIC CO., LTD., 9 12 2009. [Online]. Available:
<http://www.sparkfun.com/datasheets/Components/General/JZC-11F-05VDC-1Z%20EN.pdf>.
[Accessed 30 4 2012].
- [36] 4dSystems, "PICASO-SGC Command Set," 4dSystems, 28 3 2012. [Online]. Available:
<http://www.4dsystems.com.au/downloads/Semiconductors/PICASO-SGC/Docs/PICASO-SGC-COMMANDS-SIS-rev11.pdf>. [Accessed 30 4 2012].
- [37] STMicroelectronics, "STMF10xx Flash Memory Microcontrollers," [Online]. Available:
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/PROGRAMMING_MANUAL/CD00283419.pdf.

10 APPENDIX

10.1 FULL SCHEMATIC

[Schematic on next page]

PYROBOT



| | |
|--------------------------|--------------|
| TITLE: PyroBot Schematic | PEUg |
| Document Number: | |
| Date: 5/1/12 11:11 AM | Sheet: 1/401 |

10.2 PART LIST AND COST ANALYSIS

| Laser Driver: | | | | |
|----------------------|----------------------|-----------|--------------------------|----------------|
| Part Number | Name | Quantity | Price | Total Price |
| 1N5235B-TR | Zener Diode | 2 | \$ 0.07 | \$ 0.14 |
| MF1/4DC1000F | Resistor 100Ω | 1 | \$ 0.06 | \$ 0.06 |
| MF1/4DC T52R3301F | Resistor 3.3kΩ | 5 | \$ 0.06 | \$ 0.30 |
| CFR25J680R | Resistor 680Ω | 2 | \$ 0.01 | \$ 0.02 |
| CFR25J1K0 | Resistor 1kΩ | 1 | \$ 0.01 | \$ 0.01 |
| | Resistor 4.7kΩ | 1 | \$ 0.21 | \$ 0.21 |
| | Resistor 1Ω | 1 | \$ 1.10 | \$ 1.10 |
| | Resistor 3Ω | 1 | \$ 0.85 | \$ 0.85 |
| LM338 | Regulator 5A | 1 | \$ 2.45 | \$ 2.45 |
| HS352-ND | LM338 Heatsink | 1 | \$ 1.25 | \$ 1.25 |
| 1122 | Opamp | 2 | \$ 6.63 | \$ 13.26 |
| | Potentiometer Max 5k | 1 | \$ 1.34 | \$ 1.34 |
| 296-11933-1-ND | Schmitt Trigger | 1 | \$ 0.23 | \$ 0.23 |
| Total: | | 20 | <input type="checkbox"/> | \$21.22 |

| Laser: | | | | |
|---------------|-----------------------|----------|----------|----------------|
| Part Number | Name | Quantity | Price | Total Price |
| NDB7875 | Blue Laser Diode 1.6W | 1 | \$ 60.00 | \$ 60.00 |
| -- | Laser Diode Heatsink | 1 | \$ 16.00 | \$ 22.50 |
| 259-1568-ND | Heatsink Fan | 1 | \$ 6.95 | \$ 6.95 |
| 259-1565-ND | Smoke Fan | 1 | \$ 3.48 | \$ 3.48 |
| Total: | | 2 | | \$89.45 |

| Step Motor: | | | | |
|--------------------|--|----------|----------|----------------|
| Part Number | Name | Quantity | Price | Total Price |
| ROB-10846 | Stepper Motor - 58 oz.in (400 steps/rev) | 3 | \$ 16.95 | \$ 50.85 |
| Total: | | 3 | | \$50.85 |

Step Motor Driver:

| Part Number | Name | Quantity | Price | Total Price |
|--------------------|----------------------------------|-----------|---------|----------------|
| DRV8825PWP | DRV8825 Stepper Motor Controller | 3 | \$ 7.25 | \$ 21.75 |
| EVM-3WSX80B14 | 10KΩ Trimpot | 3 | \$ 0.99 | \$ 2.97 |
| GRM216R71H103KA01D | .01uF 50V Capacitor | 3 | \$ 0.39 | \$ 1.17 |
| GRM216F51E104ZA01D | .1uF 16V Ceramic Capacitor | 12 | \$ 0.10 | \$ 1.20 |
| GRM216F51E474ZA01D | .47uF 6.3V Ceramic Capacitor | 3 | \$ 0.26 | \$ 0.78 |
| EEE-FT1C101AR | 100uF 16V Aluminum Capacitor | 3 | \$ 3.23 | \$ 9.69 |
| SML-LXT0805GW-TR | Green LED | 3 | \$ 0.43 | \$ 1.29 |
| SML-LXT0805SRW-TR | Red LED | 3 | \$ 0.38 | \$ 1.14 |
| ERJ-6GEYJ511V | 510Ω Resistor | 3 | \$ 0.10 | \$ 0.30 |
| ERJ-6GEYJ101V | 100Ω Resistor | 3 | \$ 0.10 | \$ 0.30 |
| ERJ-6GEYJ332V | 3.3KΩ Resistor | 6 | \$ 0.10 | \$ 0.60 |
| ERJ-1TRQFR30U | 300mΩ (2512) | 6 | \$ 1.05 | \$ 6.30 |
| | Driver Heatsinks | 5 | \$ 6.95 | \$ 34.75 |
| | PCB Fabrication | 3 | \$ 3.33 | \$ 9.99 |
| 296-11933-1-ND | Schmitt Trigger | 3 | \$ 0.23 | \$ 0.69 |
| Total: | | 33 | | \$27.09 |

Safety Hardware:

| Part Number | Name | Quantity | Price | Total Price |
|--------------------|---------------------------|-----------|---------|----------------|
| 254-35C1-ROX | Piezoelectric Buzzer | 1 | \$ 3.53 | \$ 3.53 |
| RMCF0603FT4K30 | 4.3k ohm Resistor | 1 | \$ 0.04 | \$ 0.04 |
| 2N2222 | NPN transistor | 1 | \$ 1.07 | \$ 1.07 |
| QRD1114 | Optical Sensor | 1 | \$ 1.13 | \$ 1.13 |
| 150-01020 | 1k ohm Resistor | 2 | \$ 0.10 | \$ 0.20 |
| LM34CZ | Temperature Sensor | 1 | \$ 6.60 | \$ 6.60 |
| Lm10CN | Opamp & Voltage Reference | 1 | \$ 3.16 | \$ 3.16 |
| MRS25000C2002FRP00 | 20k ohm resistor | 1 | \$ 0.33 | \$ 0.33 |
| MF1/4DCT52A3002F | 30k ohm resistor | 1 | \$ 0.06 | \$ 0.06 |
| PV37W104C01B00 | 100k ohm potentiometer | 1 | \$ 2.26 | \$ 2.26 |
| MF1/4DC1003F | 100k ohm resistor | 1 | \$ 0.06 | \$ 0.06 |
| OD105JE | 1M ohm resistor | 1 | \$ 0.42 | \$ 0.42 |
| 291-10M-RC | 10M ohm resistor | 1 | \$ 0.10 | \$ 0.10 |
| MKP1848520454K2 | 2uF capacitor | 1 | \$ 2.18 | \$ 2.18 |
| TVA1208.7 | 400uF capacitor | 1 | \$ 3.56 | \$ 3.56 |
| Total: | | 16 | | \$24.70 |

Interface with User

| Part Number | Name | Quantity | Price | Total Price |
|------------------|---|----------|----------|----------------|
| μLCD-32PT(SGC) | 3.2" Serial LCD-TFT Display Module | 1 | \$ 79.99 | \$ 79.99 |
| PRT-09385 | Jumper Wires Premium 12" M/F Pack of 10 | 1 | \$ 4.50 | \$ 4.50 |
| PRCSA1-20L-BB000 | Rocker Switch | 1 | \$ 3.58 | \$ 3.58 |
| UB15KKW015C-CC | Emergency Stop Button | 1 | \$ 10.10 | \$ 10.10 |
| Total: | | 2 | | \$98.17 |

Power Management:

| Part Number | Name | Quantity | Price | Total Price |
|---------------|--------------------------------|-----------|----------|----------------|
| XL125-1 | Power Supply | 1 | \$ 17.95 | \$ 17.95 |
| 09-50-8031 | Molex connector | 1 | \$ 0.40 | \$ 0.40 |
| 09-50-8141 | Molex connector | 1 | \$ 1.22 | \$ 1.22 |
| 22-01-3037 | Molex connector | 1 | \$ 0.44 | \$ 0.44 |
| 08-52-0123 | Crimp Terminal | 3 | \$ 0.34 | \$ 1.02 |
| 08-50-0106 | Crimp Terminal | 17 | \$ 0.29 | \$ 4.93 |
| 2N2222 | NPN transistor | 1 | \$ 1.07 | \$ 1.07 |
| | Rechargeable NiMH Battery Pack | 1 | \$ 6.49 | \$ 6.49 |
| G3VM-352C | MOS FET Relay | 1 | \$ 1.95 | \$ 1.95 |
| UC3610 | Dual Shottcky Diode Bridge | 1 | \$ 6.16 | \$ 6.16 |
| 150-01020 | 1k ohm resistor | 1 | \$ 0.20 | \$ 0.20 |
| Total: | | 27 | | \$41.83 |

System Structure:

| Part Number | Name | Quantity | Price | Total Price |
|---------------|--------------------------------|------------|-----------|-----------------|
| -- | MakerSlide Kit | 1 | \$ 120.00 | \$120.00 |
| 259-1558-ND | Structural Fans | 3 | \$ 3.78 | \$ 11.34 |
| RSN-10/04-M | 1/4" Nylon Spacer | 6 | \$ 0.02 | \$ 0.12 |
| B000FPBU9K | 3/4" Nylon Spacer | 10 | \$ 0.06 | \$ 0.60 |
| ME058-5 | M5 Flat Washer | 90 | \$ 0.01 | \$ 0.90 |
| 90592A012 | M5 Hex Nut | 20 | \$ 0.02 | \$ 0.40 |
| 99055A125 | 200mm Threaded Rod | 1 | \$ 0.86 | \$ 0.86 |
| 91290A137 | M3 x 50mm SHCS | 3 | \$ 0.20 | \$ 0.60 |
| 91290A222 | M5 x 8mm SHCS | 22 | \$ 0.06 | \$ 1.32 |
| 91290A254 | M5 x 30mm SHCS | 16 | \$ 0.09 | \$ 1.44 |
| 91290A264 | M5 x 55mm SHCS | 4 | \$ 0.22 | \$ 0.88 |
| 90695A040 | M8 Thin Hex Nut | 2 | \$ 0.05 | \$ 0.10 |
| 91290A232 | M5 x 16mm SHCS | 2 | \$ 0.08 | \$ 0.16 |
| 1394A31 | Belt Anchors | 4 | \$ 0.20 | \$ 0.80 |
| HNKK5 | Insertion Nuts | 2 | \$ 0.16 | \$ 0.32 |
| 6412K11 | M5 - M8 Coupler | 1 | \$ 3.89 | \$ 3.89 |
| SMZ002D | Delrin Nut | 1 | \$ 5.00 | \$ 5.00 |
| SMS001 | Motor Mount Plates | 3 | \$ 5.00 | \$ 15.00 |
| SMZ003P | Z Mount Plate | 1 | \$ 5.00 | \$ 5.00 |
| -- | System Structure Custom Plates | 1 | \$ 200.00 | \$200.00 |
| Total: | | 193 | | \$368.00 |

Totals:

| | |
|------------------|-----------------|
| Quantity: | 296 |
| Price: | \$721.31 |

10.3 USERS GUIDE

The user guide is divided into the host guide and the system guide.

10.3.1 HOST

Precondition: PyroBot is connected to the host computer.

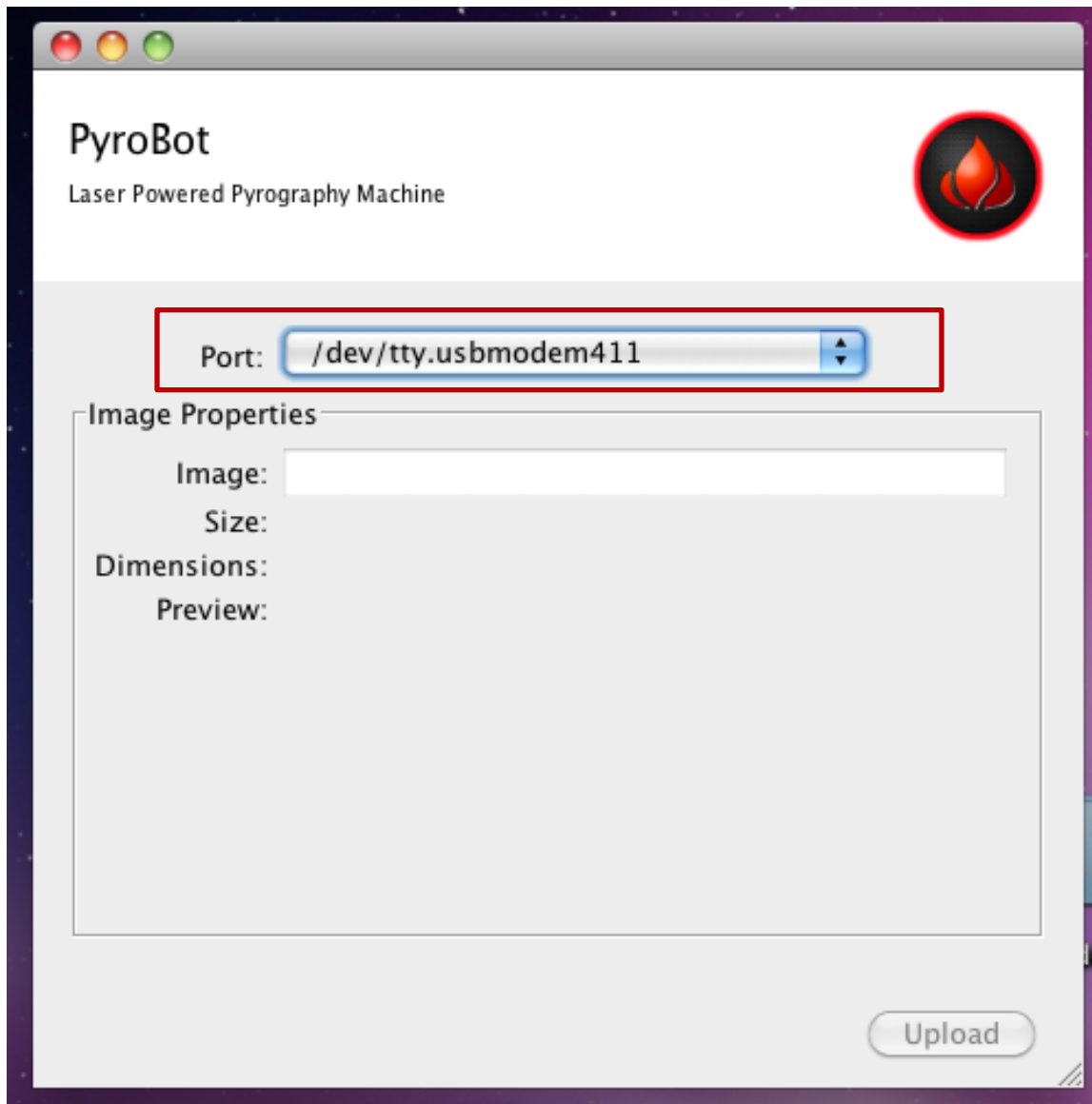
10.3.1.1 START APPLICATION

To start, locate the application jar file on your computer. Click on it to start the application.

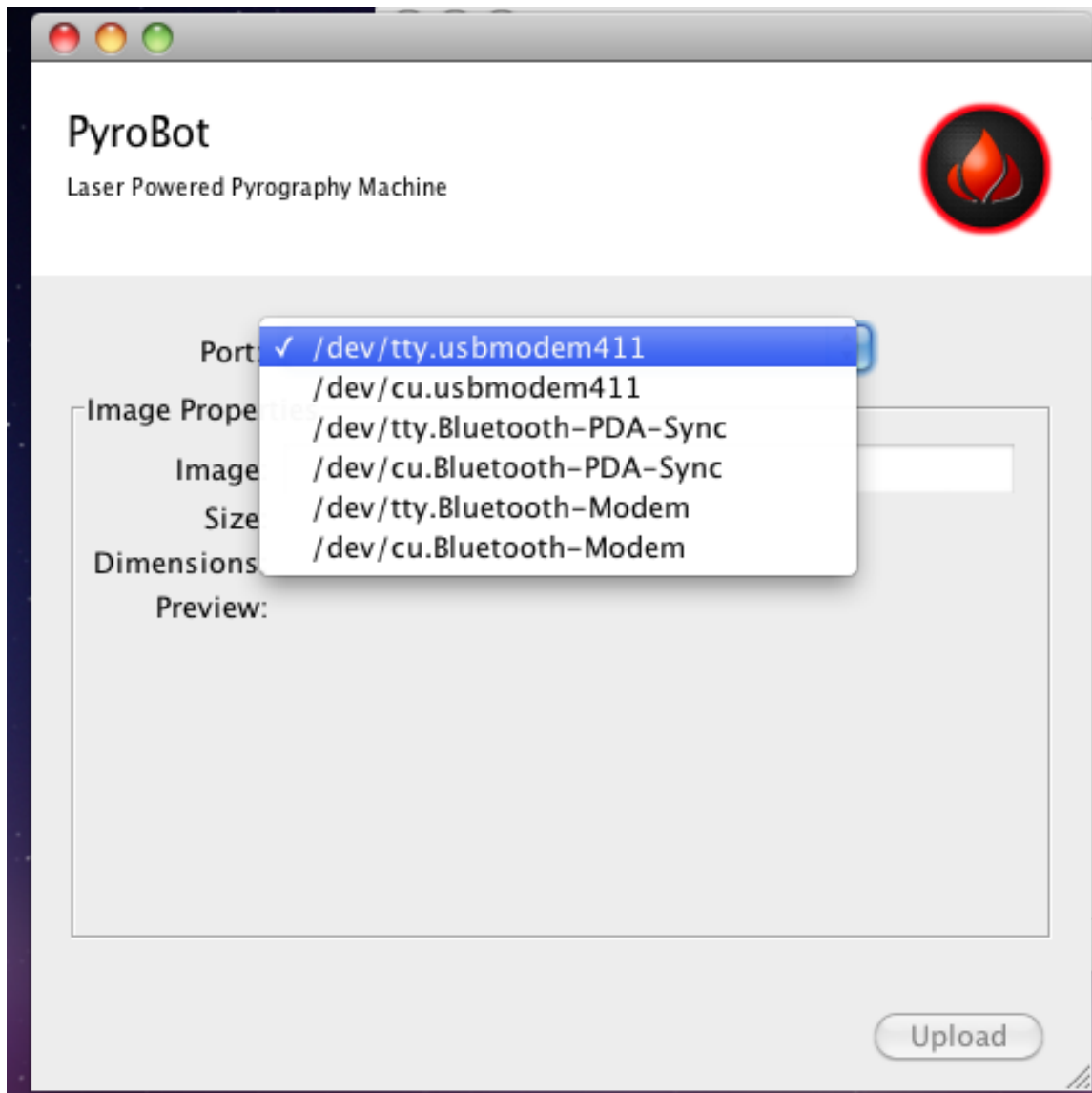


10.3.1.2 CONNECT TO SERIAL PORT

After launching the application, a screen similar to the following should appear. On startup, the application will preload the Port Dropdown Box with the names of all available serial ports. To begin interacting with the system, click on the Dropdown to see which ports are available.

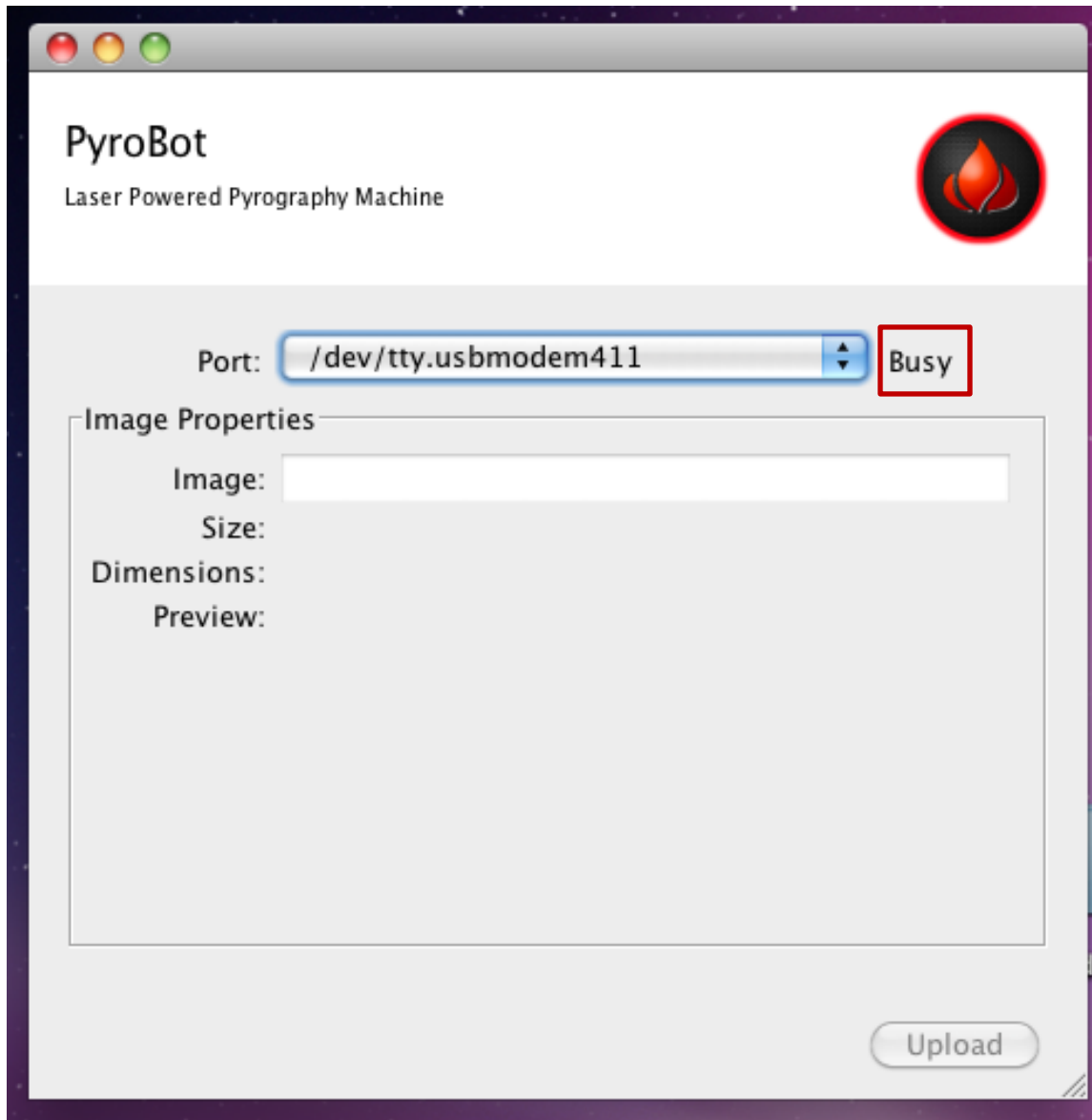


Clicking on the Dropdown will result in all available ports being displayed. Select the one that corresponds to the port PyroBot is connected to. If you are unsure as to which one it is, refer to the "Finding out which port PyroBot is connected to" section below.



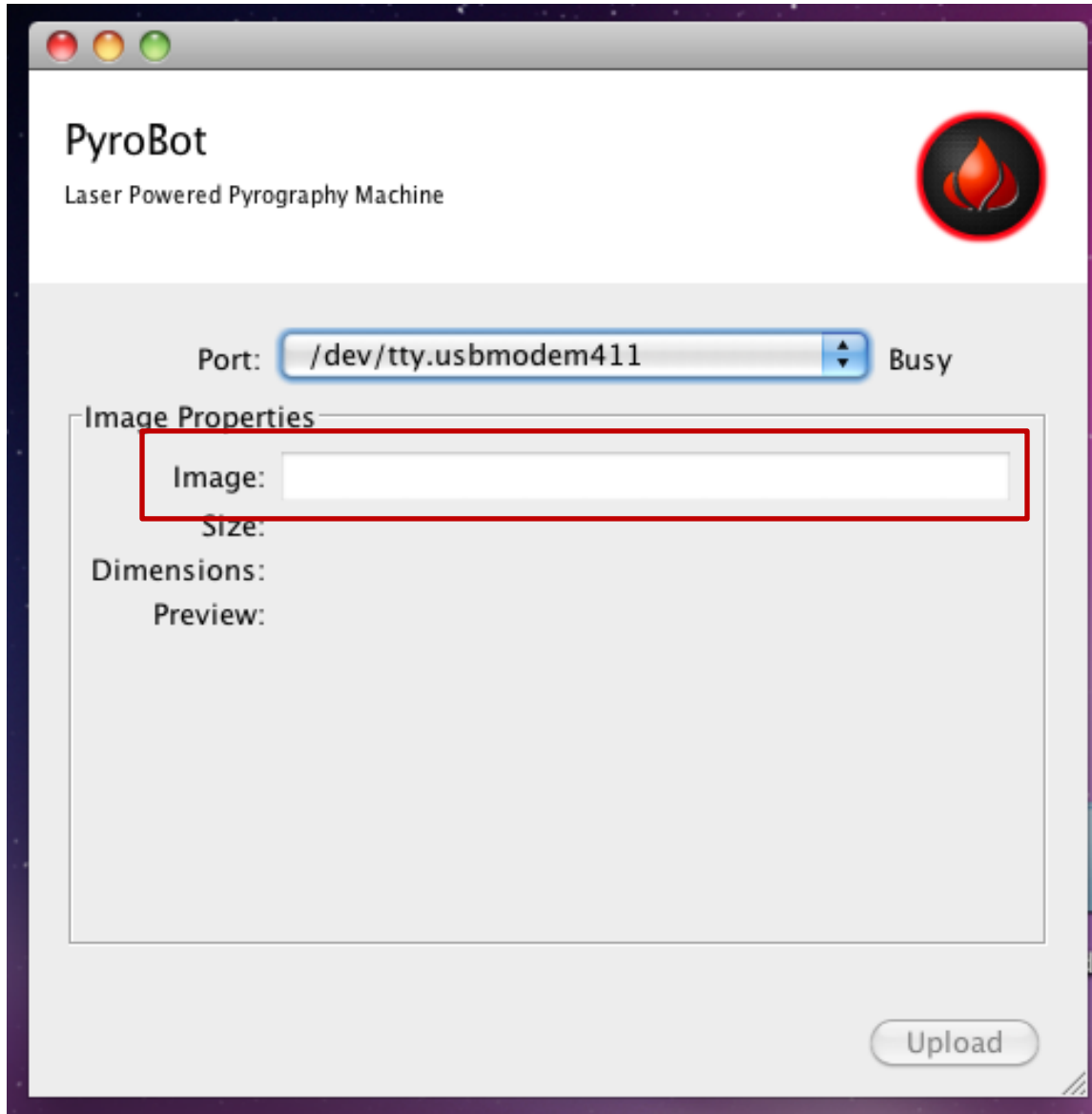
After a port is selected, wait while the application tries to communicate with the system. After about two seconds, a label should appear next to the Dropdown indicating the system's state.

Right now, the label should read "Busy".

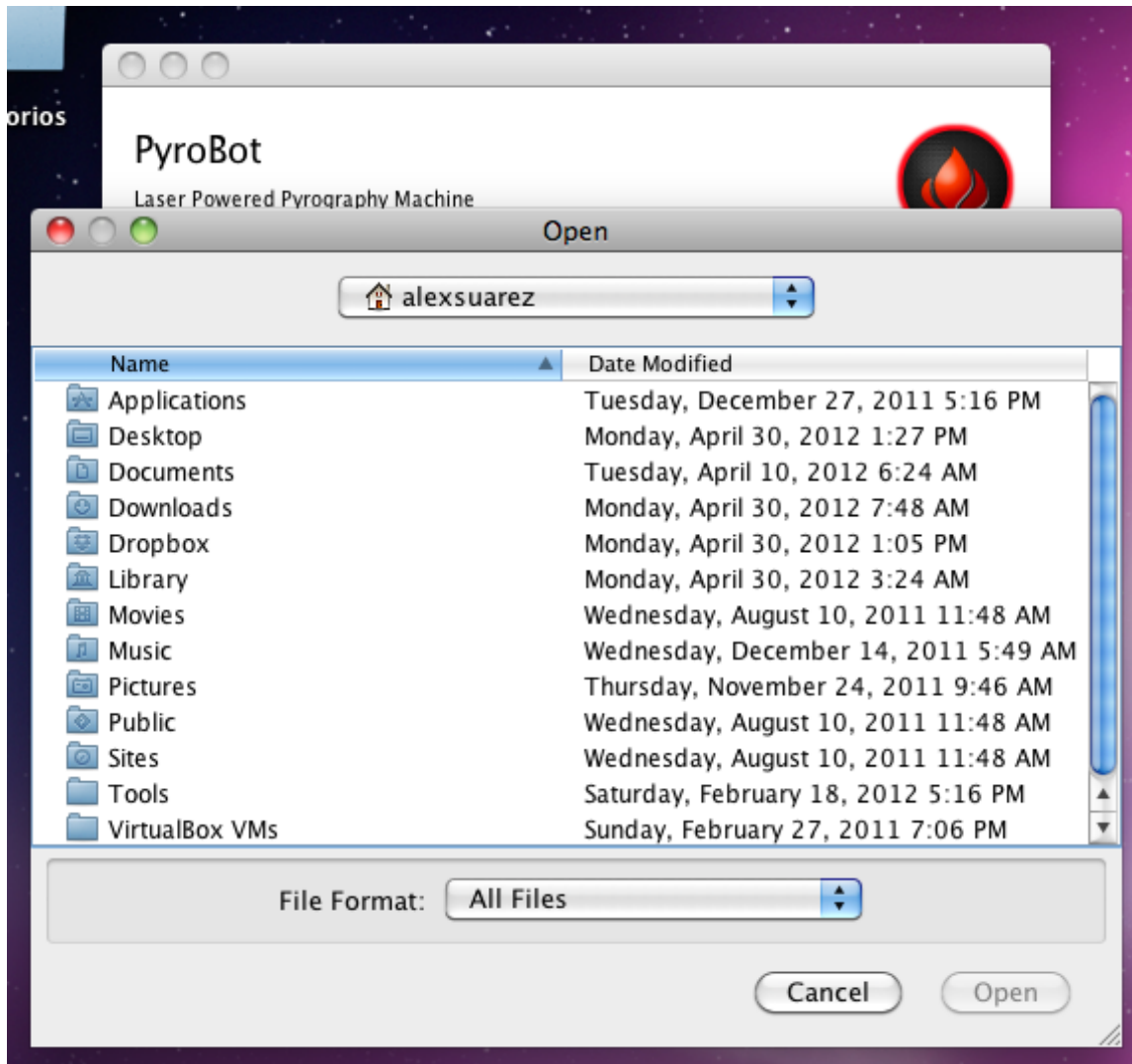


10.3.1.3 SELECTING AN IMAGE

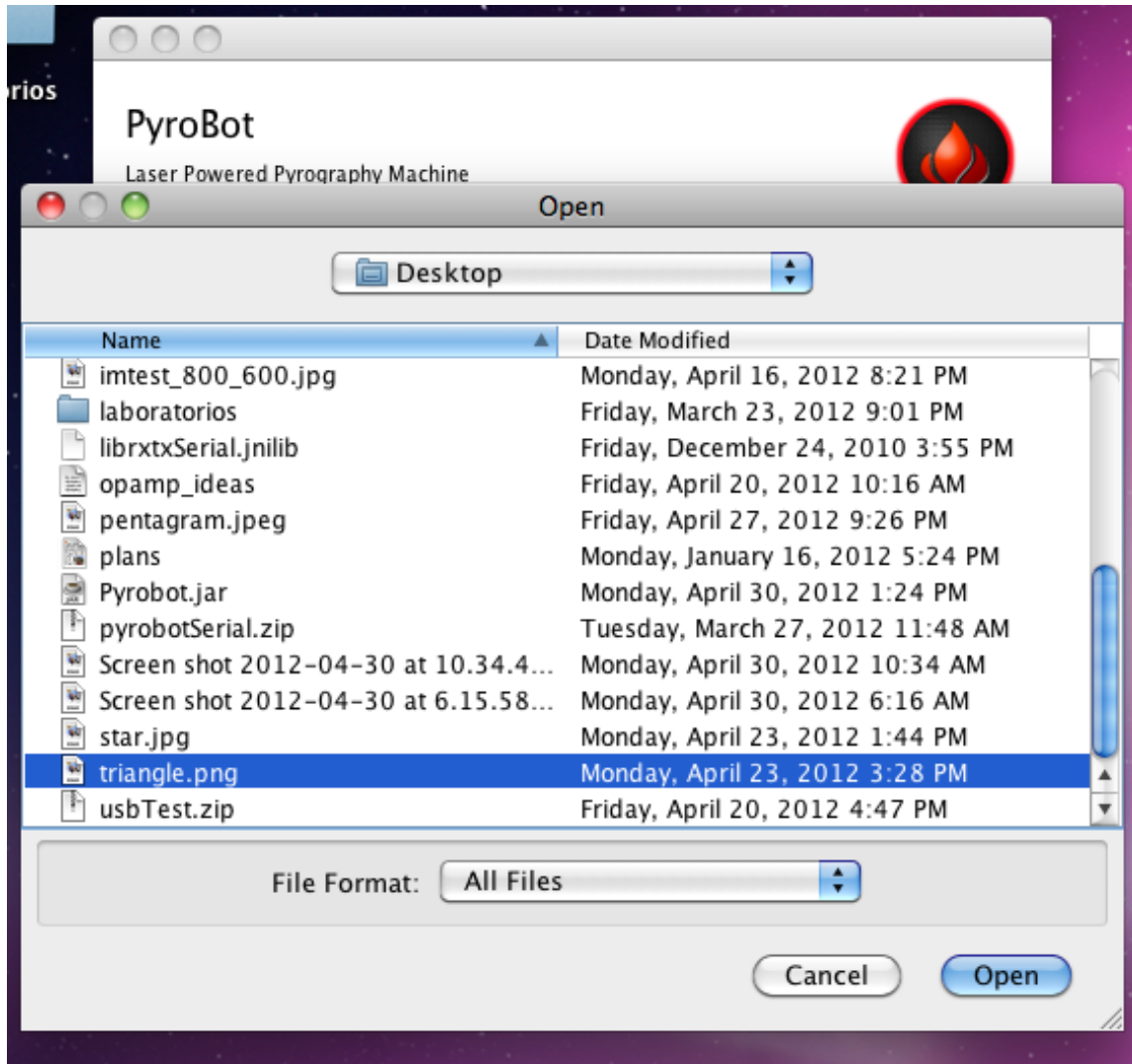
To select an image to burn, begin by clicking on the Textbox next to the word "Image:"



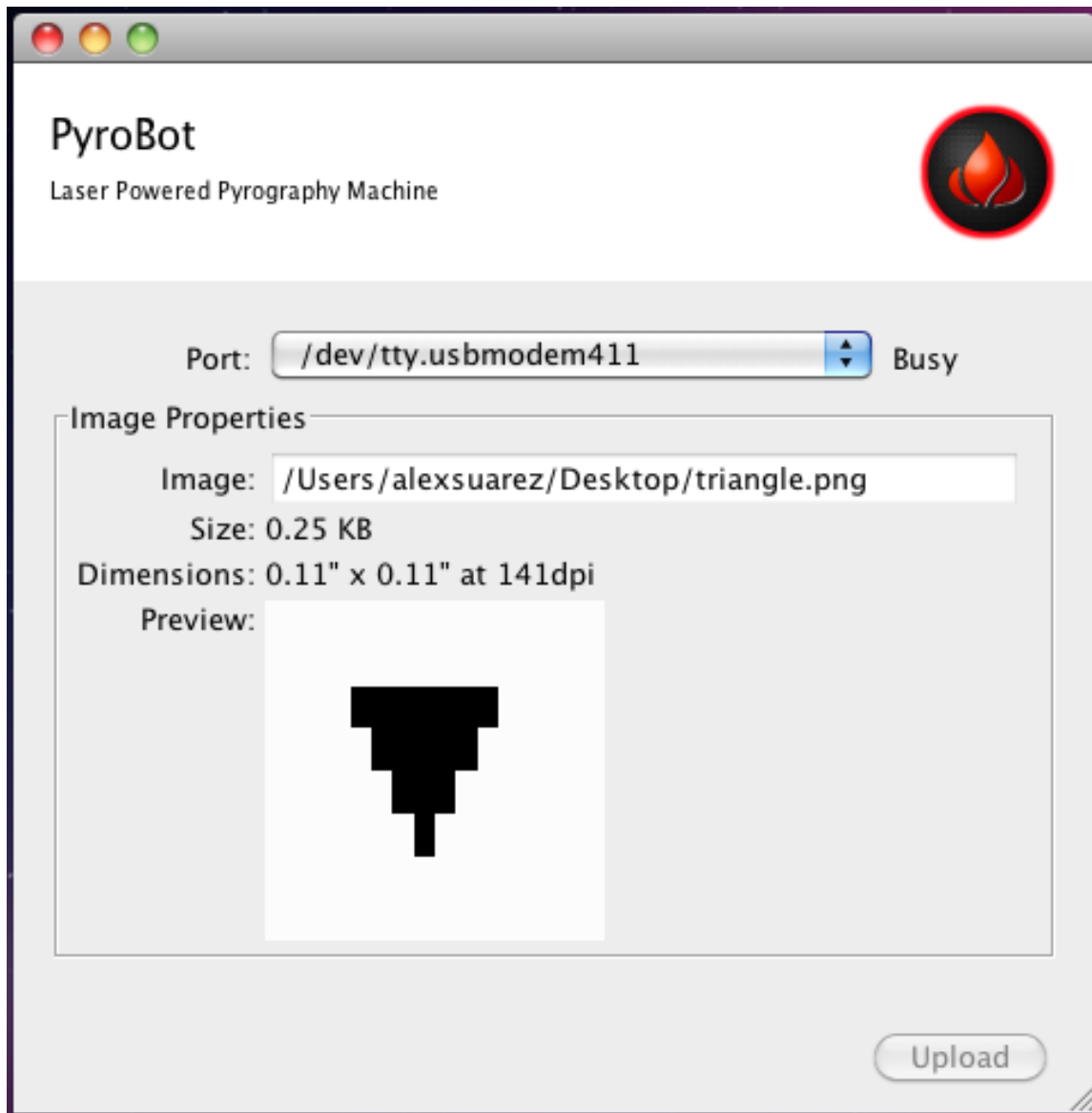
Upon clicking the Textbox, a new window should display. Use this window to browse your computer and find the image you would like to print.



Once you've found an image, highlight it and click "Open" to select it. Be sure that the image you select is of type JPG, PNG, GIF, BMP or WBMP.



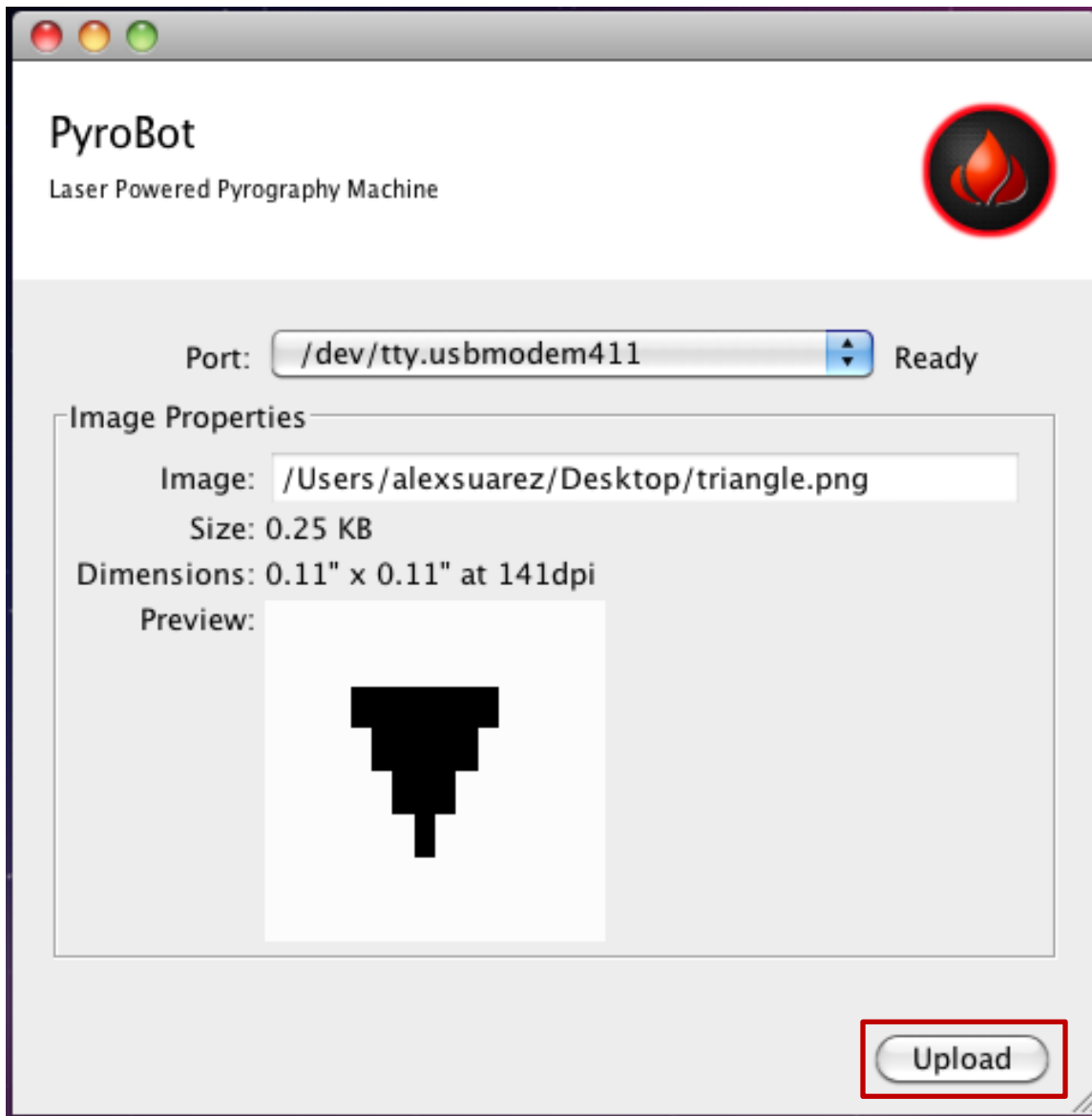
After selecting the image, the application will display onscreen the selected image along with its location on your computer, its size* and the dimensions (in inches) of the image if it were to be printed on a wooden plate at the current DPI*.



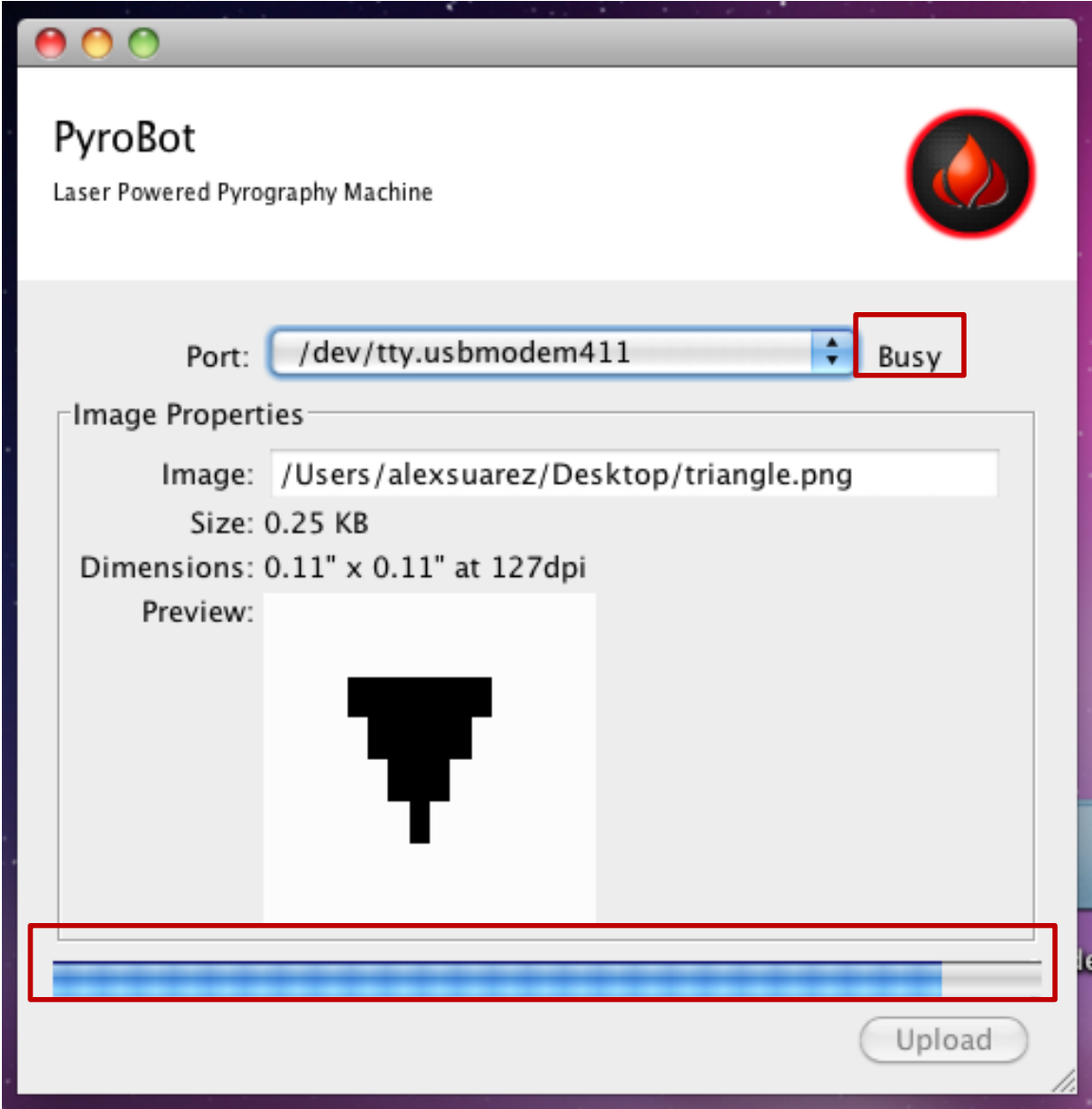
* The size presented by the application may not necessarily coincide with the size your computer reports; it is no cause for alarm.

10.3.1.4 UPLOADING AN IMAGE

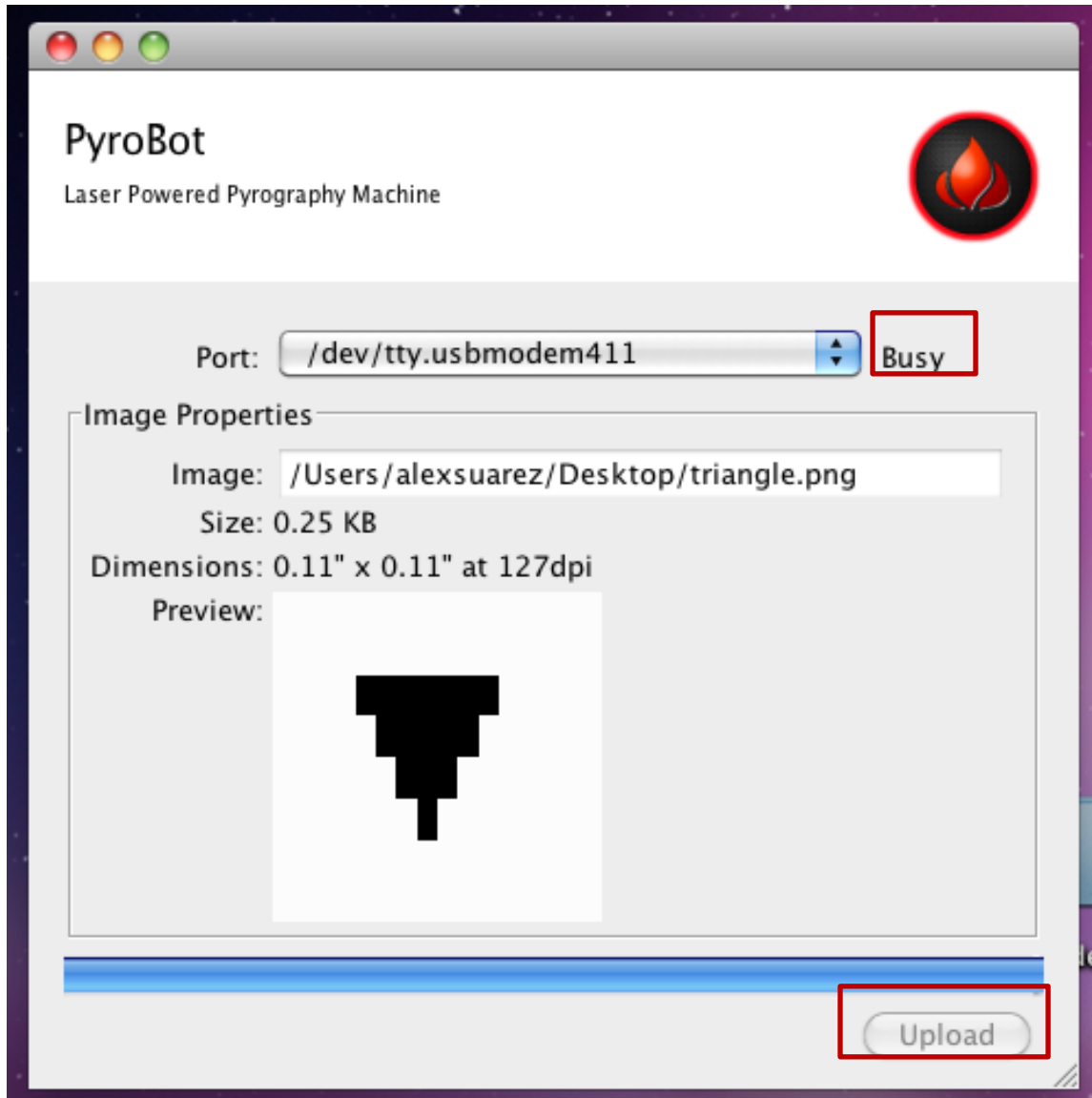
Once the application has established a connection with the system, a serial port has been opened and an image selected, the application will enable the upload button automatically. To begin uploading the image, simply click on the Upload Button.



A progress bar indicating completion percentage will display while the upload is happening. While this happens, notice that both the Upload Button is disabled and that the system state label reads 'Busy'.

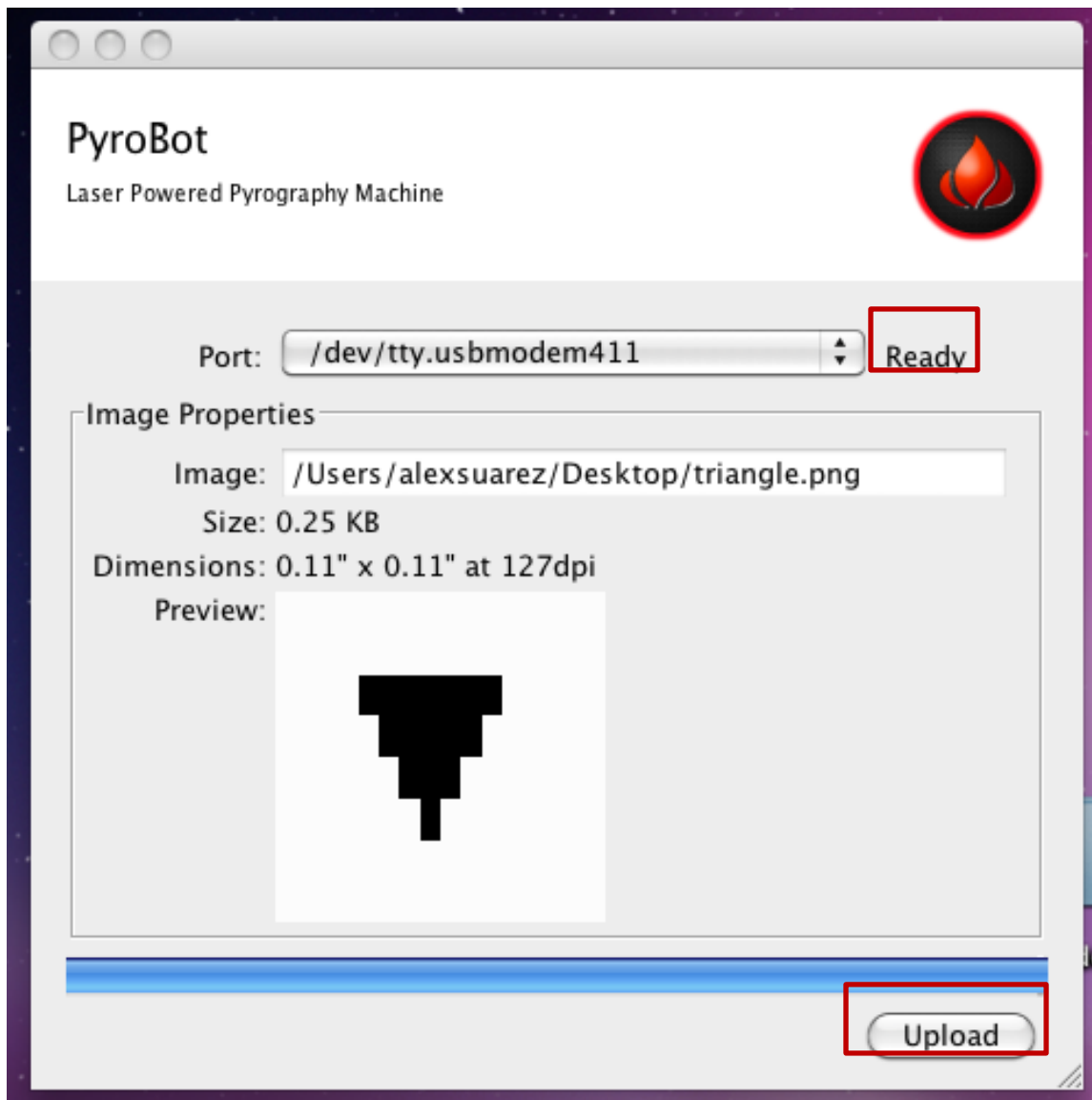


On upload completion, the upload bar will appear filled. Notice that the state label continues to read 'Busy' and that the Upload button is still disabled; this is normal. To upload a new image, you will need to select 'Upload' from PyroBot's UI. Operating PyroBot is covered in another section.



10.3.1.5 UPLOADING A DIFFERENT IMAGE

As was mentioned in the previous section, you need to first select 'Upload' from PyroBot's UI before being able to upload a different image. If the application is able to upload a new image, the state label will change from 'Busy' to 'Ready' and the Upload Button will be enabled. To upload a different image, follow the same steps as before.



10.3.1.6 FINDING OUT WHICH PORT PYROBOT IS CONNECTED TO

On connection, your computer will recognize PyroBot as a serial device. In order to find out which port the system is connected to, you can do the following:

10.3.1.7 ON WINDOWS 7

Open the Device Manager

Go to Start > Control Panel

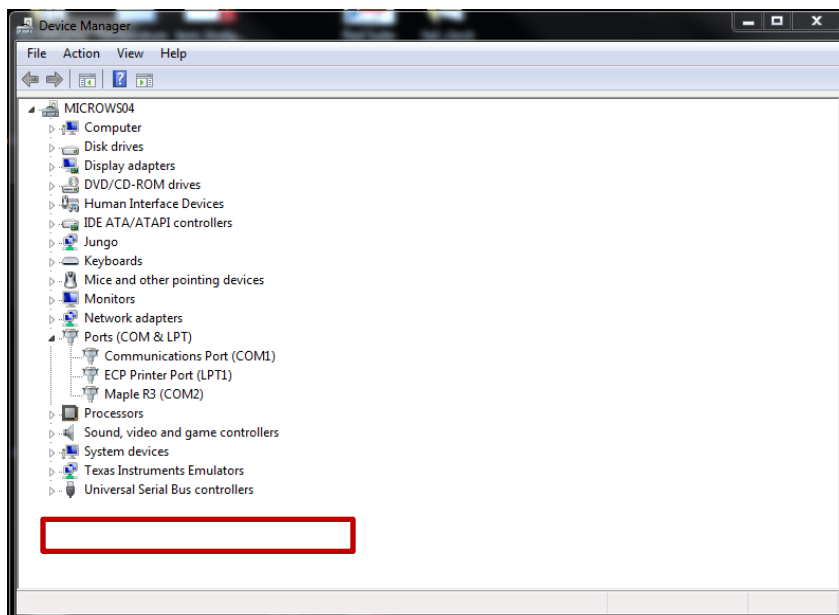
Click on Hardware and Sound

Under Devices and Printers, click on Device Manager

Under Ports (COM & LPT)

Look for a port with the name 'Maple' in it. The port name should appear next to it. This is the port name you need to select within application.

Ex. On this computer, the Maple board appears under COM2.



10.3.2 SYSTEM

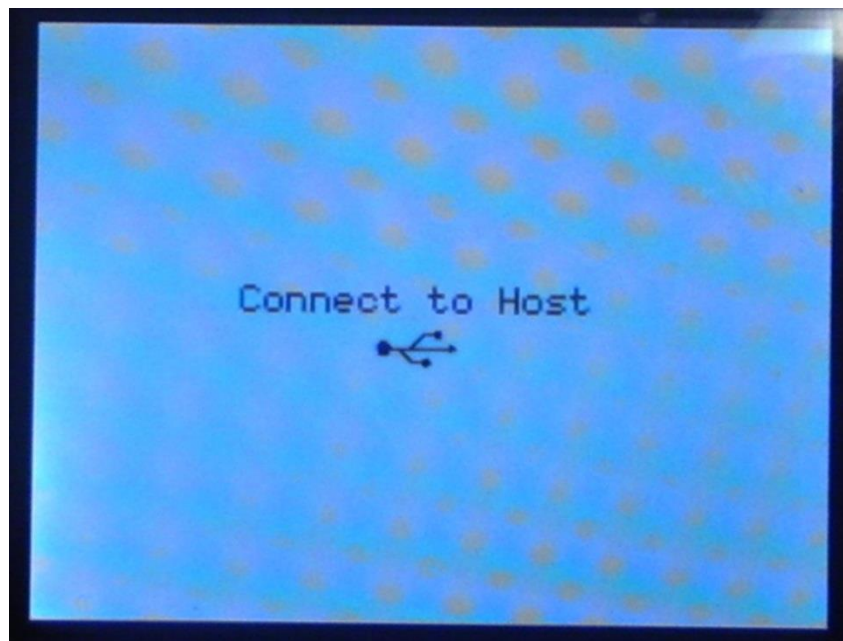
When the system is initially turned on you are presented with a splash screen.



The initial PyroBot splash screen.

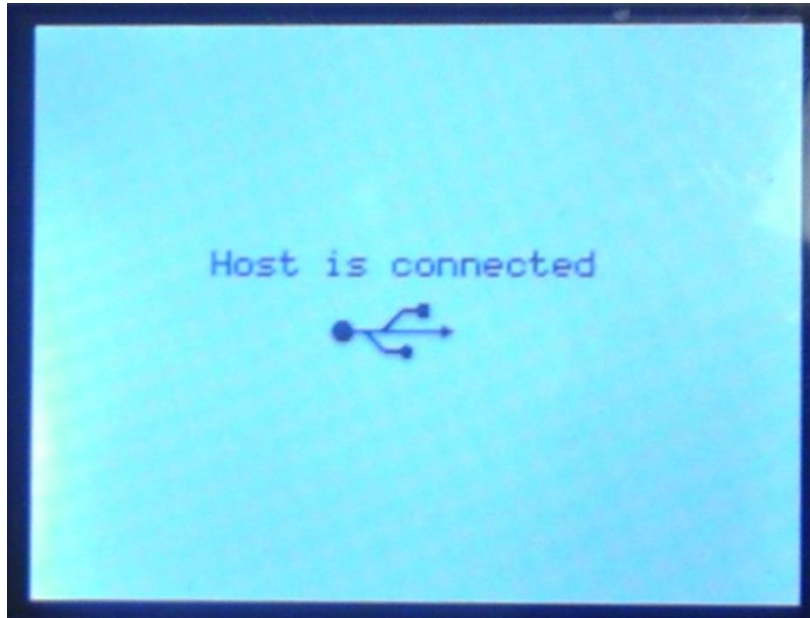
This splash screen prompts you to tap on the screen to continue. This is done as a security measure to avoid the machine from executing tasks when you are not ready.

Once you have tapped you are prompted to connect a host computer to PyroBot.



The host prompt screen.

If the host was already connected, or after you connect a host, you will get the following screen.



Host connected screen.

From here on all actions are done on the host system.

While an image is being saved to memory, the following screen will display.

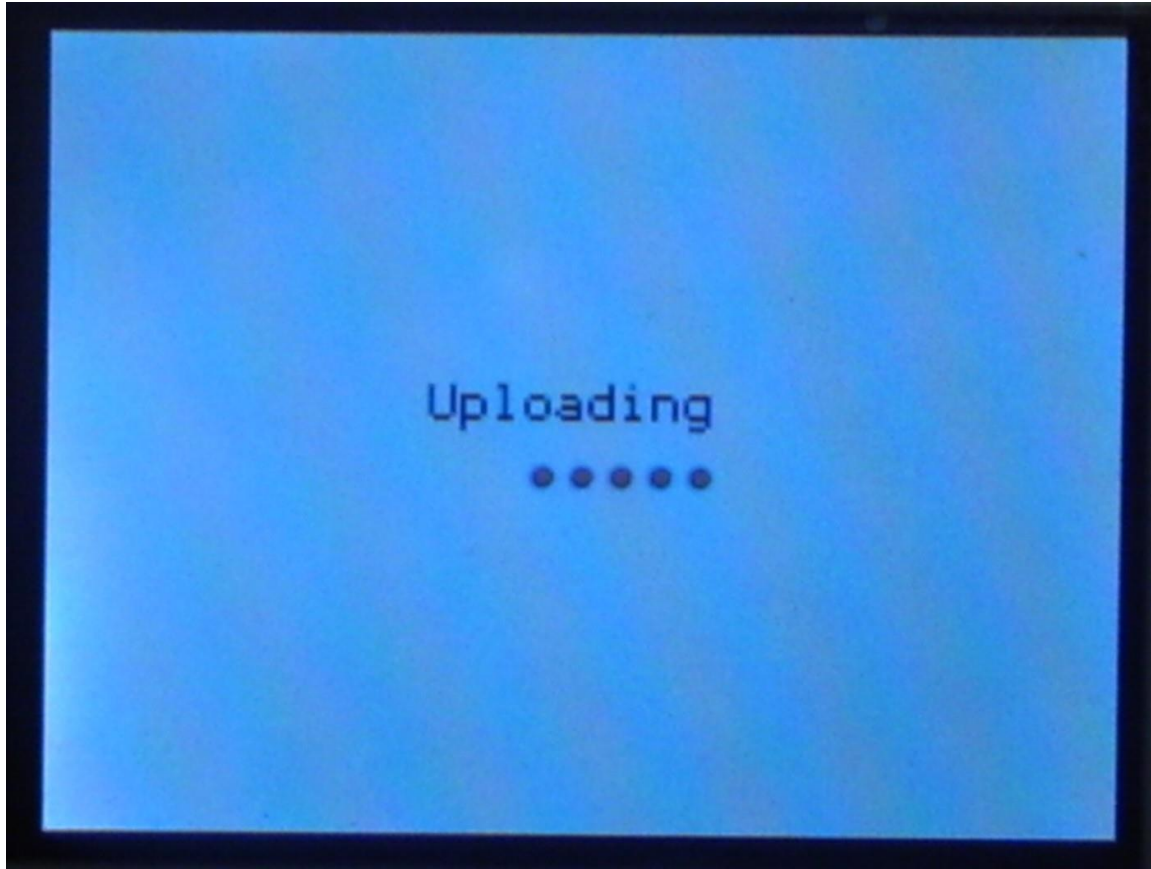
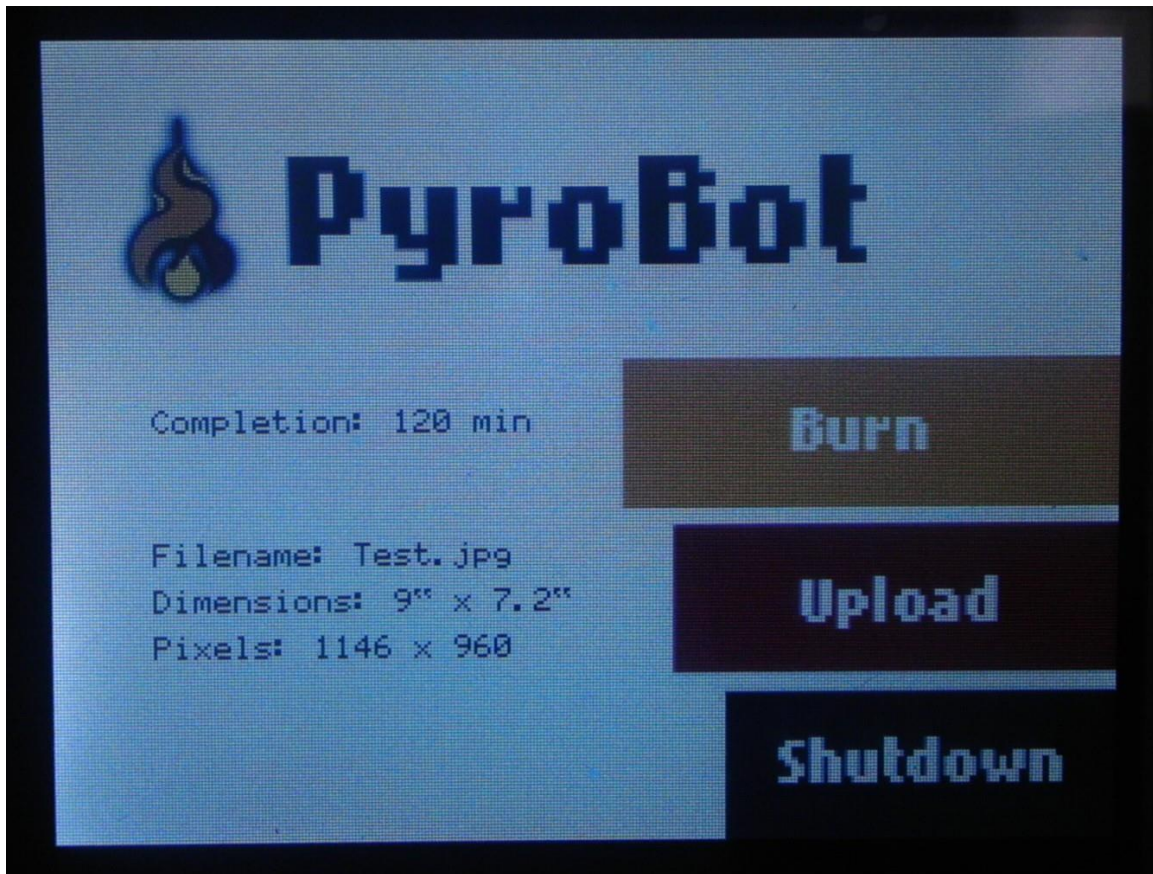


Image uploading screen.

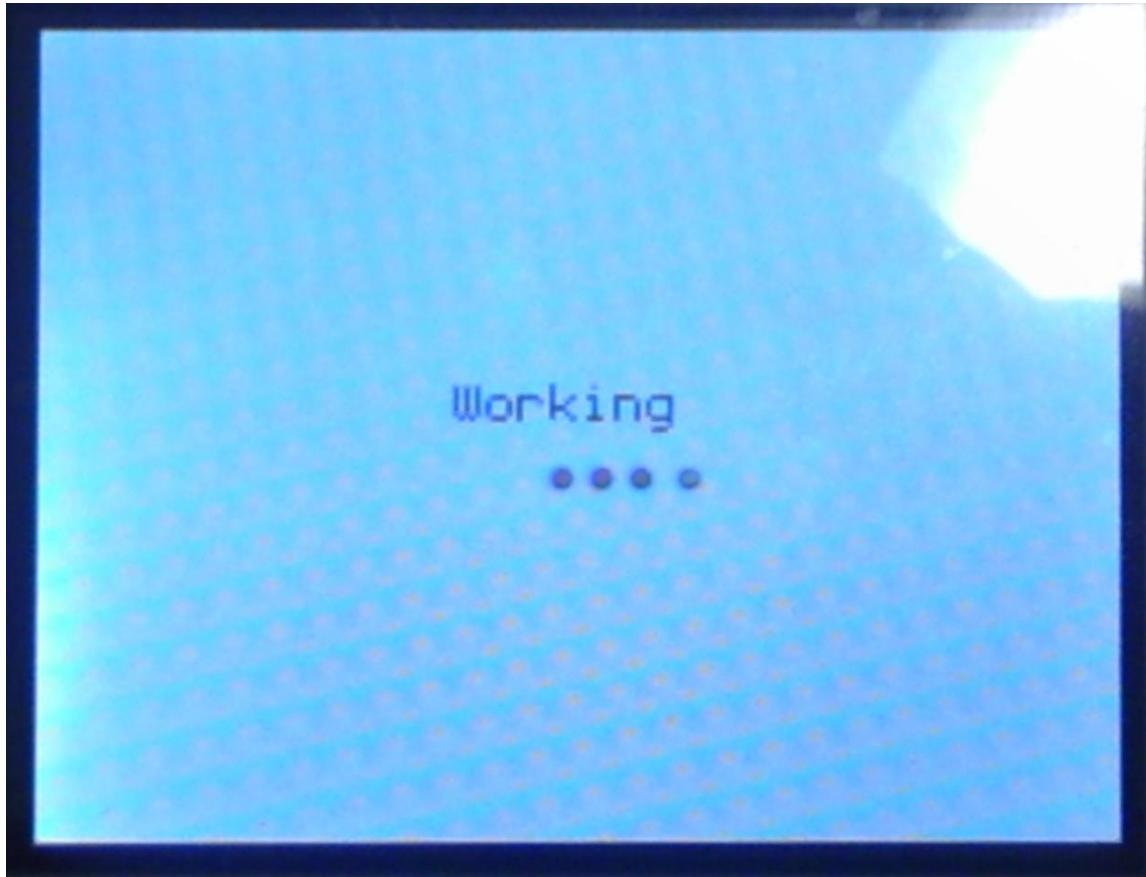
After an image has been transferred successfully by the host application, you will be presented with the following screen. Select any of the three buttons to perform an action.

Pressing the upload button returns you to the connecting screen. But the other two work as described next.



Main menu screen.

Pressing the burn button will cause the system to attempt to burn the transmitted image. While the burning is taking place, this screen is displayed.



On completion, the screen returns to the main menu.

Pressing the shutdown button presents you with this screen. Once it is shown, you will need to restart the system in order to utilize it again.

