Remote, Underground Powervault Inspection Robot

ECE 4012 Senior Design Project

Section L8A, Team P.V.I.R
Project Advisor: Lukas Graber

Stephanie Chan, schan40@gatech.edu
Elizabeth Fuller, efuller3@gatech.edu
Adrian Muñoz, am262601@gatech.edu
Nelson Raphael, nraphael7@gatech.edu
Lemek Robinson, lrobinson61@gatech.edu

Submitted

2019 May 2

Executive Summary

P.V.I.R is a remotely operated, recon robot that uses non-destructive testing methods (NDT) to retrieve information from powervaults. P.V.I.R will consist of a water-tight aluminum body for movement, a mechanical arm for positioning the camera and other sensors, a sensor package to retrieve information in the powervault, and a user interface for interacting with the robot. The robot must be able to read gauges in a poorly lit environment, and relay information to a remote server. The base of the robot has been built by a previous mechanical team. This project will focus on designing and implementing the sensor package, the control system, and the data collection and display system of the robot. A mechanical arm will support the camera and other sensors.

P.V.I.R is designed to be used to gather information in situations where powervaults are not safe for humans to enter. The pilot of the P.V.I.R can inspect the powervault without putting themselves in a dangerous situation. The P.V.I.R has visual sensing capability for reading gauges in the powervault, audio sensing to record high voltage, electrical discharge, gas sensors to test air quality, and an RGB-D sensor to map the terrain of the powervault. The P.V.I.R will be controlled remotely by a user interface program that runs on a computer or tablet. The program will also organize and record the information received.

The total price of all the components used in design are approximated to be around $775. Some of the sensors and microcontrollers used in this project are legacy parts and are do not contribute to the actual cost of development. This P.V.I.R will be sold for $2,500.

Table of Contents

Powervault Inspection Robot

1. Introduction

Team P.V.I.R is designing a power vault inspection robot. The team is requesting $775 from Georgia Institute of Technology - College of Electrical and Computer Engineering. The $775 will be used to develop a sensor package, a control GUI, a mechanical arm, and a robot testing environment.

1.1 Objective

The objective of P.V.I.R is to evaluate the environment of a transformer vault. This will reduce the risk that workers must take while doing routine inspections. P.V.I.R is a durable, easy to control, mobile, recon robot that gathers information specific to power vaults. It will retrieve and display visual data, infrared data, data concerning air quality and a live feed from the robot's camera. The main body of the powervault inspection robot was designed by a previous team. This project will focus on designing and building the sensor package, the software package and the mechanical arm that supports the sensor package.

1.2 Motivation

The motivation behind P.V.I.R consists of a need to speed up the downtime of the power grid due to repairs and to minimize the likelihood of injuries for utility workers in a dangerous environment. An inspection robot will cut down the number of workers hurt while on the job. In 2016, the number of fatalities on the job was 5,190 people [1]. This is an average of about 14 people a day. Electrical parts such as power lines and transformers are responsible for 80% of electrocution deaths among workers [2] Inspection robots will help utility workers understand and troubleshoot problems in a safer and more efficient manner. Previous senior design teams and the EPRI have designed and tested robot vehicles that navigate in underground vaults [3]. Underground inspection robot designers are aware that the end users are utility workers with prior experience performing vault inspections and scheduled maintenance routines. Both procedures require different types of information that will be expected from the robot. The improvements offered by our design include a more efficient and convenient means of communication from the user to the robot, as well as, a more enjoyable user interface by including a mobile recording system.

1.3 Background

Previous groups have completed a few tasks related to our project. The current project builds upon a previous team's robot. The base is made of welded aluminum, so it is negatively buoyant. A picture of the existing robot base is shown in figure 1 below. The previous sensor package was mounted on a static, L-shaped arm made from wood.

More than one form of mobile robot can survey a powervault, ranging from watertight treaded vehicles to aerial drones. Each of these iterations were tested and some were equipped with control systems and sensor packages. This provided the group with the option of either building a robot from scratch or starting with a pre-existing robot as a base, which is an essential building block. The pre-existing robot bodies were intended to be waterproof, dust resistant, and durable on uneven terrains. The current tasks of the group are based on a continued design problem to create a suitable, teleoperated, mobile robot to survey power vaults of varying sizes and conditions. These conditions include but are not limited to varying vault dimensions, gas content, standing water levels, and varying locations of instrument gauges in the vault. A starting point already exists regarding the project, but the team will be focused on developing the sensor package, setting up an efficient control system, and developing a servo arm. The servo arm can adjust the height for the camera to read the gauges accurately. The final design solution is imagined to be a two person system with one operator and one observer, since most power vault inspections are typically two person endeavors due to safety regulations.

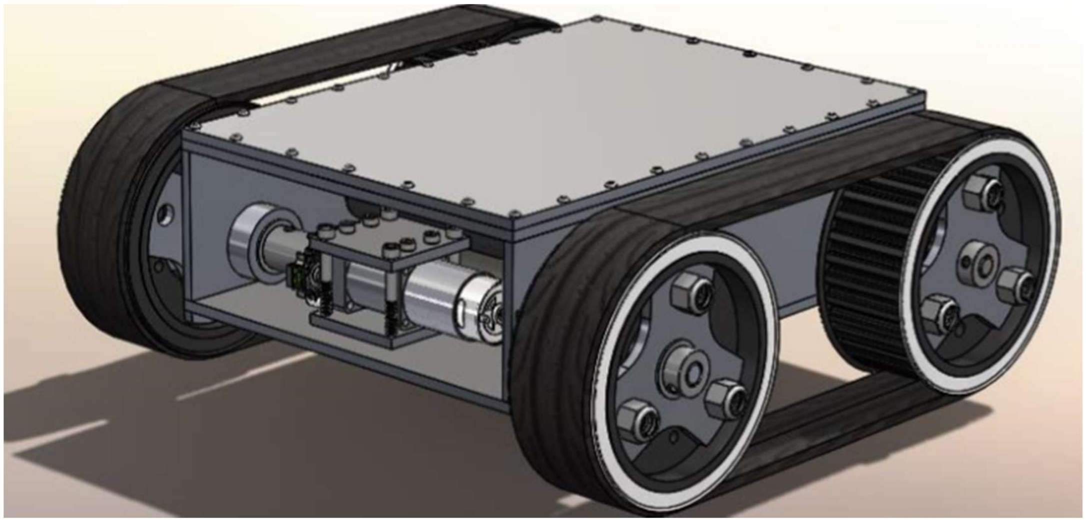


Figure 1. Robot base from previous team.

2. Project Description and Goals

The team will be prototyping a sensor package, software package, and a mechanical arm. A testing environment will also be designed and built. The sensor package will contain an IR camera, gas, temperature, humidity, and air quality sensors. A camera will be anchored on a mechanical arm which will be able to move vertically and rotate 360°. All of the sensor data will be displayed on a control GUI that the team is also developing. This GUI will be displayable on a tablet or laptop. The GUI will display all important information desired by the utility workers operating P.V.I.R to optimize the power vault procedure efficiency by providing quick access to data for educated decisions and improvised actions.

The goals for the project are as follows:

- Create an efficient sensor setup to take necessary readings in the power vault.
- Build a motor powered arm that is attached to the top of the robot, which allows the sensor setup to be raised and lowered to the needed height.
- Program a control system that manages the motor speed and maneuverability in an easily understood and secure manner that functions through a user interface an operator can interact with on a PC.

- Provide real time video feed for the observer from the robot's point of view.
- Build a testing environment comprised of 7ft tall boards with installed dummy gauges to simulate a power vault setup. This will be used to demonstrate the robot at the Design Expo.

The total estimated cost for the project development is around $266,000. This accounts for the cost of labor of five engineers working on average of 40 hours a week each for 12 weeks, the costs of the hardware purchased, and the cost of the website creation. This is interpreted as if these are research and development engineers since the solution would implemented many times over by the company. The main audience for this design solution would be utility companies such as Georgia Power.

3. Technical Specifications & Verification

Table 1: Technical Description: Hardware

Hardware Component	**Description/Function**	**Values**
DC Brushed Motors	Robot movement	Powered from ~16V, 30A power supply
Nema 17 Stepper Motors	Rotation and lift	12V, PWM inputs
Arduino Uno	Controls motors, relays sensor data	12V
Raspberry Pi 3	Communicates to GUI, night vision camera attached	5V
Mechanical Arm	3 T-slots mounted on a turntable with a moveable platform via a pulley system in the middle	Height: 6 feet Turntable diameter: 8 inches
Arduino MQ2 Gas Sensor	Detects H2, LPG, CH4, CO, Alcohol, Smoke and Propane	A reading of 10 is clean air quality. As the reading gets farther from 10 the air quality is worse
Grove Infrared Temperature Sensor	Gives an object temperature reading and ambient temperature reading	Gives temperature readings in degrees Celsius

Table 2: Technical Description: Power

Power Component	**Description/Function**	**Values**
Samsung 30T 21700 Batteries	Provide power for the robot. Rechargeable Batteries	Each battery can output 4V at full charge. Used 4 in series.
20A Fuse with Waterproof Holder	Protect circuit components from inrush current above 20 A	20A
DROK LM2577 Adjustable Buck Converter	Provide 12V power to Qunqi L298N Motor Drive Controller from 16V power supply	DC IN: 5.5-32 V to DC Out: 12V
Buck Converter Step-down Power Supplies Regulator	Provide 5V power to Raspberry Pi	10A DC IN: 12/24V to DC Out: 5V

Table 3: Technical Description: Software

Software Component	**Description**	**Testing**
Communication from GUI to Pi	MQTT	Successfully sent and receive information to and from the Pi
Communication from Pi to Arduino	Serial Connection	Successfully sent and receive information to and from the Arduino
GUI	Human interface for operator	GUI is visually appealing and gathers all of the data through MQTT
GUI: Air Quality	Air Quality display	Visual gauge displays air quality information
GUI: Video Stream	Video Stream Display	Successful video stream and display with full screen functionality

4. Design Approach and Details

4.1 Design Approach

The Graphic User Interface (GUI) was built in Python using a library called Kivy for task organization and some graphics building. the Kivy library also allows for the GUI to be exported into a mobile platform such as Android or IOS format. This adds a portable functionality to the GUI design that is necessary for its use in the field. In addition to the Kivy library, the Matplotlib library was used to create many of the live graphics seen on in the program. Numpy and Pandas were used for data formatting and processing while Opencv and the Python serial library was used for video streaming and image processing capabilities.

The GUI is currently made up of 3 pages. The first page is the login page, the second page is the main page and the third page is where the static graphs are housed. The GUI has a secure login page, a live video feed, an air quality gauge graphic, an ambient temperature bar gauge graphic, two live line plots for object temperature and audio information and two static graphs.

The GUI is the main point of human interaction to the robot therefore specific attention was given to the visual appeal and layout of the program. Figures 2, 3, and 4 below shows the final layout of the GUI. Figure 1 shows a login page. This adds a layer of security to the controls of the robot.

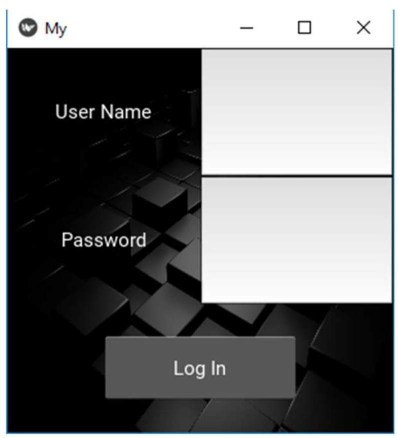


Figure 2. Login page of GUI.

Figure 2 shows the main page that follows the login page. The main page has deviated dramatically from the original proposed design, instead of having the video feed in the center of the display, the video feed has been moved to the top left corner. The original design placed the camera feed in the center because the live camera feed is the most important information coming to the GUI from the robot. The previous design was meant to make sure to get the most functionality out of the camera however the previous design did not give enough space to make reading the live graphs possible. to solve this problem, the camera feed is placed in the top left corner to give the live graphs enough space and the camera is given the ability to toggle full screen. The live graphs and the statics graphs have a toolbar that allows for them to be moved, zoomed, formatted and saved as images. The live graphs can hold 100 data points before the graph starts to scroll. The graphs can also be frozen by pressing the “f” key.

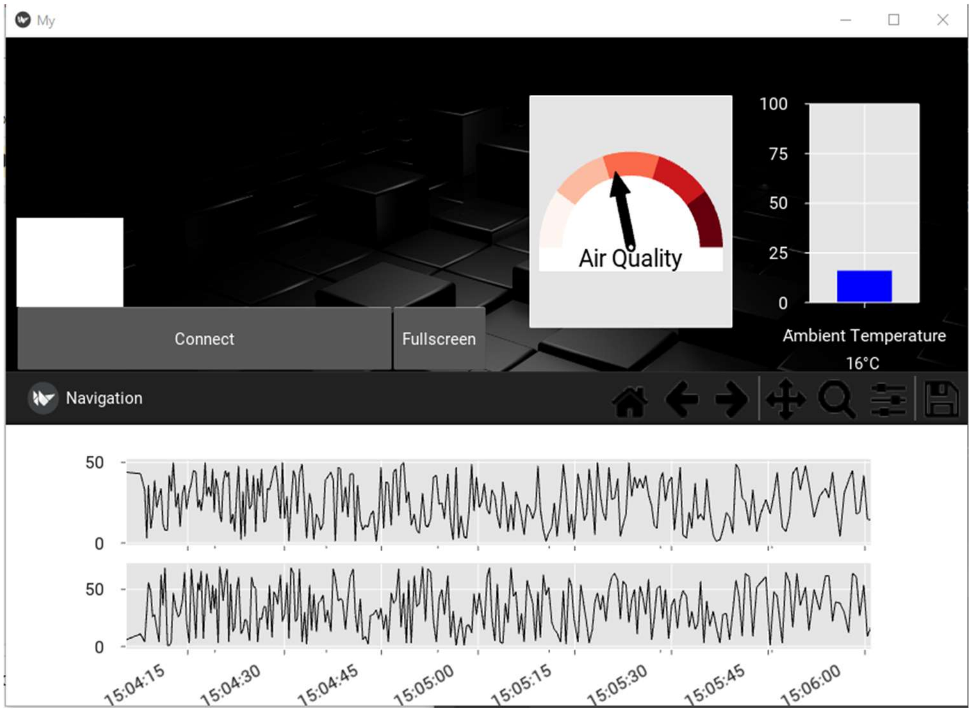


Figure 3. GUI main page.

Figure 3 below shows the static graphing page. This page can be reached by pressing the “t” key while the live graphs are selected on the main page. These graphs have the same navigation bar as the main page graphs and can also be manipulated and saved. The graph will display the last 800 data points when the button on the bottom of the screen is pressed.

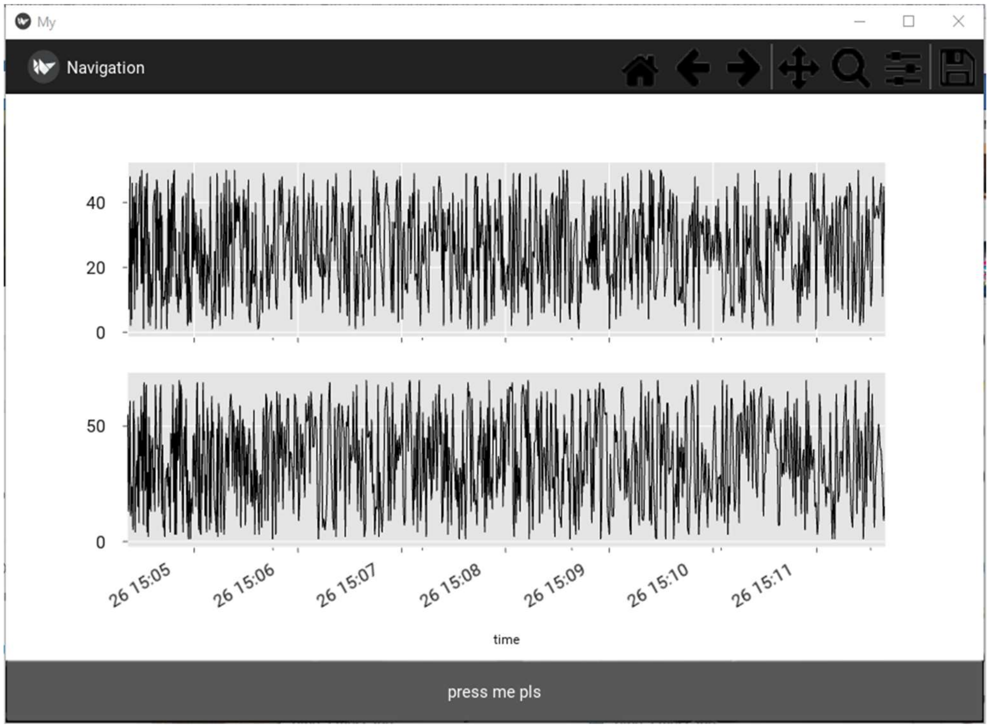


Figure 4. Static graph page.

The communication protocol between the GUI, the Raspberry Pi, and the Arduino was accomplished using the Eclipse Mosquitto client messaging service. It functions by making the device sending the signal the broker, while the device that is receiving the signal is called the subscriber. When creating the software, the Eclipse Mosquitto client was used and called as an object to implement the methods that would create brokers and have them publish to created topics and create subscribers, which would subscribe to the custom topics. Organization was necessary in order to improve troubleshooting and streamlining. Therefore, individual classes dedicated to creating clients and setting the required parameters, building subscribers that take custom strings of data and parse it, and creating publishers to send the controller input based on button actions performed on the controller. These classes and methods were implemented in a main script that initialized and executed particular methods based on, whether it was on the laptop side or the Raspberry Pi side.

When ensuring the correct data was being sent over the correct channel the topics were used. The information is sent over a certain topic; thus, it helps to separate the data. For example, the data from the temperature sensors was sent from the Arduino, read from the serial port connection between the Arduino Due and Raspberry Pi. The Raspberry Pi acts as a broker and then published the sensor data to the ‘guiData’ topic. The laptop which runs the gui would subscribe to the ‘guiData’ topic. However, since the data has been packaged in a custom string format another method is used to separate each of the data values of all the sensor readings. These values are stored in a dictionary based on the sensor type and then parsed by the gui to be displayed on the various sensor readings.

Conversely the control input sent from the Xbox controller was sent from the laptop over the broker/ subscriber connection, where the laptop acted as the broker and the Raspberry Pi acted as the subscriber. The controller input was published to the ‘controllerTest’ topic and once the subscribed data was received it was written to the Arduino Due via the serial port. To improve efficiency of data transmission over the serial port another method was developed to write information to the Arduino Due serial port only once the Raspberry Pi has successfully read from it. This confirms that the connection exists, while reading the sensor data from the port and shortly after wrote the controller input to the Arduino Due as well. Below a diagram can be found depicting the relationship from broker to subscriber.

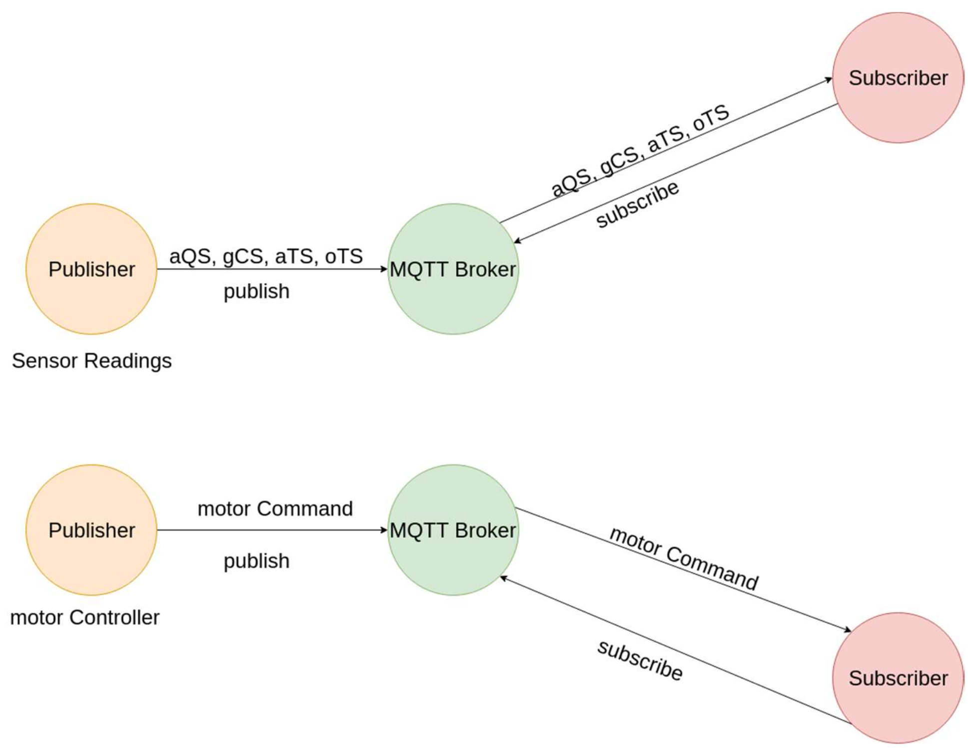


Figure 5. Communication Scheme between the laptop and Raspberry Pi

The Arduino sends and receives information via serial communication. Sensors are read using analog inputs on the Arduino. The movement motors are controlled by vex motor controllers that take a pulse width modulation (pwm) signal from the Arduino. The stop pwm value is 185, the forward pwm value is >185 and the reverse value is <185. In order for these values to be accurate the pwm frequency of the Arduino module needs to be around 960-980 Hz. The L298N motor controller boards control the lift and rotation stepper motors. The Arduino uses the stepper library to move the stepper motors. The temperature sensor needs for the Arduino to have an analog reference of 1.1V in order for it to be accurate. This was difficult to do on the Arduino DUE so the team ended up using an Arduino UNO to control the robot.

Figure 6 below shows the final design setup. The Xbox controller was not used in the final design because of the unreliable communication. Also, because the CO2 sensor needed a board that was not purchased, the CO2 sensor also was not used. Figure 6 shows the top-level function of each piece of the project.

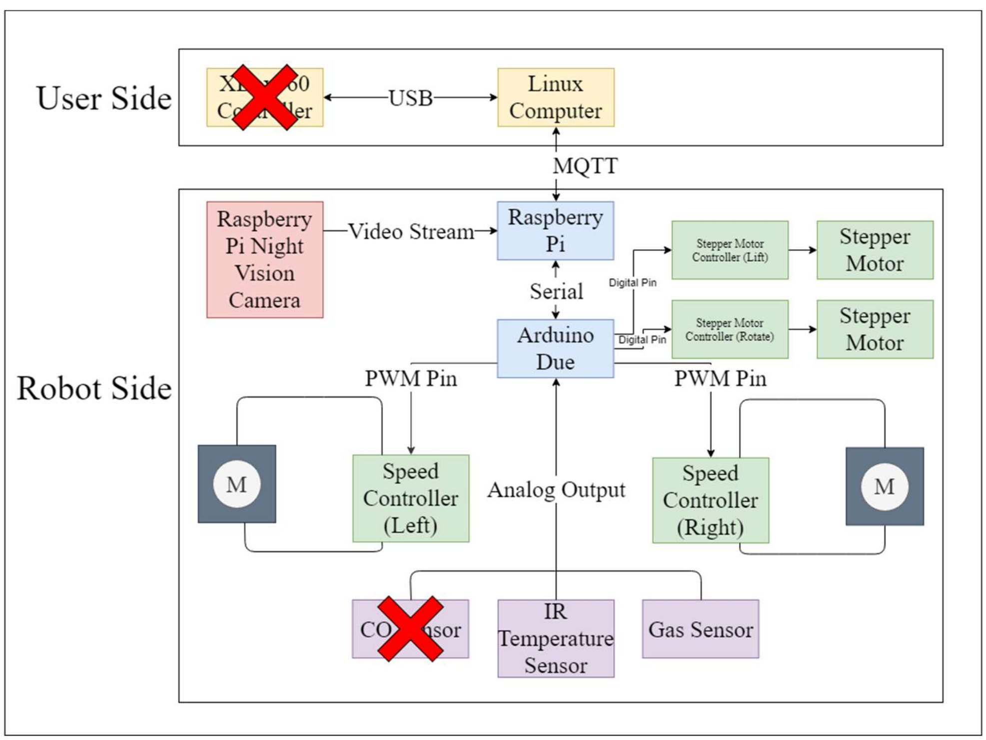


Figure 6. Final top-level design

The sensor package consists of the IR temperature sensor, a gas sensor, and the Raspberry Pi Night Vision camera. The IR temperature sensor used provided ambient and object temperatures. The MQ2 gas sensor was able to detect H_2, LPG, CH_4, CO, alcohol, smoke and propane. The Raspberry Pi camera was chosen due to its potential to work well in low light.

Everything was powered off the same 16V (4 batteries in series) power supply. The Raspberry Pi needed a +12V to 5V converter to prevent damage the Pi because the power and ground GPIO pins were used and do not protect against power surges. The Arduino, and stepper motor controllers required 12V to operate, so buck converters were used. A 20A fuse was used to protect the circuit in case the motors ended up running at full power. A full schematic of the design can be found in appendix D.

The electrical components mostly all fit inside the body of the robot with the exclusion of the Raspberry Pi, IR temperature sensor, gas sensor, camera, and stepper motors. All of the wires for these components were bundled together and fed through a waterproof cable gland through the thick top plate of the robot. To prevent tangling, the wires that lead to the lifting platform were coiled with a heat gun.

The mechanical arm design consisted of three, 6' tall, T-slots mounted on top of an 8" turntable. Figure 7 below shows a design overview of the arm. The proportions of the arm are not an issue because the weight of the robot offsets the probability of the robot tipping over. The arm was able to turn the entire platform by using a NEMA 17 stepper motor and a large flexible O-ring. Another stepper motor was mounted to the base of one of the T-slots and provided the lift functionality via a pulley system. A lot of machining was required for this project and was out of the team's scope of expertise. The CAD drawings for the mechanical arm and sensor package is included in appendix C.

Figure 7. Mechanical arm design overview

Unresolved issues include full integration of all the parts. During the testing period, each of the individual components worked smoothly. The stepper motor could turn the turntable providing rotation to the arm. The other stepper motor could lift and lower the sensor platform via a pulley system. The Raspberry Pi could stream video capture to the GUI with minimal lag. The DC motors could move with PWM inputs through the Arduino Uno. The sensors could send data to be interpreted by the GUI. However, when attempting to thread all these processes together, problems developed due to lack of processing power with the Raspberry Pi and the Arduino Uno.

It is recommended that future teams find a different alternative for controlling robot movement. It requires too much time for an input on the GUI side to then be read by the Arduino.

Some sort of Bluetooth module could be very helpful in streamlining the inputs directly to the Arduino to ensure inputs are received and executed promptly.

4.2 Codes and Standards

1. Wireless: IEEE 802.11
    - This standard ensures wireless local area network (WLAN) communication between separate systems are compatible [4].
2. Serial Bus Communication: RS-232, RS-422 and RS-485
    - RS-232 is the oldest and most widely used serial communication protocol and is used in most laptop serial interfaces. RS-232 was developed in 1962 [5].
    - RS-422 and RS-485 are newer and faster communication protocols [5].
3. USB 1.1 2.0, 3.0/3.1
    - USB 1.1 was the first edition of the USB standard, developed in 1988. Since then, the standard has been upgraded to 2.0 and 3.0/3.1 which have faster speeds [6].
    - This allows for universal ports that enable communication.
4. MQTT (Message Queuing Telemetry Transport): ISO/IEC PRF 20922
    - Client Server publish/subscribe messaging transport protocol. [7]

4.3 Constraints, Alternatives, and Tradeoffs

Some alternative designs that had varying interactions between microcontrollers, sensors, and the user interaction (VR, Tango Tablet, or Kinect Camera) were considered. The iterations were compared by looking at the pros and cons. One of the designs ruled out uses both a Raspberry Pi and Arduino. This is shown in figure 8. Using OpenCV on a Raspberry Pi is possible but requires a large amount of space on the microSD that serves as the boot disk for the microcontroller, thus requiring the user to adjust the partition and remove many large libraries that will be unnecessary in the project. Webcam integration with a Raspberry Pi via USB hub is possible, but the stream rate has an extreme amount of latency and would cause the system to be slightly behind for the observer versus the remote operator. A Pi camera functions better, but when dealing with the conditions in the vault the Pi camera does not have a very large chance of surviving without a protective casing, thus a GoPro has been considered. This may cause the original problem to reoccur due to the GoPro not being designed for a Raspberry Pi. Interfacing the Raspberry Pi with a microcontroller that runs C++ is possible already via a serial connection using a master slave connection. There are a lot of pre-existing examples. The aspects of the project left to verify are the interfacing the Tango Tablet with the Raspberry Pi, integrating the VR apparatus with the Raspberry Pi and threading all of these processes together so the design parts can operate in parallel. This iteration was not used due to the large number of components that will have to work and communicate with each other as well as draw power.

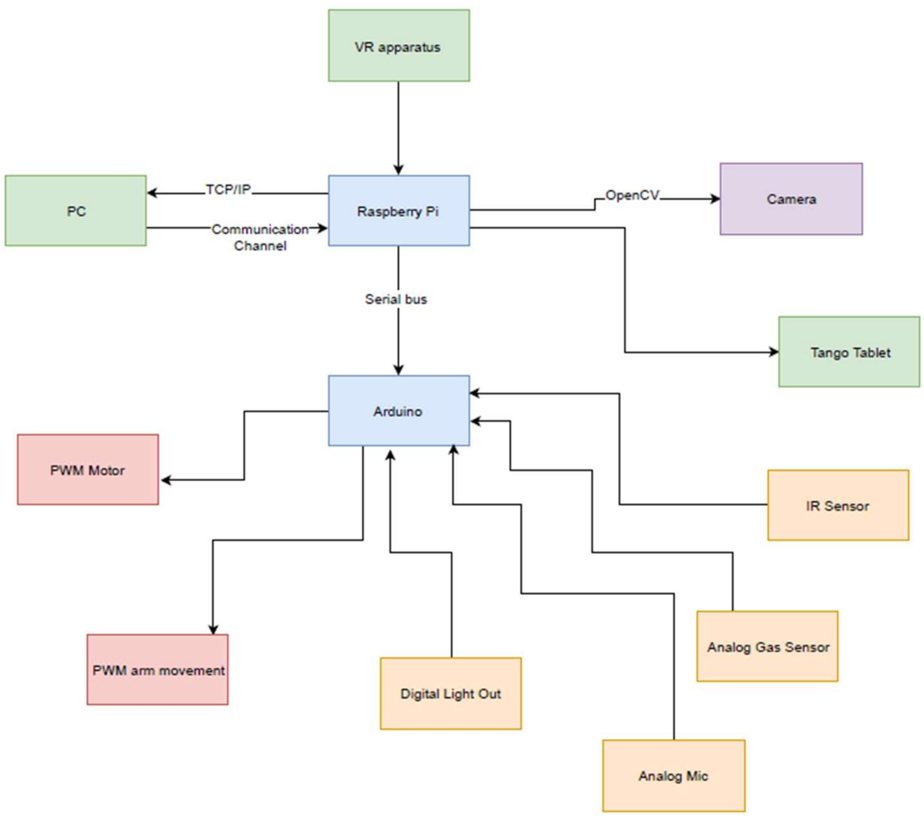


Figure 8. First iteration showing all connections using a Pi and Arduino with VR

Another iteration that was taken into consideration has every device and every I/O of the system had to go through a single Raspberry Pi as shown in Figure 9. While the exclusion of the Arduino microcontroller implied one less communication platform to worry about, there were a few trade-offs. Major trade-offs with this approach were that Python was the main programming language, which not many of us are familiar with. Only one device controlling all aspects of the robot increases the concern for scheduling parallel work. Furthermore, an ADC would have been necessary to receive input from the analog sensors.

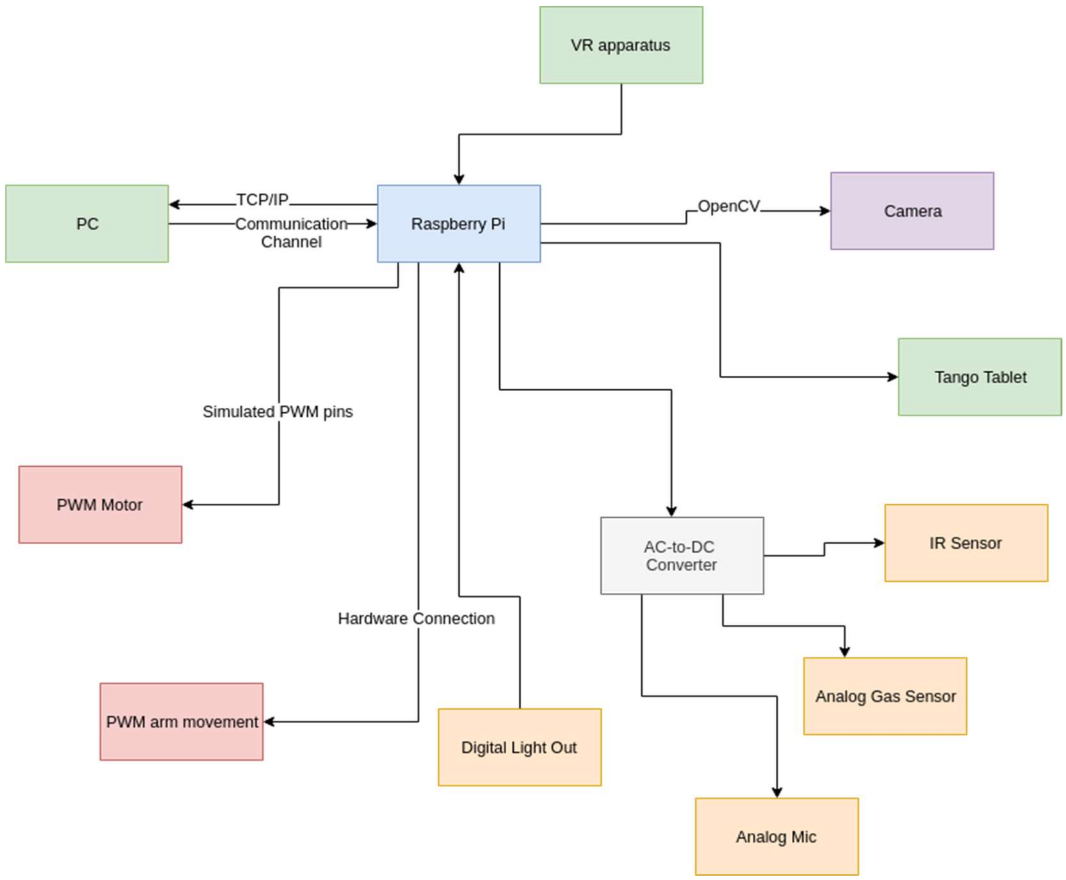


Figure 9. Second iteration showing all connections going through a single Pi.

Another design iteration not selected utilizes both the Raspberry Pi and the Arduino. In this iteration, the VR headset was not included. This iteration is shown in Figure 10. It should be noted that the VR headset will require more labor as another device and user interface will need to be designed and coded. This will result in a higher overall cost. The trade-off from the exclusion of the VR headset was the novelty brought by the virtual experience from the VR headset.

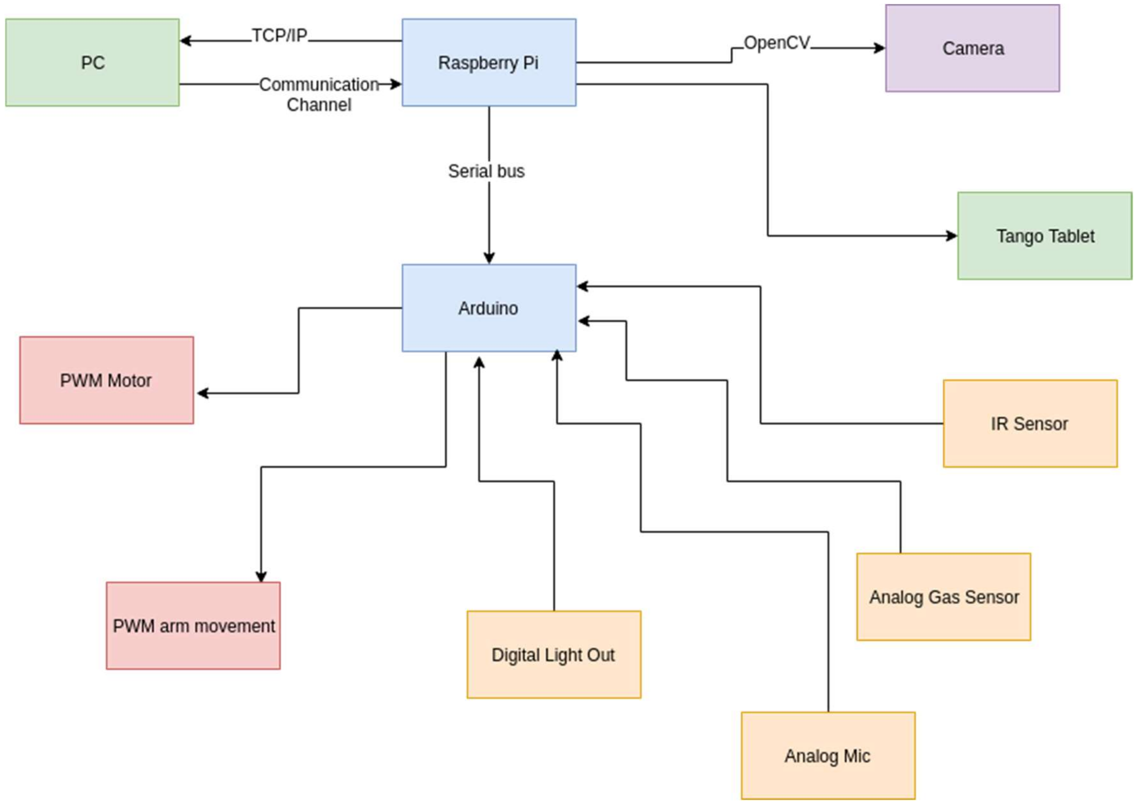


Figure 10. Third iteration showing all connections using a Pi and Arduino without VR.

The next design iteration not selected has a single Raspberry Pi controlling every aspect of the project much like the iteration shown above in Figure 8. The difference is that the VR headset is not included in this design shown in Figure 11. Once again, the major trade-off from this approach is a high dependability on the Raspberry Pi.

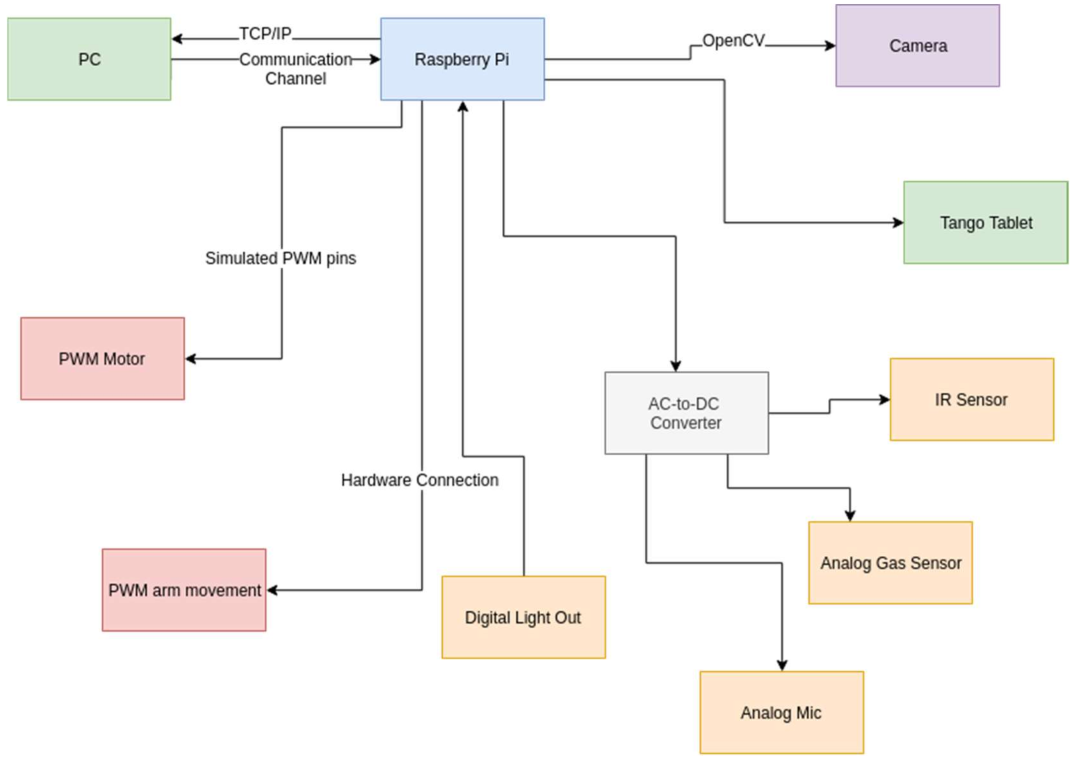


Figure 11. Second iteration showing all connections going through a single Pi without VR.

The last design iteration utilizes the Microsoft Kinect sensor instead of the Tango tablet as shown in figure 12. The Microsoft Kinect sensor is the same RGB-D camera that the Tango tablet uses. In order to access the information on the Tango tablet, one would have to design a C# program using the Unity 3D engine. This is a large effort considering the 3D mapping capability of the robot is peripheral. The Kinect sensor can be accessed by the Raspberry Pi directly and processes the image information itself. This makes it easier to export the information obtained from the depth camera to the GUI running on the remote computer. The Python program can utilize Libfreenect or OpenKinect to process the 3D information. Using the Kinect sensor, however, increases the computational load of the Raspberry Pi and makes prioritization of information an issue.

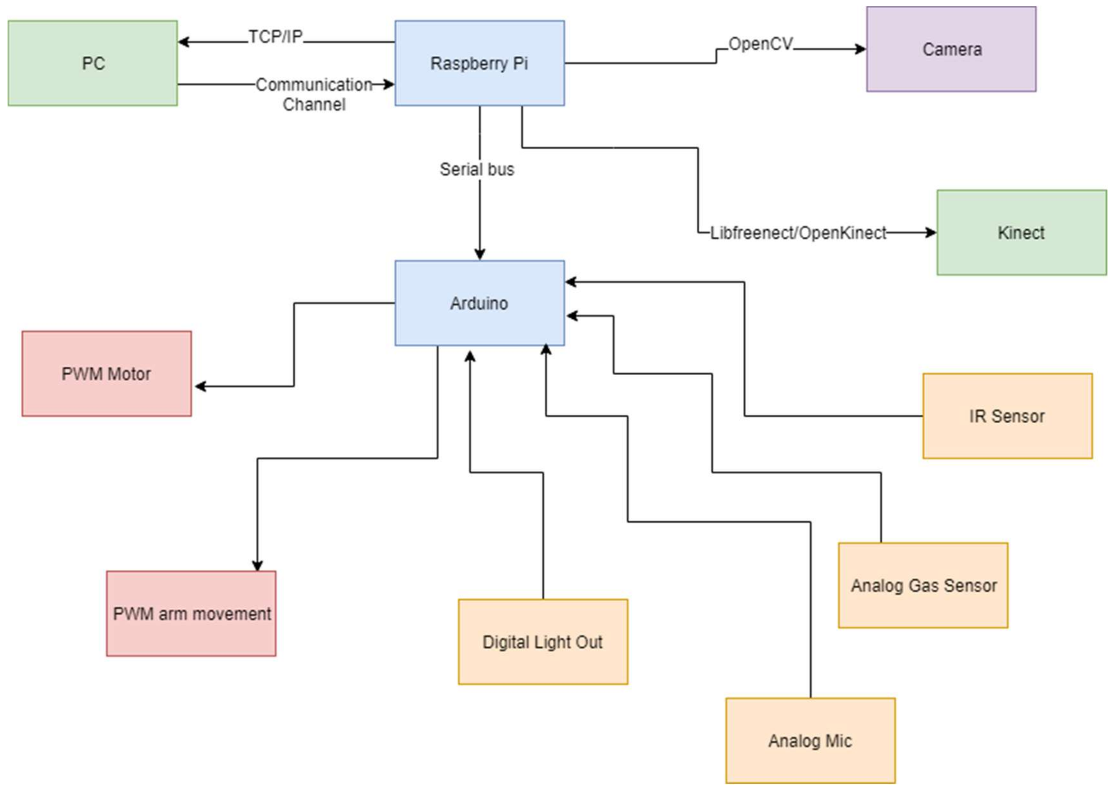


Figure 12. Last iteration showing Kinect.

At the beginning of the design process, the team did not elect to go with only one microcontroller/processor. This design was not chosen because of lack of processing power. There were too many motor controllers and sensors for just the Raspberry Pi to handle by itself which is why the dual processor design was chosen.

An alternative design considered for the arm mechanism was to use a lead screw which is well known for being a precision machine component. The benefit of using a lead screw is that it is a perfectly straight, higher load, with no backlash. The higher load benefit characteristic of the lead screw permits transfer of load without jamming. Nevertheless, accurate positioning of the sensor package was not possible due to the inherent rotating nature of the mechanism. The implementation of more complexity to the design was necessary to be added to enable the appropriate 360-degree view specified under our goals and technical specifications. Furthermore, a price comparison analysis showed that 6 feet lead screws were not within our budget and T-slots were a much better alternative for that matter.

Another component that was considered during the design brainstorming phase was deciding what type of motor to use. While RC servos were an option, they were discarded because they lack a wide angle of rotation, have less holding torque, take more current. Next, a DC motor was considered because they were robust, cheap, and use less current. However, like servos their torque is lower, and have bad speed control. As a result, we opted to go with the stepper motor for the unlimited angle of rotation, step control, lower current, and a higher holding torque.

Various Gear iterations were considered to create a rotating platform for the arm. The vast majority of the iterations were centered with the end goal of improving the output torque of the rotating platform. A planetary gearbox system was considered where the ring is fixed while the sun gear has freedom of rotation and this is where the rotating platform is connected. This planetary gearbox system enables a higher output torque. While the gearbox system was a design with a variety of great features, the team came to the conclusion that the design should be kept simpler and the turntable connected to a round belt design was sought after.

The table below represents a points system used to rank three different alternatives considered for the rotating mechanism. The points are awarded based on ease of execution where five points are given to each selection factor. from here, each solution is awarded from one point low quality execution to five points being exceptional outstanding execution. Two selection factors were applied to the three solution alternatives. We decided to go with the turntable design for its’ optimal effect and low cost.

Table 4: Rotating Mechanism Alternatives Ranking

Solution	Effect	Cost	Points
Turntable	High	Low	8
Gears	Medium	Low	6
Hinge Pulley Mechanism	Low	High	4

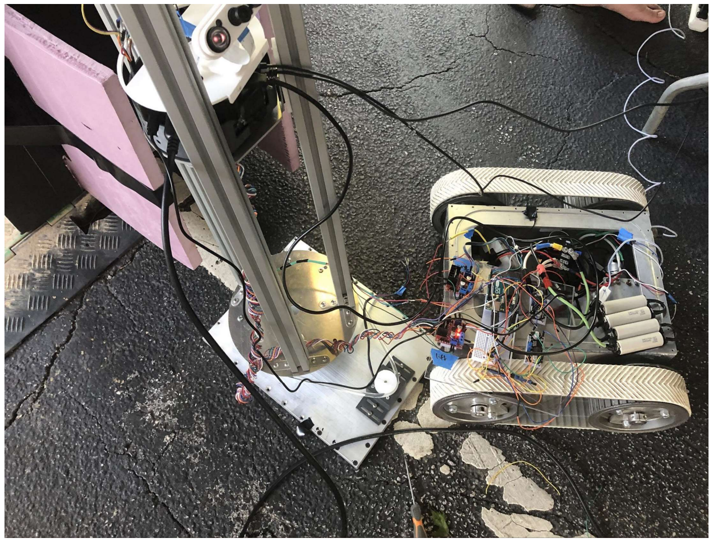


Figure 13. Inside view of robot at Expo.

Constraints limiting the design options and the overall robot performance were the robot size cannot exceed the size of the manhole. The limited budget to purchase parts limiting our freedom to buy higher quality parts with perhaps better performance, which also limits our ability to buy extra accessories for the robot. The robot needs to be flexible and intuitive for the user which implies that the user interface needs to be accessible without a steep learning curve for non-technical utility workers. Robustness, waterproof capabilities, and wireless communication through walls were other major constraints taken into consideration primarily due to obstacles on the ground and the hazardous environment P.V.I.R. will be exposed to. These constraints were driving motives to lower the robot at the manhole and use an antenna as a means of wireless communication between the robot and the manhole location. Another constraint that was taken into consideration was that the robot needed to be waterproof. Waterproofing during prototyping stage was difficult because it caused for organization during the assembly process to be inconvenient.

5. Schedule, Tasks, and Milestones

The Gantt chart with an estimated timeline of project milestones is shown in appendix A. The Gantt chart lists break down of the milestones including the designing, planning, building and testing of the mechanical arm, sensor package, and user interface. The Gantt chart shows the start time, end time, and duration of all of the tasks. Next to each task is the name of the student that will take lead on that task. Table 4 below also list the team members, their title if applicable and the task they will take lead on.

Table 4: List of tasks that each member of the team will take lead on

Task Lead	Tasks
Stephanie Chan - Team Leader	• Team Leader tasks • Order motors/controllers for control system • Order materials for mechanical arm • Order sensors • Assist with GUI and mechanical arm assembly
Elizabeth Fuller	• Sensor package schematic • Control system design • Testing sensor parts • Power
Adrian Muñoz	• Mechanical arm CAD design • 3D prototype of mechanical arm • Mechanical arm assembly
Nelson Raphael - Web Master	• Project website • Communication between microcontrollers • Integrating control system into robot shell • Test Environment CAD design
Lemek Robinson	• GUI Design • Communication between microcontrollers • Testing communications with GUI
All	• Scrapping robot and testing parts for usability • Assemble mechanical arm onto robot • Soldering PERFBoard • Mounting PERFBoard to robot • Test Environment Build

Team leader tasks not listed in the table include: sending weekly reports, updating gantt chart for the weekly meetings, keeping track of tasks and when they need to be started or finished to progress the project, keep track of in class due dates, and being the point of contact for the Design Expo. Deliverables include Oral Presentation, Final Design Proposal, Final Project Demo, and Design Expo. All members of the team will also be responsible for contributing to the deliverables and the final testing of the robot.

Appendix B includes a PERT chart that identifies the critical path. The tasks included in the critical path are associated with the development and testing of the software needed. The technical risk associated with the software component was predicted because the team's skill set was less competent in the software. An unforeseen technical difficulty was the mechanical arm CAD design. The members of our team had very little to no experience in CAD drawing and design so that became the largest technical rick of the project.

6. Final Project Demonstration

The robot was demonstrated with the testing environment, as seen in Figure 14, at the Capstone Design Expo on April 23, 2019. The robot was judged on the following aspects:

- Robot Arm Functionality
    - Rotation
    - Lift
- GUI Demonstration
    - Live Video Stream
    - Sensor Data
- Control Functions
    - Control Motors with Xbox Controller
    - Control Motors with Keyboard

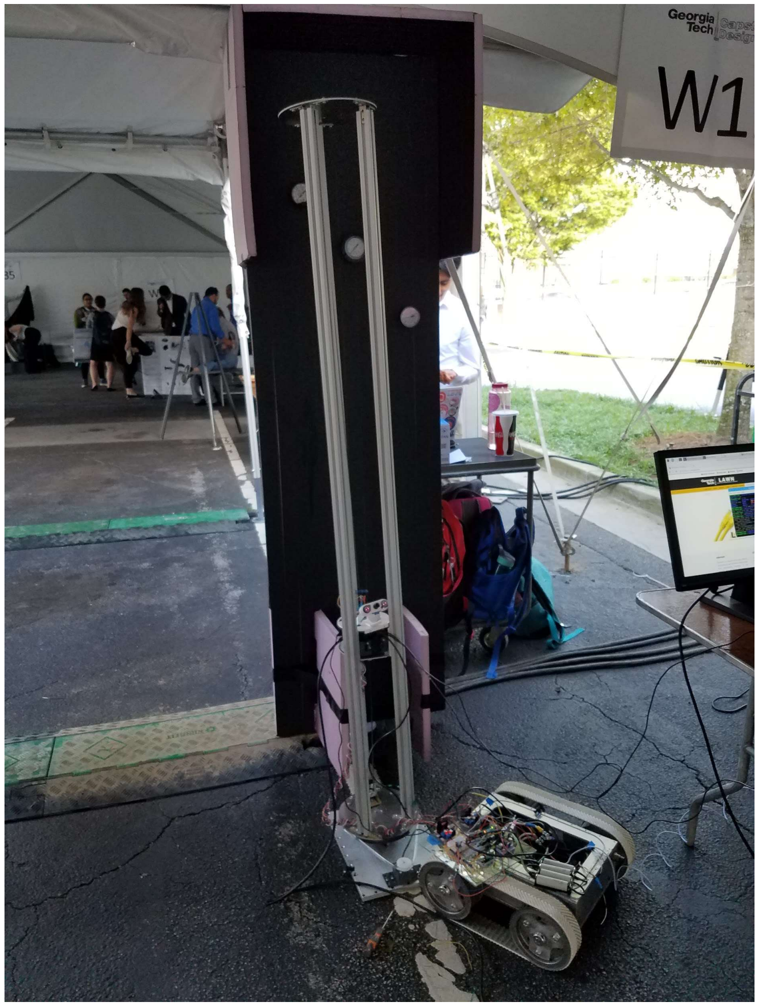


Figure 14. Robot with testing environment.

A list of the original specifications is shown in Table 5. The team was able to meet all of the specifications as promised at the beginning of the semester except for a microphone on the sensor package. The plan for the sound recording hardware was to use the existing microphone from a previous team but in testing we found that the microphone was broken. At that point we had not budgeted the money to buy a new one so the team decided to exclude that hardware from the design. However, the designed GUI is programmed to display audio data. This way when the robot is used in the future by utility workers, they would just have to plug in their own microphone that would be able to pick up corona discharge sounds.

Table 5: List of Original Specifications

Mechanical Arm Specs	
Height: Variable height from 1 - 6 feet	✔
Degrees of Freedom: 2 degrees of movement	✔
Base Size: Fits in a 760mm Diameter Manhole	✔
Sensor Package Specs	
Video: Can Stream Video to GUI	✔
Gas Sensor: Check Air Quality	✔
IR Thermal Camera: Record Thermal Images	✔
Microphone: Record Sounds	
GUI Specs	
Mobile: Usable on Mobile Platforms	✔
Control Capabilities: Remotely Controllable Robot	✔
Logging: Log Information	✔
Data Streaming: Real-Time Data Streaming	✔

A majority of the prototype testing of the mechanical arm was done during the CAD design process. The mechanical arm was designed with the purpose to lift the sensor package using two idlers and one pulley system. When testing, team members noticed unwanted friction with one of the idlers at the top of the arm that caused the belt to slip at the motor. Therefore, that idler was removed from the design in order to reduce the friction of the timing belt when moving the sensor package vertically. Otherwise, the demonstration of the robot was like previously mentioned. To enable transportation to the Capstone Design Expo, the body of the robot and the arm were carried separate from each other and assembled there.

The sensors that made up the sensor package were tested individually using the Arduino Uno and printed in the serial monitor of the Arduino. This is how the DUE inability to have a 1.1V analog reference was discovered.

In order to test each section of the GUI random numbers were used to take the place of the incoming data from the Arduino. Using this, decisions about maximum and minimum values, the number of points to graph before scrolling and calibrating the gauges were made. The video display was also prototyped outside and inside the GUI to find the fastest and most resilient method to displaying the image information.

Links to videos from the demo are as listed in table 6:

Table 6: Demo Video Links

Video Title	Video Link
mqtt Setup	https://youtu.be/shiEYz6oGXM
mqtt Demo 1	https://youtu.be/OsKsYtkzx7M
mqtt Demo Keyboard	https://youtu.be/86Fh0tne35E
Object Oriented Programming Demo	https://youtu.be/ZXin4JpDj1
mqtt GUI Depackage Demo	https://youtu.be/vKcjTv2oNbM
mqtt Controller Sender Demo	https://youtu.be/UA4RlVIEmt4
Robot Movement Test using Arduino Due	https://youtu.be/RKCFxmx3-9Q
Robot Movement Part 2	https://youtu.be/OAb53kK3GZ4
Elevation Test	https://youtu.be/Chi9t13gta4
Nema Motor Serial Test	https://youtu.be/ieaHjYvU39A
Serial Demo	https://youtu.be/Q-efTZSCp5s

7. Marketing and Cost Analysis

7.1 Marketing Analysis

Robots and drones are becoming a more popular option for industrial inspection [8]. Regular inspection is important to maintain equipment and worker safety. Many robots today can inspect pipeline, buildings, equipment, and more. Because of modern day technology inspection robots are faster and more efficient than the average utility worker. They are safer because the utility worker will no longer have to go into a power vault or other potentially dangerous environments. AETOS, GE Inspection Robotics, and Honeybee Robotics are in the top companies in the inspection robot market [9]. Currently AETOS has a wide range of unmanned land inspection robots. Their robots are equipped with different types of sensors and imaging technology that can inspect for corrosion, cracks, defects and other damages [10]. The AETOS robot uses nondestructive testing (NDT) methods and can be configured to conduct different types of land based inspections, specifically for oil and gas, petrochemical, and infrastructure. Like the AETOS robot, P.V.I.R also utilizes many different sensors and imaging systems for inspection. The AETOS website does not list the different sensor capabilities but P.V.I.R uses gas sensors, infrared array sensors, sounds sensors, a camera and an RGB-D camera for inspection. P.V.I.R also uses NDT methods to inspect the power vault. Unlike the AETOS land robot, P.V.I.R is not unmanned and is instead controlled using a gaming system controller. This allows the utility worker to have full control over what they want to see in the power vault and where they want to drive it. Additionally, while the AETOS land robot can be configured for different types of inspections, the P.V.I.R is designed specifically for the utility workers to inspect the power vaults that they are working in and to ensure that they are in a safe work environment.

7.2 Cost Analysis

The P.V.I.R is made up of three parts, a robot base, a mechanical arm, and a sensor package. The project also includes a test environment that mimics a transformer that would be found in a typical powervault. For the purposes of this project, the cost of the robot base development is not accounted for because the base was developed by a previous team. The needed components for the development of the mechanical arm and robot lid replacement in shown in table 7. The part used for the sensor package is shown in table 8. Other material used for the development process along with the testing environment is shown in table 9.

Table 7: Mechanical Arm and Robot Lid Cost

Type	Item	Qty	Unit Price	Total Cost
Arm	8" Lazy Susan	1	$17.00	$17.00
Arm	6061 Aluminum Sheet 1/8" Thick, 12" x 24"	1	$47.11	$47.11
Arm	T-Slot Slider	4	$5.50	$22.00
Arm	T-Slots 6ft (72in)	3	$19.23	$57.69
Arm	Timing Belt, Connectors, Pully, etc.	1	$12.99	$12.99
Arm	L - Brackets 2" x 2"	5	$0.92	$4.60
Arm	1 16 count Alternate L-Braket Set from Amazon	1	$9.99	$9.99
Arm	6061 Aluminum Sheet 1/4" Thick 12" x 12" For Arm base	1	$43.12	$43.12
Arm	Nema 17 Stepper Motor	2	$19.90	$39.80
Arm	Pulley for 1/4" Diameter Round Belt, 1.5" OD	1	$10.38	$10.38
Arm	Round Belt 1/4" Diameter - 33" circum.	1	$9.26	$9.26
Arm	6061 Aluminum Sheet 1/8" Thick, 12" x 12"	1	$27.71	$27.71
Arm	Zinc-Plated Steel Corner Bracket, 5/8" x 1" x 1/2"	4	$0.63	$2.52
Arm	Zinc-Galvanized Low-Carbon Steel Rod - 3ft	1	$2.70	$2.70
Robot Top Lid	GiBot Cable Glands - 25 Pack Plastic Waterproof 3.5-13mm Cable Glands Joints Wire Protectors, PG 7/9/11/13.5/16, Black	1	$9.49	$9.49
Arm	304 Stainless Steel Screw and Nut 515pcs, M3 M4 M5 Metric Socket Head Bolt and Nut Assortment Set	1	$19.99	$19.99
Arm	Male-Female Threaded Hex Standoff	4	$4.11	$16.44
Arm	10-32 x 1-1/2" Hex Head Cap Screw Bolts, External Hex Drive, Stainless Steel 18-8, Full Thread, Bright Finish, Flat Point, Quantity 50 By Fastenere	1	$9.69	$9.69
Robot Top Lid	6061 Aluminum 1/2" Thick x 10 inch wide - 2 ft	1	$73.16	$73.16
Arm	Qunqi L298N Motor Drive Controller Board Module Dual H Bridge DC Stepper For Arduino	2	$6.89	$13.78
Robot Top Lid	#6-40 x 3/4", Flat, Socket Head Cap Screw, Alloy Steel, Steel, Black Oxide Finish, 100PK	1	$9.84	$9.84
Robot Top Lid	#6-40, Tap, Right Hand, Plug, 2 Flutes, High Speed Steel, TiN Tap Finish	1	$9.46	$9.46
Arm/Robot Lid Total Cost				$468.72

Table 8: Sensor Package and Power Components

Type	Item	Qty	Unit Price	Total Cost
Sensor	Arduino Gas Sensor	2	$7.53	$15.06
Sensor	Grove - Infrared Temperature Sensor	2	$9.90	$19.80
Information Storage	Sandisk 32gb microSD card	1	$8.90	$8.90
Sensor	Parallax Carbon Dioxide Sensor MQ-7	2	$5.99	$11.98
Power	Samsung 30T 21700 3000mAh 35A Battery	8	$7.99	$63.92
Power	EFAN 4 Channel Battery Charger	2	$9.97	$19.94
Power	IM2X20700 MosMax dual 20700/21700 battery tray/holder/sled injection molded	4	$5.25	$21.00
Control	Xbox 360 Controller, Wired USB Controller for PC & Microsoft Xbox 360 Black	1	$16.99	$16.99
Power	Aweking Waterproof DC/DC 12V Step Down to 5V 3A 15W Voltage Buck Converter Regulator Transformer Power Supply for Car Truck Vehicle CE listed	1	$16.59	$16.59
Power	AutoEC 20A Inline ATC ATO Waterproof Fuse Holder (1 Set)	1	$7.99	$7.99
Cable	Curlynet Spiral-Coiled Ethernet Cable (2-9 ft.)	1	$12.99	$12.99
Cable	Tripp Lite 6 ft. Hi-Speed USB 2.0 to USB Micro-B Cable (M/M), Coiled, USB Type-A to Micro-B (U050-006-COIL)	1	$7.94	$7.94
Sensors/Microcontroller				$98.72
Power/Cabling				$150.37

Table 9: Lab testing and Test Environment Components

Type	Item	Qty	Unit Price	Total Cost
Testing	30 Pcs Double Sided PCB Board Prototype Kit for DIY, 4 Sizes by Paxcoo	1	$10.85	$10.85
Testing	Jumper Wires Male to Male, Solderless Breadboard and Arduino	1	$7.89	$7.89
Testing	Breadboard Solderless - 3 ct 400 Pin with 4 Power Rails for Arduino and Pi	1	$7.99	$7.99
Testing Enviroment	2-1/2" Test Pressure Gauge, 0 to 4000 psi	1	$1.61	$1.61
Testing Enviroment	2-1/2" General Purpose Pressure Gauge, 0 to 30 psi	1	$3.59	$3.59
Testing Enviroment	2-1/2" General Purpose Pressure Gauge Liquid Filled	1	$4.85	$4.85
Testing Enviroment	FOAMULAR 150 1 in. x 4 ft. x 8 ft. R-5 Scored Square Edge Rigid Foam Board Insulation Sheathing	1	$19.55	$19.55
Lab Testing				$26.73
Testing Enviroment				$29.60

The total cost for the development and build of the mechanical arm, sensor package, and testing environment is about $775. The breakdown of the total cost for each portion of the project is shown below in table 10.

Table 10: Total Cost Breakdown

Type	Total Cost
Arm/Robot Lid	$468.72
Sensors/Microcontroller	$98.72
Power/Cabling	$150.37
Lab Testing	$26.73
Testing Enviroment	$29.60
Total Cost	**$774.14**

The total cost of labor shown in Table 11 was determined with an assumed salary of $60,000 per engineer. This gives an average of $28.85 per hour. Assuming an engineer is working the typical 40 hour work week for 12 weeks, the total number of hours worked will be 480 hours. This gives a total cost of labor to be $13,848.00.

Table 11: Total Labor Hours and Cost Per Engineer

Project Areas	Labor Hours	Labor Cost
Project Design		
Hardware Design	20	$577.00
Software Design	80	$2,308.00
Mechanical Arm Design	15	$432.75
Build and Testing		
Debugging	60	$1,731.00
Mechcanical Arm Build	20	$577.00
Control System Build	10	$288.50
Testing	90	$2,596.50
Robot Assembly	30	$865.50
Demo Preparation		
Presentation	3	$86.55
Reports and Proposal	10	$288.50
Website	40	$1,154.00
Weekly Group Meetings and Lab time	102	$2,942.70
Total Labor Cost	480	**$13,848.00**

The development cost shown in Table 12 is calculated using 30% of the total labor cost as the fringe benefit and 120% of materials and labor as the overhead. The total development cost comes out to be $265,592.00.

Table 12: Development Cost

	Cost
Parts	$775.00
Labor	$69,240.00
Fringe Benfits	$20,772.00
Subtotal	$90,787.00
Overhead %	$84,018.00
Total	**$265,592.00**

Over a 5-year period, there will be 4,950 units produced and sold. This number is chosen because there are about 3,300 power utility companies in the United States [11] and it is assumed that the average number robots per company is 1.5. The components that make up the sensor package and mechanical arm can be bought in bulk for a discounted price of $700 per unit. The manufacturing employees will be paid $15 an hour to assemble the robot. The engineers responsible for testing the robot will be paid $28 an hour. The sales expense will make up 10% of the final selling price of $2,500.00. This selling price is competitive with other inspection robots on market currently. SuperDroid Robots have an inspection robot with the cost of $1,836 [12]. As shown in Table 13, the estimated profit per unit is $310.56. This is calculated with fringe benefits being 30% of labor and overhead being 120% of labor, materials, and fringe benefits. The estimated amortized development cost is $25. Over a 5-year period, the total profit will be $1,537,272.00.

Table 13: Selling Price and Profit per unit with 5-year profit for 4950 units

	Cost
Parts Cost	$700.00
Assembly Labor	$60.00
Testing Labor	$14.00
Total Labor	$74.00
Fringe Benefits, % of Labor	$22.20
Subtotal	$870.20
Overhead, % of Material, Labor, & Fringe Benefits	$1,044.24
Subtotal, Input Costs	$1,914.44
Sales Expense	$250.00
Amortized Development Costs	$25.00
Subtotal, All Costs	$2,189.44
Profit	$310.56
Selling Price	**$2,500.00**
5-Year Profit	**$1,537,272.00**

8. Conclusion

Overall, the project did not work as well as intended due to the aforementioned constraints and issues. The robot was unable to hand all the processes together but was able to individually execute each one. The presentation of the robot at the Capstone Expo was troublesome. During transit, parts in the arm shifted causing extra friction in the sensor package platform preventing it from sliding smoothly. The pin connections in the stepper motor controllers were insecure. Therefore, once the top plate was moved onto the robot; the jumper wires would disconnect from very little jostling. Having extra slack in the wires would have been helpful in this situation.

One thing that the team would have done differently is the design of the mechanical arm. A foldable arm design would have made for easier transportation and the lifting function for the sensor package platform would have been smoother. However, scissor lifts are hard to produce because of the precision needed while machining and pre-made ones were out of the budget range. Another thing the team would have done differently is how the parts communicated to one another. Having the controller inputs go through the GUI to the Pi and then to the Arduino caused a lot of problems. Inputs were delayed and also dropped entirely. A more reliable way of communication directly to the Arduino would have helped the control system run smoother.

A lot of different tools were used for the first time by team members. Having a project that was half mechanical work resulted in a lot of learning points. More parts should have been cut with the water jet to increase precision instead of using hand drills to make holes when needed. The team did not allow for enough time for hardware and software integration. This complex of a system should have had more time allotted for testing in general. This project felt like a 3-in-1; the GUI, the robot control with an Xbox controller, and the mechanical arm; some of which required a lot of effort and was not in the team's area of expertise. More time could have been dedicated in the previous semester to the mechanical drawings and schematic side of the project so that parts could have been ordered more promptly rather than in batches. In review, lessons learned are as follows:

- Reduce the number of design iterations and assure at least one is complete and make changes accordingly.
- Don't worry about ordering lab testing materials and check lab for assembly parts before ordering
- Complete the CAD design, assembly and schematics before ordering parts
- Provide more slack on cables then one might think is needed
- Allow for more time to integrate software with hardware

- Simplicity is best - trying to integrate too many parts results in things not working last minute

Upon completion, the robot did meet the specifications from the original proposal. The details of how the specifications are met are shown in table 14 below. As mentioned in a previous section of this report, a microphone was not mounted in the final sensor package but the design GUI has the ability to display auto data once a microphone is plugged in.

Table 14: Specification and Description

Mechanical Arm Specs	**Description**
Height: Variable height from 1 - 6 feet	Sensor package can be varied from the bottom to the top of the 6ft T-slot arm.
Degrees of Freedom: 2 degrees of movement	360° rotation and 6ft vertical lift
Base Size: Fits in a 760mm Diameter Manhole	The robot base is small enough to the lowered down into a 760mm diameter.
Sensor Package Specs	
Video: Can Stream Video to GUI	Mounted Rpi infrared camera
Gas Sensor: Check Air Quality	MQ2 sensor able to detect H2, LPG, CH4, CO, Alcohol, Smoke and Propane
IR Thermal Camera: Record Thermal Images	Grove infrared sensor gives an object temperature reading and room temperature reading
Microphone: Record Sounds	----
GUI Specs	
Mobile: Usable on Mobile Platforms	Can be used on computer, laptop, tablet, or phone
Control Capabilities: Remotely Controllable Robot	Can be controlled by Xbox controller input and keyboard inputs
Logging: Log Information	Logs real time data from the sensor package with the ability to freeze the data and take screenshots
Data Streaming: Real-Time Data Streaming	Camera streaming from Rpi camera

In conclusion, while parts of this project were challenging, the team learned a lot and accomplished what we set to do. The various parts push us out of our comfort zone but opened many opportunities to learn different software and tools. As a whole, the project was enjoyable to work on and taught us skills that will be used going forward in future works.

9. Leadership Roles

The leadership roles for the team members of Team P.V.I.R is shown in Table 13 below. The tasks of the team leader include Expo Coordinator and Documentation Coordinator. The role of the team leader is to be the point of contact between the team and the advising professor and submitting necessary forms to order or checkout materials from the senior design lab. The team came together at the end of the project for integration testing but primary worked in sub teams of hardware assembly and software design.

Table 13: Leadership Roles

Team Member	Leadership Role
Stephanie Chan	Team Leader, Hardware Assembly Team
Elizabeth Fuller	Head of Sensor Package and Power Design, Hardware Assembly Team
Adrian Muñoz	Head of Mechanical Arm Design, Hardware Assembly Team
Nelson Raphael	Webmaster, Software Design Team
Lemek Robinson	GUI Design and Programming Specialist, Software Design Team

10. References

[1] "UNITED STATES DEPARTMENT OF LABOR," *Occupational Safety and Health Administration*. [Online]. Available: https://www.osha.gov/oshstats/commonstats.html. [Accessed: 15-Apr-2019].

[2] "Chart Book (6th edition): Fatal and Nonfatal Injuries - Fatalities from Contact with Electricity in Construction," *CPWR*. [Online]. Available: https://www.cpwr.com/chart-book-6th-edition-fatal-and-nonfatal-injuries-fatalities-contact-electricity-construction-0. [Accessed: 15-Apr-2019].

[3] T.Zhao, A.J Phillips, and S. W. Eckroad, "Underground Utility Vault Inspection System and Method," U.S. Patent 8,706,340, 22 Apr., 2014.

[4] "802.11-2016 - IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE-SA - The IEEE Standards Association - Home*. [Online]. Available: https://standards.ieee.org/standard/802_11-2016.html. [Accessed: 29-Apr-2019].

[5] electronics radio, "Serial data transmission standards," *Electronics Notes*. [Online]. Available: https://www.electronics-notes.com/articles/connectivity/serial-data-communications/transmission-standards.php. [Accessed: 29-Apr-2019].

[6] electronics radio, "USB Universal Serial Bus Standards," *Electronics Notes*. [Online]. Available: https://www.electronics-notes.com/articles/connectivity/usb-universal-serial-bus/standards.php. [Accessed: 29-Apr-2019].

[7] 20922:2016, I. (2019). *ISO/IEC 20922:2016*. [online] ISO. Available at: https://www.iso.org/standard/69466.html [Accessed 1 May 2019].

[8] "Guide to Industrial Inspection Robots," *Global Electronic Services*, 11-Jun-2018. [Online]. Available: https://blog.gesrepair.com/guide-inspection-robots-used-industrial -sectors/. [Accessed: 29-Apr-2019].

[9] Morton, "Global Inspection Robots Market Analysis 2018 – AETOS, GE Inspection Robotics, Honeybee Robotics, Inuktun Services," *Technology 24*. [Online]. Available: https://ittechnology24.com/global-inspection-robots-market-analysis-2018-aetos-ge-inspection-robotics-honeybee-robotics-inuktun-services/. [Accessed: 29-Apr-2019].

[10] AETOS, "Robotic Inspection Solutions," *Mistrasgroup.* [Online]. Available: https://www.mistrasgroup.com/services/company/publications/AETOS_Robotic_ Inspection_Solutions.pdf. [Accessed: 25-Apr-2019].

[11] "U.S.: largest utility companies 2018," *Statista*. [Online]. Available: https://www.statista.com/statistics/237773/the-largest-electric-utilities-in-the-us-based-on-market-value/. [Accessed: 25-APR-2019].

[12] "Wireless Tracked Inspection Robot with GoPro camera," *Oz Robotics*. [Online]. Available: https://ozrobotics.com/shop/sgt-32p-wireless-tracked-inspection-robot-with-gopro/?gclid=Cj0KCQiA8_PfBRC3ARIsAOzJ2urFLzgB4Zd8cdeFt695BLK9tSg-YjAIKI7-CBIOE6x9qK2-eQMhg2AaAr33EALw_wcB. [Accessed: 25-Apr-2019].

Appendices

Appendix A - Project Gantt Chart

See next page for project Gantt Chart

Underground Power Vault Robot Project Schedule

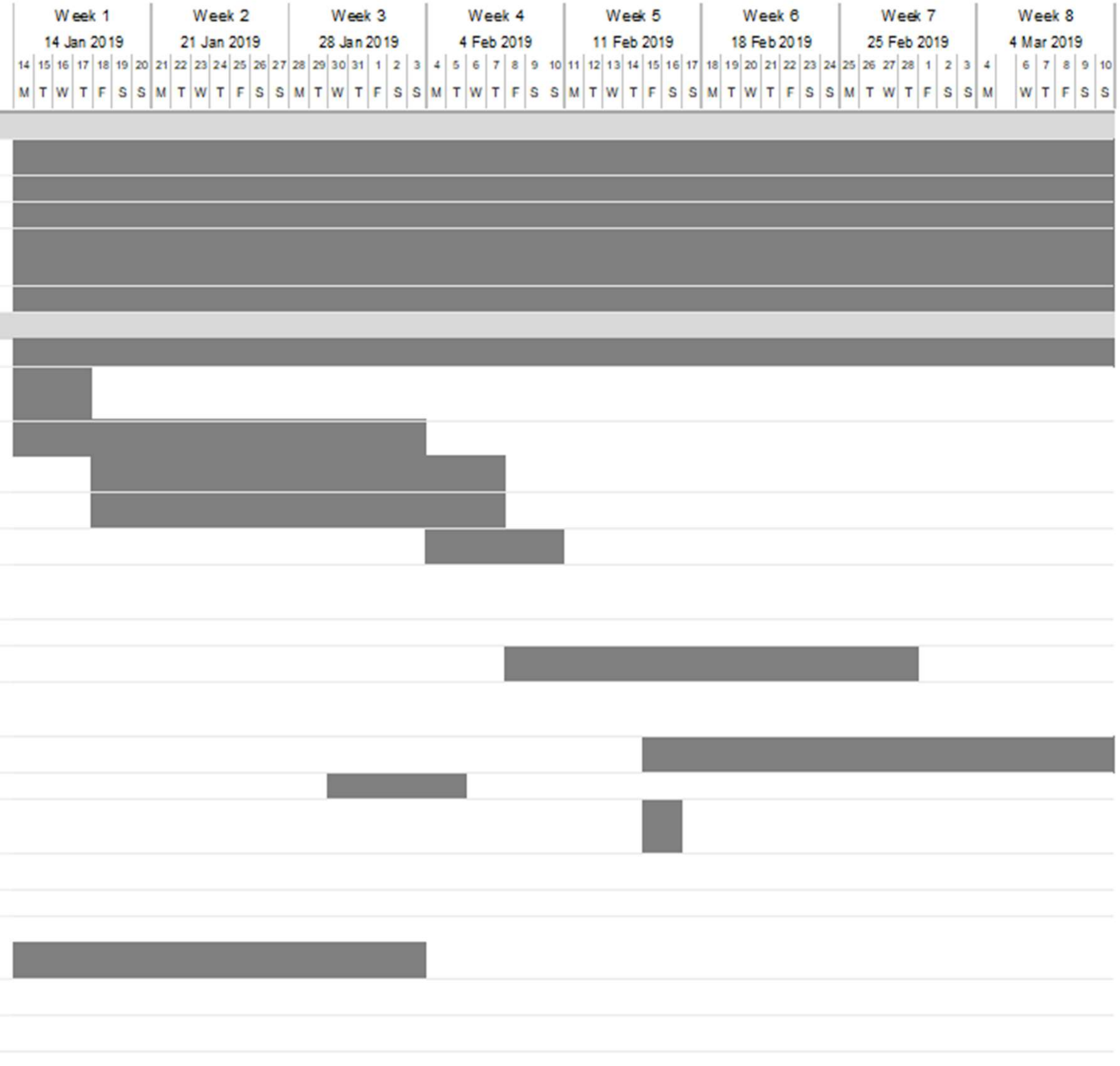
Georgia Tech ECE 4011

Gantt Chart Template © 2006-2018 by Vertex42.com.

Project Start Date: 1/14/2019 (Monday)   Display Week: 1

Advisor: Lukas Graber

WBS		TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	D	**Delivarables**			Thu 5/02/19			-
1.1		Final Oral Presentation	All	Mon 1/14/19	Wed 4/24/19	101	100%	73
1.2		Final Project Summary	All	Mon 1/14/19	Wed 5/01/19	108	100%	78
1.3		Final Project Report	All	Mon 1/14/19	Wed 5/01/19	108	100%	78
1.4		Final Project Demo	All	Mon 1/14/19	Tue 4/23/19	100	100%	72
1.5		Capstone Design Expo	All	Mon 1/14/19	Tue 4/23/19	100	100%	72
1.6		Website	Nelson	Mon 1/14/19	Wed 5/01/19	108	100%	78
2		**Design and Build Robot**			Sun 4/14/19			-
2.1	A	GUI Design	Lemek	Mon 1/14/19	Sun 3/10/19	56	100%	40
2.2	B	Scrapping robot/testing parts for usability	All	Mon 1/14/19	Thu 1/17/19	4	100%	4
2.3	C	Mechanical Arm CAD Design	Adrian	Mon 1/14/19	Sun 2/03/19	21	100%	15
2.4	E	Sensor Package Schematic	Elizabeth	Fri 1/18/19	Thu 2/07/19	21	100%	15
2.5	F	Control System Design	Elizabeth	Fri 1/18/19	Thu 2/07/19	21	100%	15
2.6	G	3D Prototype of Mechanical Arm	Adrian	Mon 2/04/19	Sun 2/10/19	7	100%	5
2.7	H	Testing Communications with GUI	Lemek and Nelson	Mon 3/11/19	Sun 3/31/19	21	100%	15
2.8	I	Testing Sensor Parts	Elizabeth	Mon 3/11/19	Tue 3/12/19	2	100%	2
2.9	J	Board Design for Sensor Package	Stephanie	Fri 2/08/19	Thu 2/28/19	21	100%	15
2.10	K	Order Motor/controllers for control system	Stephanie	Tue 3/12/19	Mon 3/18/19	7	100%	5
2.11	L	Order materials for mechanical arm	Stephanie	Fri 2/15/19	Tue 3/12/19	26	100%	18
2.12	M	Order Sensors	Stephanie	Wed 1/30/19	Tue 2/05/19	7	100%	5
2.13	N	Integrating Control System into Robot Shell	Nelson	Fri 2/15/19	Sat 2/16/19	2	100%	1
2.14	O	Assemble Mechanical Arm onto Robot	All	Tue 4/09/19	Thu 4/18/19	10	100%	8
2.15	P	Soldering Board	All	Wed 3/20/19	Thu 3/21/19	2	100%	2
2.16	Q	Mounting Circuit	All	Mon 4/15/19	Tue 4/16/19	2	100%	2
2.17	R	Test Enviroment CAD Design	Nelson	Mon 1/14/19	Sun 2/03/19	21	100%	15
2.18	S	Order Test Enviroment Materials	Stephanie	Tue 3/12/19	Mon 3/18/19	7	100%	5
2.19	T	Assemble Test Envirment	All	Fri 4/19/19	Mon 4/22/19	4	100%	2
2.20	U	Final Testing	All	Tue 4/16/19	Mon 4/22/19	7	100%	5



Underground Power Vault Robot Project Schedule

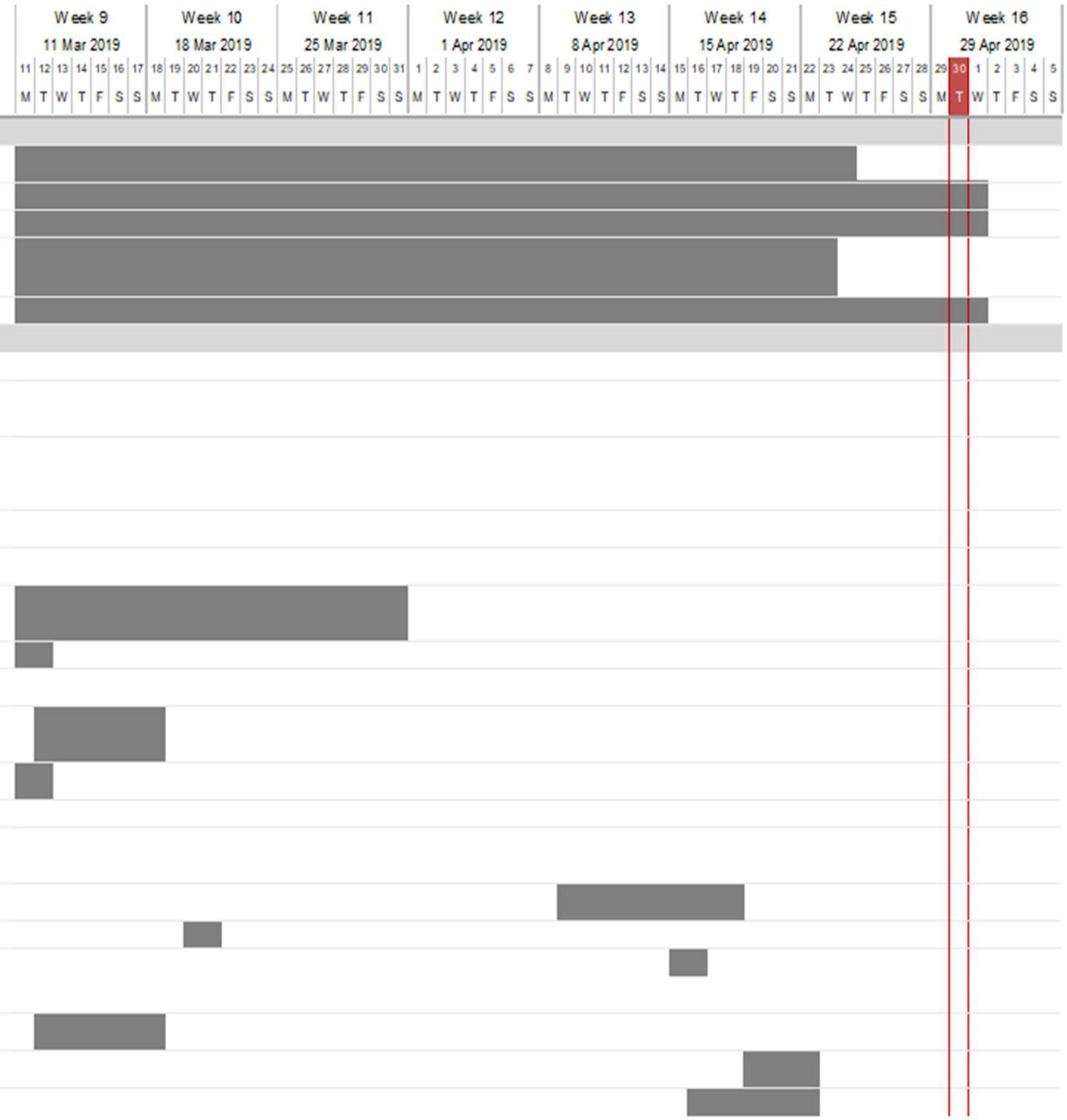
Georgia Tech ECE 4011

Gantt Chart Template © 2006-2018 by Vertex42.com.

Project Start Date: 1/14/2019 (Monday)   Display Week: 9

Advisor: Lukas Graber

WBS		TASK	LEAD	START	END	DAYS	% DONE	WORK DAYS
1	D	**Delivarables**			Thu 5/02/19			-
1.1		Final Oral Presentation	All	Mon 1/14/19	Wed 4/24/19	101	100%	73
1.2		Final Project Summary	All	Mon 1/14/19	Wed 5/01/19	108	100%	78
1.3		Final Project Report	All	Mon 1/14/19	Wed 5/01/19	108	100%	78
1.4		Final Project Demo	All	Mon 1/14/19	Tue 4/23/19	100	100%	72
1.5		Capstone Design Expo	All	Mon 1/14/19	Tue 4/23/19	100	100%	72
1.6		Website	Nelson	Mon 1/14/19	Wed 5/01/19	108	100%	78
2		**Design and Build Robot**			Sun 4/14/19			-
2.1	A	GUI Design	Lemek	Mon 1/14/19	Sun 3/10/19	56	100%	40
2.2	B	Scrapping robot/testing parts for usability	All	Mon 1/14/19	Thu 1/17/19	4	100%	4
2.3	C	Mechanical Arm CAD Design	Adrian	Mon 1/14/19	Sun 2/03/19	21	100%	15
2.4	E	Sensor Package Schematic	Elizabeth	Fri 1/18/19	Thu 2/07/19	21	100%	15
2.5	F	Control System Design	Elizabeth	Fri 1/18/19	Thu 2/07/19	21	100%	15
2.6	G	3D Prototype of Mechanical Arm	Adrian	Mon 2/04/19	Sun 2/10/19	7	100%	5
2.7	H	Testing Communications with GUI	Lemek and Nelson	Mon 3/11/19	Sun 3/31/19	21	100%	15
2.8	I	Testing Sensor Parts	Elizabeth	Mon 3/11/19	Tue 3/12/19	2	100%	2
2.9	J	Board Design for Sensor Package	Stephanie	Fri 2/08/19	Thu 2/28/19	21	100%	15
2.10	K	Order Motor/controllers for control system	Stephanie	Tue 3/12/19	Mon 3/18/19	7	100%	5
2.11	L	Order materials for mechanical arm	Stephanie	Fri 2/15/19	Tue 3/12/19	26	100%	18
2.12	M	Order Sensors	Stephanie	Wed 1/30/19	Tue 2/05/19	7	100%	5
2.13	N	Integrating Control System into Robot Shell	Nelson	Fri 2/15/19	Sat 2/16/19	2	100%	1
2.14	O	Assemble Mechanical Arm onto Robot	All	Tue 4/09/19	Thu 4/18/19	10	100%	8
2.15	P	Soldering Board	All	Wed 3/20/19	Thu 3/21/19	2	100%	2
2.16	Q	Mounting Circuit	All	Mon 4/15/19	Tue 4/16/19	2	100%	2
2.17	R	Test Enviroment CAD Design	Nelson	Mon 1/14/19	Sun 2/03/19	21	100%	15
2.18	S	Order Test Enviroment Materials	Stephanie	Tue 3/12/19	Mon 3/18/19	7	100%	5
2.19	T	Assemble Test Envirment	All	Fri 4/19/19	Mon 4/22/19	4	100%	2
2.20	U	Final Testing	All	Tue 4/16/19	Mon 4/22/19	7	100%	5

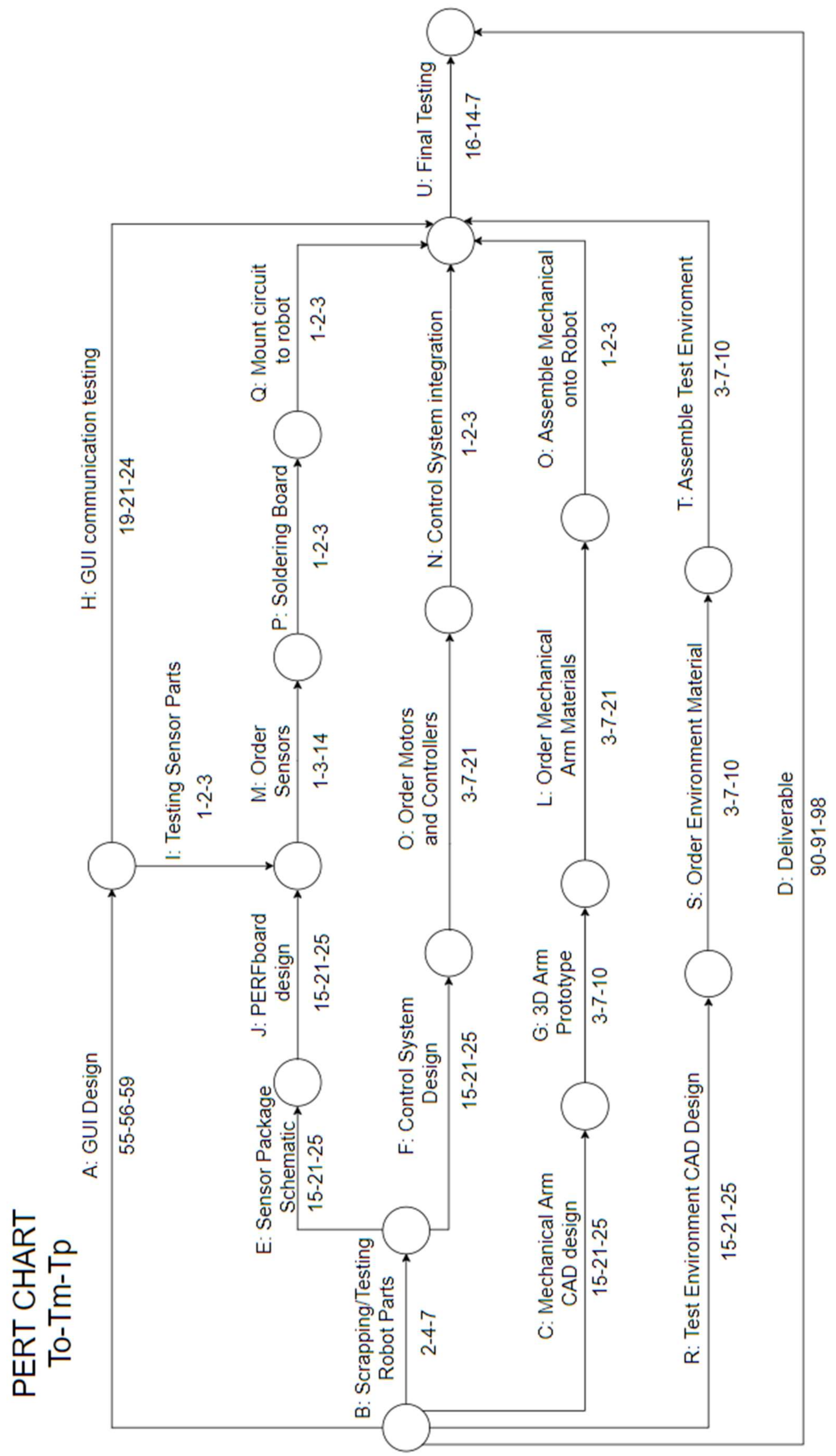


Appendix B - Project PERT Chart

See next page for project PERT Chart

Note:

Deliverables include Oral Presentation, Final Design Proposal, Final Project Demo, and Design Expo


PERT CHART
To-Tm-Tp
A: GUI Design
55-56-59
B: Scrapping/Testing Robot Parts
2-4-7
C: Mechanical Arm CAD design
15-21-25
D: Deliverable
90-91-98
E: Sensor Package Schematic
15-21-25
F: Control System Design
15-21-25
G: 3D Arm Prototype
3-7-10
H: GUI communication testing
19-21-24
I: Testing Sensor Parts
1-2-3
J: PERFboard design
15-21-25
L: Order Mechanical Arm Materials
3-7-21
M: Order Sensors
1-3-14
N: Control System integration
1-2-3
O: Order Motors and Controllers
3-7-21
O: Assemble Mechanical onto Robot
1-2-3
P: Soldering Board
1-2-3
Q: Mount circuit to robot
1-2-3
R: Test Environment CAD Design
15-21-25
S: Order Environment Material
3-7-10
T: Assemble Test Enviroment
3-7-10
U: Final Testing
16-14-7

Appendix C - CAD Drawings for Mechanical Arm and Sensor Package

Images Start on next page

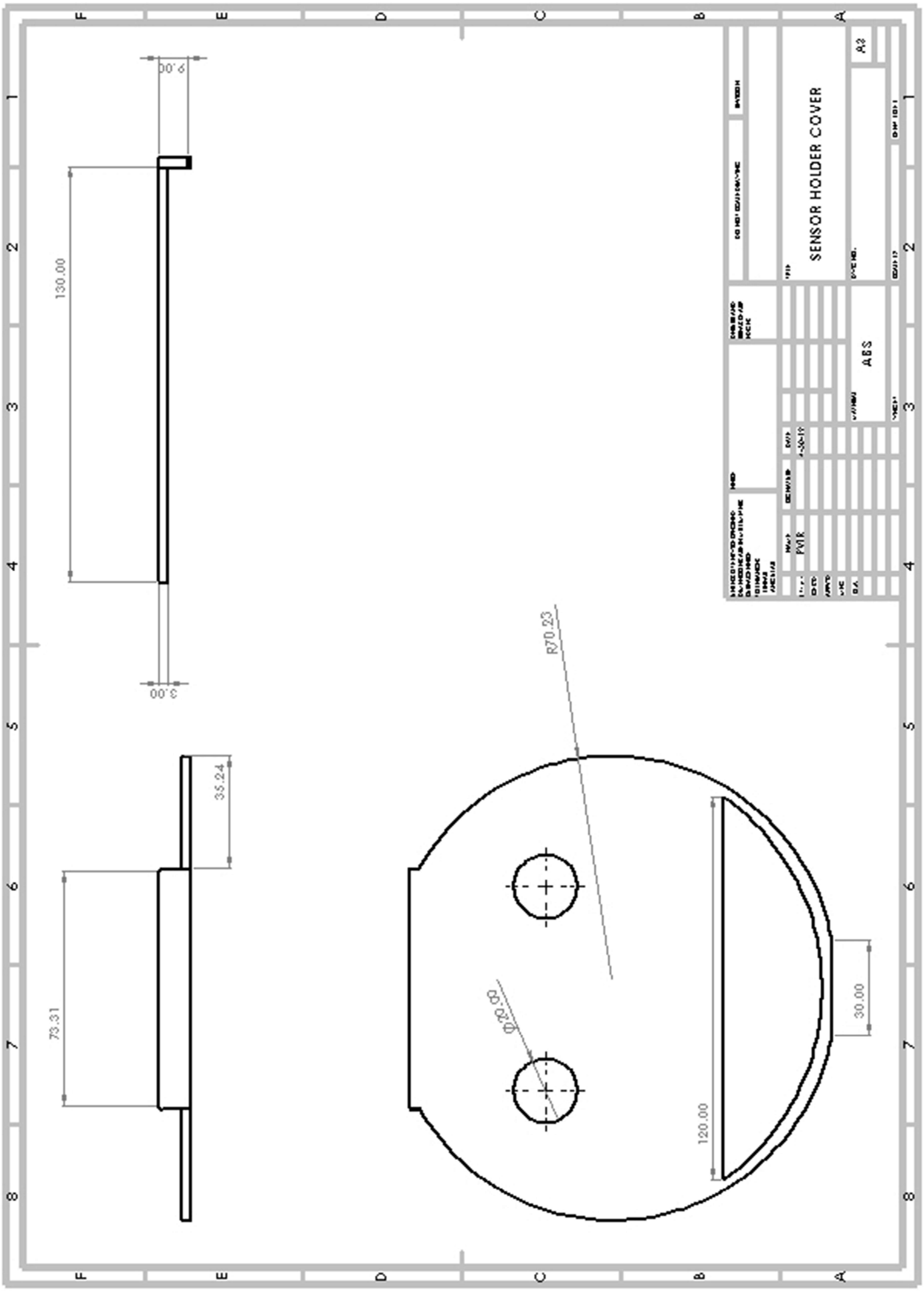
Figure 1: Sensor Holder Cover

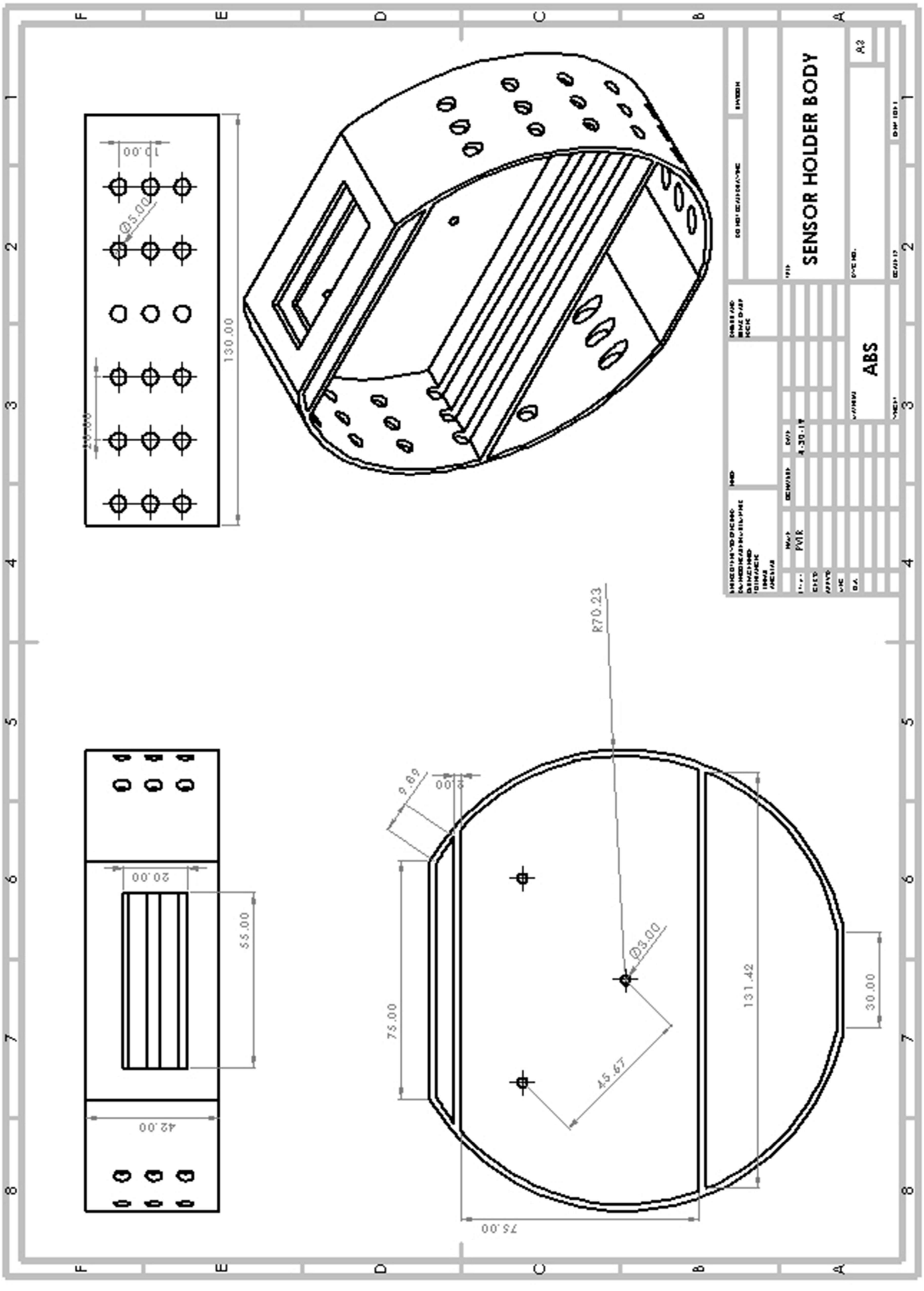
Figure 2: Sensor Holder Body

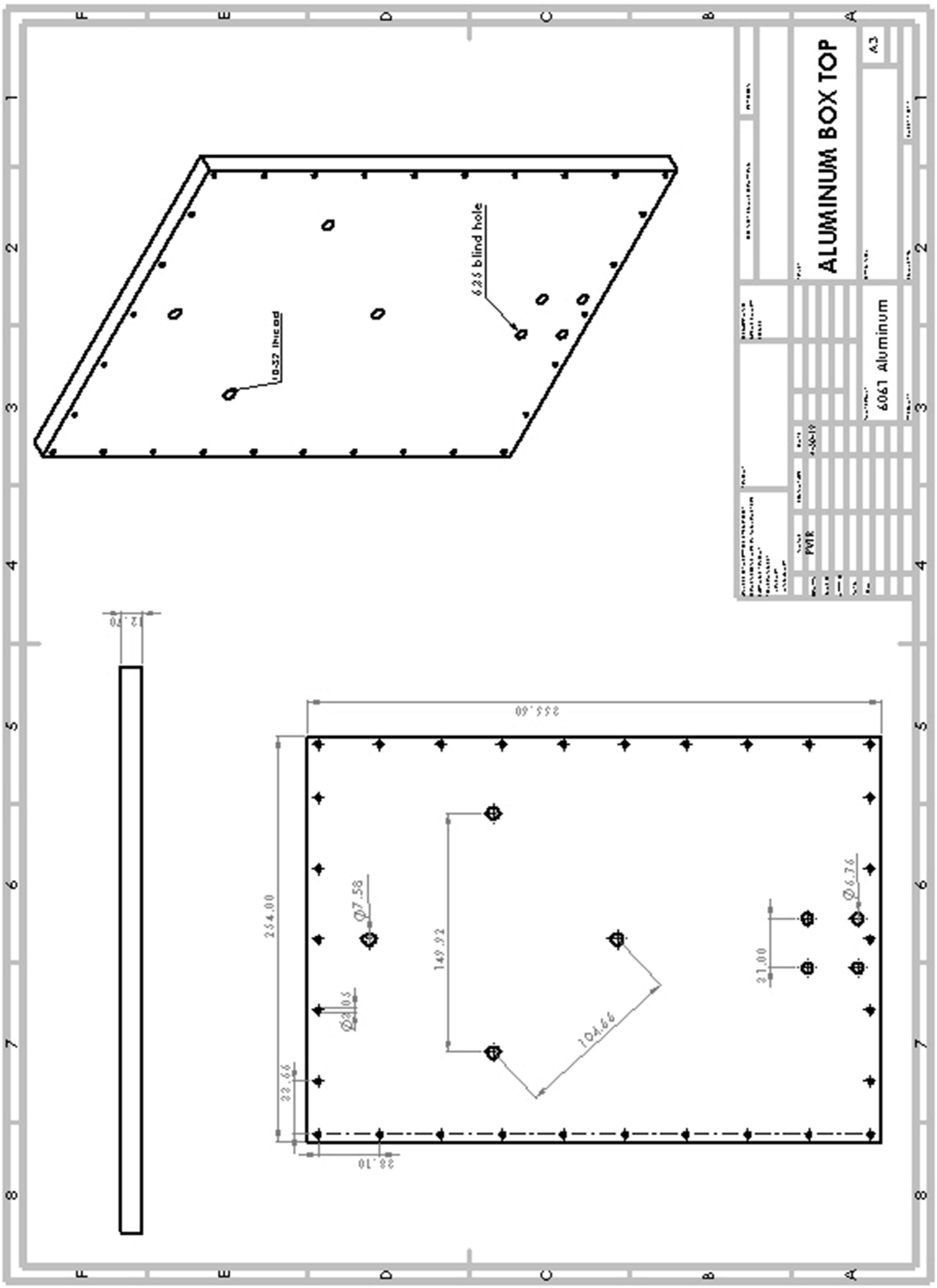
Figure 3: Aluminum Box Top (Robot Base Lid)

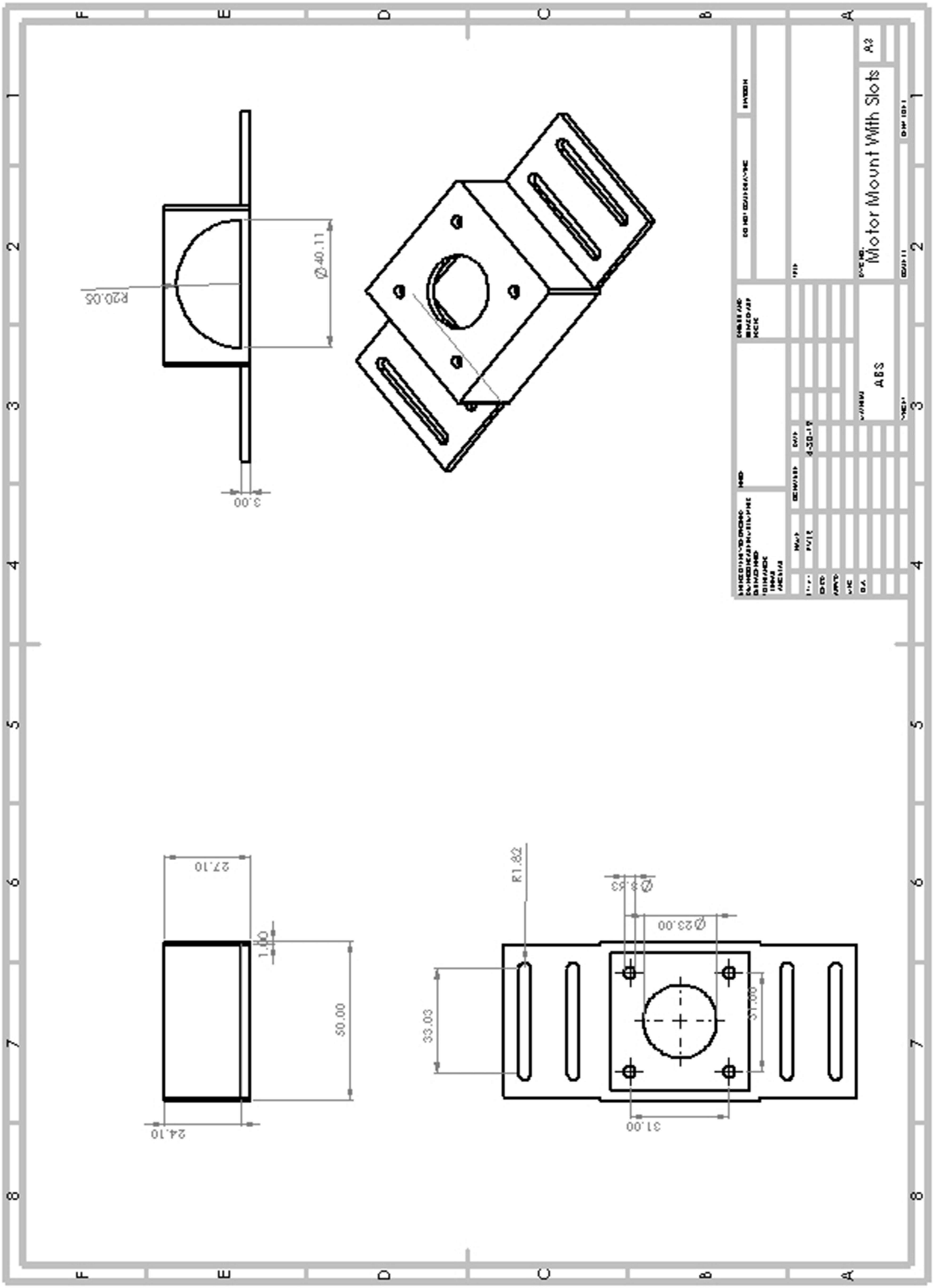
Figure 4: Motor Mount with Slots (Turntable Movement)

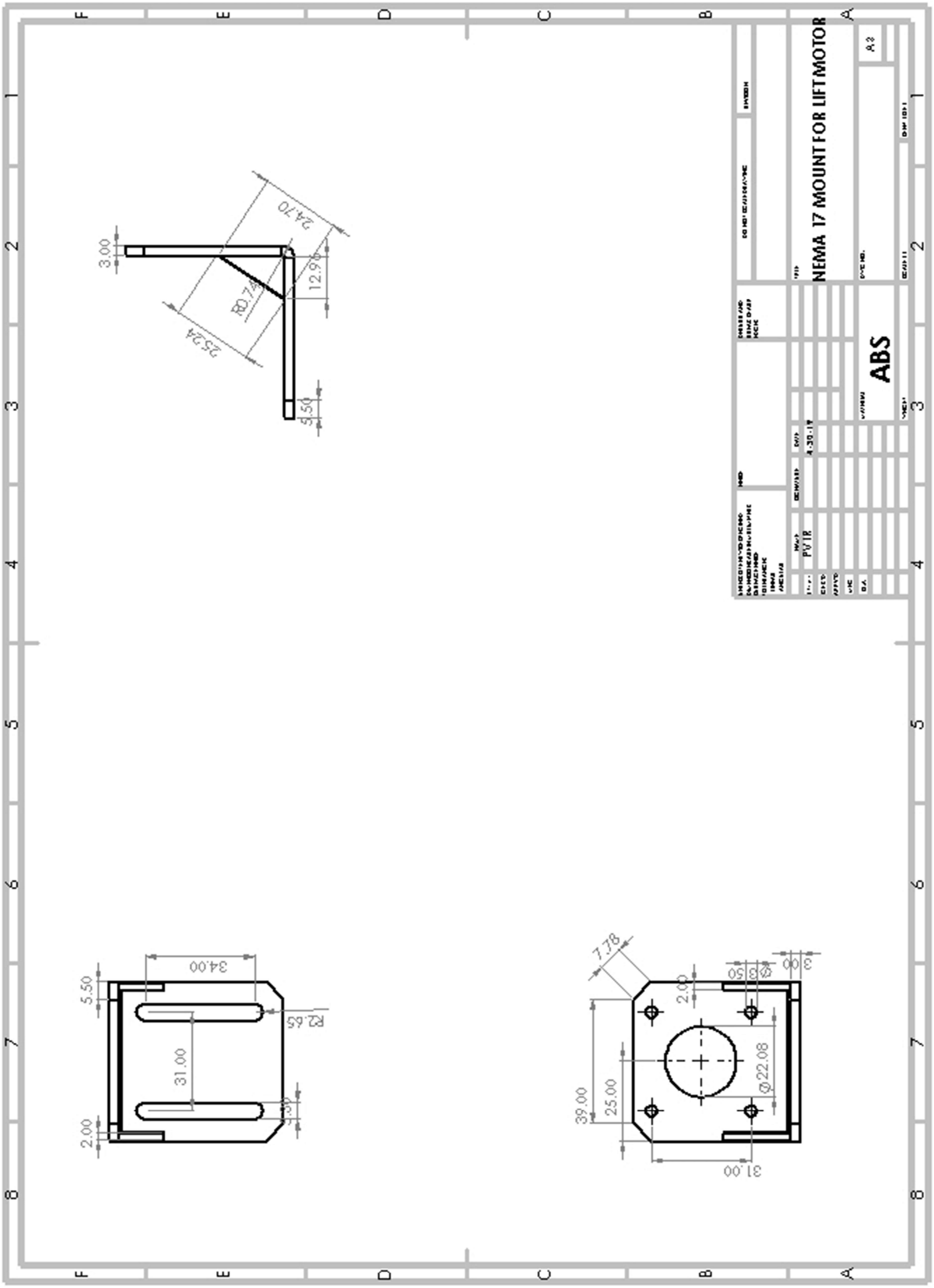
Figure 5: Motor Mount with Slots (Sensor Package Vertical Movement)


SENSOR HOLDER COVER
ABS
PVIR
A3
9.00
130.00
3.00
35.24
73.31
R70.23
Φ20.00
30.00
120.00


SENSOR HOLDER BODY
ABS
A3
PVIR
130.00
10.00
Ø5.00
20.00
55.00
42.00
R70.23
Ø3.00
131.42
30.00
75.00
45.67
9.89
75.00


ALUMINUM BOX TOP
A3
6061 Aluminum
PVIR
10-32 thread
.625 blind hole
355.60
254.00
149.92
104.66
31.00
12.70
Ø7.58
Ø6.36


Motor Mount With Slots
ABS
4-30-19
A3
R20.05
Ø40.11
3.00
27.10
1.00
50.00
24.10
R1.82
Ø3.63
Ø23.00
33.03
31.00
31.00


NEMA 17 MOUNT FOR LIFT MOTOR
ABS
A3
PVIR
4-30-17
24.70
25.24
10.74
12.95
3.00
5.50
34.00
R2.65
31.00
5.50
2.00
7.78
39.00
25.00
Ø22.08
31.00
2.00

Appendix D - Schematic