Optical Communications Low-Cost Payload
System Design Document
Version 2.4
03/06/2025

Document Control

Distribution List

The following list of people will receive a copy of this document every time a new version of this document becomes available:

Teaching assistants:

Clay Pate

Luke Newcomb

Walter Hernandez

Jose Alejandro Gonzalez Nunez

Rajagopal Sugumar

Customer(s):

Eduardo Rojas

Project team members:

Korey Kelley

Alexandria Colebank

Jessica Sammons

Mosely Tector

Brian Barker

Change Summary

The following table details changes made between versions of this document:

Version	Date	Modifier	Description
1.0	10/31/2024	Brian Barker	Initial document creation.
1.1	11/14/2024	Jessica Sammons	Section 3, Section 6 first drafts.
1.2	11/18/2024	Jessica Sammons	Section 5 first drafts.
1.3	11/21/2024	Brian Barker	Formatting, Sections 1 and 2 corrections.
1.4	11/21/2024	Jessica Sammons	Completed sections 1.2, 1.4, and 2.2. Updated images and figure numbers.
2.0	02/06/2025	Lexi Colebank	Added all comments from last revisions into the whole document.
2.1	02/06/2025	Jessica Sammons	Updated Section 1.1 Purpose and Scope according to comments. Rewrote Section 1.2.3 Future Contingencies so that it made sense. Added specific section numbers to Section 1.3 Document Organization. Added document

			descriptions to each of the documents to Section 1.4 Project References.
2.3	03/04/2025	Lexi Colebank	Finished Section 6, adding new paragraphs related to safety protocols. Updated Section 4 and 3 with new information regarding the system and added diagrams.
2.3	03/04/2025	Brian Barker	Updated font consistently throughout the document for better cohesion. Updated figure numbers throughout the document.
2.4	03/06/2025	Brian Barker	Repositioned Figure 7 in Section 4.2.1 to the top of the text. Reformatted the table of contents. Fixed Section 5 and Section 6 incorrect headings.
2.4	03/06/2025	Jessica Sammons	Created new Software Architecture Diagram to replace Figure 3. peer reviewed all figures and formatted for final submission
2.4	03/06/2025	Lexi Colebank	Created new Hardware Diagram High Level to replace Figure 2 and Figure 12.
2.4	03/06/2025	Brian Barker	Final Peer Review completed. Implemented diagram comments from TA and corrected backgrounds for Figures 2, 3, 13. Corrected figure spacing throughout the document for readability.

Table of Contents

1. Introduction

1.1. Purpose and Scope

The purpose of this SDD is to describe the Low-Cost Optical Communication Payload system and its subsystems of Commercial-Off-The-Shelf (COTS) parts including the Nvidia Jetson Nano Developer Kit, the Nvidia Jetson TX2 Developer kit, and the Laser Driver being used to accomplish optical communication. The system will achieve the customer goal of a 1 Mbps transmission speed using laser diodes to transmit and receive data across a 1-foot range. The project received at the start of development by the previous senior design team was guided by Dr. Rojas. The previous team consisted of Troy Clifford, Andrew Marinello, Sarah Shiffer, Daniel Unger, Max Wilson, and Ashley Young. The previous senior design team achieved a transmission rate of 55 bps using the Jetson Nano and Jetson TX2 Developer kits. The legacy team (the previous senior design team) had more budget and time constraints than us, so they were not able to achieve 1 Mbps transmission rates. Upon researching the components, the team has decided to implement a new laser driver to increase the data rates. Our current scope includes transmission of greater than 55 bps.

A low-cost optical communication payload system is beneficial for allowing data transfers that are greater than 55 bps. We are not able to reach 1 Gbps with the equipment we have been provided, so our project scope has been decreased to the scope of 1 Mbps. Through optical communication, there is an opportunity for loss-correction algorithms and data redundancy that cannot be used in Radio Frequency (RF) transmissions alone. This SDD will describe the system in its entirety as well as each subsystem being used and implemented into the design.

1.2. Project Executive Summary

Laser communication has a lower power consumption, uses lighter antennas, has higher data rates, is more secure, and has a wide range of bandwidths when compared to microwave systems. This project aims to modify the current optical communication and increase the transmission speed to 1 Gbps over a 1-foot range. The project improves upon the previous iteration. Each component has been retested or switched to commercial-off the shelf (COTS) products that increased the efficiency. The software was revised for a more robust data transmission.

1.2.1. System Overview

The Low-Cost Optical Communication Payload system uses laser-based technology to perform high-speed data transmission. The system is designed for compact environments such as small satellites and aims to achieve transmission rates of 1 Mbps at a range of 1 foot. This system has advantages over radio-frequency systems including higher data rate transmission, reduced power consumption, and enhanced security.

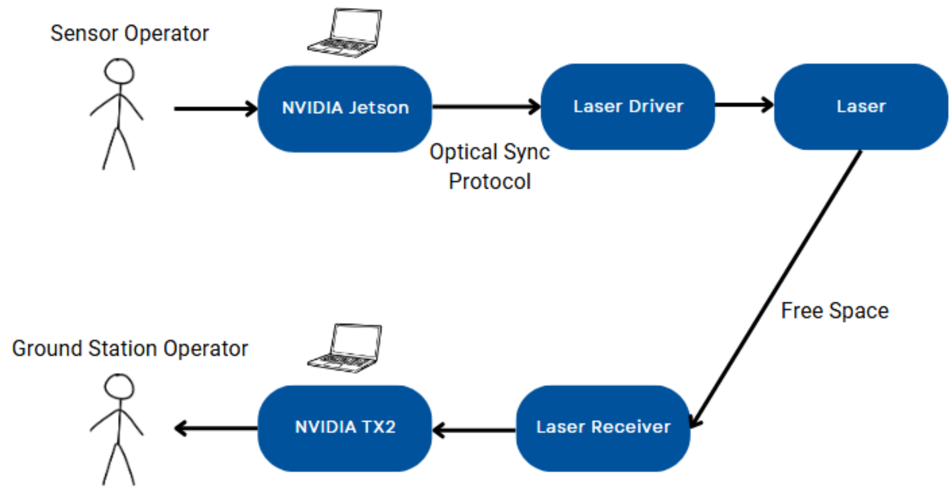

Figure 1. Graphical Data Flow Diagram

1.2.2. Design Constraints

The Optical Communication System (OCS) has constraints in budget, size, power, and wavelength. The system has constraints to be a low-cost payload with a budget of $1,000 to $3,000. Some of the electrical equipment needed to reach 1 Gbps can range from $1,000 to $5,000. The OCS needs to be lightweight and compact to fit on a satellite to make more room for other equipment. The size of the payload needs to be within 10x10x10 cm and be capable of transmission over a 1-foot range.

1.2.3. Future Contingencies

Challenges that may arise during development of the Optical Communication System include software performance, signal reception with distance constraints, hardware compatibility, and data transmission limitations.

First, the initial prototype of the system was written in Python, which was changed to C++ computer language in order to increase performance of real-time processing. Next, the signal reception depends on the distance between the laser driver and the photo sensor device. In order to reduce human error of the distance calibration, the team is implementing a distance calibration tool which is a 3D printed track to improve accuracy and reliability. Third, the hardware compatibility of the Jetson Nano and Jetson TX2 with the computer monitors relies on an HDMI cable connection. Additionally, Jetson Nano and Jetson TX2 require specific software installations that may not be compatible with all machines. The team has ensured that the proper cables and computers are available for the system. Finally, the current laser driver is unable to reach the target data rate of 1 Gbps, so a more capable laser driver will need to be designed or sourced in order to meet this requirement. The team has updated the requirement of transmission speed of 1 Gbps to 1 Mbps due to the laser driver limitations.

1.3. Document Organization

The organization of the Systems Design Document is as follows: First, the main parts of optical communication are introduced in Section 1. Then, there is a detailed discussion of the overall system architecture in Section 2. Next is a breakdown of the human-machine interface in Section 3, followed by a detailed design of the system in Section 4, including both hardware and software aspects. Succeeding this is an examination of the system's external interfaces in Section 5. Finally, there is an evaluation of the system integrity controls.

1.4. Project References

The Software User Manual.docx is a document overview of the Software written and provided by the legacy Optical Communication System team. This document includes information pertaining to the use of the Python Imaging library, the GPIO pins, and Binary File Format.

The Hardware User Manual.docx document describes the original hardware equipment and hardware set up of the legacy Optical Communication System provided to us by the previous senior design team. This document describes in detail the photo detector, laser driver, Jetson Nano, and Jetson TX2 used in the system.

The CS165CU-Manual.pdf describes the specifications and setup of the photodetector used in the system.

The PM400-Manual.pdf describes the specifications and setup of the handheld optical power and energy meter used to measure power and energy in the system for testing purposes.

The JetsonNano_DataSheet_DS09366001v1.1.pdf describes the specifications and setup of the Jetson Nano device being used to transmit data in the system.

The LD15CHA Datasheet.pdf describes the specifications, setup, and use of the laser driver used in the system.

The LDS9-SpecSheet.pdf describes the 9V linear power supply specifications and use in relation to the laser diode component.

1.5. Definitions, Acronyms, and Abbreviations

1.5.1. Definitions

This section lists terms used in this document and their associated definitions.

Table 1: Definitions

Term	Definition

1.5.2. Acronyms

This section lists the acronyms used in this document and their associated definitions.

Table 2: Acronyms

Term	Definition
COTS	Commercial-Off-the-Shelf
Gbps	Gigabits per second
Mbps	Megabits per second
nm	Nanometers
mW	Milliwatts
OOK	On-Off Keying

OCS	Optical Communication System
UART	Universal Asynchronous Receiver/Transmitter
GPIO	General Purpose Input/Output
SPI	Serial Peripheral Interface
OBC	On-Board Carrier

1.5.3. Abbreviations

This section lists the abbreviations used in this document and their associated definitions.

Table 3: Abbreviations

Term	Definition
RX	Receive
TX	Transmit

2. System Architecture

2.1. System Hardware Architecture

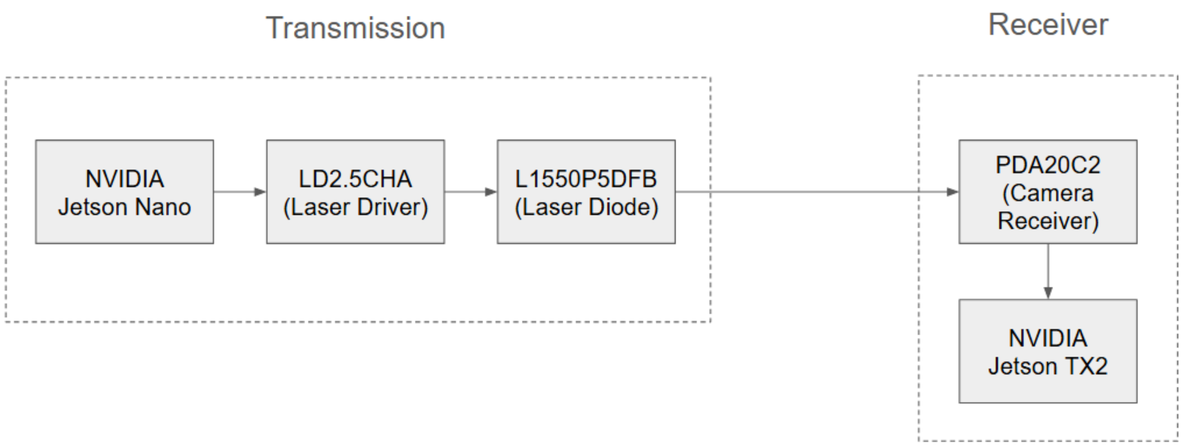

Figure 2: High-level Hardware Architecture Diagram

The hardware architecture for this optical free-space communication system consists of five primary components, each playing a crucial role in the transmission, reception, and processing of data. These components include the Jetson Nano, laser driver, laser diode, camera-based optical receiver, and Jetson TX2. Figure 2 provides a high-level block diagram illustrating the interconnections between these devices.

The Jetson Nano serves as the primary control unit responsible for generating and encoding binary data to be transmitted. It processes digital signals and modulates the data using On-Off Keying (OOK) before passing it to the laser driver. The Jetson Nano operates at 3.3V logic levels and interfaces with the laser driver via a UART signaling mechanism.

The laser driver acts as an intermediary between the Jetson Nano and the laser diode, ensuring proper current and voltage regulation for stable optical output. It converts the digital modulation signals from the Jetson Nano into precise electrical pulses that drive the laser diode. This regulation is crucial in maintaining signal integrity, especially at high data rates. The laser driver in our configuration has a modulation limit of 1 MHz. This drives the ultimate transmission speed of our system, capped at 1 Mbps.

The laser diode is the core transmission element, converting electrical signals into optical pulses that propagate through free space. The laser operates at 1550 nm with a power output of 5 mW and a modulation capability of up to 2.5 Gbps, ensuring high-speed data transmission with minimal divergence. Since the system relies on free-space optical communication, factors such as beam alignment, atmospheric interference, and optical power levels must be considered to maintain link reliability.

On the receiving end, a camera-based optical receiver (PDA20C2) captures the transmitted laser pulses and converts them back into an electrical signal. The receiver's design involves high-speed photodetection and signal processing to recover the original binary data.

The Jetson TX2 functions as the data processing unit on the receiver side. It decodes the demodulated signal from the camera, reconstructing the original binary stream. The TX2 is also responsible for error correction and further processing of the received data, enabling real-time evaluation of transmission performance.

Together, these components form a complete optical free-space communication system, enabling high-speed, low-latency data transfer between two computing platforms. This architecture provides a foundation for our wireless free-space system.

2.2. System Software Architecture

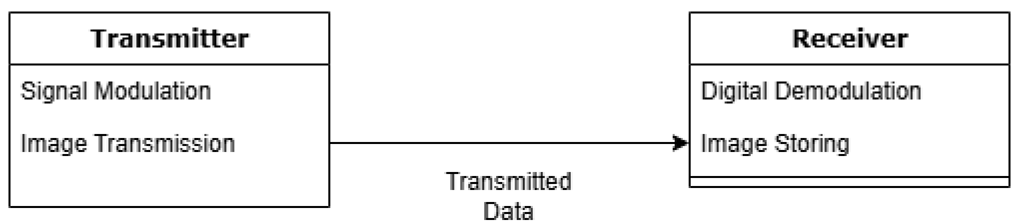

Figure 3. Software Architecture

The software architecture consists of transmission software and reception software. The transmission software takes in a picture file from a directory on the transmitting machine and loads its bytes into memory. The machine then sends the image name and size to a UART-based serial device. Finally, the bytes are sent to the same serial device using one system call. The reception software, on the contrary, begins by waiting for the transmitting machine to send the image's name and size through its own UART-based serial device. The software then reads the bytes serially from the same serial device in chunks sized 4095 bytes. The chunks are stored contiguously in memory. Finally, when the device has read the size of the file, it writes the bytes into an image file by the same name to the receiving machine's desktop.

2.3. Internal Communications Architecture

2.3.1. Device Interconnections

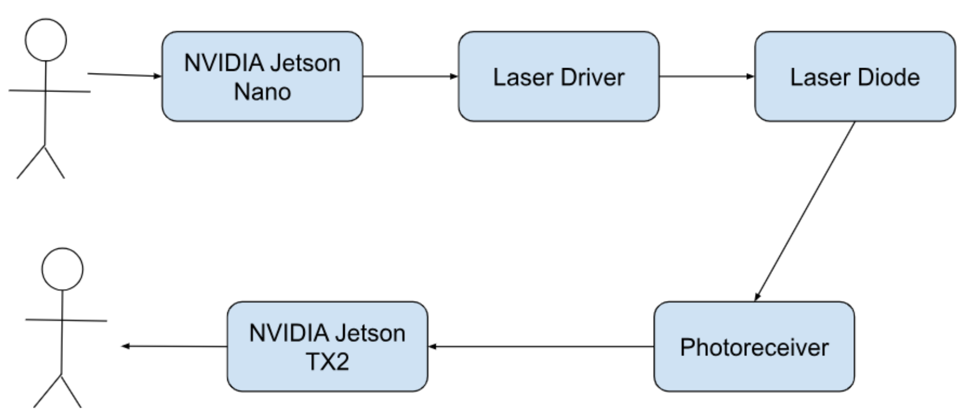

Figure 4: Connections Between High Level System Components

2.3.2. Communication Interfaces

The laser driver interfaces with the laser diode to transmit data. The Photodiode Detector receives the signal sent from the laser and converts the data to electrical signals for processing.

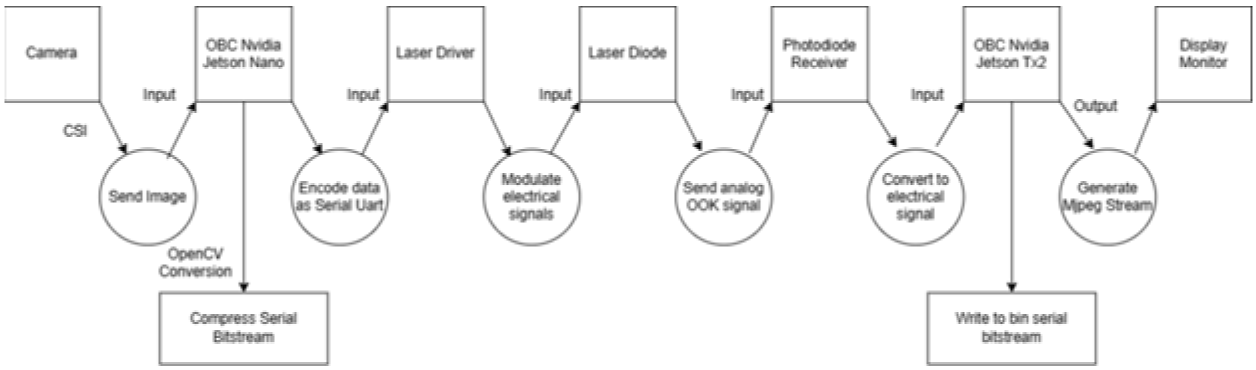

Figure 5: DFD Level 1 Diagram

3. Human-Machine Interface

3.1. Beginning Transmission

There must be a user who runs the system's software on each machine. The reception software on the reception machine must be run before the transmission software begins. Both machines follow the same process of requiring each user to open a Terminal window and run the command "cd ~/Desktop/OPTICS Fall 2024". The users must then run "cd transmit" or "cd receive" for the transmission and reception machines, respectively. To start the system, the user must run the command "sudo ./receive.sh" on the reception machine. Then, the user must run "sudo ./transmit.sh" on the transmission machine. When run in this order, these commands will send a default file stored in the "transmit" directory of the transmitting machine to the "receive" directory of the reception machine.

The bash command "transmit.sh" has options to customize the system's operation. Running "sudo transmit.sh -h" or "sudo transmit.sh --help" will provide the user with a step-by-step guide on how to send an image between the machines, as well as which flags "transmit.sh" has implemented. Running "sudo transmit.sh --from", followed by a directory path, specifies where to get the image file from on the transmitting machine. Specifying the flag "--image" followed by an image file name of the correct type specifies the name of the image to transmit. With no flags specified, the image sent will be a default image stored in the folder containing "transmit.sh".

The bash command "receive.sh" also has options to customize the system's operation. The "-h" or "--help" flag gives the user a step-by-step guide on how to run the machine, as well as what option flags are available to "receive.sh". The option "--to" followed by a directory path specifies where in the receiving machine's file system to save the image. The option "--image" followed by an image file name of the correct type specifies the name of the image to save the transmitted data into. With no flags specified, the image will be stored by the same name as the file on the transmitting machine in the folder containing "receive.sh".

3.2. Viewing the Image

With no flags specified when running the transmitting or receiving software, the image can be seen in "~/OPTICS Fall 2024/transmit" on the transmitting machine, and in "~/OPTICS Fall 2024/receive" on the receiving machine. With "--to" specified for "receive.sh", the image can be viewed in the directory corresponding to "--to"; the same goes for the transmitting machine when "--from" is specified. The image files can be opened using the file explorer GUI that comes with Ubuntu. The images are visible to the user through monitors hooked up to both machines.

4. Detailed Design

4.1. Hardware Detailed Design

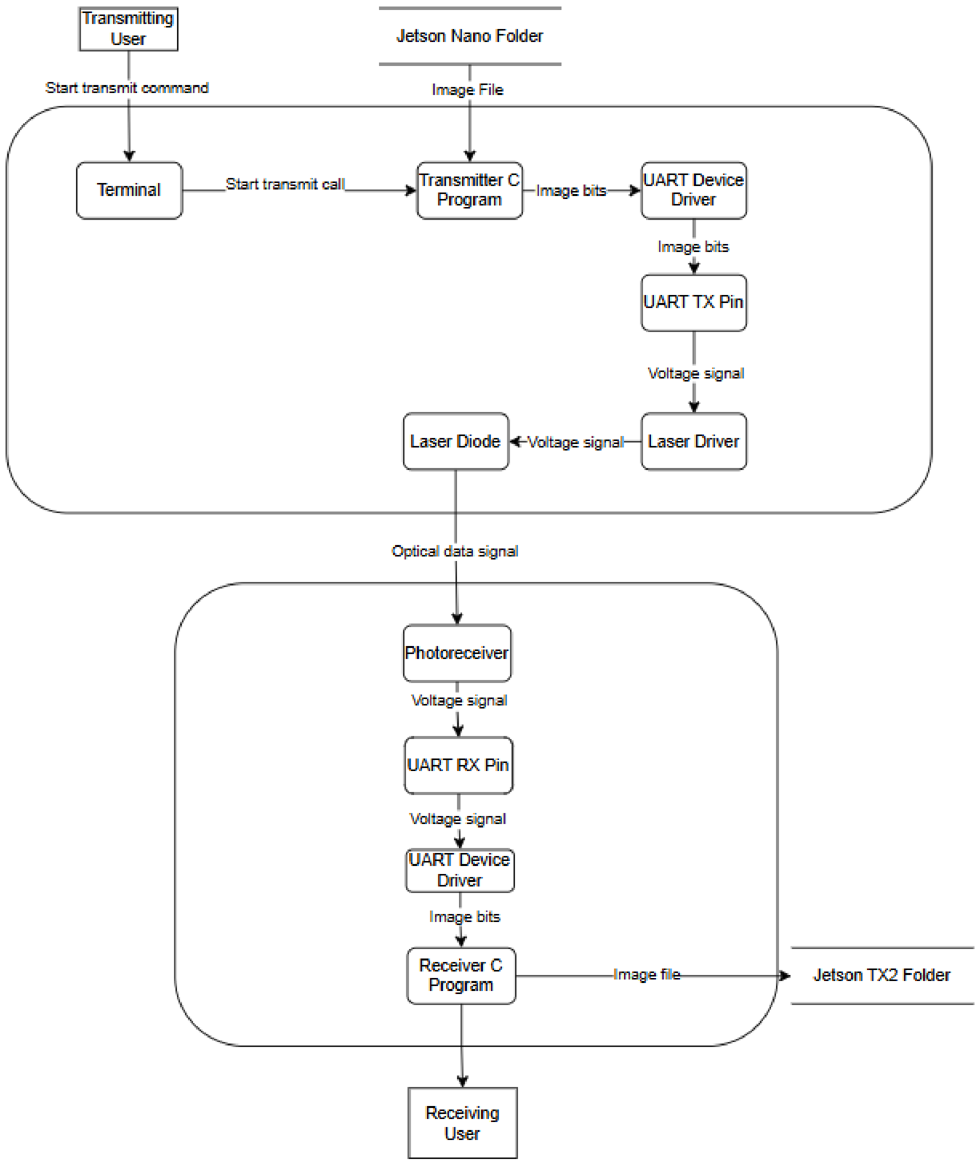

Figure 6: Detailed Hardware Layout

The optical communication system (OCS) can be summed up as a two-section system comprised of a transmitter (TX) and a receiver (RX). The final project will include all the components listed in Figure 2. The TX consists of a Jetson Nano, laser driver, and laser diode. The RX consists of a camera and a Jetson TX2. The TX uses the onboard computer NVIDIA Jetson Nano to modulate the video into an optical carrier signal that is fed to the laser driver to push the laser to emit light pulses with the characteristics of the modulated video signal. The laser is transmitted through free space and captured in the RX. The received optical signal is converted into an electrical signal. The on-board computer demodulates the signal into the original digital data and displays it on a video monitor.

4.1.1 Laser Driver Setup

The laser driver voltage must be manually adjusted by the operator based on the specifications of the laser diode in use. The laser driver operates in two modes: constant current mode and constant power mode. Constant current mode offers greater electrical stability, protecting the diode from damage, though it is less precise with temperature variations. In contrast, constant power mode ensures optical stability and delivers a more accurate optical signal, but it may shorten the diode's lifespan due to continuous current adjustments to maintain consistent power. Given that the team has only one laser diode, constant current mode is selected to minimize the risk of damage. The laser diode has an absolute maximum current limit of 120 mA, with a forward operating current range of 20 mA to 40 mA. These current values are used in the equations shown in Figure 3 to determine the required voltages for the trimpots on the laser driver. According to the datasheet, the transfer function is 1 A, leading to the setpoint trimpot being adjusted to 40 mV and the limit trimpot to 79.28 mV. While the limit trimpot could be set as high as 120 mV, it was configured to a lower value since the team does not intend to operate the laser driver at or near the diode's absolute maximum voltage.

4.1.2 Hardware Setup

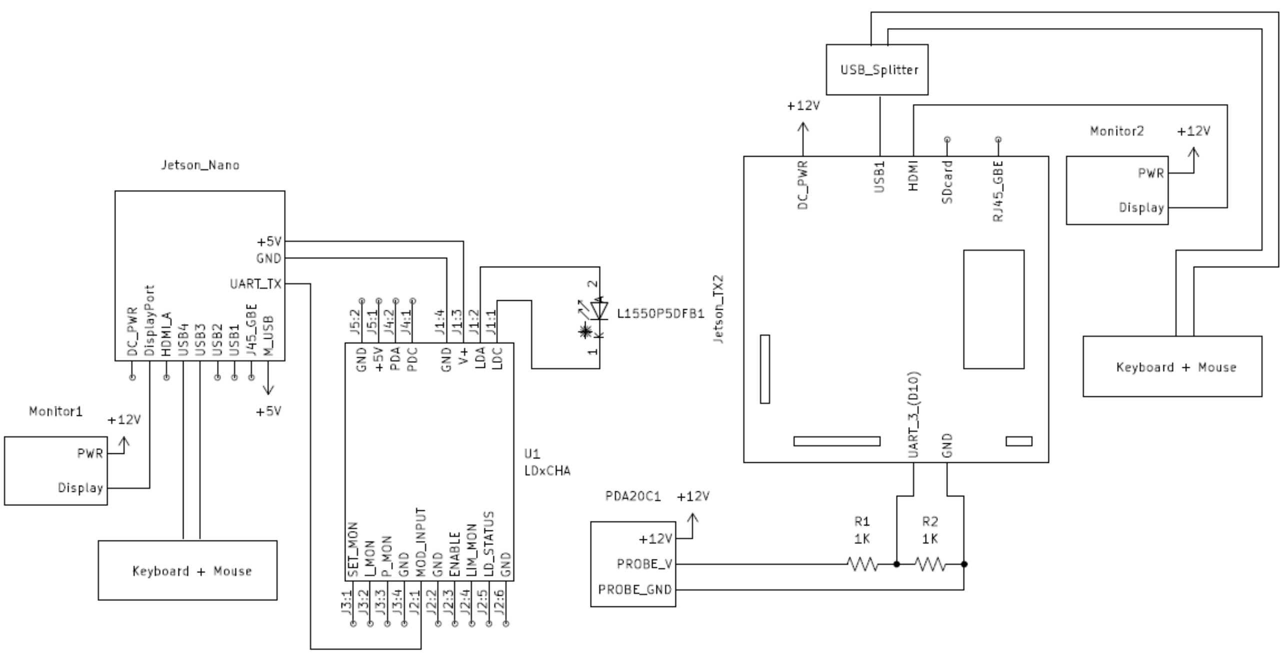

Figure 7: Wiring Diagram

After completing the laser driver setup, the laser driver is connected to the Jetson Nano and the laser diode on a breadboard, with the wiring configuration detailed in the wiring diagram below, as shown in Figure 4 and Figure 5. The Jetson Nano provides 5 V of power to the laser driver by connecting its 5 V pin and GND pin to the V+ pin and GND pin on the laser driver. The laser driver interfaces with the laser diode via the photodiode anode (PDA), photodiode cathode (PDC), laser diode anode (LDA), and laser diode cathode (LDC) pins, with PDA linked to pin 2, PDC to pin 4, LDA to pin 3, and LDC to pin 1. The camera receiver (PDA20C2) is powered by a 120 V power supply and connected to the Jetson TX2 through a cable tied to ground and a UART pin. When activated, the laser transmits binary data—0 and 1—as low and high voltage levels, respectively, per the programmed implementation. The receiver captures the analog signal, demodulates it, and sends it as a digital signal to the Jetson TX2. The full system design is illustrated in the wiring diagram below, as depicted in Figure 6 and Figure 7.

OUTPUT CURRENT LIMIT:
Refer to the datasheet for your laser to determine the maximum forward current, and calculate the current limit monitor voltage ($V_{ILIMMON}$) using this equation and the transfer function found in **Table 2 on page 5**:

$$V_{ILIMMON} = I_{LIMIT} / \text{Transfer Function}$$

OUTPUT CURRENT SETPOINT:
In Constant Current Mode, determine your desired current level and calculate the current setpoint monitor voltage ($V_{ISETMON}$) using the transfer function found in **Table 2**:

$$V_{ISETMON} = I_{SET} / \text{Transfer Function}$$

Note: In Constant Power Mode, the output current setpoint is dependent on the photodiode response.

Figure 8: Equations to Calculate Trimpot Voltages for Laser Driver

The camera receiver (PDA20C2) is powered by a 120 V power supply and is connected to the Jetson TX2 via a cable that ties to the ground and a UART pin. When the laser is activated, it transmits binary data—0 and 1—as low and high voltage levels, respectively, according to the programmed implementation. The receiver captures the analog signal, demodulates it, and forwards it as a digital signal to the Jetson TX2. The complete system design is illustrated in Figure 6 and Figure 7.

After completing the laser driver setup, the laser driver is integrated with the Jetson Nano and the laser diode on a breadboard. The wiring configuration is detailed in the wiring diagram below, as shown in Figure 4 and Figure 5. The Jetson Nano supplies the laser driver with 5 V of power by connecting its 5 V pin and GND pin to the V+ pin and GND pin on the laser driver. The laser driver interfaces with the laser diode through the photodiode anode (PDA), photodiode cathode (PDC), laser diode anode (LDA), and laser diode cathode (LDC) pins, with PDA linked to pin 2, PDC to pin 4, LDA to pin 3, and LDC to pin 1.

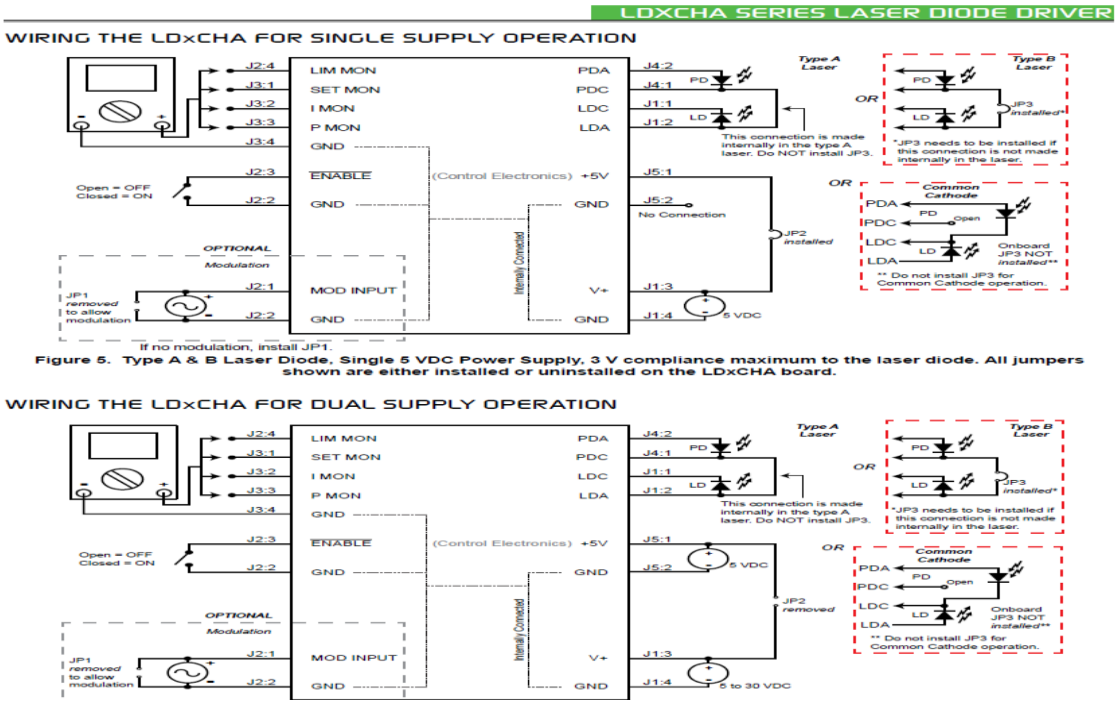

Figure 9: Laser Driver Pin Diagram

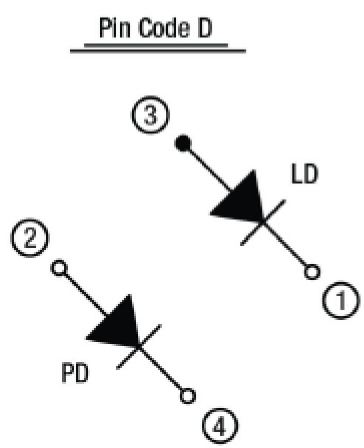

Figure 10: Laser Diode Pin Diagram

4.2. Software Detailed Design

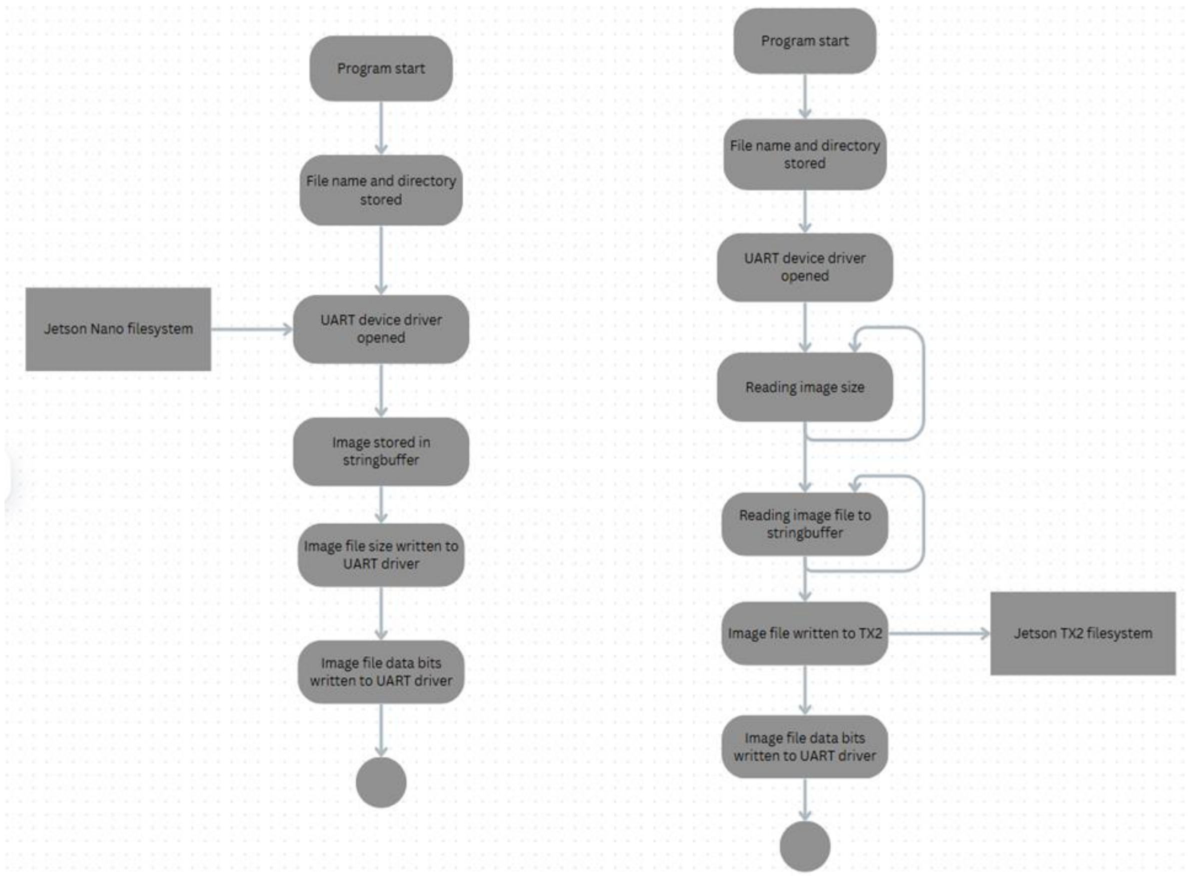

Figure 11: Left: Transmission Software Detailed Design Diagram, Right: Reception Software Detailed Design Diagram

The OCS software is made up of two subsystems: the transmitting software and the receiving software. The programs are kept on the Jetson Nano and Jetson TX2 respectively. Each program contains only the main function; no other classes or functions are user written. The program takes auxiliary functions from the library "simple_uart" written by AndreRenaud.

The transmitting main function opens and configures the "ttyS0" device driver for UART use using the function "simple_uart_open" from the "simple_uart" library. It then opens and reads the selected image file into a string buffer using the built-in Ubuntu C-standard library. The program then writes to the UART device driver the image file size, the file name, and the image file (image file sent with one function call) using three separate calls to the function "simple_uart_write".

The receiving main function opens and configures the "ttyTHS2" device driver for UART use using the function "simple_uart_open" from the "simple_uart" library. The program then halts program functionality until the UART device driver has received data. This is accomplished by starting a while loop with the condition statement calling the function "simple_uart_has_data", which returns the amount of data stored in the UART device driver. When "simple_uart_has_data" is non-zero, the program continues. The program then reads in the image size and name by calling "simple_uart_read" twice in succession. The program follows by reading the whole image in chunks with a maximum size of 4095 bytes. The program uses the same while loop as before, waiting until the device driver has information before it calls the read function. The data gets read into a string buffer in the main function. The image file is then written to an image file of the read-in name using the built-in Ubuntu C-standard library function "write".

4.3. Internal Communications Detailed Design

The Laser Driver and OBC input data into the system. The data sent into the system includes the signal, voltage (power), and the minimized file data. Board 2 receives data from the system that allows the maximized file to be returned back into the system. The receiver and screen only take in data from the system. The receiver takes in the data from the laser after the system modifies it and the screen is given the display data.

The internal communication for the system takes place in the software for both the receiver and transmitters. The system interfaces between laser hardware and the Jetson boards using UART communication. The transmitting and receiving laser machinery communicates with each other through an optical data signal. The receiver acquires UART data based on the Jetson on Serial On-Off Keying (OOK) and processes the packets to generate a JPEG. This process is shown in Figure 9.

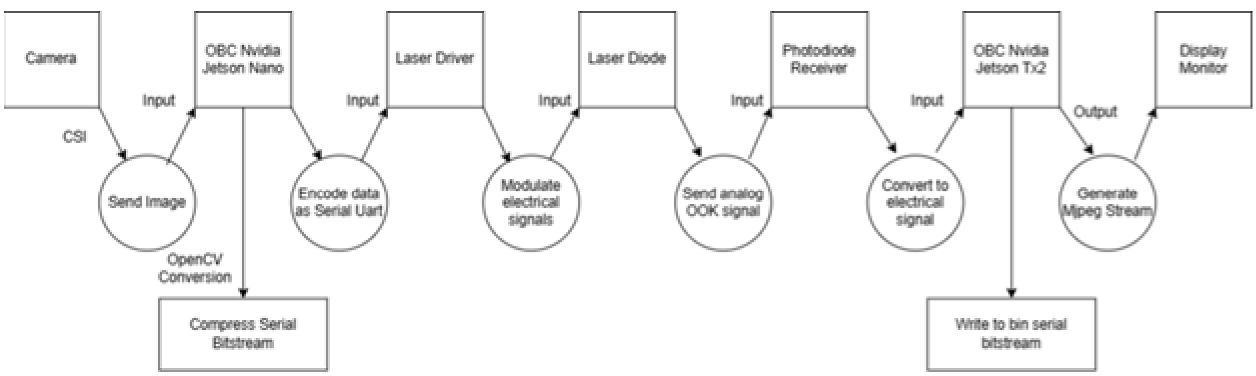

Figure 12: DFD Level 1 Diagram Covering the Overall System

Data transmitted from the Jetson Nano is formatted into binary packets suitable for optical transmission. The Jetson Nano then sends the data through Serial UART to the laser driver. The settings for the UART are 8N1 data transmission rate of 1 Mbps. The laser driver modulates the electrical signal and supplies current to the laser diode. The laser diode sends an analog OOK signal which is the input for the RX system. The Raspberry Pi camera captures the laser transmission, which is then reformatted by the Jetson TX2. The Photodiode receiver detects the OOK signals and converts them back into electrical signals. The electrical signals converted by the photodiode are the input to the Jetson TX2 which writes to the bin serial bitstream. The Jetson TX2 generates the JPEG image, displaying the image on the monitor connected to the Jetson TX2.

5. External Interfaces

5.1. Hardware Interface Architecture

The system contains an Nvidia Jetson Nano which transmits data through a laser and laser driver optically.

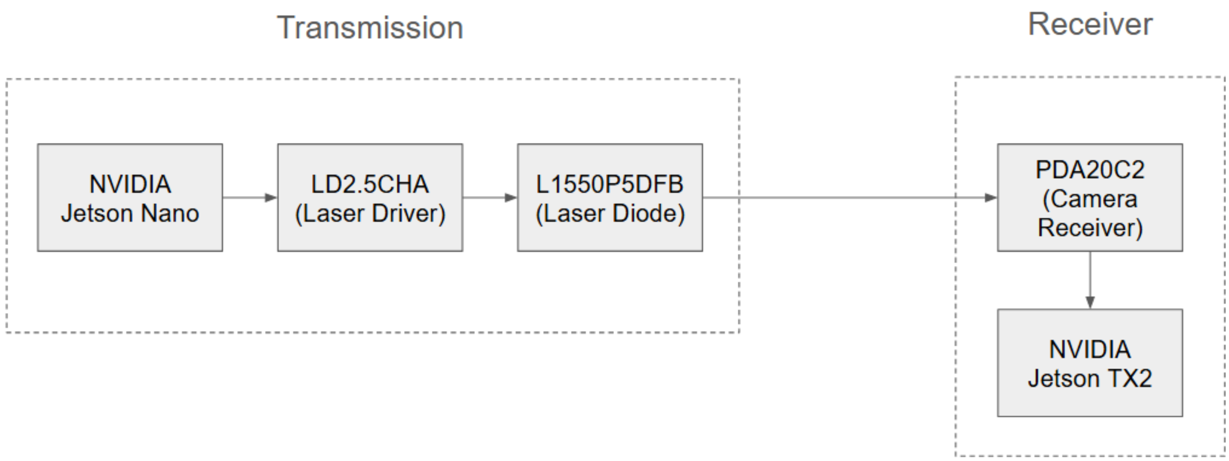

Figure 13: Hardware Interface Diagram

Users interface with the Jetson Nano and the Jetson TX2 using a keyboard, a mouse, and a monitor.

5.2. Hardware Interface Detailed Design

5.2.1. Data Formatting Requirements

Data transmitted from the Jetson Nano is formatted into binary packets suitable for optical transmission. The Jetson Nano then sends the data through serial UART transmission to the laser driver. The laser driver modulates the electrical signal and supplies current to the laser diode. The laser diode sends an analog OOK signal which is the input for the RX system. The photodiode receiver detects the OOK signals and converts them back into electrical signals. The electrical signals converted by the photodiode are the input to the Jetson TX2 which writes to the bin serial bitstream. The Jetson TX2 generates the JPEG image, displaying the image on the monitor connected to the Jetson TX2.

5.3. Software Interface Architecture

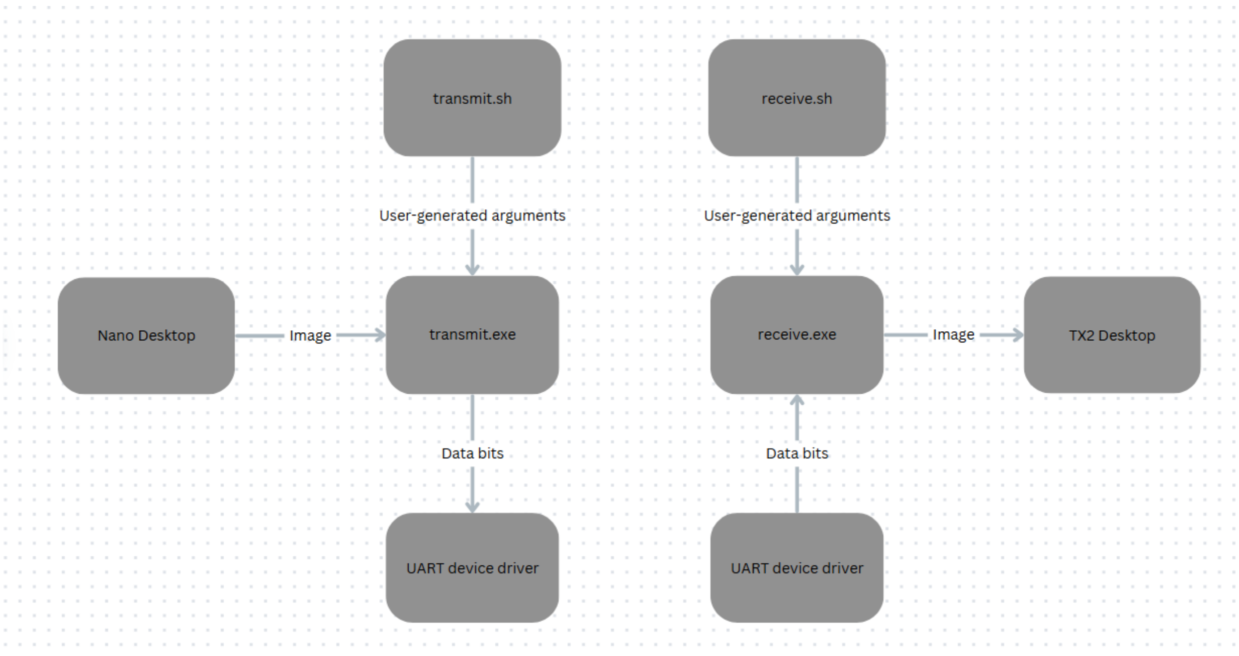

Figure 15: Software Interface Architecture Diagram

5.4. Software Interface Detailed Design

5.4.1 Sending and Receiving Data

The transmit software opens the UART device driver on the Jetson Nano using the Ubuntu C-standard library “open” command on the device driver “/dev/ttyS0”. It sends data to the device driver using the Ubuntu C-standard “write” function. The entire file is sent to the driver at once. The image to be sent is pulled from the file system by first opening the image and then reading it into a C++ string buffer. The Ubuntu C-standard library functions “open” and “read” are used to create a file pointer to the image object and to read in its data. The directory and name of the file are decided by user arguments passed to the main function when transmit.sh is called.

The reception software opens the UART device driver on the Jetson TX2 using the Ubuntu C-standard library “open” command on the device driver “/dev/ttyTHS2”. It receives data from the device driver using the Ubuntu C-standard “write” function. The entire file is read by the driver in chunks with a max size of 4095 bytes. The image is written to the TX2’s file system by first opening the image and then writing the raw image data from a string buffer to the file system. The Ubuntu C-standard library functions “open” and “read” are used to create a file pointer to the image object and to read in its data. The directory and name of the file are decided by user arguments passed to the main function when receive.sh is called.

6. System Integrity Controls

To protect our communication system from data loss, misuse, unauthorized access, and modifications, we will implement several security measures. Data validation checks will ensure that all received data is processed accurately and completely before being used by the payload. Additionally, backups of critical configuration files will be maintained to allow for quick recovery in case of corruption or accidental deletion.

To ensure only authorized users can access and operate critical system functions, strict access controls will be implemented to secure the room. These measures will prevent unauthorized individuals from interfering with system operations or improperly activating hardware. For example, in the future there will be safeguards in place to prevent unauthorized usage of high-risk components, such as lasers, to eliminate potential hazards.

Furthermore, all software and hardware changes will require approval and verification from multiple team members before implementation. These measures will help maintain system security, prevent misuse, and ensure data integrity moving forward in system development.