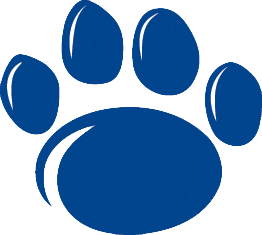


Capstone Final Report



Last Updated: May 2nd 2022

Electrical and Computer Engineering Department

Penn State Erie – The Behrend College

Super Mileage Vehicle Camera System

Jill Johnson

Dr. Myung Cho

Team Project Number: 561

Submitted by:

Andrew Breier

Blake Pahler

Anthony Mazzocco

Submission date

(Format, May 1, 2016)

Submitted in partial fulfillment of the requirements for senior design project

Electrical and Computer Engineering Department

Penn State Erie – The Behrend College

[Table 1: List of Requirements 4](#_Toc102407858)

[Table 2: List of Specifications 4](#_Toc102407859)

[Table 3: List of implementation materials 5](#_Toc102407860)

[Table 4: List of hardware components 6](#_Toc102407861)

[Table 5: List of software components 6](#_Toc102407862)

[Table 6: List of acquired technical skills 8](#_Toc102407863)

[Table 7: List of learning outcomes 9](#_Toc102407864)

[Table 8: List of generated media 10](#_Toc102407865)

1. Problem Statement

The problem presented to the team was that the driver of the Super Mileage Vehicle (SMV) had extremely poor visibility out of the windshields. The current SMV has small windows on either side of the vehicle and curved front windshield. The curved front windshield creates a fisheye lens effect, which distorts the driver’s visibility. The side windows are approximately three inches and are partially covered by the wheels of the vehicle. In order to increase the driver’s safety and visibility, we must implement a safer system through the use of cameras and digital technologies.

Table 1: List of Requirements

|  |  |  |
| --- | --- | --- |
| ***Serial*** | ***Engineering Requirement*** | ***Explanation /details*** |
| 001 | Camera’s View | The cameras must be able to see objects at least 5 feet from the ground and at least 20 feet from a distance and must have a field of view of at least 80 degrees. |
| 002 | Structure | The system must not affect the aerodynamics/performance of the car and must be low profile. |
| 003 | Power | The system must not be powered by any other source except its own given power supply. |
| 004 | Upgrades | Any additions or changes to the camera system must be applicable if needed in the future. |

Table 2: List of Specifications

|  |  |  |
| --- | --- | --- |
| ***Serial*** | ***Engineering Specifications*** | ***Explanation /details*** |
| 001 | Resolution | The resolution of the cameras must be above 720p in order to provide a clear image, although higher resolution (1080p) is desired. |
| 002 | Latency | The latency of the system must be low enough (0.3 seconds) for the driver to clearly see oncoming objects. |

Table 3: Project tasks breakdown and contributions

|  |  |  |
| --- | --- | --- |
| ***Team***  ***Member’s Name*** | **Title of the System**  **Subcomponent/Subtasks** | **Weight of contribution**  **(%)** |
| Andy Breier | Script | 100 |
| Andy Breier | Python Code | 40 |
| Andy Breier | Mock Mount | 10 |
| Blake Pahler | Python Code | 40 |
| Anthony Mazzocco | Python Code | 20 |
| Anthony Mazzocco | Mock Mount | 90 |

Table 4: Breakdown of the course deliverables and corresponding contributions

|  |  |  |  |
| --- | --- | --- | --- |
| **Title of the System**  **Subcomponent/Subtasks** | ***Andrew Breier***  **contribution**  **(%)** | ***Blake Pahler***  **contribution**  **(%)** | ***Anthony Mazzocco***  **contribution**  **(%)** |
| Gantt chart and the progress reports updates | 40 | 30 | 30 |
| Midterm presentation | 36 | 32 | 32 |
| Midterm Video  Submission | 10 | 80 | 10 |
| Initial poster version for poster competition | 33.3 | 33.3 | 33.3 |
| Final Poster Version | 33.3 | 33.3 | 33.3 |
| Final Video Submission | 45 | 10 | 45 |
| Final Presentation | 36 | 32 | 32 |
| Final Report | 36 | 32 | 32 |

1. Proposed Solution
   1. Background Research

To start this project, we first had to research the rules of the competition. The competition does not say that having windows is a requirement, so this project is able to be implemented into the vehicle at some point. We then had to ensure that it was possible to find cameras that have an 85-degree field of view, as this is a rule of the competition. It is also stated that this must be a separate system from the vehicle’s system. So, this was why it was required to obtain a portable battery for the entire system.

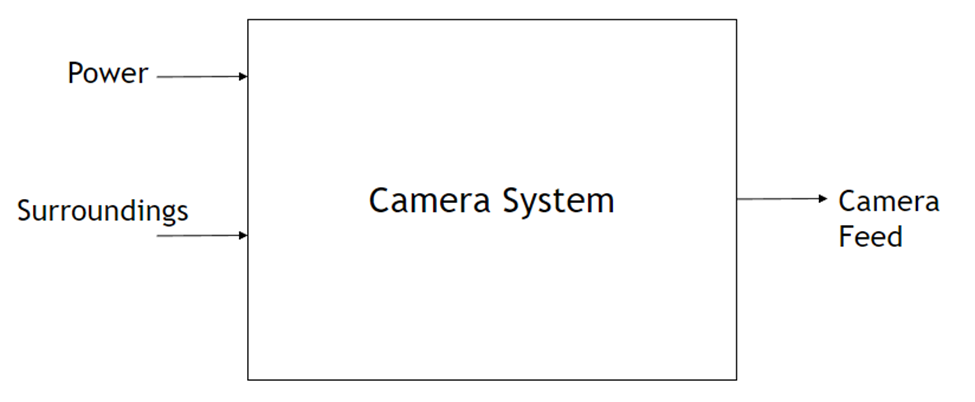
Once the research for the rules was completed, we started looking into ways to show three different images onto a monitor. Hence, we started researching different minicomputers that would have the ability to complete this task. We then researched if it was possible to use python and the OpenCV library on the Raspberry Pi 4 to implement the system. Finally, we had to research if it was possible to have three cameras plugged into the Raspberry Pi 4 and show the different images. We found that the cameras required too much power and then had an externally powered USB hub plug into the Pi to power and pass data from the cameras. Along with this, we also had to research the best camera for this task. Once our team had completed the research, we decided this solution was the best way to solve the problem.

We chose to go with this solution with the Raspberry Pi 4 because it has a simple OS and can be powered by a portable battery. Also, python is already installed on this system and only needed the OpenCV library to make the system work properly. And, the Raspberry Pi 4 has enough USB ports to connect all the cameras to. Also, this was chosen because it would be able to help our team improve upon skills that we have learned throughout the years.

This is a good fit for the problem, because it covers all the requirements and follows all the rules of the competition. Due to the ability of the Raspberry Pi 4 to be able to show all images from the camera, and for the USB hub to power all the cameras and that the Pi can connect to a USB monitor this seemed like the best solution. Also, all these subsystems can be powered by a portable battery, making this the best solution.

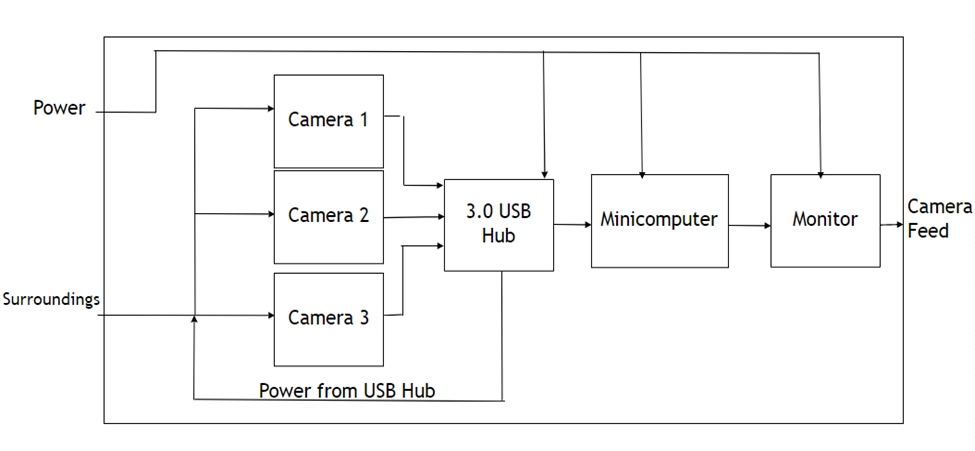
* 1. **Proposed System Design**

Figure 1 below shows the Level 0 design of the entire system. Essentially there is power and surroundings going into the Camera System, and the output of the system is the camera feed to the user



**Figure 1**

Figure 2 shows the Level 1 diagram of the system, showing all the connections between systems. As shown, power is supplied to the 3.0 USB Hub, the Raspberry Pi 4 and the monitor. The power from the USB Hub is used to power the cameras and then feed data back to the USB Hub. Also, the cameras take in each of the surroundings from the three different views. The USB Hub then feeds this data to the RPi 4, Which is then passed along to the monitor and is then shown to the user as camera feed.

**Figure 2**

* 1. **System Implementation**

The technology we used to implement this system consisted of a few different mediums. To start, we decided to use the python version of OpenCV. To use python on the Raspberry Pi, we also needed to an IDE, so we decided to use the Thonny interface that is already included in the Pi OS. We also decided that the best way to connect all the components to one another was to use USB cables. Whether this be USB A to USB C, USB A to DC 5v. We also used a USB 3.0 Hub for each of the cameras. Finally, we used a micro-HDMI to HDMI cord for the display. This is how our team decided to implement the technology. To power this system, we used a 26,800 mAh, 5.4A power bank with three USB ports.

We decided to use the python version of the OpenCV library because it is much simpler to work with and user friendly. This also ties into why we used Thonny as well. It is a lightweight IDE and is simple to understand. Also, it comes included with the Pi OS so there is no need to download extra software. The cords are used so that once implemented into the actual vehicle the team can plug in the cords how they like them and put them in places that would not be uncomfortable to the driver. The USB Hub had to be used, there was no other solution for the power consumption issue. This also holds true for the HDMI cable as well, there is no other technology for this. The battery bank was used because it can power the entire system for up to 5 hours, a bigger bank can be used in the future if needed by the team.

Table 5: List of implementation materials

|  |  |  |
| --- | --- | --- |
| ***Item***  ***Serial*** | ***Item***  ***Name*** | ***Item URL (if the item is delivered as a copy on flash drive, please indicate so)***  ***(Github, Google Drive URL, etc)*** |
| 001 | Github | <https://github.com/BlakePahler/Senior-Design> |
| 002 | Pi SD card | N/A |

Table 6: List of hardware components

|  |  |  |  |
| --- | --- | --- | --- |
| ***Item***  ***Serial*** | ***Item***  ***Name*** | ***Item***  ***Purchase Link*** | ***Price***  ***($)*** |
| 8541535469 | DC 5v to USB A Cord | [USB A to DC 5 volt](https://www.amazon.com/Charging-5-5x2-1mm-Connector-4-0x1-35-3-5x1-35/dp/B07J6NQ1KN/ref=sr_1_1?crid=3FQ6VONEVRMN8&keywords=8541535469&qid=1649878609&sprefix=8541535469%2Caps%2C79&sr=8-1) | $7.99 |
| B07W3ZMVP1 | Raspberry Pi 4 case | [Pi4 Case](https://www.amazon.com/MazerPi-Raspberry-Cooling-HeatsinkModel/dp/B07W3ZMVP1/ref=sr_1_2?crid=16J7AAC2SDA92&keywords=raspberry+pi+4+case+mazerpi&%20qid=1649878746&sprefix=raspberry+pi+4+case+mazerpi%2Caps%2C67&sr=8-2) | $9.99 |
| 0791313705156 | On/Off Switch for Raspberry Pi4 | [On/Off Switch for Pi](https://www.amazon.com/iUniker-Raspberry-Switch-Supply-TypeC/dp/B07V8G2SYZ/ref=sr_1_2?crid=10B44FH893PGR&keywords=iuniker+pi+switch&qid=1649878844&%20sprefix=iuniker+pi+switch%2Caps%2C64&sr=8-2) | $7.99 |
| B07P5ZP943 | Portable Battery for Rpi4 | [Portable Battery](https://www.amazon.com/Charmast-26800mAh-Portable-Li-PolymerCompatible/dp/B07P5ZP943/ref=sr_1_9?crid=336BHQLCPPO2X&keywords=raspberry%2Bpi%2B4%2Bba%20ttery&qid=1649882613&sprefix=raspberry%2Bpi%2B4%2Bbattery%2Caps%2C78&sr=8-9&th=1) | $39.99 |
| B06WWQ7KLV | Micro HDMI to HDMI cord (3ft) | [Micro to full HDMI](https://www.amazon.com/UGREEN-Adapter-Ethernet-CompatibleRaspberry/dp/B06WWQ7KLV/ref=sr_1_3?keywords=micro+hdmi+to+hdmi&qid=1649883931&sprefix=%20micro+hdmi%2Caps%2C94&sr=8-3) | $8.99 |
| 0818707029503 | USB 3.0 Extension Cable with On Off Switch 6 ft, Support Data and Power | [On/Off Switch power + data](https://www.amazon.com/dp/B08M45QM3J?psc=1&ref=ppx_yo2ov_dt_b_product_details) | $11.99 |

Table 7: List of software components

|  |  |  |  |
| --- | --- | --- | --- |
| ***Item***  ***Serial*** | ***Item***  ***Name*** | ***Download***  ***URL*** | ***Price***  ***($)*** |
| 001 | OpenCV library | <https://pypi.org/project/opencv-python/> | N/A |

1. Results

* Successfully implemented features:
  + Simple OS
  + Easily Upgradable
  + All is powered by portable battery
  + Power for each camera via USB Hub
  + Working code for cameras
  + Working script to work on bootup
  + Proof of concept on mock mount

* Features left for future work:
  + Improve code to make better use of CPU
  + Mount into vehicle
  + Obtain USB powered monitor
  + Easier to make sure cameras are on/plugged in
  + Ensure everything continually works on rough terrain
  1. Experiments, Testing, or Analyses

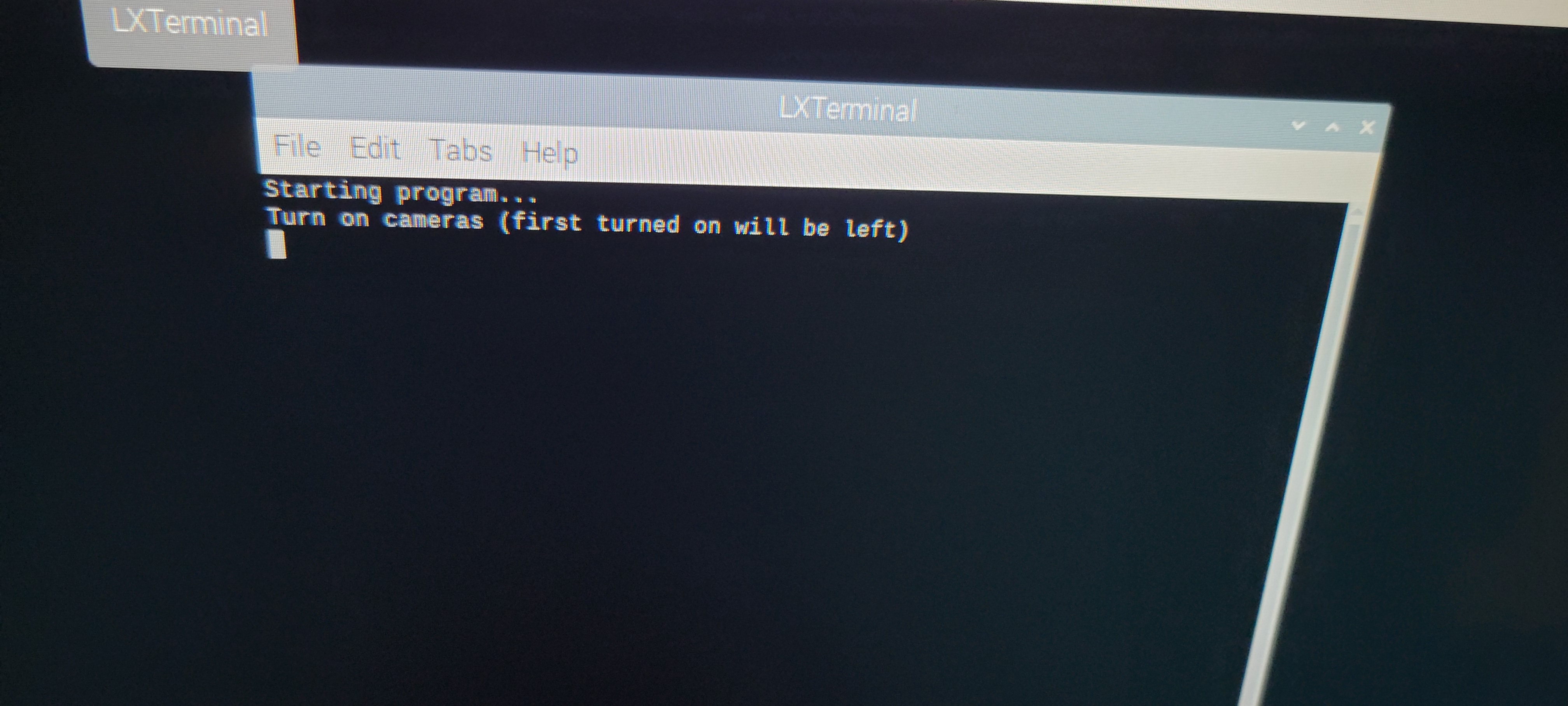
While working on this project, there was a lot of testing that needed to be done to ensure that the cameras would work simultaneously and, on a boot, up of the RPi4. To begin our first test, we started with only using one camera. This first test had one error. This error was caused by the system not knowing what USB file to look at. Once our team found the proper USB file, we were able to get the camera code working, however, with only one camera. When testing with more cameras, we ran into an issue where only one window for one camera would open at a time. So, we tested different codes to get more than one camera working. Eventually we decided to reinstall the OpenCV library, which ended up resolving our issue. Now that we had all the cameras simultaneously working, we were able to start working on the boot up script.

The boot up script gave us many issues when testing. While these are not errors per say, we were just unable to find a way to get the system to run the python code on boot up. After doing extensive research into Linux type systems and how they operate on a reboot, we found that we could write into what is called the .bashrc file. This file is used to run certain code as different programs are opened. In our case, this would run the python code, that we direct to run, when the LXTerminal is opened on the system. After knowing this, we tried to find ways to open the LXTerminal as the system booted. So, we settled on what is called LXSessions. The user can create a file that will start up the system in a certain way, or open certain programs. We used this to open the LXTerminal program which would then in turn run our python code. This was tested and retested many times to ensure that every time the system was booted up, the program would run without fail.

The final test was using the three cameras to open three different windows in three separate places on the system. This was to be done so no windows would overlap with each other, blocking the view of one camera to the user. To use the three different cameras, we had to obtain a USB Hub to power all the cameras, which after testing, was found to work. The last test was placing the cameras on the monitor in a way that would not block the drivers’ view from the cameras. We tested many different areas on the monitor to make sure the cameras were placed and showed each view correctly. The below images show the final output of the system along with LXTerminal opening and running the code.



**Final Output**



**Output from LXTerminal on boot**

1. Technical Skills and Learning Outcomes
   1. Technical Skills

Table 8: List of acquired technical skills

|  |  |  |  |
| --- | --- | --- | --- |
| ***Student***  ***Name*** | ***Skill Title*** | ***Your level before this project***  ***(none, beginner, intermediate, advance)***  ***(0,1,2,3)*** | ***Your level after this project***  ***(none, beginner, intermediate, advance)***  ***(0,1,2,3)*** |
| Andrew Breier | This project gave me more knowledge in the way the Pi OS works along with other types of Linux based systems. I also improved my understanding of python and the OpenCV library. | 1 | 2 |
| Blake Pahler | This project allowed me to gain valuable experience with the Raspberry Pi Minicomputer and coding in Python. I also learned more about techniques such as multithreading and multiprocessing for python functions. | 1 | 2 |
| Anthony Mazzocco | After this project, I was able to get my feet wet a little bit by experiencing one of the ways a Raspberry Pi minicomputer can be utilized and programmed in a design. | 0 | 1 |

* 1. Learning Outcomes

Table 9: List of learning outcomes

|  |  |  |  |
| --- | --- | --- | --- |
| ***Student***  ***Name*** | ***Skill Title*** | ***Your level before this project***  ***(none, beginner, intermediate, advance)***  ***(0,1,2,3)*** | ***Your level after this project***  ***(none, beginner, intermediate, advance)***  ***(0,1,2,3)*** |
| Andrew Breier | From this project I have improved on my teamwork and how to manage my time more wisely. Also, I have improved on implementation of a system and improved my presenting of data skills. | 1 | 2 |
| Blake Pahler | A few skills that I have learned throughout this project are designing and implementing a system to completion, managing time more effectively, and better teamwork. | 1 | 2 |
| Anthony Mazzocco | After finishing this semester, I was able to go through the experience of problem solving several times throughout this project and can say I gained confidence as one who can come up with solutions. | 0 | 2 |

1. Project Media

Table 10: List of generated media

|  |  |  |
| --- | --- | --- |
| ***Item***  ***Serial*** | ***Item***  ***Name*** | ***Item***  ***URL*** |
| 001 | Initial project demo | https://www.youtube.com/watch?v=0EbrMHLVzsA |
| 002 | 2-minute video presentation | https://youtu.be/8X7ZzPwCrNI |
| 003 | Final project demo | https://www.youtube.com/watch?v=IYKmZZ0L8KE |

1. References

[1] Raspberry Pi, “Raspberry pi 4 model B,” Raspberry Pi. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. [Accessed: 20-Dec-20221].

[2] A. Rosebrock, “Multiple cameras with the Raspberry Pi and opencv,” PyImageSearch, 17-Apr-2021. [Online]. Available: https://pyimagesearch.com/2016/01/18/multiple-cameras-with-the-raspberry-pi-and-opencv/. [Accessed: 21-Dec-2021].

[3] “Five ways to run a program on Your Raspberry Pi at startup,” Dexter Industries. [Online]. Available: https://www.dexterindustries.com/howto/run-a-program-on-your-raspberry-pi-at-startup/. [Accessed: 15-Apr-2022].