



# DESIGN DOCUMENT

Team May15-03

Shawn LaGrotta

Matt Eckes

Jacob Mayer

Trevor Boone

Jacob Schulz

[lagrotta@iastate.edu](mailto:lagrotta@iastate.edu)

[mweckes@iastate.edu](mailto:mweckes@iastate.edu)

[jdmayer@iastate.edu](mailto:jdmayer@iastate.edu)

[tdboone@iastate.edu](mailto:tdboone@iastate.edu)

[jschulz@iastate.edu](mailto:jschulz@iastate.edu)

## TABLE OF CONTENTS

1	List of Figures .....	2
2	Definitions.....	3
3	Introduction .....	3
3.1	Background & Problem Statement .....	3
3.2	Acknowledgements.....	4
3.3	Operating Environment .....	4
3.4	Intended Use and Intended Users .....	4
3.5	Assumptions.....	4
3.6	Limitations.....	5
3.7	Expected End Product and Other Deliverables.....	5
4	Design Requirements .....	5
4.1	Functional Requirements.....	5
4.2	Non-Functional Requirements.....	6
5	Project Design .....	6
5.1	System Overview.....	6
5.2	Hardware .....	9
5.2.1	Raspberry Pi .....	9
5.2.2	Arduino with RAMPS Motor Shield.....	9
5.2.3	LCD .....	11
5.2.4	LED Status Indicator .....	11
5.3	Mechanical Design .....	11
5.3.1	3D Printer .....	11
5.3.2	HDD Jig .....	12
5.3.3	Probe Holder .....	13
5.4	Software.....	14
5.4.1	Interface.....	16
5.4.2	Backend Server .....	16
5.4.3	GUI Design.....	16
5.4.4	G-Code Generator.....	17
5.4.5	Host Software & G-Code Interpreter .....	18
5.4.6	Firmware .....	18
5.5	Calibration.....	19

6	Testing and Validation .....	20
6.1	Software Communication .....	20
6.2	File Parsing Gerber Files.....	20
6.3	Motor Controls.....	20
6.4	X, Y, and Z Motion .....	20
6.5	Transition Time .....	21
6.6	Locating Test Points .....	21
6.7	Risks .....	21
7	Project Schedule .....	22
8	Conclusion.....	22
9	Engineering Drawings .....	23

## 1 LIST OF FIGURES

---

Figure 1: System Diagram Block Overview .....	7
Figure 2: Two models of the Raspberry Pi .....	9
Figure 3: Arduino Mega 2560 with and without RAMPS 1.4 shield.....	10
Figure 4: RepRap Smart Adapter LCD .....	11
Figure 5 3D Printing Platform Use .....	12
Figure 6 RepRap Prusa i2 Model.....	12
Figure 7 HDD Jig Design .....	13
Figure 8: Spring Barrel Latch .....	13
Figure 9: Pin Protection Covers.....	13
Figure 10: Probe Holder .....	14
Figure 11: Software Data Flow Block Diagram.....	15
Figure 12: GUI Mockup .....	17

## 2 DEFINITIONS

---

Term	Description
HDD	Hard Disk Drive
HDD Jig	Device used to hold the HDD in place
PCBA	Printed Circuit Board Assembly
Oscilloscope	Device used to view electrical signals
LCD	Liquid-Crystal Display
LED	Light Emitting Diode
HTTP	Hypertext Transfer Protocol
G-Code	Numerical control programming language used in the controlling automated machine tools
Gerber Files	The standard used by printed circuit board industry software to describe the printed circuit board images: copper layer, solder mask, legend, etc.
Drill Files	Defines hole locations, tool numbers and finished hole sizes (X&Y coordinates) for PCBA design
Arduino	A single board microcontroller
Raspberry Pi	A single board computer
RAMPS	RepRap Arduino Mega Pololu Shield
UI	User Interface
GUI	Graphical User Interface
Stepper Motor	Motors which operate only in discrete increments of rotation
Carriage	The moving middle assembly on the x-axis of a RepRap which holds the probe
Via	Electrical connection between layers on a circuit board
GPIO	General Purpose Input/Output

## 3 INTRODUCTION

---

The purpose of this document is to give an overview of the Robotic Probe Positioning Senior Design Project. A background of the design problem, operating environment, use, and expected deliverables will be covered in this section.

### 3.1 BACKGROUND & PROBLEM STATEMENT

Hitachi Global Storage Technologies (HGST) is a company that designs hard disk drives (HDD), solid-state drives, and external storage products and services. During the development of a HDD, a significant amount of debugging is necessary to troubleshoot problems with the operation. A key tool for an engineer to capture and examine the waveforms that describe the devices behavior is an oscilloscope.

In some instances, an engineer is required to work on a HDD that is at another location. The current strategy for placing a probe at a remote location, on the correct via, is to employ a technician to assist the engineer. This operational partnership can become troublesome when

considering time zone changes and lack of clear communication. In order to streamline the development process and save time this project was suggested by two engineers at HGST, Brian Joyce and Zach Wygant.

The goal of this project is to develop a robotic system that can move an oscilloscope probe to any via on a 2.5" form factor HDD. The robot should be able to operate remotely, allowing the engineer to control the positioning of the probe without intervention from a local technician. For phase one of the project, we are designing to control only one probe. With time permitting the system will be expanded to a two probe design, offering the engineer a more useful tool in capturing waveforms.

A prototype of this robot will be available by April 2015. If the robot is deemed useful in assisting the engineers, multiple units will be produced.

### 3.2 ACKNOWLEDGEMENTS

HGST, Inc. has allocated \$1,500 dollars for the development of this project. Our client contacts, Brian Joyce and Zack Wygant, will provide oscilloscope probes and HDDs in order to correctly model the probe holder and HDD jig. In addition, Gerber board and drill files are provided so that the HDD vias can be located.

Our group's ISU Faculty advisor, Mr. Leland Harker, will provide technical support. This will include suggestions on design concepts, part modeling, part manufacturing, and design standards.

### 3.3 OPERATING ENVIRONMENT

The robot will be placed on a debugging cart in a HGST laboratory. The lab is climate controlled, therefore the testing platform can be open to the environment. Design requirements for harsh conditions do not need to be considered. The client's interaction with the GUI is through a HGST's intranet and will need to be as non-intrusive to the computers software as possible.

### 3.4 INTENDED USE AND INTENDED USERS

The intended use for this project is to control an oscilloscope probe remotely by commands given by an engineer. The HDD will be placed into the robot by a technician, then control of the probe placement is given to the engineer working remotely. The engineer will issue commands by interacting with a GUI created by our team.

The technician placing the HDD and mounting the probe will work directly with the robot. User friendly design will allow consistent HDD placement.

### 3.5 ASSUMPTIONS

For this project, we assumed that the robot will be controlled by only one engineer at a time. Communication with the robot is done over the HGST intranet. We assumed that conflict with accessing the company's network would not be a problem. In addition, we assumed all HDD will

follow the 2.5" HDD industry standards for sizing. The robot location is assumed to have access to 120VAC power. The user is expected to comprehend English.

Once the project is completed we assumed that the company will be able to create multiple units, as the robotic system is deemed to be useful in diagnosing a malfunctioning HDD. We will provide quality documentation so that a replica of the system can be created with limited modification.

### 3.6 LIMITATIONS

Typically, an engineer uses multiple probes simultaneously while troubleshooting a HDD. The system we designed is limited to operate only one probe, with the ability to scale up to two probes in the future. The engineer will be limited to capturing one signal at a time while working remotely, and may still need the assistance of a technician to hold other probes in conjunction to the robotic system.

### 3.7 EXPECTED END PRODUCT AND OTHER DELIVERABLES

At the end of the project, May 2015, we are expected to deliver a robot system that is able to take remote commands and place the probe onto the proper HDD via. Documentation will also be provided so that the system can be replicated and debugged by the company as more units are needed.

## 4 DESIGN REQUIREMENTS

---

This section contains the design requirements, both the functional and non-functional requirements. These requirements will govern any solution brought on during the course of this project.

### 4.1 FUNCTIONAL REQUIREMENTS

- There shall be a probe positioning device and a control system to control it.
- The probe positioning device shall:
  - Consistently hold a 2.5" form factor HDD in the same position
  - Hold, move, and manipulate the LeCroy Oscilloscope Probes
  - Place probes on HDD PCBA test points with enough force to make a good electrical contact, yet not enough to damage the PCBA
  - Take commands from the control system and act accordingly
  - Allow probes to be grounded to the test fixture and HDD
  - Support positioning at least one probe
- The control system shall:
  - Be remotely accessible over the HGST intranet
  - Send commands to the probe positioning device (client)
  - Receive commands from the remote operator over the intranet (server)
  - Translate commands from remote operator on intranet to probe positioning device

## 4.2 NON-FUNCTIONAL REQUIREMENTS

- The device shall move the probe to any point on the board within 60 seconds
- The device shall have LEDs that show the operational status of the device
- The device shall have an LCD to display the coordinates of the probe and operational status
- The robot will return status information back to the client
- It shall take up a volume less than 4 ft<sup>3</sup>
- The device shall be easily maintained and built from standard parts when possible
- The device should support a manual operation mode

# 5 PROJECT DESIGN

---

The design approach is laid out in this section. The overall system design and network structure including the individual components are described in this section. The reasoning behind each design choice will be explained.

## 5.1 SYSTEM OVERVIEW

The user will manually put the 2.5" HDD into the secure jig and then plug in the HDD development electrical harness. The user will then access the web page through the HGST Intranet using a web browser. Upon logging in, the user will next import the Drill and Gerber board files into the web application, the Raspberry Pi web server will then parse the files; thus finding all the via locations.

Next, the web application will plot all test via locations onto a template HDD on the GUI. Once the file is parsed and the test points are displayed, the user can then proceed to select a via to test. The web server will then take the test point via and current probe coordinates to generate the corresponding G-code commands to move the probe to the via. The Arduino receives these commands, converts the commands to a waveform, then sends the signal to the RAMPS 1.4 motor board. The RAMPS board sends power to the stepper for the correct duration.

Upon receiving a test command, the Raspberry PI will activate the status LEDs and indicate that the robot is in use. The current coordinates of the oscilloscope probe are displayed on the LCD by the Raspberry Pi. The operational status messages will be sent through the web server back to the client at all times.

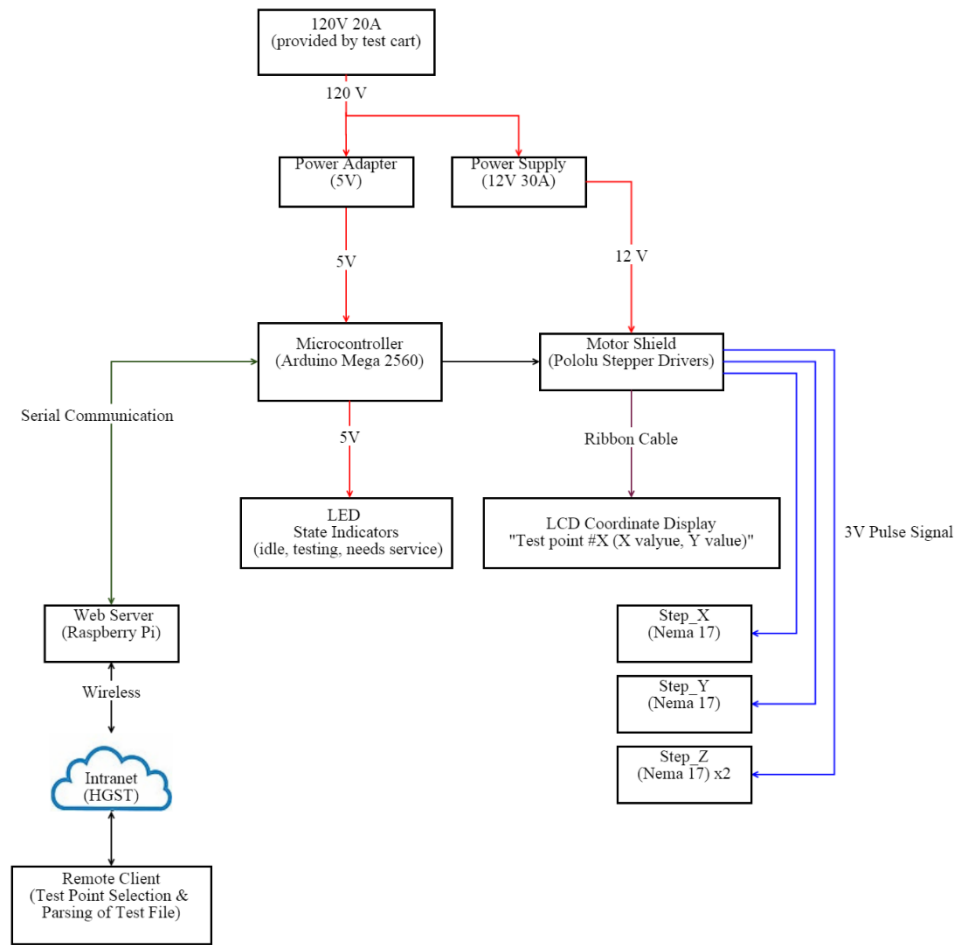


Figure 1: System Diagram Block Overview



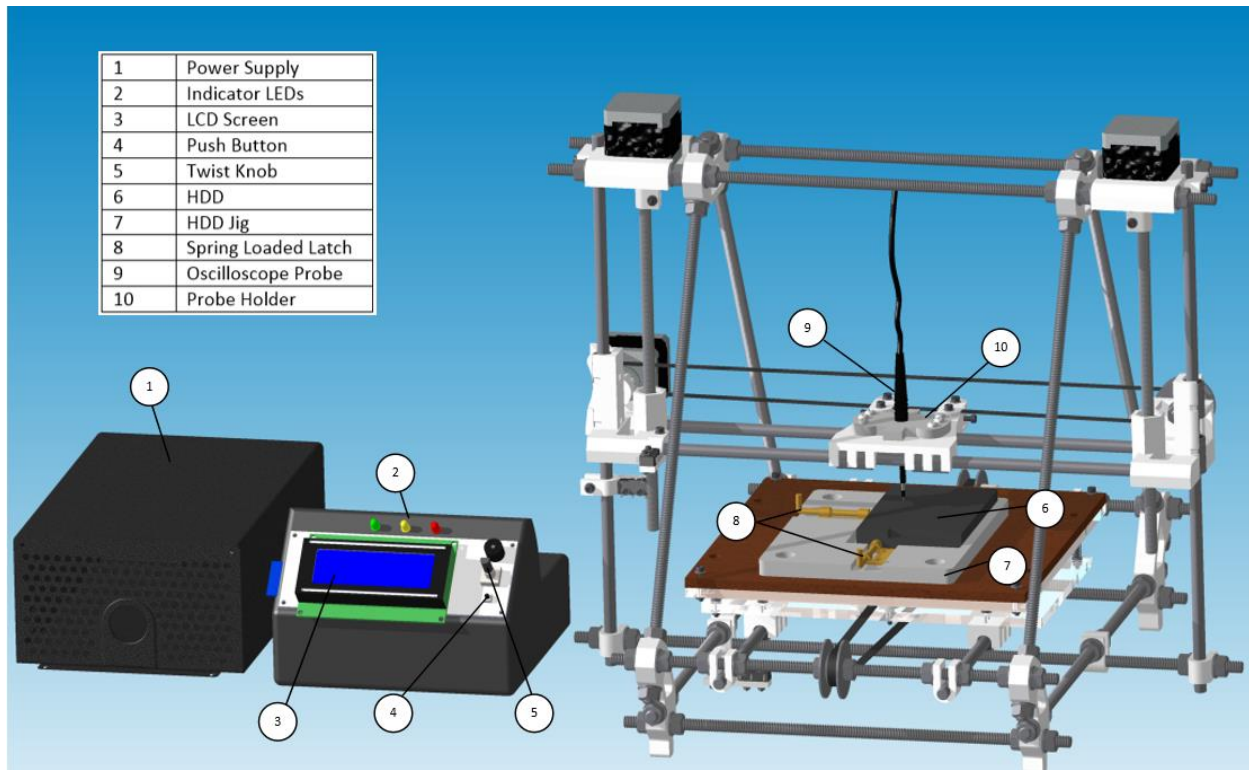


Figure 2 Concept Sketch

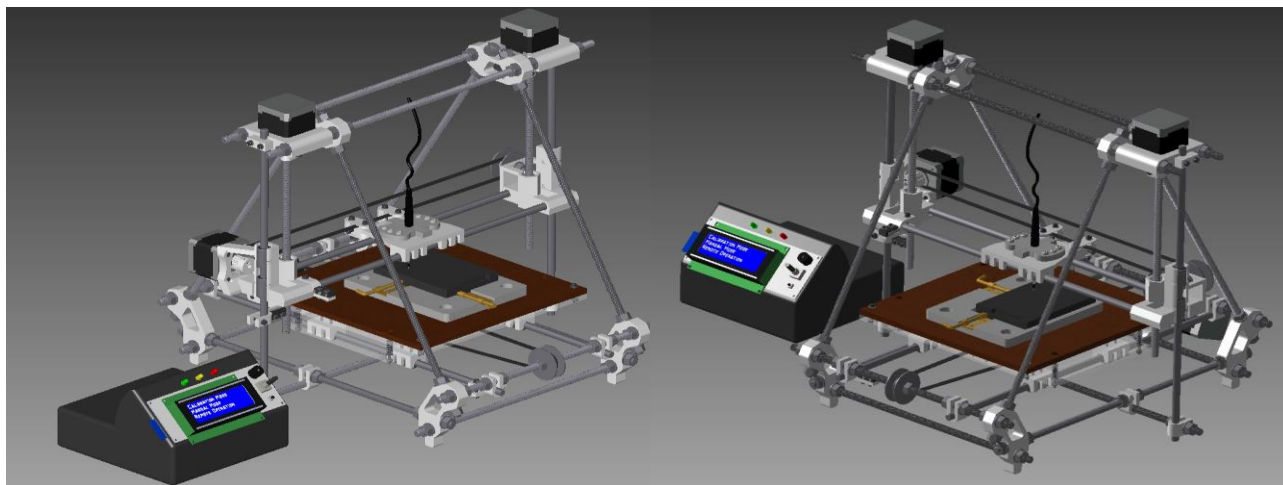


Figure 3: Images of robot LH and RH sides



Figure 4: Image of jig LH and RH sides

## 5.2 HARDWARE

The hardware section includes all components for motor control. This includes the microcontroller, motor drive, displays, and web server.

### 5.2.1 Raspberry Pi

The Raspberry Pi will be the web server for connecting the client to the robot. It will be responsible for generating G-code and sending it to the Arduino via USB. The Pi will be connected to the HGST intranet via a PI Wifi.

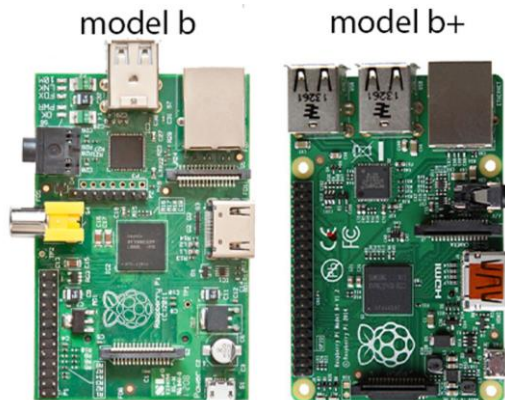


Figure 5: Two models of the Raspberry Pi

### 5.2.2 Arduino with RAMPS Motor Shield

The motors will be controlled by an Arduino Mega with a RAMPS connected to it. The RAMPS will have stepper drivers to drive the stepper motors for the X, Y, and Z axis. The RAMPS motor can support up to six stepper drivers. The Arduino is responsible for converting G-code to the correct waveforms in order to control the stepper motors.



Figure 6: Arduino Mega 2560 with and without RAMPS 1.4 shield

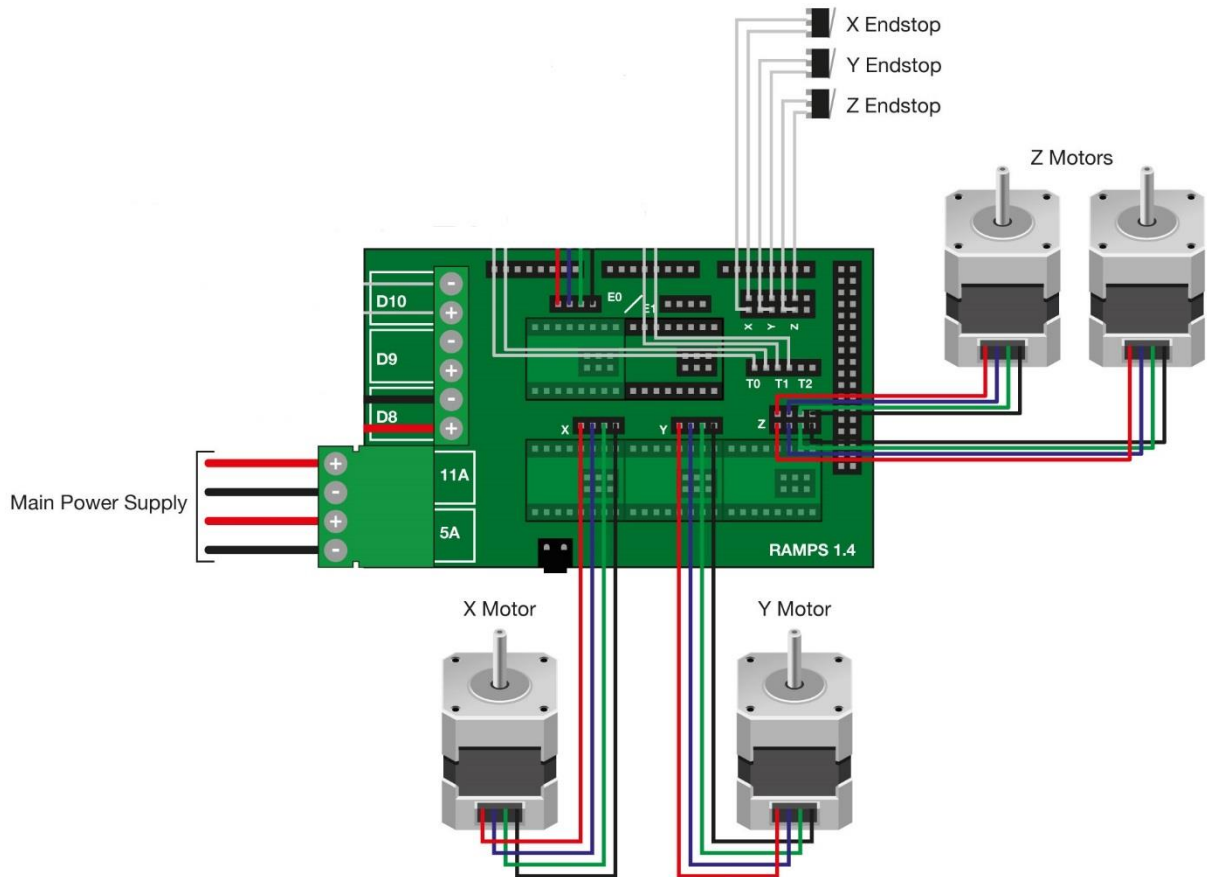


Figure 7 Ramps 1.4 Wiring Schematic

### 5.2.3 LCD

A LCD will be connected to the GPIO pins of the Raspberry Pi to display the current coordinates of the oscilloscope probes. Status and error messages will also be provided to the user through the LCD. The RepRap LCD Smart controller supports a push button and a rotary encoder. This will enable the probe to be controlled manually for calibration.



Figure 8: RepRap Smart Adapter LCD

### 5.2.4 LED Status Indicator

LEDs will be connected to the GPIO of the Raspberry Pi to indicate the status of the robot. The LEDs will be mounted in such a location so as to make them easily visible.

## 5.3 MECHANICAL DESIGN

The mechanical design is laid out in this section. Several components are listed both professionally fabricated components, as well as, custom designed components created by our team. The need for the company to have readily available parts was considered in the design process.

### 5.3.1 3D Printer

In order to achieve the accuracy and precision needed to complete the task and still manage to stay within budget we decided to leverage 3D printing technology. The robot would be able to control the probe in all three Cartesian axis with high accuracy. Third party manufacturing was chosen, as opposed to self-manufacturing the components, due to the group's limited knowledge of mechanical design. The RepRap Prusa Mendel i2 was chosen because it met the accuracy requirements, fell within budget, and has the most third party support out of all 3D printing platforms.

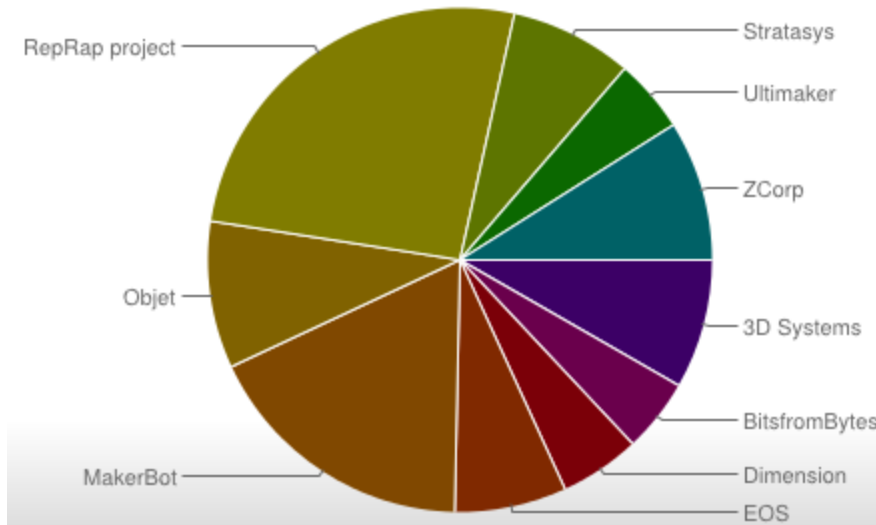


Figure 9 3D Printing Platform Use

Source: Moilanen, J. & Vadén, T.: *Manufacturing in motion: first survey on the 3D printing community*, *Statistical Studies of Peer Production*

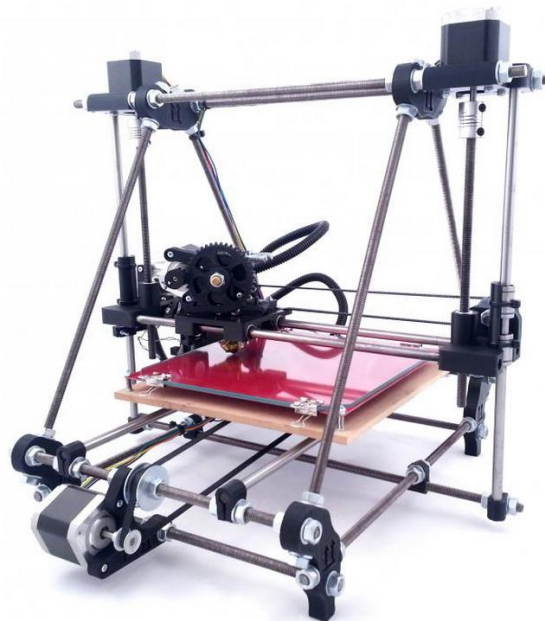


Figure 10 RepRap Prusa i2 Model

Source: NWRepRap.com, 2013

### 5.3.2 HDD Jig

The jig will hold 2.5" HDD in the recessed rectangular area of the jig. Spring barrel latches will be used to apply pressure from the bottom side (where the harness connects) of the HDD and the side not touching the recessed wall. The latch pin ends will be protected by rubber screw covers to protect the HDD and also increase the friction between the pin and chassis. The purpose of the latches is to hold the HDD in one corner of the recessed rectangle and prevent any movement

when testing .The whole jig will be fastened down to the X-axis platform using three M8 bolts. The jig will be fabricated out of 3/8" aluminum stock which will allow proper grounding of the oscilloscope to the HDD chassis.

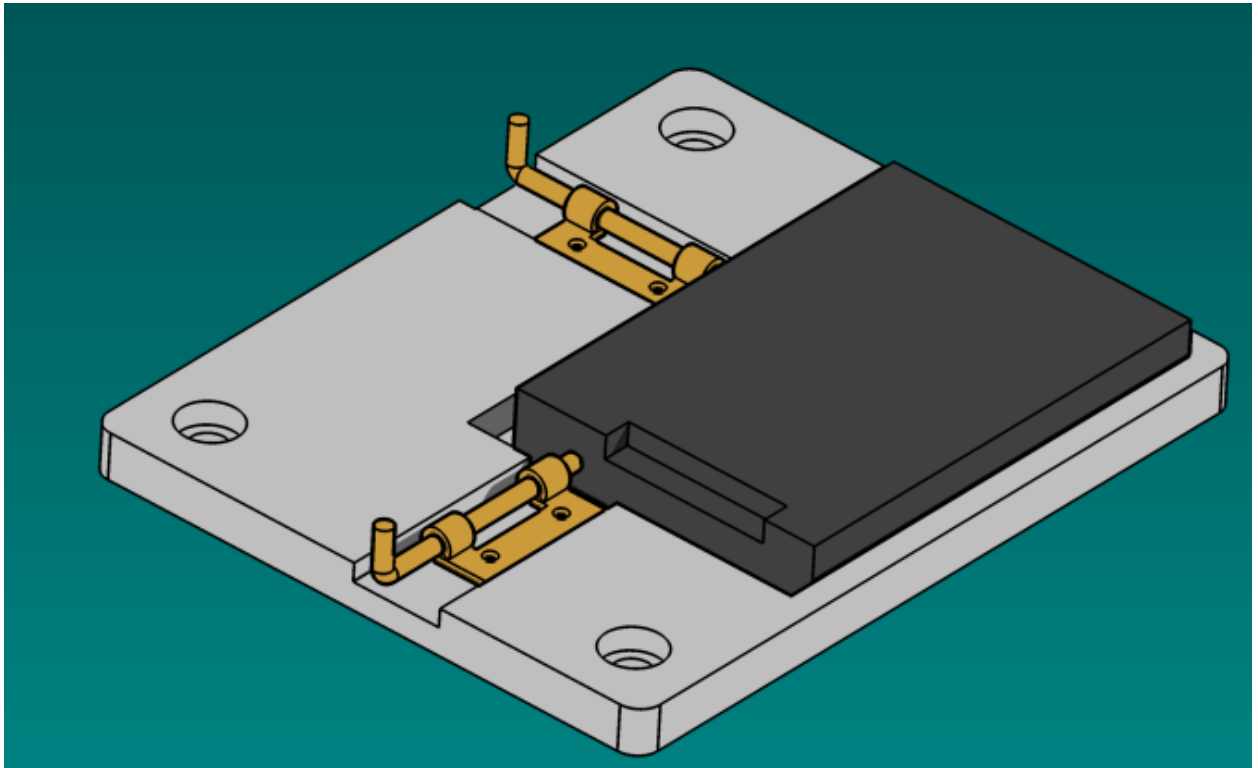


Figure 11 HDD Jig Design



Figure 12: Spring Barrel Latch



Figure 13: Pin Protection Covers

### 5.3.3 Probe Holder

The probe holder will securely hold the probe during testing. The holder itself will be fastened to the 3D printer carriage. To avoid damaging the probes and the HDD the probe will be equipped with a spring loaded tip. The holder will be fabricated via 3D printing.

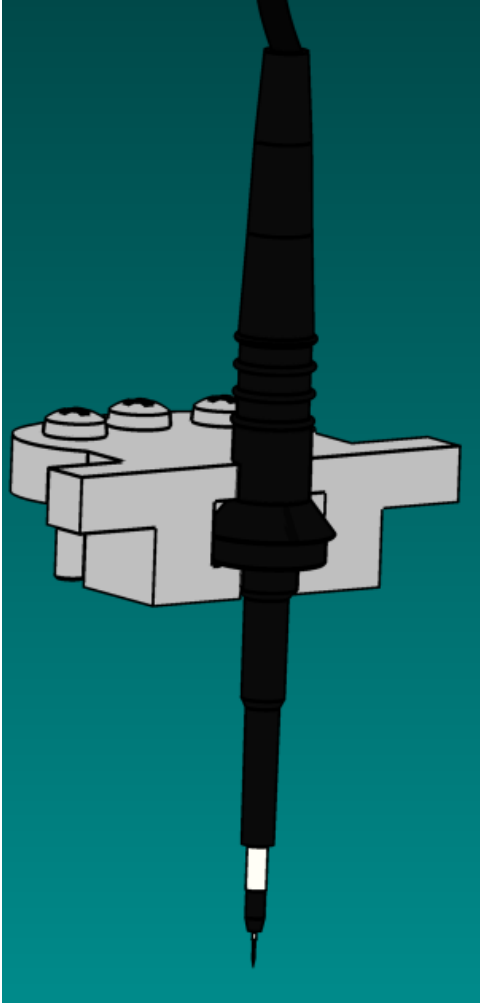


Figure 14: Probe Holder

## 5.4 SOFTWARE

This section describes the design of the software. A diagram of the high-level structure is provided below, which is comprised of several stages between the interface presented to the client and control of robot's hardware. Rationale for the inclusion and modification of each module is discussed in addition to their details. Each module could have existing software and/or custom pieces.

# Software Modules and Data Flow

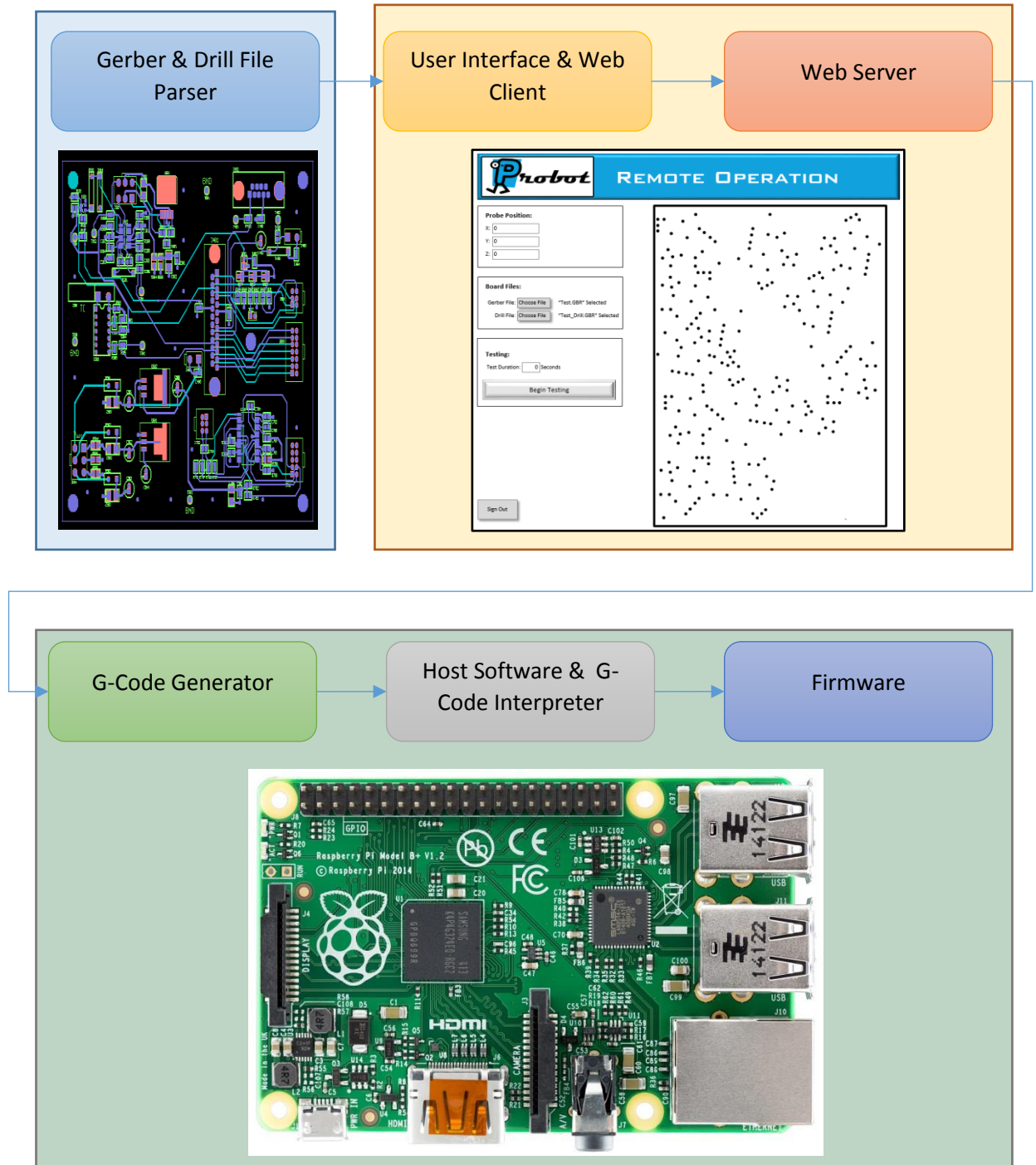


Figure 15: Software Data Flow Block Diagram



#### 5.4.1 Interface

For the interface we decided to build a web application and serve it off of a computer that is connected by a serial connection to the hardware controller. A web interface has an advantage over native apps in the fact that there is already rich ecosystem of tools to help develop web apps. This allows for a solution that is cross platform and doesn't require the user to install any extra software on their computer. On the implementation level the plan is to program the web server in python using the tornado library, as it is both lightweight and allows for asynchronous handling of requests. This server will serve a simple HTML/CSS page with the GUI functionality implemented in JavaScript.

#### 5.4.2 Backend Server

Since we decided to do a web application we needed some web server to serve the webpages. To accomplish this we decided to use the tornado library because it is designed for small projects and will allow for us to efficiently develop the backend.

##### 5.4.2.1 Long Polling

Operations on the machine will take some time to complete, and there could be multiple clients connected to the server at the same time; therefore a need exist for an asynchronous way to send messages between the server and the client. Because it is a webserver, the clients can always post data to the server but the web server can only send data back when the clients request it. We decided that a long polling solution was the most efficient answer to this problem. Essentially this means that the client will just keep sending requests but the server will only respond to them if there is new data. If the client is always sending a request then the server can send messages back at any time. This asynchronous messaging system also makes it easier to have more clients as you can ensure that every client gets the same message.

#### 5.4.3 GUI Design

The overall goal of the GUI is to be able to move probes to the desired points. The implementation for this has two major parts, the uploading and parsing of files, and the selection of test points. Both of these functions are implemented in JavaScript combined with asynchronous requests to the server.

##### 5.4.3.1 File Storage/Parsing

File parsing will be done client side in JavaScript and stored in files on the server. We decided to include file parsing into our program because it makes our tool much easier to use by not having to get the coordinates of test points from another program. We also decided to store them on the server to make it so it isn't necessary to always have to upload the files every time and also it makes it so once it is uploaded once it can be used by different people even if they do not have the file.

##### 5.4.3.2 Point Selection

Point selection is the main functional component of the interface. Ease of selection of points is one of the top priorities of the GUI. To facilitate this, we are planning on putting points to be selectable on a 2d representation of the circuit board and also, if possible, have points selectable by name. Once the points are selected the client will post data to the server and the server will parse the data and move accordingly.

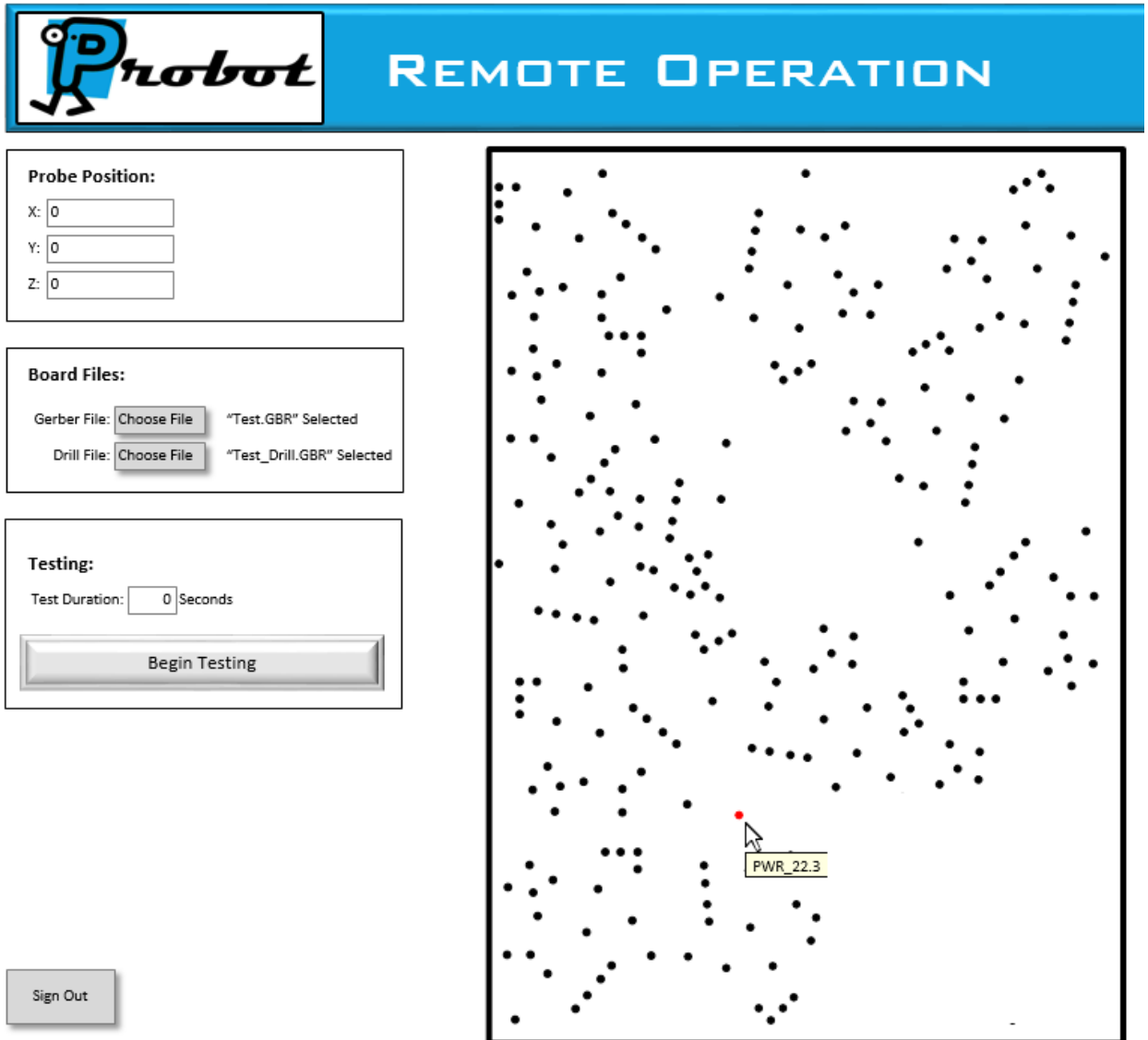


Figure 16: GUI Mockup

#### 5.4.4 G-Code Generator

In the 3D printing world, where the RepRap originates, G-Code is usually generated by an entirely separate utility. The input to this utility is a file (usually STL file), which describes a dimensioned 3D model and is created by an engineer using CAD software. Using this file, the model would be translated into a long list of G-Code commands for the printer.

Our project, however, doesn't get its input from a 3D model. G-Code for movement needs to be generated dynamically based on commands send by the client and processed by the web server. For this our team will use a combination of custom code and existing libraries. The custom code will handle project-specific modifications to the original printer, and a library designed for the

RepRap, called MeCode, will translate high level motion commands (move from point a to point b) into a series of G-Code instructions.

#### 5.4.5 Host Software & G-Code Interpreter

Once G-Code instructions have been generated, they are submitted to a module called “host software” for the RepRap. The job of the host software is to provide machine configuration (sometimes through a GUI), managing a job queue, and supervising jobs while they execute. The host software is also responsible for communicating with the microcontroller over a serial connection. This functionality is necessary in order to send G-Code to the firmware and read the statuses of hardware sensors. Following is a list of several choices of RepRap host software. RepSnapper will be chosen for its speed, simplicity, and customization.

Name	Description
Pronterface	Pronterface is the visual host by Kliment. It features an intuitive user interface, a 3D model to G-Code translator, and track separate G-Code groups.
RepSnapper	Fast, barebones host program. It has a fast, simple G-Code generator and focuses on providing easy customization.
Replicator G	Well-rounded compared to the others. Also has a 3D model to G-Code translator built in as well. Designed primarily for 3D printers, so many of its special features aren't of use for this project.
Repetier	Has a simple interface to two different 3D model to G-Code translators (which have different strengths depending on the type of model).

#### 5.4.6 Firmware

The firmware's responsibility is to interpret G-Code and control the hardware. It is tailored to the specifications of the hardware and is able to take the abstract G-Code instructions sent from the host software, incorporate calibration details specific to the hardware, and generate the necessary electrical signals for manipulating the hardware (e.g., moving motors). Again, there are several options of firmware available for the RepRap. As before, they follow this paragraph. Marlin will be chosen for its calibration, customization, accuracy, and safety features.

Name	Description
Sprinter	Simple to setup and start using. Simple, first-time calibration. Support for most electronics and has acceleration control.
Marlin	A feature-extension to sprinter with more control over calibration, motor control, safety features, and better customization. The downside is that it is more complex to set up.

Teacup	Smallest firmware size, so it can fit on smaller Arduinos than any other firmware. Has no dependencies with Arduino libraries. However, these properties mean a reduction in features.
Repetier	Built alongside its host controller sibling for an all-in-one solution. However, is incompatible with other host controllers. Communication between host and firmware is more robust and reliable.

### 5.5 CALIBRATION

With 3D printing, calibration of the machine is simplified because the models can start anywhere within the platform and only the relative changes in position matter. With our project, we will be securing and grounding a HDD to the platform, and so we need to be able to consistently match the origin of our machine with the origin of the HDD design. Without this guarantee the machine will fail to read the contacts on the HDD correctly. This necessity requires a more complex and robust form of calibration.

We will take advantage of the PCBA board manufacturing accuracy to define an accurate origin to base a coordinate system off of. Through manually moving the probe to a specific point and then identifying to robot which point the probe is really at, the software will be able to compute calibration coefficients using the data from the board files. For example, if we identified the left-most and bottom-most via as the origin and then moved the probe to that position, the software could reference the via in the board files and then “zero” the probe out.

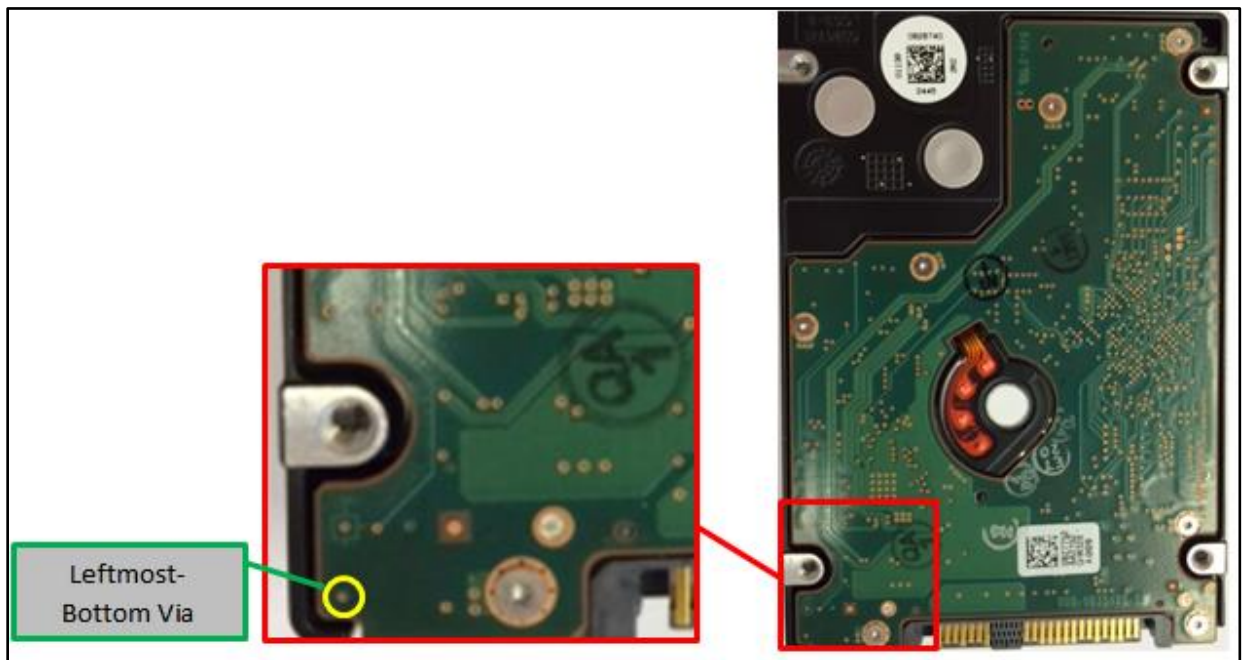


Figure 17 “Left-most and Bottom-most” via on HDD PCBA - Calibration explanation

## 6 TESTING AND VALIDATION

---

### 6.1 SOFTWARE COMMUNICATION

Without communication between the client and the server, the user's command will never make it to the robot. To test that the communication module is working correctly, first a connection will be obtained and verified. From there, the client and server agree to go into test mode in which each client command will be sent to the server, the server will note the command, and return a successful status to the client. This will ensure that communication between the two is correct and that the server can handle any requests the client is capable of sending.

### 6.2 FILE PARSING GERBER FILES

The success of this project lies in the ability to accurately locate test vias on a HDD. If we are unable to accurately locate the test points with regard to a reference point then the whole system fails. Due to the importance of this step, we must be able to show that the GUI accurately represents the test vias on the HDD. After the software generates a representation of the HDD with the test points located, the coordinates will be compared to the locations given in both the gerber file and the measured values.

### 6.3 MOTOR CONTROLS

Testing of motor controls is an important step in ensure all components are working properly. Before significant time is spent is mounting the motors to the robotic system, we first would like to check that they are operating correctly. RepRap software has an application, Pintrun, that allows for motor commands to be sent to the Arduino, and then to the specified motor (X, Y, or Z). For this testing procedure the motor control components (Arduino, RAMPS, 3 Nema 17 stepper motors) will be connected. Commands, sent through Pintrun, will control each individually and test for clockwise and counter clockwise motion. This test is simply to ensure that all motor controls components are working correctly and the Arduino motor pins are correctly identified.

### 6.4 X, Y, AND Z MOTION

Accuracy and precision in the X, Y, and Z direction is an absolute must for this project. The robotic system needs to provide significant reliability in order to be useful. For this reason extensive testing will be done to ensure both accuracy and precision. For this testing procedure, we will give the robot a command to travel in one cardinal direction. The movement will then be accurately measured using calipers. The actual moving distance will be compared to the expected value. To get a high degree of statistical importance a large sample size will be taken. This procedure will be done at a variety of lengths as well as compared against different speed ratings. Once the accuracy and precision are documented, we will be able to publish an optimum speed rating that will allow for accurate movement with limited traversing time.

## 6.5 TRANSITION TIME

A non-functional requirement the company expects is the movement from one test point to another to take no more than 60 seconds. The accuracy of movement trumps this speed requirement, but if we can show that the robot can move quickly from one test point to another, the robots efficiency will greatly increase. For this test we will create a random list of test points the robot must travel to. The time it takes the robot to move from one test point to another will be measured. Once a large sample size is taken the average will be published.

## 6.6 LOCATING TEST POINTS

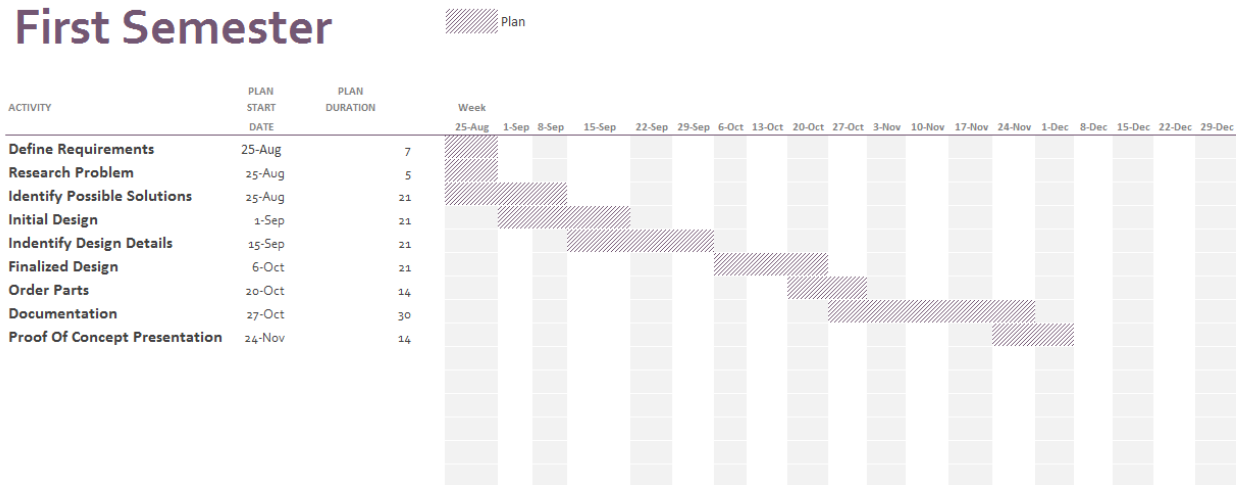
The final test of the system is a complete test of the devices capabilities. A HDD will be placed in the HDD jig. The HDD Gerber files will be parsed and the GUI will indicate where the test points are located. The user will then select randomly a test point to measure. The robot's performance will then be measured on the basis of locating the test point and making sound electrical contact with the test via. We will do a large sample size in order to simulate the debugging process a HGST employee may encounter. We will then find an operational limit in how precise and accurate we can be over number of test points and recommend how many vias a user can accurately test before recalibrating the robot.

## 6.7 RISKS

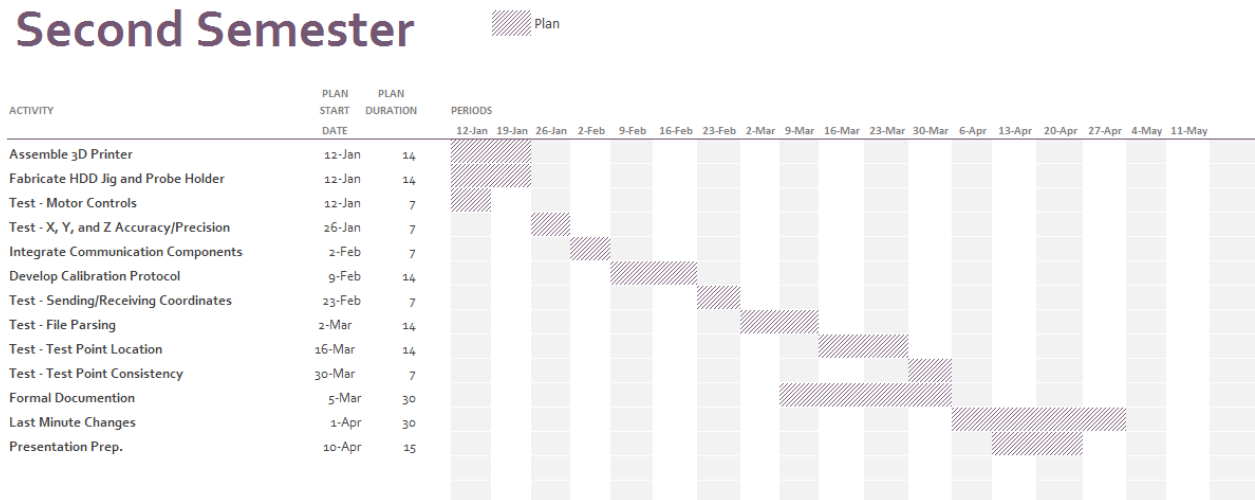
- **Calibration:** It is crucial that the system stay calibrated as the user is likely to not be around the machine to fix the problem. Additionally, small calibration errors (< 1mm) cannot be tolerated as it would exceed minimum feature separation. However, our team has no experience with calibration methods and we don't know how well our equipment will cooperate in regards to both being calibrated and staying calibrated.
- **Accuracy:** The machine has to be accurate within less than 1 mm. This means that our algorithms, motors, and construction must meet strict tolerances that might be difficult to obtain.
- **Customization:** The foundational bulk of the work is going to be purchased in order to best utilize our time and improve the quality of our final solution. However, in order to build exactly what we need, there is some custom mechanical work that needs to be done. Due to the lack of practical mechanical engineering experience on our team, it is possible that problems will arise with our design that we did not foresee.

## 7 PROJECT SCHEDULE

### First Semester



### Second Semester



## 8 CONCLUSION

Overall, our robot is designed to leverage the highly accurate but cheap technology of 3D printing to design a HDD development tool. The robot will hold an HDD in a jig securely, all while an engineer at a remote location commands the robot to move the probe to different vias for debugging of the HDD. The via locations will be identified by parsing Gerber and Drill files and then displayed on a GUI so that the user can pick which one to test. The system will be responsive and send status feedback to the engineer, indicating the operational success of the systems function.

# 9 ENGINEERING DRAWINGS

