

|  |  |
| --- | --- |
| Submitted By: | Mason Primus, Cole Licklider, Logan Boehm |
| Date: | 12-16-2024 |

Real-Time Car Projector Final Engineering Project Report

Senior Design ELE 4902

# **Executive Summary**

This document contains the design of the Real-Time Car Projector (RTCP) project. It describes the problem the team set out to solve and the steps taken to design a product that solves the chosen problem over 30 weeks (about 7 and a half months). The problem the team chose is the existence of windows on self-driving vehicles. The team wanted to produce a solution that would allow luxury car companies to remove windows from their self-driving vehicles while avoiding any potential psychological harm to the passengers that would come from riding in a vehicle without a view of the outside world. While the team couldn’t find any proof that this would cause psychological harm, they believe that it’s highly likely, and friends and family have agreed when talking about the project.

The RTCP will allow passengers in a windowless, self-driving car to still be able to view the outside world. It will use cameras on the outside of the vehicle to collect video data of the outside world, then the video will be projected inside of the car. This way, companies will be able to remove windows from self-driving cars without potentially risking the psychological safety of their consumers. The system will also run in real-time so what the passengers see is what’s currently outside of the car. The RTCP project is made up of five subsystems: Input, Image Quality, Processor, Output, and Power Supply. Of the five, the team has decided not to design the Power Supply subsystem. The document contains descriptions of each subsystem and the rationale for not designing the Power Supply subsystem. The designed subsystems would, in theory, be passed off to another team that would design the Power Supply subsystem. The document also describes the system the team designed, developed, and built to meet the specifications.

In the end, the RTCP system was unable to meet the specifications it set out to meet. It was found that the developed Processor subsystem did not have enough processing power to accurately perform the stitching algorithm used by the Image Quality subsystem. The Processor subsystem was able to process the individual camera feeds from the Input subsystem but was unable to accurately create one output video feed in real-time for the Output subsystem to project.

# **Table of Contents**

[**Executive Summary** ii](#_Toc185170375)

[**Table of Contents** iii](#_Toc185170376)

[**Table of Figures** iv](#_Toc185170377)

[**Table of Tables** v](#_Toc185170378)

[**Introduction** 1](#_Toc185170379)

[**Description of the Problem** 1](#_Toc185170380)

[**Problem Statement** 1](#_Toc185170381)

[**Stakeholders** 2](#_Toc185170382)

[**Alternate Solutions** 3](#_Toc185170383)

[**Solution Approach** 4](#_Toc185170384)

[**Real-Time Car Projector** 5](#_Toc185170385)

[**System Diagram** 5](#_Toc185170386)

[**Subsystem Specifications** 6](#_Toc185170387)

[**Input Subsystem** 6](#_Toc185170388)

[**Processor Subsystem** 6](#_Toc185170389)

[**Image Quality Subsystem** 7](#_Toc185170390)

[**Output Subsystem** 9](#_Toc185170391)

[**Subsystem Selection Processes** 10](#_Toc185170392)

[**Input Subsystem** 10](#_Toc185170393)

[**Processor Subsystem** 11](#_Toc185170394)

[**Image Quality Subsystem** 13](#_Toc185170395)

[**Output Subsystem** 14](#_Toc185170396)

[**Bill of Materials** 16](#_Toc185170397)

[**System Verification Results** 17](#_Toc185170398)

[**Summary** 17](#_Toc185170399)

[**Bibliography** 18](#_Toc185170400)

[**Appendix A – Wiring Diagram** 22](#_Toc185170401)

[**Appendix B – Data Sheets** 22](#_Toc185170402)

[**Appendix C – 3D-Printed Files** 22](#_Toc185170403)

[**Appendix D – Input Subsystem Testing Results** 22](#_Toc185170404)

[**Appendix E – Processor Subsystem Testing Results** 26](#_Toc185170405)

[**Appendix F – Output Subsystem Testing Results** 30](#_Toc185170406)

[**Appendix G – Image Quality Subsystem Testing Results** 34](#_Toc185170407)

[**Appendix H – Full System Testing Results** 36](#_Toc185170408)

# **Table of Figures**

Figure 1: System Block Diagram 5

Figure 2: RTCP Wiring Diagram 22

Figure 3: Input Subsystem Camera Operation Testing Setup 23

Figure 4: Input Subsystem Overall FOV Testing Setup 23

Figure 5: Raspberry Pi 5 Power Test Setup 26

Figure 6: Raspberry Pi Zero W Power Test Setup 27

Figure 7: Processor Subsystem Operation Test Setup 29

Figure 8: Output Subsystem AC Adapter 31

Figure 9: Output Subsystem Operation Test Setup 32

Figure 10: Output Subsystem Split Image Test 32

Figure 11: Input Subsystem Position Setup 37

# **Table of Tables**

Table 1: List of acronyms that are used throughout this document vi

Table 2: Stakeholder Requirements 2

Table 3: Requirements and Specifications for the Real-Time Car Projector 4

Table 4: Requirements and Specifications for the Input Subsystem of the RTCP 6

Table 5: Requirements and Specifications for the Processor Subsystem of the RTCP 7

Table 6: Requirements, Specifications, and Transforms for the Image Quality Subsystem of the RTCP 8

Table 7: Requirements, Specifications, and Transforms for the Output Subsystem for the RTCP 9

Table 8: Decision Matrix for Camera Choice for the Input Subsystem 10

Table 9: Decision Matrix for Processor Choice for the Processor Subsystem 11

Table 10: Decision Matrix for the Cooling Solution Choice for the Raspberry Pi 5 12

Table 11: Decision Matrix for the Projector Choice for the Output Subsystem 14

Table 12: BOM for building a projector. 15

Table 13: BOM 16

Table 14: Input Subsystem Camera Operation Testing 24

Table 15: Input Subsystem Overall FOV Testing 25

Table 16: Processor Subsystem Power Verification Plan 28

Table 17: Processor Subsystem Operation Verification Plan 30

Table 18: Output Subsystem Power Verification Plan 31

Table 19: Output Subsystem Operation Verification Plan 33

Table 20: Radial, or fisheye, distortion test. 34

Table 21: Dead Zone Test 35

Table 22: RTCP Setup 37

Table 23: RTCP System Power Verification Plan 38

Table 24: Processor Operation Verification Plan 39

Table 25: Processor and Output Subsystems Compatibility Verification Plan 39

Table 26: RTCP System Operation Verification Plan 40

Table 1: List of acronyms that are used throughout this document

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| API | Application Programming Interface |
| AV | Audio Visual |
| BOM | Bill of Materials |
| CSI | Camera Serial Interface |
| FOV | Field of View |
| FPS | Frames Per Second |
| GPIO | General Purpose Input/Output |
| HDMI | High-Definition Multimedia Interface |
| IDE | Integrated Development Environment |
| LCD | Liquid-Crystal Display |
| LED | Light Emitting Diode |
| MIPI | Mobile Industry Processor Interface |
| MSOE | Milwaukee School of Engineering |
| NVMe | Non-Volatile Memory Express |
| OBS | Open Broadcasting System |
| OpenCV | Open Source Computer Vision Library |
| ORB | Oriented FAST and Rotated Brief |
| OS | Operating System |
| RAM | Random Access Memory |
| RTCP | Real-Time Car Project |
| SAE | Society of Automotive Engineers |
| SATA | Serial Advanced Technology Attachment |
| SD | Secure Digital |
| SSD | Solid State Drive |
| SURF | Speeded Up Robust Features |
| UART | Universal Asynchronous Receiver-Transmitter |
| USB | Universal Serial Bus |
| UVC | USB Video Class |

# **Introduction**

This document details the design process that the team has taken to design the RTCP. The first section describes the problem the team is setting out to solve, and the next section outlines the approach taken. The following section discusses the design process the team has taken to develop the RTCP project. It also goes over the BOM that the team has purchased. The next section analyses the system verification results. The section afterwards gives a summary of the whole report. Following that is the bibliography of references the team has used to develop the RTCP. In the appendices are the schematic diagram of the RTCP, 3D models of parts that have been printed for the project, datasheets of chosen components, the results of testing each subsystem, and the results of testing the full RTCP system.

# **Description of the Problem**

### **Problem Statement**

Safety in the automotive industry has been steadily improving since cars became widespread. We currently sit at the advent of the greatest safety improvement since seat belts: Level 5 autonomous driving. The SAE describes Level Five automation as full driving autonomy without input from the driver [1]. Currently, many companies are designing driverless cars. When Level 5 automotive travel becomes widespread, will cars need windows? With an automated vehicle, windows are no longer required in a physical sense. *Windows increase manufacturing costs, constrain where safety airbags are installed, and provides an access point for vehicles and personal theft.* However, the main issue with removing windows is the psychological vulnerability of the passengers. Riding in an autonomous car with no outside view could potentially damage a passenger’s psychological safety. The team will focus on processing the video signals that will later be shown to the passenger.

### **Stakeholders**

The top four key stakeholders are the luxury car companies that would be implementing the RTCP, the passengers that would be riding in the self-driving cars, the team designing the RTCP, and the team designing the power system of the car. Telsa’s Autopilot was the first mass-produced self-driving accessory and was marketed as a luxury. It is safe to assume that the solution will also start out as a luxury accessory.

Table 2: Stakeholder Requirements

|  |
| --- |
| **Stakeholder 1: Luxury Car Companies** |
| *The solution must:*   * Latency of < 20 ms * Refresh rate of >90Hz * Resolution of 1080p or equivalent quality |
| **Stakeholder 2: Passengers** |
| *The solution must:*   * Latency of < 20 ms * Refresh rate of >30Hz |
| **Stakeholder 3: Real-Time Car Projector Design Team** |
| *The solution must:*   * Be capable meeting automotive environmental specifications: -45 to 85 degrees Celsius * Relative humidity: 0-95% * Refresh rate of >30Hz * Latency of < 20 ms * Resolution of 1080p or equivalent quality |
| **Stakeholder 4: Team Designing the Power System of the Car’s Accessories** |
| *The solution must:*   * Powered from the vehicle’s electrical system * Power efficient |

### **Alternate Solutions**

The team investigated products that solved the chosen problem. While there weren’t any that fully solved the problem as Level 5 autonomous vehicles aren’t widespread, there are products that solve problems like the team’s chosen problem. The team found five alternate solutions that were similar in nature and scope.

Collins Aerospace developed a similar solution for fighter jet pilots. They were tasked with letting the pilot see through the fuselage. It was achieved with the input being an array of sensors on the skin of the aircraft and the output being a part of the helmet that the pilot wears [2].

There have also been discussions about commercial aircraft becoming windowless. In 2019, airline Emirates released a design that would have first-class suites with no windows, instead using real-time fiber-optic cameras to provide passengers with a view of the outside world [3]. While the company has yet to remove any windows from their planes, they have begun using this virtual window technology for first-class suites in the middle of the plane. This way, even if you have the middle suite, you can still have a view of the outside world.

The Las Vegas Sphere uses some similar technologies that would apply to our potential solution. Sphere Entertainment, the developers of the Las Vegas Sphere, utilize a system called Blue Sky. Blue Sky is the ultra-high-resolution camera system used to capture content for the Vegas Sphere. Working with STMicroelectronics, a team from 7thSense developed and manufactured an 18K sensor capable of capturing images at the scale and precision necessary for the Sphere’s display [4]. This solution demonstrates the complexity around capturing an image in a way that fits to the display, which is something for us to take into consideration.

Kithara Software is responsible for developing top-of-the-market real-time image capture. Camera Standards for image capture, image processing and subsequent control reactions in the same real-time context include the GigE Vision(C) and the USB3 Vision (C). Both cameras are pre-developed solutions to a live time display problem [5].

Apple Vision Pro, a virtual and augmented reality headset, has similar technologies to the ones we plan to include in our solution. Many of the specs for the image capture and display will be replicated in our design, such as refresh rate, latency, and resolution. For the image capture on the Vision Pro, there are two high-resolution main cameras, six world-facing tracking cameras, a TrueDepth camera, a LiDAR scanner, a flicker sensor, and an ambient light sensor. The display is a 23-million-pixel, 3D display system, made up of Micro-OLED. Refresh rates, latency, and resolution can be found in the solution requirements section of this document as we are also using the respective specs for our design [6].

# **Solution Approach**

Passengers of the future could expect a panel that will replace windowpanes. Cameras from the outside of the car will collect video data and project it onto the panels to allow passengers to see the outside world. From a manufacturing perspective, this will simplify the frame's manufacturing by not cutting out the windshield or the rear-view window. Airbags can be placed in a more optimal position in the car without the constraints of window placement. Windowless vehicles provide improved security in terms of vehicle break-ins. While improving the physical safety of passengers and their belongings and making manufacturing easier and more affordable, the projection of the outside world would have to be sufficient to not cause harm to a passenger’s mental or psychological health.

Table 3: Requirements and Specifications for the Real-Time Car Projector

|  |  |  |  |
| --- | --- | --- | --- |
| Description of Requirement | Expected Value | Importance\* | Compliance Verification Method |
| Refresh rate | ≥ 30 Hz | H | Measured |
| Latency | < 20ms | H | Test/Measured |
| Resolution | 1080p | E | Measure |
| Operating Temperature | −45℃ to 85℃ | M | Datasheet |
| Humidity | 0-95% | M | Datasheet |
| Power Consumption | ≤ 200 Watt | L | Measured |
| Input Voltage | 20VDC ± 10% | L | Measured |

\*Importance of the requirement: E=essential, H=high, M=moderate, L=low

# **Real-Time Car Projector**

The RTCP system is made of five subsystems, four of which the team focused on developing. The subsystems are the Input, Image Quality, Processor, Output, and Power Supply subsystems with the team choosing not to develop the Power Supply subsystem.

### **System Diagram**

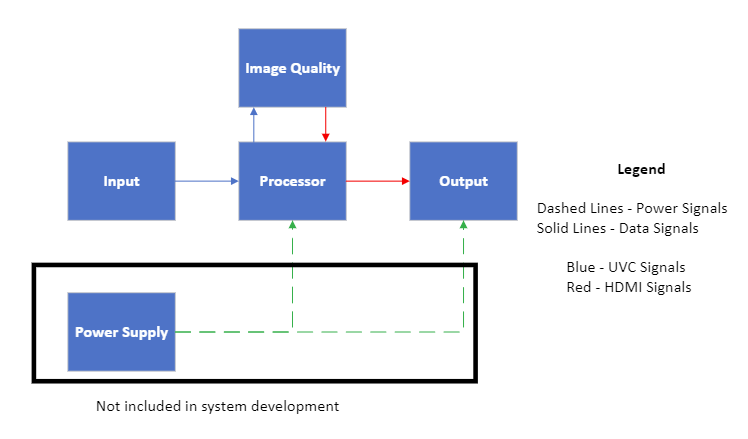


Figure 1: System Block Diagram

The Input subsystem refers to the cameras that will be used to collect video images of the outside world and turn them into digital data that can be read by the Processor subsystem. The Processor subsystem is the brain of the RTCP, holding the software in the Image Quality subsystem. The Processor subsystem will receive data from the Input subsystem, and the data collected will be given to the Image Quality subsystem. The Image Quality subsystem is entirely software-based. It takes the data from the Processor subsystem, analyzes, and processes it, then sends it back to the Processor. It will perform processes like color correction, fixing edge blending, splicing images together, and more. The Processor subsystem will send out this data to the Output subsystem, which is comprised of some sort of method of displaying the image of the outside world so that passengers can see it. The Input, Processor, and Output subsystems all include hardware that will need to be powered. The power supply subsystem is outside of the RTCP’s scope. RTCP’s focus is to prove that windows will not be a requirement for self-driving cars as the system will be able to properly provide a clear and accurate view of the outside world. Being able to power the system is outside of the team’s scope.

### **Subsystem Specifications**

#### **Input Subsystem**

The Input subsystem consists of three cameras with the appropriate specs in Table 1 and a 3D printed mount of the cameras. The cameras will be supplied with power over USB. This subsystem will capture the video data that will be processed or displayed in other subsystems.

Table 4: Requirements and Specifications for the Input Subsystem of the RTCP

|  |  |  |  |
| --- | --- | --- | --- |
| Description of Requirement | Specification | Importance\* | Compliance Verification Method |
| Resolution | At least 720p\*\* | H | Test/Measure |
| Number of Cameras | 3 | M | Measure |
| FOV | 180° | M | Measure |
| FPS | At least 30\*\* | H | Test/Measure |
| Operating Temperature | −45℃ to 85℃ | L | Datasheet |
| Output Signal Type | UVC 1.5 | E | Datasheet |
| Power Consumption | ≤1.5 W | L | Test/Measure |
| \*Importance of the requirement: E=essential, H=high, M=moderate, L=low  \*\*The resolution and fps are intertwined. For a resolution of 720p, the fps would have to be at least 60 fps. For a resolution of 1080p, the fps would have to be at least 30 fps. | | | |

#### **Processor Subsystem**

The Processor subsystem consists of two processors, a primary and a secondary. The primary processor handles signal processing and output of two video feeds. The secondary processor handles the output of a single video feed. The primary and secondary processors output their signals to the Output subsystem as HDMI. The processors communicate with each other using UART. UART will also be the method to send a video feed from the primary to the secondary. Both processors need to adhere to the specifications in Table 3.

Table 5: Requirements and Specifications for the Processor Subsystem of the RTCP

|  |  |  |  |
| --- | --- | --- | --- |
| Description of Requirement | Specification | Importance\* | Compliance Verification Method |
| Input Signal | Process ≥ 3 UVC signals | E | Controller Chipset |
| Inter-Processor Communication | UART | E | Controller Chipset |
| Price | < $100 | M | N/A |
| GPIO Pins | At least one for an error indicator | L | Datasheet |
| FPS | 60\*\* | E | Test/Measure |
| Resolution | 1080p | E | Test/Measure |
| Output Signal | 3 HDMI outputs | H | Controller Chipset |
| Power Consumption | ≤ 197 W | L | Test/Measure |
| Processor Speed | >1 GHz | H | Datasheet |
| Storage | >5 GB | H | Datasheet |
| RAM | >512MB | H | Datasheet |
| Operating System | 64-bit | E | OS Compliance |
| \*Importance of the requirement: E=essential, H=high, M=moderate, L=low  \*\*Able to support the lowest FOV of the components attached | | | |

#### **Image Quality Subsystem**

The Image Quality subsystem is completely software-based. The purpose of this subsystem is to read the video feeds, process them, then output them. A signal for inter-processor communications is required for debugging issues that may arise.

Table 6: Requirements, Specifications, and Transforms for the Image Quality Subsystem of the RTCP

|  |  |  |  |
| --- | --- | --- | --- |
| Description of Requirement | Specification | Importance\* | Compliance Verification Method |
| Supported Input Signals | 3 UVC | E | Driver Compliance |
| Resolution | 1080p | E | Test/Measure |
| Supported Output Signal | 3 HDMI | H | Driver Compliance |
| Inter-Processor Communication | UART (PL011) | E | Driver Compliance |
| GPIO Capability | Has embedded software to utilize GPIO pins with interrupt handler | L | Driver Compliance |
| Rotation | Each video must be able to rotate at least 180° degrees | M | Datasheet\*\* |
| Color Correction | Brightness, Contrast, Hue, Saturation, Sharpness | H | Datasheet\*\* |
| Distortion Correction | Fisheye Correction\*\*\* | H | Datasheet\*\* |
| Required Programming Language | C++ | H | Driver Compliance |
| Operating System | 64-bit & ARM compatible | E | OS Compliance |
| IDE | VScode | H | Driver Compliance |
| \*Importance of the requirement: E=essential, H=high, M=moderate, L=low  \*\*Datasheet refers to the documentation either of the OpenCV’s API or standards of signals.  \*\*\*OpenCV has a built-in function for fisheye correction that can be found in its documentation [7] | | | |

#### **Output Subsystem**

The Output subsystem consists of the projection system of three projectors. The purpose of this subsystem is to take the processed image and project it to the front interior panel of the car.

Table 7: Requirements, Specifications, and Transforms for the Output Subsystem for the RTCP

|  |  |  |  |
| --- | --- | --- | --- |
| Description of Requirement | Specification | Importance\* | Compliance Verification Method |
| Output Resolution | 1080p | E | Datasheet |
| Output Frame Rate | 30 FPS | E | Datasheet |
| Latency | 11 ms | E | Datasheet |
| Price | < $500 | H | Datasheet |
| Power | ≤150 W | H | Datasheet |
| Output Signal Type | HDMI | E | Datasheet |
| Aspect Ratio | 16:9 | E | Datasheet |
| \*Importance of the requirement: E=essential, H=high, M=moderate, L=low | | | |

### **Subsystem Selection Processes**

#### **Input Subsystem**

Table 8: Decision Matrix for Camera Choice for the Input Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Arducam [8] | EV2U-RMR2-MMC1-C1 [9] | OV3660 [10] |
| Resolution | 1080p or 720p | 1080p | 720p |
| FOV | 80° | 86°/72°/38° (D/H/V) | 66.5° |
| FPS | 30 or 60, depends on the resolution | 30 | 45\* |
| Operating Temperature | N/A\*\*\* | -20°C ~ 70°C | -20°C ~ 70°C |
| Price Per Unit | $24.95 | $69.43 | $22.67 |
| Power Consumption | N/A\*\*\* | ~1 W | 14 mAh battery |
| Output Signal Type\*\* | RAW 10 | UVC | RAW 10 |
| \*The FPS at 1080p is 20 and decreases further as the resolution increases.  \*\*Cameras can output multiple different formats the listed ones offer the most benefit.  \*\*\*Couldn’t find this information online. | | | |

The chosen solution for the Input subsystem is the EV2U-RMR2-MMC1-C1 camera. This camera was chosen because, while it’s the most expensive of the three, it works the best for the team. The FOV, resolution, and fps are well-matched for the subsystem, and the UVC output signal type makes it possible to test the cameras before developing other subsystems.

If the team were doing a design project in industry instead of MSOE, the camera choice would be different. In industry, the operating temperature of the camera would be much more important as the system would be used in very cold and extremely hot areas. Whereas, for this design project, the operating temperature isn’t as important as it won’t be tested at those extremes. Continuing, as the RTCP would be a system that goes within an autonomous vehicle, the Input subsystem would use the same cameras that are used for the self-driving system and there likely wouldn’t be a need for separate cameras for the RTCP.

#### **Processor Subsystem**

The first decision to be made was if the Processor subsystem would be designed using digital logic (FPGA) or an embedded system. After some research, the team found that most FPGA options were too expensive for the team. Also, the team has much more experience with embedded systems, so they felt much more comfortable working with an embedded system than an FPGA system. Due to these reasons, the team chose to investigate embedded system options to design the Processor subsystem.

Table 9: Decision Matrix for Processor Choice for the Processor Subsystem

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria | Raspberry Pi 5 [11] | Rock 3 Model C 2GB [12] | Raspberry Pi Zero W [13] |
| USB Input | 4 ports (2 3.0 and 2 2.0) | 4 ports (1 3.0 and 3 2.0) | 1 Micro USB GTO port |
| Power Consumption | 5 V/5 A | 5 V/3 A | 5 V/2.5 A |
| Cost | $80 | $42.94 | $15 |
| GPIO Pins | 40 | 40 | 40 |
| Resolution | 4k | 1080p | 4k |
| FPS | 60 | 60 | 60 |
| Output Signal Type | 2 Mini HDMI and 2 4-lane MIPI | 1 HDMI and 1 2-lane MIPI | 1 Mini HDMI and 1 CSI Camera Connector |
| Processor Speed | 2.4 GHz | 1.6 GHz | 1 GHz |
| Storage | Micro SD card | SD card, NVMe SSD, SATA SSD | Micro SD Card |
| RAM | 8 GB | 2 GB | 512 MB |
| Operating System | 64-bit | 64-bit | 64-bit |

The chosen solution for the Processor subsystem is the Raspberry Pi 5 working in tandem with the Raspberry Pi Zero W. The team could not find any low-cost processors with three HDMI outputs, so they were forced to use two processors. The best processors they could find that would take in 3 UVC inputs were the Raspberry Pi 5 and the Rock 3 Model C 2GB. The boards are similar, and the differences that the team cared the most about were the cost and the output signals. While the Raspberry Pi 5 is about twice the price of the Rock 3 Model C 2GB, it also has twice the outputs, two of which are HDMI. As for getting a 3rd HDMI output, the Raspberry Pi Zero W will be the secondary processor that would easily be able to take in data from one of the Raspberry Pi 5’s 4-lane MIPI ports and output it through HDMI, and it costs $15. So, despite the higher cost, the team found the Raspberry Pi 5 to work for the purposes of the RTCP much better.

If the team were doing a design project in industry instead of MSOE, the processor(s) choice would be different. In industry, the car’s computer will receive input from the cameras. Depending on the processor used the signal processing might need to be done on a secondary processor. If that is the case, then the Raspberry Pi 5 or similar could be utilized.

Table 10: Decision Matrix for the Cooling Solution Choice for the Raspberry Pi 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Criteria | Official Pi 5 Case [14] | 52Pi ICE Tower [15] | Cooling Fan and Heatsink Set [16] | Geek Pi/52 Pi Case [17] |
| Cost | $10.39 | $17.31 | $8.22 | $12.99 |
| Size (LxWxH, in) | 3.86x2.77x1.30 | 2.95x2.60x2.17 | 1.57x0.39x1.57 | 3.46x2.2x0.94 |
| Type of cooling | Passive and Active | Passive and Active | Passive and Active | Passive |

While testing the Raspberry Pi 5, it was determined that a cooling solution was required to use it as the team needed to. After some research, it was discovered that it would be hard to cause thermal throttling on the Raspberry Pi Zero W even if somebody tried to, so it was decided that buying a cooling solution for the Raspberry Pi Zero W was unnecessary. The chosen solution was the Official Pi 5 Case. The official case isn’t the cheapest solution, but as it’s the official case from Raspberry Pi, it was guaranteed that it would work with the Raspberry Pi 5. Also, being a case, it offers good protection for the Raspberry Pi 5 along with a fan that connects directly to the Raspberry Pi 5 and a heatsink, offering both passive and active cooling.

#### **Image Quality Subsystem**

The first choice was to pick a Computer Vision library that could fulfill all requirements from the Subsystem Specifications section. OpenCV, an open-source computer vision and machine learning software library, was selected due to its abundance of enterprise-use , documentation, and compatibility [18]. The computing processor that was chosen was the Raspberry Pi 5 (see the above section). The operating system that was chosen was the Raspberry Pi OS, which is recommended by the Raspberry Pi Foundation. Raspberry Pi OS is a fork of the Debian distribution of Linux. This would allow the team to utilize Debian’s package manager, bookworm to download and install well-known programs like VScode and OpenCV. An early issue the team ran into was OpenCV compiling and running on their Windows computers, while not being able to run on the Raspberry Pi OS. The solution was to generalize the generation of build files for the project. CMake was selected due to its vast compatibility and references to it inside of the OpenCV documentation.

The majority of the processing that that image quality subsystem was the stitching of three real-time video feeds. First, each camera was tested and confirmed to have the ability to output a live feed with a resolution of 1920x1080 and a frame rate of 30 frames per second. The Raspberry Pi 5 did not have an issue with meeting this goal when each camera was outputting to a separate window. After each camera was verified, three cameras were stitched together using three video stitching algorithms. All stitching algorithms are based on finding like pixels between two images and overlapping the images in a way that there are no repeat pixels. The first algorithm tested was OpenCV’s panoramic stitching. The output was a live feed that had too slow of a framerate to measure. The next algorithms tested were the ORB and then the SURF algorithms were tested. The SURF algorithm appeared to be promising, but it suffered from the same issues as the panoramic stitching, too slow of a framerate to measure. The ORB stitching had issues with detecting images from two of the three cameras resulting in two-thirds of the live image not displaying anything. OpenCV’s fisheye distortion algorithm was utilized to great effect to undistort the images that the cameras were collecting. Due to the unsuccessful stitching algorithms, it could only be verified when viewing each camera individually.

The solution to the stitching problem is to use better hardware that is purpose-built for solving matrix algebra fast. Some development boards were initially considered but fell well outside of the project’s development budget causing it not to be considered as a realistic choice.

#### **Output Subsystem**

Table 11: Decision Matrix for the Projector Choice for the Output Subsystem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Specification | VOPPLS [17] | Elephas [18] | Build our own [19] | VISSPL [20] |
| Output Resolution | 1080p | 1080p | 4k | Up to 4k | 1080p |
| Output Refresh Rate | 30 fps | 30-60 fps [21] | 30-60 fps | 30-60 fps | 30-60 fps |
| Latency | 11 ms | Low\* | Low\* | Low\* | Low\* |
| Price | < $500 | 3x$64 | 3x$79.99 | 3x$329.96 | 3x$85.79 |
| Power | ≤150 W | 50 W [22] | 50 W | ~ 100 W\*\* | 50 W |
| Input Signal | HDMI | HDMI/USB/AV | HDMI | HDMI | HDMI/USB/AV |
| Aspect Ratio | 16:9 | 16:9 | 16:9 | 16:9 | 16:9 |
| \*Low as in not enough to make a visible difference while projecting. This passes specs as 11 ms is comparable to the projector’s latency performance.  \*\*~100 W is an estimate. | | | | | |

The chosen solution for the Output subsystem is the VISSPL projector as it is reasonably priced and meets all criteria for the desired performance. The price to build our own was priced out as follows:

Table 12: BOM for building a projector.

|  |  |  |
| --- | --- | --- |
| Light Source | 100 W LED | $45.06 |
|  | Heat Sink | $54.95 |
|  | Power Board | $8.49 |
| Light Control | Fresnel Lens | $15.09 |
| Focusing System | Glass Panel, adjusting pulley and belt system | Miscellaneous Components\* |
| Image Source | Smart Phone LCD Display | $56.38 |
| Projection Lens | Large Format Camera Lens | $49.99 |
| Miscellaneous Components\* | - | ~ $100 |
| TOTAL | - | ~ $329 |
| \*Cheaper items | | |

### **Bill of Materials**

Table 13 shows the BOM for the RTCP. However, the purchase of the USB C pd and cord was made by one of the team members, so that cost isn’t counted against the budget.

Table 13: BOM

|  |  |  |  |
| --- | --- | --- | --- |
| Component | Cost Per Unit | Number of Units | Total Cost |
| EV2U-RMR2-MMC1-C1 | $69.43 | 4 | $277.72 |
| USB Hub | $17.99 | 1 | $17.99 |
| JST SH 5 pin cord | $9.49 | 1 | $9.49 |
| Raspberry Pi 5 | $80.00 | 1 | $80.00 |
| Raspberry Pi Zero W | $15.00 | 1 | $15.00 |
| Mini HDMI to HDMI | $8.02 | 3 | $24.06 |
| SD card | $9.99 | 2 | $19.98 |
| VISSPL Projector | $71.49 | 3 | $214.47 |
| Mirco USB cable | $7.69 | 1 | $7.69 |
| MIPI cable | $2.35 | 1 | $2.35 |
| HDMI to micro-HDMI (HDMI 2.1) | $7.99 | 1 | $7.99 |
| USB to micro-USB (USB 2.0) | $4.99 | 1 | $4.99 |
| USB C pd and cord | $15.99 | 1 | $15.99 |
| Official Raspberry Pi 5 Case | $10.00 | 1 | $10.00 |
| Cooling Fan & Heatsink Set | $8.21 | 1 | $8.21 |
| Micro HDMI to HDMI cord | $7.99 | 2 | $15.98 |
| Shipping |  |  | $36.48 |
| Total Spent: $768.39 | | Remaining Budget: $247.60 | |

# **System Verification Results**

The full system results can be seen in Appendix H. The RTCP System Operation Verification Plan noted in Table 26 is very interesting because OpenCV prioritizes resolution over everything else. This allows confidence that the USB cameras are always at 1080p. The framerate’s maximum value can be set by OpenCV, but the actual value can differ depending on the amount of processing the processor is doing at any given time. Despite the framerate being set to 30 fps, the framerate can drop below this value. There are two potential solutions to the framerate issue. The first would be to use cameras that allow framerates greater than 30 fps. This could solve the issue due to the software throttling the framerate to not exceed the maximum. The second and most probable solution would be to use a processor that has more processing power. As shown through running the stitching test (step 5), the processor cannot keep up with three-camera stitching. Sadly, the latency of the system could not be tested due to the lack of testing equipment. The testing procedure would involve the use of a high-speed camera. Software starts the camera’s recording and sends a signal to the display at the same time. The timestamp in the recording of when the signal is displayed on the monitor is the latency of the system.

# **Summary**

This document contains the problem and solution that RTCP sets out to solve. RTCP sets out to make cars with Level 5 automation safer and reduce the access points of personal theft. It outlines the stakeholders in this project, including a luxury car company, passengers, the RTCP team, and the team designing the power system of the car’s accessories. Alternate solutions that are currently on the market were being looked at with the most notable one being Apple Vision Pro. The approach to the solution can be taking the Apple Vision Pro and scaling it up to the size of a car. The full list of system requirements can be seen in Table 3. The system can be broken into four subsystems: Input, Processor, Image Quality, and Output. The Input subsystem collects three video streams of the outside work and sends them to the Image Quality subsystem through the Processor subsystem. The Processor subsystem holds the hardware components of the Image Quality subsystem. The Image Quality subsystem utilizes OpenCV to perform transformations on the video signals and send them to the Output subsystem through the Processor subsystem. The Output subsystem projects the edited video streams for the passengers to view. All non-labor expenses for developing RTCP are shown in Table 13. The final section in this report analyses the system’s verification testing results. The RTCP team is grateful to Dr. Kelnhofer for his guidance throughout the project and to Field Theory Consulting Inc and Bob Radke for their financial support. The RTCP team would also like to thank Mr. Jim Frommell for his help with component acquisition.

# **Bibliography**

|  |  |
| --- | --- |
| [1] | Faist, "What are the 6 levels of autonomous vehicles?," 11 January 2023. [Online]. Available: https://www.faistgroup.com/news/autonomous-vehicles-levels/. [Accessed 8 February 2024]. |
| [2] | Collins Aerospace, "F-35 Gen III Helmet Mounted Display System (HMDS)," [Online]. Available: https://www.collinsaerospace.com/what-we-do/industries/military-and-defense/displays-and-controls/airborne/helmet-mounted-displays/f-35-gen-iii-helmet-mounted-display-system. [Accessed 8 February 2024]. |
| [3] | F. Street and M. Ros, "Are windowless planes the future of travel?," 27 March 2019. [Online]. Available: https://www.cnn.com/travel/article/emirates-windowless-planes/index.html. [Accessed 20 January 2024]. |
| [4] | AVNetwork Staff, "Everything We know about the Vegas Spere, So Far-A Running Blog," [Online]. Available: https://www.avnetwork.com/news/sphere-las-vegas-immersive#:~:text=Big%20Sky%20is%20the%20ultra,live%20entertainment%20experiences%20for%20Sphere. [Accessed 18 January 2024]. |
| [5] | Kithara Software, "Image Capturing and Processing in Real Time-Kithara Software," [Online]. Available: https://kithara.com/en/solutions/realtime-image-processing. [Accessed 18 February 2024]. |
| [6] | A. Inc, "Apple Vision Pro," 2024. [Online]. Available: https://www.apple.com/apple-vision-pro/specs/. [Accessed 25 January 2024]. |
| [7] | OpenCV, "Fisheye Camera Model," [Online]. Available: https://docs.opencv.org/4.x/db/d58/group\_\_calib3d\_\_fisheye.html. [Accessed 2 May 2024]. |
| [8] | ArduCam, "High-Resolution Autofocus Camera for Raspberry Pi," September 2021. [Online]. Available: https://cdn.sparkfun.com/assets/4/3/2/f/6/arducam-16mp-autofocus-camera-product-brief.pdf. [Accessed 15 February 2024]. |
| [9] | Mouser Electronics, "EV2U-RMR2-MMC1-C1," [Online]. Available: https://www.mouser.com/ProductDetail/Innodisk/EV2U-RMR2-MMC1-C1?qs=HoCaDK9Nz5eHrsWUaGgKGg%3D%3D. [Accessed 15 February 2024]. |
| [10] | OmniVision Technologies, Inc., "OV3660 Datasheet," 13 May 2011. [Online]. Available: https://m5stack.oss-cn-shenzhen.aliyuncs.com/resource/docs/datasheet/unit/OV3660\_CSP3\_DS\_1.3\_sida.pdf. [Accessed 15 February 2024]. |
| [11] | Raspberry Pi, "Raspberry Pi 5," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-5/?resellerType=industry. [Accessed 4 April 2024]. |
| [12] | Arduino Store, "Rock 3 Model C 2GB," [Online]. Available: https://store-usa.arduino.cc/collections/boards-modules/products/rock-3-model-c-2gb?\_pos=28&\_fid=547efdc2d&\_ss=c. [Accessed 5 April 2024]. |
| [13] | Raspberry Pi, "Raspberry Pi Zero W," [Online]. Available: https://msoe365-my.sharepoint.com/:w:/g/personal/kelnhofer\_msoe\_edu/EWRi8gJgBY9BvExLlfJqL24BosD\_k7SE3eG0i-wIiu8bSA?e=5X3YU0. [Accessed 24 April 2024]. |
| [14] | Raspberry Pi, "Raspberry Pi Case for Raspberry Pi 5," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-5-case/. [Accessed 23 October 2024]. |
| [15] | The Pi Hut, "ICE Tower CPU Cooler for Raspberry Pi 5," [Online]. Available: https://thepihut.com/products/ice-tower-cpu-cooler-for-raspberry-pi-5. [Accessed 23 October 2024]. |
| [16] | The Pi Hut, "40mm Cooling Fan & Heatsink Set for Raspberry Pi 5," [Online]. Available: https://thepihut.com/products/4010-black-cooler-fan-with-heatsinks-for-raspberry-pi-5. [Accessed 23 October 2024]. |
| [17] | Ali Express, "GeeekPi Raspberry Pi 5 Case with heatsink Raspberry Pi 5 Aluminum with Fan and Copper heatsink Pack," [Online]. Available: https://www.aliexpress.us/item/3256805962551448.html?gatewayAdapt=glo2usa4itemAdapt. [Accessed 23 October 2024]. |
| [18] | Open CV, "Computer Vision software and services," [Online]. Available: https://opencv.org/. [Accessed 29 March 2024]. |
| [19] | FFmpeg, "About FFmpeg," [Online]. Available: https://ffmpeg.org/about.html. [Accessed 1 April 2024]. |
| [20] | Stackshare, "FFMPEG vs OpenCV: What are the differences?," [Online]. Available: https://stackshare.io/stackups/ffmpeg-vs-opencv. [Accessed 1 April 2024]. |
| [21] | Amazon, "Mini Projector, VOPLLS 1080P Full HD Supported Video Projector, Portable Outdoor Home Theater Movie Projector, 50% Zoom, Compatible with HDMI, USB, AV, Smartphone/Tablet/Laptop (Small Projector)," [Online]. Available: https://www.amazon.com/Projector-VOPLLS-Supported-Compatible-Smartphone/dp/B0BG2GFCGD?source=ps-sl-shoppingads-lpcontext&ref\_=fplfs&smid=A1RYXKDLCJN3E0&th=1. [Accessed 2 April 2024]. |
| [22] | Walmart, "Mini Projector, ELEPHAS 8000LM Portable Projector 1080P HD Movie Projector Compatible with Android/iOS/Windows/TV Stick/HDMI/USB," [Online]. Available: https://www.walmart.com/ip/Mini-Projector-ELEPHAS-8000LM-1080P-HD-Projector-LED-Technology-200-Display-Support-Home-Outdoor/2659210314?wmlspartner=wlpa&selectedSellerId=101083583. [Accessed 2 April 2024]. |
| [23] | D. Perks, "Building a TRUE 4k home cinema projector (it's awesome)," 15 June 2020. [Online]. Available: https://www.youtube.com/watch?v=YfvTjQ9MCwY. [Accessed 4 April 2024]. |
| [24] | Amazon, "Mini Projector, VISSPL Full HD 1080P Video Projector, Portable Outdoor Projector with Tripod, Kids Gift, Home Theater Movie Phone Projector Compatible with Android/iOS/Windows/TV Stick/HDMI/USB," [Online]. Available: https://www.amazon.com/Projector-VISSPL-Portable-Outdoor-Compatible/dp/B0CD2BS3QX?tag=unboxtechi0da-20&crid=1BM4PZ99DAN2E&keywords=visspl%2Bmini%2Bprojector&qid=1707454524&sprefix=visspl%2B%2Caps%2C143&sr=8-3&ufe=app\_do%3Aamzn1.fos.006c50ae-5d4c-4777-9bc0. [Accessed 2 April 2024]. |
| [25] | C. D. Systems, "christiedigital.com," 2024. [Online]. Available: https://www.christiedigital.com/about/display-technology/high-frame-rates/#:~:text=The%20ideal%20frame%20rate%20for%20a%20projector%20depends%20on%20the,benefit%20from%20higher%20frame%20rates.. [Accessed 23 April 2024]. |
| [26] | Nebula, "How Many Watts Does a Projector Use? (Full Explanation)," 05 December 2023. [Online]. Available: https://us.seenebula.com/blogs/buying-guides/how-many-watts-does-a-projector-use#:~:text=A%20good%20projector%20typically%20uses,advanced%2C%20high%2Dbrightness%20units.. [Accessed 23 April 2024]. |
| [27] | ANY-maze, "Connecting multiple USB cameras to a computer," 10 September 2021. [Online]. Available: https://www.any-maze.com/support/guides/connecting-multiple-usb-cameras-to-a-computer/. [Accessed 11 April 2024]. |
| [28] | U. D. o. Transportation, "US Department of Transportaion, Automated Vehicles: Comprehension Plan," January 2021. [Online]. Available: https://www.transportation.gov/sites/dot.gov/files/2021-01/USDOT\_AVCP.pdf. [Accessed 11 April 2024]. |
| [29] | SCN Staff, "Behind the Scenes of the Company Helping Power the Video Systems at Sphere," 20 December 2023. [Online]. Available: https://www.avnetwork.com/news/behind-the-scenes-of-the-company-helping-power-the-video-systems-at-sphere. [Accessed 25 January 2024]. |
| [30] | D. Hawley, "How Many Volts Are In A Car Cigarette Lighter?," 13 March 2023. [Online]. Available: https://www.jdpower.com/cars/shopping-guides/how-many-volts-are-in-a-car-cigarette-lighter#:~:text=A%20car%20cigarette%20lighter%20has,%2C%20mobile%20phones%2C%20and%20laptops. [Accessed 2 February 2024]. |
| [31] | trejan, "Raspberry Pi 5 MIPI to HDMI," 12 April 2024. [Online]. Available: https://forums.raspberrypi.com/viewtopic.php?t=368965&e=1&view=unread#p2211455. [Accessed 12 April 2024]. |
| [32] | ekidney, "9 displays output," 18 December 2023. [Online]. Available: https://forums.raspberrypi.com/viewtopic.php?p=2171399. [Accessed 12 April 2024]. |

**Appendix A – Wiring Diagram**

A diagram of a computer network

Description automatically generated

Figure 2: RTCP Wiring Diagram

# **Appendix B – Data Sheets**

See [this website](https://hero.page/nosamthewise/real-time-car-projector) for the devices used. It also has the LinkedIn profiles of the team members and the Github for the project.

# **Appendix C – 3D-Printed Files**

See [this folder](https://msoe365-my.sharepoint.com/personal/kelnhofer_msoe_edu/_layouts/15/onedrive.aspx?FolderCTID=0x012000EA2ABF16E089B0459FB5071A4BBB41A9&id=%2Fpersonal%2Fkelnhofer%5Fmsoe%5Fedu%2FDocuments%2FSenior%20Design%20Teams%2FAY24S2%2FTeam%204%2F3D%20Printing%20Files) for the 3D-printed files.

# **Appendix D – Input Subsystem Testing Results**

List of needed equipment:

* Computer with the latest version of OBS
  + OBS does not support multi-video inputs from a singular USB hub. One computer with OBS installed can be used for the FOV testing assuming it has at least 3 USB ports. Otherwise, multiple computers with OBS must be used.
* Camera rig
* 3 EV2U-RMR2-MMC1-C1 cameras
* Current clamp
* Oscilloscope with at least two channels
* Modified camera wire
* Marker
  + i.e., cone, flag, etc.
* Protractor

A diagram of a computer system

Description automatically generated

Figure 3: Input Subsystem Camera Operation Testing Setup

**A diagram of a computer component

Description automatically generated**

Figure 4: Input Subsystem Overall FOV Testing Setup

Table 14: Input Subsystem Camera Operation Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Set up test equipment as shown in Figure 3. | N/A | N/A | N/A |
| 2 | Measure voltage from the camera using an oscilloscope | ~5 V | 5.043 V | Pass |
| 3 | Measure current from the camera using a current clamp and oscilloscope | ≤200 mA\* | 182 mA | Pass |
| 4 | Calculate power consumption from the camera using the voltage and current measurements (P=IV) | ≤1 W | 918 mW | Pass |
| 5 | Use OBS to see the resolution from the camera | 1080p | 1080p | Pass |
| 6 | Use OBS to see the FPS of the camera | 30 Hz | 30 Hz | Pass |
| 7 | Repeat Steps 1-6 for the remaining 2 cameras. | N/A | N/A | N/A |

\*Calculated using expected power divided by expected voltage.

Table 15: Input Subsystem Overall FOV Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Set up test equipment according to Figure 4. | N/A | N/A | N/A |
| 2 | Rotate the arms of the test rig until the overall FOV is about 180°. | N/A | N/A | N/A |
| 3 | Starting at 0°, place a marker at ~45° intervals until 180° is reached. | N/A | N/A | N/A |
| 4 | Measure the angle between the marks to find the FOV of the cameras | ≥180° | >180° | Pass |
| 5 | Measure the angle of the hinges on the camera rig to determine the most optimal angle between cameras. | N/A | Side arms at 130° in reference to the middle section. | N/A |

All goals the test set out were achieved. (i) The most optimal angle between each adjacent camera is 130°. (ii) Each camera can output video with a resolution of 1080p at 30 FPS. (iii) Each camera uses 918 mW of power. While the results did match up to the expectations, no experiment can produce perfect results. For example, the measured voltage the camera received wasn’t exactly 5 V. This is due to the imperfections in both the measurement equipment and the device under test. In the camera operations test, the wires being measured had been partially cut and soldered so that the voltage and current through the USB cord could be measured without interference from the data wires. The solder could introduce imperfections to the circuit, causing the voltage and current to vary from normal. Similarly, the equipment used to test the voltage and current isn’t perfect and has measurement errors, which would cause the results to be incorrect. Continuing with the FOV testing, it’s likely that the markers weren’t placed perfectly at every 45° interval due to natural human error. Also, the protractor used to measure the angle of the hinges on the camera rig isn’t perfectly accurate, both due to the limits of the protractor and the human eye. Both could introduce errors in the angles seen during testing.

# **Appendix E – Processor Subsystem Testing Results**

List of needed equipment:

* 1 Raspberry Pi 5
* 1 Raspberry Pi Zero W
* Power supply that can output at least 5 V/5 A
  + Need either 1 with 2 outputs or 2
* 2 cords or converters that connect to USB C
  + Needed to connect the Raspberry Pi 5 to its power supply
  + One will be modified to test power consumption
* 2 cords or converters that connect to Mirco USB
  + Needed to connect the Raspberry Pi Zero W to its power supply
  + One will be modified to test power consumption
* Current clamp
* Oscilloscope with at least two channels
* Wires for UART communication
* Keyboard with USB connectivity
  + Used to connect to and code the Raspberry Pis
* Mouse
  + Not required but it makes using the Raspberry Pis a lot easier

A diagram of a computer system

Description automatically generated

Figure 5: Raspberry Pi 5 Power Test Setup

A diagram of a diagram

Description automatically generated

Figure 6: Raspberry Pi Zero W Power Test Setup

Table 16: Processor Subsystem Power Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Download benchmark script to test power consumption from the Raspberry Pis. | N/A | N/A | N/A |
| 2 | Set up test equipment according to Figure 5. | N/A | N/A | N/A |
| 3 | Run the benchmark script on the Raspberry Pi 5. | N/A | N/A | N/A |
| 4 | Measure voltage from the Raspberry Pi 5 using an oscilloscope. | ≤5 V\* | 4.73 V | Pass |
| 5 | Measure current from the Raspberry Pi 5 using a current clamp and an oscilloscope. | ≤3 A\* | 0.9 A | Pass |
| 6 | Calculate power consumption from the camera using the voltage and current measurements (P=IV) | ≤67 W\* | 4.257 W | Pass |
| 7 | Set up test equipment according to Figure 6. | N/A | N/A | N/A |
| 8 | Run the benchmark script on the Raspberry Pi Zero W. | N/A | N/A | N/A |
| 9 | Measure voltage from the Raspberry Pi Zero W using an oscilloscope. | ≤5 V\* | 4.95 V | Pass |
| 10 | Measure current from the Raspberry Pi Zero W using a current clamp and an oscilloscope. | ≤1 A\* | 0.44 A | Pass |
| 11 | Calculate power consumption from the camera using the voltage and current measurements (P=IV) | ≤5 W\* | 1.298 W | Pass |
| 12 | Add the calculated powers from steps 6 and 11 to get the total subsystem power consumption. | ≤197 W | 5.555 W | Pass |
| \*These values are from the wall wart supply used in the test. | | | | |

A diagram of a computer program

Description automatically generated

Figure 7: Processor Subsystem Operation Test Setup

Table 17: Processor Subsystem Operation Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Set up test equipment according to Figure 7. | N/A | N/A | N/A |
| 2 | Run a script on the Raspberry Pi 5 that sends a message to the Raspberry Pi Zero W. | N/A | N/A | N/A |
| 3 | Check if the Zero W received the message. | N/A | N/A | Pass |
| 4 | Set pin GPIO17 on the Raspberry Pi Zero W high. | N/A | N/A | N/A |
| 5 | Read the output of the GPIO pin with an oscilloscope. | 3.3 V | 3.3 V | Pass |
| 6 | Set pin GPIO24 on the Raspberry Pi 5 high. | N/A | N/A | N/A |
| 7 | Read the output of the GPIO pin with an oscilloscope. | 3.3 V | 3.3 V | Pass |

The results for the Processor subsystem are as follows. (i) The power consumption of the Raspberry Pis, and through them, the power consumption of the overall subsystem, were in spec. (ii) The Raspberry Pi 5 and Zero W were able to communicate with each other through UART. (iii) The GPIO pins on the Pis function as intended. As such, the Processor subsystem’s functionality has been validated.

# **Appendix F – Output Subsystem Testing Results**

List of needed equipment:

* 3 VISSPL projectors with the appropriate in-box components
  + 3 HDMI to HDMI connector cables
    - Needed to connect projectors to a personal computer
  + 3 Wall Wart power supplies
    - Voltage, Current, Voltage specs are listed on the AC Adapter
* White projection surface (i.e. white wall, board, or sheet)
* USBC to HDMI adapter with 2-3 HDMI connections

A picture containing text, electronics

Description automatically generated

Figure 8: Output Subsystem AC Adapter

Table 18: Output Subsystem Power Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Refer to the AC Adapter in Figure 8 for power specs\* | N/A | N/A | N/A |
| 2 | Output Current Spec. | < 3 A | 2.3 A | Pass |
| 3 | Output Voltage Spec. | < 60V | 21 V | Pass |
| 4 | Output Power Spec. | ≤ 100 W | 43 W | Pass |
| \*Power could be verified by using a modified power cable, a current probe, an oscilloscope and the relationship P=VI. The modified power cable would have to have exposed wires to measure voltage. In this case, we did not have the resources to create a modified power cable. | | | | |

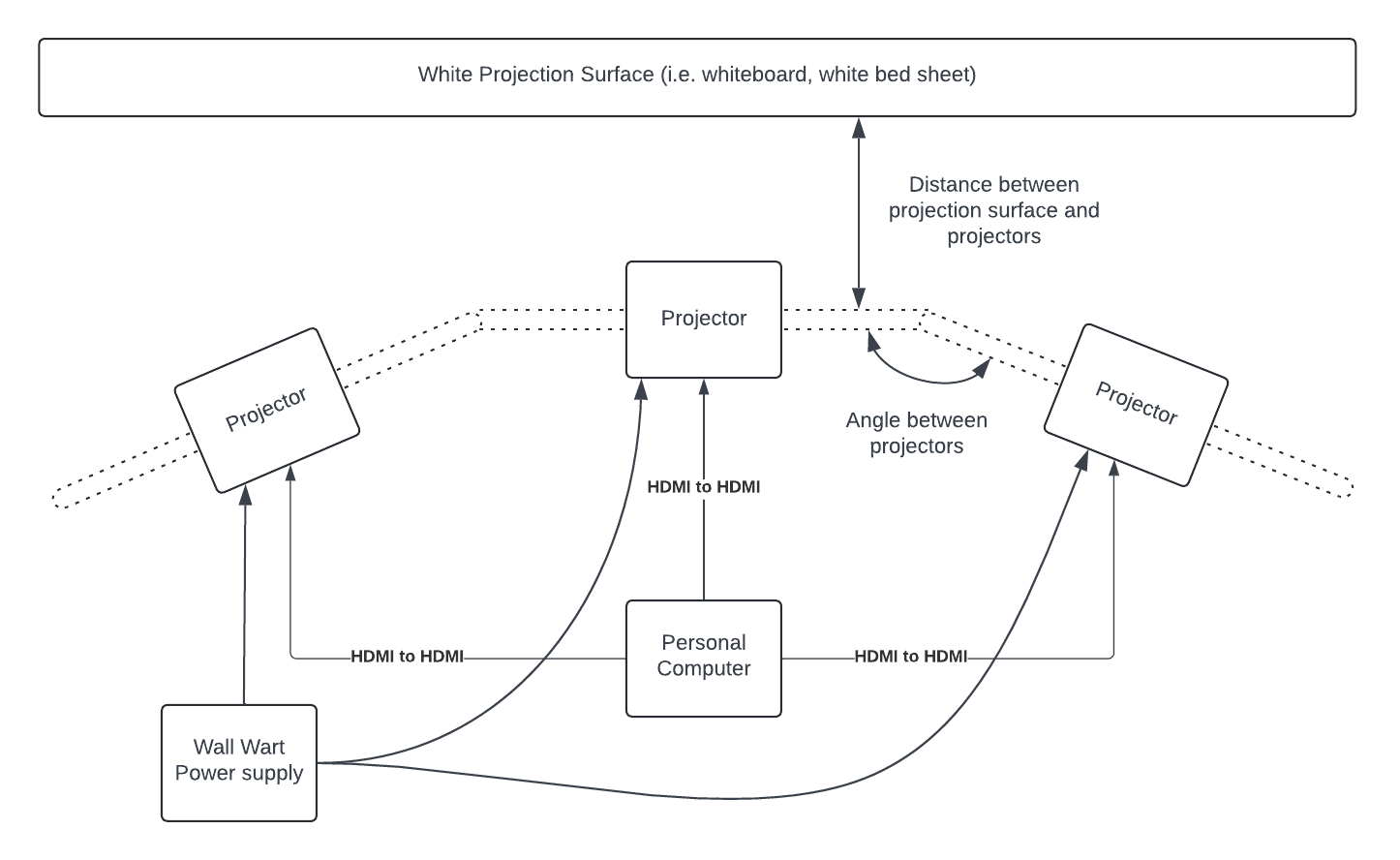


Figure 9: Output Subsystem Operation Test Setup



Figure 10: Output Subsystem Split Image Test

Table 19: Output Subsystem Operation Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Setup up this test according to Figure 9. | N/A | N/A | N/A |
| 2 | Run the provided Python Script which will split an image. Drag the windows to each projector screen. | N/A | N/A | N/A |
| 3\* | Visually verify that a projected image is apparent and clear | N/A | N/A | N/A |
| 4 | Adjust angle of projector stand arms to create one continuous image | Varies\*\* | Varies | N/A |
| 5 | Adjust focus of projector to unblur the image | Varies | Varies | N/A |
| 6 | Adjust projector distance from the projection surface to ensure a clear image | Varies | Varies | N/A |
| \*Steps 3-5 are interchangeable and can be repeated as the technician sees fit to ensure a clear projected image  \*\*These settings are up to the technician and are subject to change to provide the most optimal viewing experience | | | | |

The results for the Output subsystem are as follows. (i) The power consumption of the projectors was in spec. (ii) The projectors were able to project one image split into three segments for each projector. (iii) The optimal distance from the projection surface and from the ground will vary and will be up to the technician. Angle of the two outside projectors with respect to the middle projector and the focus of the projectors will be up to the technician as well. All projector settings are subject to change to provide the most optimal viewing experience.

# **Appendix G – Image Quality Subsystem Testing Results**

List of needed equipment:

* 1 Raspberry Pi 53 UVC Webcams
* 1 HDMI Display
* 1 USB Hub
* 1 USB Keyboard
* 1 USB Mouse
* 3 .xml Configuration Files
* 1 Raspberry Pi 5 Operation Program
* DMM with thermal probe

Table 20: Radial, or fisheye, distortion test.

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Description of Step | Objective | Measured Value |
| 1 | Confirm that each camera is connected to its corresponding USB port. | The program correlates USB ports to specific camera config files. | N/A |
| 2 | Plug a display into HDMI port 0 on the Raspberry Pi 5. | We need a way to view the camera's outputs. | N/A |
| 3 | Using the Terminal type $’./output CAM2’ into bash. Press q at any time to quit. | This will run the program and open a window for each camera. | N/A |
| 4 | Record peak temperature using the DMM and thermal probe | This allows more insight if previous tests fail | 58 °C |
| 5 | Throughout the test record the peak CPU usage using Task Manager | This allows more insight if previous tests fail | 24% |
| 6 | Hold the checkerboard calibration image 1 foot away from camera 1 and take a screen capture for further analysis. | Removing the fish-eye or barrel distortion leaves a very qualitative pass/fail test. Ideally each square should have straight lines. | N/A |
| 7 | Repeat step 5 for camera 2 and 3. | N/A | N/A |

Table 21: Dead Zone Test

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Objective | Expected Value | Measured Value |
| 1 | Confirm that each camera is connected to its corresponding USB port. | The program correlates USB ports to specific camera config files. | N/A | N/A |
| 2 | Plug a display into HDMI port 0 on the Raspberry Pi 5. | We need a way to view the camera's outputs. | N/A | N/A |
| 3 | Using the Terminal type $’./output PANO’ into bash. | This will run the program with one window for all cameras. | N/A | N/A |
| 4 | Find the camera seem, move ~3 feet away from the closest camera, and measure the space between the outputs | This measures the amount of the image that is being exclude due to the algorithm. | Exploration | Fail – fps to slow |
| 5 | Record peak CPU Usage percentage | This allows more insight if previous tests fail. | <25% | 90% |
| 6 | Record peak temperature | This allows more insight if previous tests fail. | <80 °C | 70 °C |

The results of the Image Quality Subsystem are as follows: (i) The CPU usage peaked at 24% during the radial distortion test and 90% during the dead zone test. (ii) The CPU temperature peaked at 58 °C during the radial distortion test and 70 °C during the dead zone test. (iii) The radial distortion of the cameras has been noticeably flattened from the Input subsystem test. (iv) The dead zone test failed. The dead zone test failed due to the frame rate of the output being low and unreadable. The spiking of CPU usage and thermal readings could either indicate that it does not have enough processing power, or the CPU is thermal throttling. Currently, multithreading is enabled to use all CPU cores, the next step would be to off load CPU processing to the GPU or reduce the amount of math done by OpenCV when stitching the images together.

# **Appendix H – Full System Testing Results**

List of needed equipment:

Required:

* 1 Raspberry Pi 5
* USB C PD power supply for the Raspberry Pi 5
  + Must output at least 5 V/5 A (25 W)
* 1 Raspberry Pi Zero W
* Micro USB power supply and cable for the Raspberry Pi Zero W
* 2 Micro SD cards
* Wires for UART communication
  + Can be any wires, just needs to connect to GPIO pins on both Pis
* Keyboard with USB A connection
* Multimeter
  + Must have voltage/current probes and thermal couple.
* USB A to micro-USB converter
* 3 EV2U-RMR2-MMC1-C1 cameras
* 3 wires for the cameras
  + The cameras have a JST SH 5 pin cord, and they connect to the Raspberry Pi 5 via USB A
* 3 VISSPL projectors
* 2 micro-HDMI to HDMI cables or converters
* 1 mini-HDMI to HDMI cable or converter
* 3 Wall Wart power supplies for the VISSPL projectors
  + Must output at least 21 V/2.3 A (43 W)

Optional (Not required for testing the system but can help make it easier):

* USB hub with USB A output
* USB Mouse

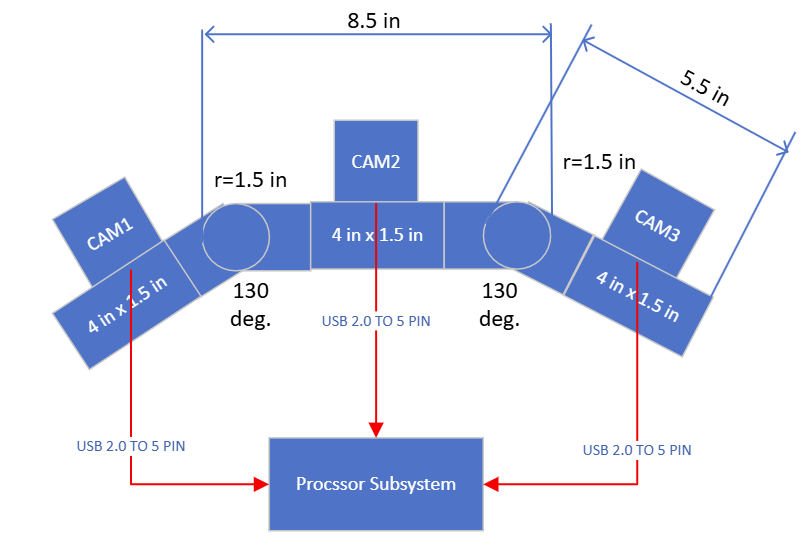


Figure 11: Input Subsystem Position Setup

Table 22: RTCP Setup

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Setup test equipment according to Figure 2. | N/A | N/A | N/A |
| 2 | Setup the camera rig according to Figure 11. | N/A | N/A | N/A |
| 3 | Setup the Output subsystem according to Figure 9. | N/A | N/A | N/A |

Table 23: RTCP System Power Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Multiply the projector power consumption by 3 to get the Output subsystem power consumption. | ≤150 W | 129 W | Pass |
| 2 | Record the power consumption of the Processor subsystem. | ≤50 W | 5.555 W | Pass |
| 3 | Add the Processor and Output powers together to get the full system’s maximum power consumption.\* | ≤200 W | 134.555 W | Pass |
| \*Only the Processor and Output subsystems will draw power from external sources, so the system power consumption only relies on those. Since the measured power consumption from the subsystems in the past has either been the maximum power consumption or a value that won’t change, the maximum power consumption of the whole system can be calculated using those previously measured values. | | | | |

Table 24: Processor Operation Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Run a script on the Raspberry Pi 5 that sends a message to the Raspberry Pi Zero W. | N/A | N/A | N/A |
| 2 | Check if the Zero W received the message. | Message received | Message received | Pass |
| 3 | Set pin GPIO17 on the Raspberry Pi Zero W high. | N/A | N/A | N/A |
| 4 | Read the output of the GPIO pin with a multimeter. | 3.3 V | 3.3 V | Pass |
| 5 | Set pin GPIO24 on the Raspberry Pi 5 high. | N/A | N/A | N/A |
| 6 | Read the output of the GPIO pin with a multimeter. | 3.3 V | 3.3 V | Pass |

Table 25: Processor and Output Subsystems Compatibility Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Send data for a still image to the Zero W from the Pi 5. | Image received | Didn’t receive the image | Fail |
| 2 | Output the still image to the projectors. | Image displayed | N/A | Fail |

Table 26: RTCP System Operation Verification Plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Step | Description of Step | Expected Value | Measured Value | Pass or Fail |
| 1 | Run the output executable with CAM3 argument. | N/A | N/A | N/A |
| 2 | Measure the Frames Per Second (FPS) of each camera window.\* | 30 fps | 30, 25, 30 | Pass, Fail, Pass |
| 3 | Measure the resolution of each camera window.\*\* | Is the frame outputting? | Yes | Pass |
| 4 | Press q to quit the CAM3 version of the executable. | N/A | N/A | N/A |
| 5 | Run the output executable with SURF argument. | N/A | N/A | N/A |
| 6 | Measure the latency of the system.\*\*\* | <20 ms | N/A | N/A |
| 7 | Check for any deadzone.\*\*\*\* | No deadzone | No deadzone | Pass |
| Steps 1-3 are to confirm that OpenCV uses 1080p @ 60 FPS inputs for the stitching algorithm.  \*The following link can be used on the Raspberry Pi 5 to measure the FPS. <https://www.testufo.com/frameskipping>  \*\*Software controls the frame’s resolution. If the resolution is under the set resolution, then it will not display anything inside the window frame and an error message in the terminal.  \*\*\*Purchasing proper testing equipment will put the project out of budget. This test can be simplified to a qualitative noticeable/unnoticeable (Pass/Fail). In industry, this would be a necessary test to pass.  \*\*\*\*The deadzone is the area between cameras that are removed by the stitching algorithm. If there is a deadzone measure the distance of the deadzone at 1 foot away from the center camera. | | | | |