LaserDrop

Authors: Anju Ito, Roger Lacson, and KJ Newman

Department of Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—This project aims to create a device capable of transmitting and receiving data at a minimum speed of 4 Mbps using laser optics. This project will be implemented by using two laser lines, one green and one infrared, to send serial digital signals to their receivers. This device hopes to add security and discreteness to conventional communication methods; laser optics rids the device of RF signals that poses a risk of being listened in on while still allowing wireless communication that is designed to work at a distance of 1m.**

***Index Terms*—FPGA, Free-Space Optical Communication, Laser, Security**

I. Introduction

SECURITY is a huge concern in today's society, especially with regards to data. Information is constantly being transferred everywhere, with people inadvertently creating RF signals in their everyday actions. Any electronic communication that involves WiFi, Bluetooth, or cellular data makes information vulnerable to snooping in which other individuals listen in on their transactions and potentially steal their data. The primary means of getting around this is using a physical, wired connection. However, this often creates an inconvenience for ordinary users and may also attract attention in public settings.

These flaws are problematic, especially for undercover agents with valuable information that they need to hand off. While our project has many potential use cases including underwater uses (where RF signals do not propagate well) and military operations in RF-denied areas, our primary focus is to satisfy the need of a secure, secretive communication method for covert agents in a hostile environment with RF surveillance.

Our project aims to create a device that will allow communication through optical lasers. This project will be able to interface with any USB-C device that can run our application, transmitting data at a minimum speed of 4 Mbps, while providing security and secrecy.

First, security is provided with the lasers' extremely narrow beam focus and fast communication rates: the target size is extremely small, and its communication rates are infeasible to be captured by the comparatively slow frame rates of cameras. Physical interception is the only viable means of stealing data, which is unlikely to succeed without being immediately noticed by the users.

Second, this communication method is covert, as it does not require any physical interaction between the two users. This is important for our use case, as undercover agents should not have any direct interaction that may alert others of their ties. Our device aims to provide these people with the opportunity to communicate in public settings, such as in a park-side bench or inside public transportation without anyone suspecting anything is happening.

The goal of our product is to provide security and secrecy to assist in espionage, which is vital to our national security and defense. Traditional means of handing off data such as dead drops and encryption exist with the downside that the former leaves a physical risk of being discovered or destroyed, while the latter leaves traces of communication from its RF signals and the opportunity of decryption by malicious parties.

II. Use-Case Requirements

The use-case requirements for LaserDrop include constraints on minimum transmission range, aiming tolerance, and safety for public use.. The use-case requirements for our project are listed below.

1. *The device must be able to transmit from a distance of 1 m*

Since this device is intended to be used for espionage tactics, it must be capable of being used in public as a communication method between two "strangers." The distance requirement of 1 m allows the users to stay at a comfortable distance from each other.

2. *The allowable angular error from a 1 m distance must be between 0.5 and 2 degrees*

This allows the users slight flexibility in terms of hand shaking and aiming accuracy, while also keeping the laser focused enough so that the surrounding people cannot simultaneously listen in on the data.

3. *The device must be able to work in constant, indoor ambient light*

The device must be usable in locations with stable lighting without the need for users to move to a special environment. This ensures that it can be used in a public setting.

4. *The total laser power from the device at any point must not exceed 5 mW*

Lasers are divided into classes to define their safety ratings. Lasers between 0.5 mW and 5 mW are considered Class 3R (previously called class IIIa), which are considered to be safe [1]. These lasers are low enough power that the human blink reflex is fast enough to prevent eye damage from the laser, making it acceptable to use in public.

5. *The device must be capable of handling one to two bit errors in data*

As LaserDrop deals with free-space optical transmission, it needs to be able to operate in everyday environments. This includes shadows that could cross over the receiver, which could create a one-bit or a two-bit error at the edges.

6. *The device must support a minimum data transmission speed of 4 Mbps*

Our use case is to discreetly and quickly send files in a public location. In these situations, the users would want to spend as little time as possible communicating with each other to avoid detection. 4 Mbps of communication speed allows one to send 50 PDF pages in a second. This speed is enough that within a few seconds, a user should be able to send any reasonably-sized individual file, keeping the time that the users are in the same place at a minimum.

7. *The device must be capable of being powered with USB 3.0 power delivery mode*

To make our device not stand out, only a single wire should connect to our device, as this is the standard for the majority of consumer products. A USB interface for a device is extremely common, ensuring that our device is not immediately suspicious and is familiar to operate for non-technical users.

III. Architecture and/or Principle of Operation

The block diagram of our device's general architecture is shown in Fig. 1 below.

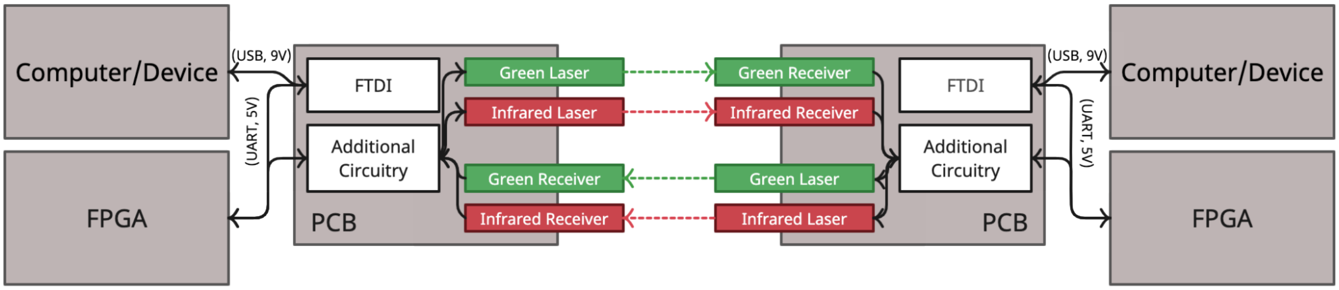


Fig. 1. System Architecture Block Diagram

The three main components of our project are the PCB, FPGA, and the user device.

The PCB will feature a USB transceiver chip that will facilitate communication between the FPGA and the user device. Furthermore, it will contain circuitry that will send digital signals from the FPGA to laser diodes for transmission, and will also convert the incoming laser signals into a digital logic. Each PCB will come with two lasers and two receivers (one green and one infrared), with each receiver only detecting their respective lasers with set wavelengths. The USB transceiver on the PCB will connect to the computer/device via a USB-C port and to the FPGA via Fast Serial protocol. While the device is connected via a USB-C, we will be using USB 2.0 protocol for data transmission.

The FPGA is responsible for controlling the lasers and processing receiver inputs. As mentioned before, it is connected to the transceiver chip and will directly control additional circuitry on the PCB board that connects to the lasers and receivers. It will have a buffer to store packets for transmission as well as re-transmission in the case of a failed transmission. The FPGA will transfer packets to and from the transceiver chip that connects to the laptop and will also signal laser responses based on input from the receivers and the user device.

Lastly, software on the user device is responsible for constructing and Hamming encoding packets and communicating with hardware components. The computer interfaces with the transceiver chip via a USB connection, establishing a virtual COM port (VCP). On the sender side, the computer initiates the data transmission. When a packet is requested, it is subsequently constructed and provided to the remaining circuitry for data transmission. On the receiver side, the packet is then received and processed before final reconstruction at the end of data transmission.

The lasers will be communicating using a custom protocol created for this project. Files will be split into smaller packets for transmission. Each packet will contain 512 bits that contain a START sequence, a message tag that increments every packet, the actual data, and Hamming encoding for error detection. If the receiver receives data with an incorrect Hamming encoding, it will request for the packet tag so the data can be resent. At the end of the transaction, both parties will send a "done" signal: the transmitter after sending its last packet, and the receiver after receiving the last packet, making sure that all its packets have the correct Hamming encoding, so that it can reconstruct the full packet at the end.

All data will be split and sent simultaneously over the 2 lasers using a UART-based protocol consisting of a start bit, 8 data bits, and a stop bit, that allows the device to differentiate between consecutive 0's versus a complete misalignment. The lasers will have three possible states of operation: an off, a low, and a high state. Using the low state to represent a logic 0 will reduce switching time drastically.

The device will draw all of its power from the user device via the USB-C port it is connected to, through the USB 3.0 Power Delivery feature. This will provide 9V3A for the board to use, which should be sufficient to power all of the PCB components as well as the FPGA.

IV. Design Requirements

A summary of the design requirements is outlined in Table 1 below:

Table 1. Design Requirements

Description	Requirement
Minimum Hardware Bitrate	5.42 Mbps
Max Rise/Fall Time	73.8 ns
Minimum Input Voltage	6.6 V
Minimum Photocurrent	5 µA
Max Bitrate Between FPGA/PCB	25 Mbps
Laser Radius at 1 m	0.87 cm to 3.5 cm

As a data transmission device, the device is concerned with transmission speeds. The total transmission speed of the lasers must be 5.42 Mbps as shown in (1), due to overhead added from UART (start and stop bit), as well as our communication protocol.

$$S_{min} = 4\,Mbps \cdot \frac{10\ bits}{8\ data} \cdot \frac{520\ bits}{480\ data} = 5.42\,Mbps \quad (1)$$

The data transmission speed puts a constraint on the rise time and fall time of the circuits, because signal integrity on the laser lines is a concern for our system. Our design requires a less than 73.8 ns for both rise and fall time in order for the square wave to be readable (we define readable as 80% of the wavelength within 2% of the intended voltage). Based on the required 5.42 Mbps bitrate, the maximum rise or fall time is calculated as shown in (2):

$$\frac{1}{\frac{5.42\times10^{6} Mbps}{2\ lasers}} \times 0.2 = 73.8\,ns \quad (2)$$

Due to the fact that the FPGA is located on a separate board, the maximum transfer speed between the two boards will be limited. As we will elaborate later, we have measured the GPIO pins from the FPGA and determined that 25 Mbps was the fastest transmission speed that the signal quality remained acceptable (Fig. 2).

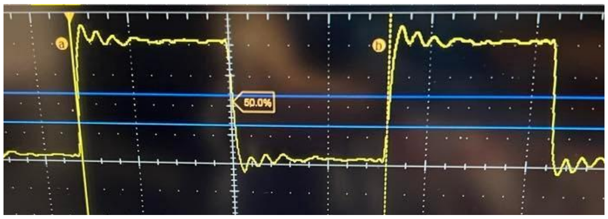


Fig. 2. 12.5 MHz signal measured on the FPGA GPIO pin

The highest voltage required by our circuits is 6.6V, which is the peak forward voltage of our green laser diode. This means our input voltage must be over 6.6V. The minimum photocurrent required from our receivers is 5 µA. This is the minimum current that the transimpedance amplifier will be able to detect where our signal size will not be too small.

We also require that the laser radius at 1 m be between 0.87 cm and 3.5 cm. This requirement comes from the desired 0.5 to 2 degree angular error from our use-case requirement (Fig. 3). This constraint ensures that the laser light source is large enough to aim but small enough to not extend beyond the receiver so that other people can also receive the signal.

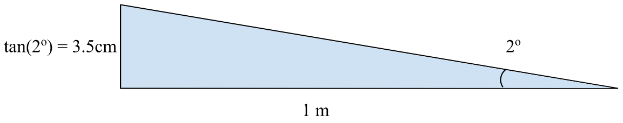


Fig. 3. Angle Error Tolerance Calculation (similar for 0.5º)

V. Design Trade Studies

A. *Lasers*

We elected to use multiple lasers in parallel to be able to transmit data faster. This adds slight complexity to the architecture in return for a greatly increased speed, which makes it easier to achieve our desired 5.42 Mbps. We elected to parallelize lasers with different colors rather than using multiple of the same color so we can more easily prevent channels from interfering with one another. This led to a need to decide on which laser colors to use, as well as how many. The primary focus is to ensure that we can easily filter and differentiate the lasers on the receiver end. The only combination of lasers that we could find that would meet our speed requirements and are far enough apart in wavelength (enough that commercial photodiodes can differentiate) was a 515 nm green laser and a 793 nm infrared laser. No combinations using three lasers could be found, largely because light detectors are generally designed to cut out either the IR spectrum as a whole or the visible light spectrum as a whole, not in between. Additionally, we do not have the budget for high-end components that have better filtering and speed characteristics. Our budget and receiver characteristics limited us to a one infrared, one green laser architecture.

Another trade related to the lasers is the modulation scheme. We considered three modulation schemes as shown in Fig. 4. We decided to use the middle scheme, which is a digital signal in which a logic 0 uses a non-zero voltage. This is because square wave modulation is much easier to implement in hardware than sinusoidal modulation, and not completely turning the laser off increases switching speed.

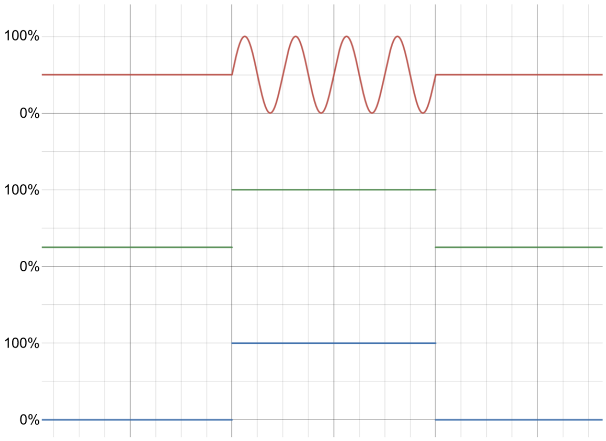


Fig. 4. Modulation Schemes

B. *Transmitter Circuit*

Since we are utilizing two lasers, each transmitter circuit needs to have a rise and fall time under 73.8 ns to ensure that data is readable for 80% of the bitlength, as shown in (2).

We also must power lasers at eye-safe levels. There are several ways to achieve this: there are commercially available ICs for time-of-flight (TOF) sensors and fiber optic transmitters, or we could create our own transmission circuit. The pros and cons of these options are listed in Table 2. We elected to use the simple transmission circuit that uses MOSFETs to switch lasers on and off. The more-precise laser control that fiber optic and TOF sensor chips offer is not needed for our application, and many of the features of these chips would need to be disabled for us to use on our PCB. Additionally, the majority of these ICs are incompatible with our lasers because of their power requirements. Using any of these chips seems to be an instance of fitting a square peg into a round hole; there are too many conflicts between these chips' intended use case and ours. Using discrete transistors will let us meet our rise and fall time requirements if we use gate driver ICs, and our output power can be adjusted using resistors as well.

TABLE 2. TRANSMITTER CIRCUIT OPTIONS

Design	Pros	Cons
MOSFET	Simple Guaranteed to work Cheap	No dynamic optical power control
Fiber Optic Transmitter	Dynamic optical power control	Expensive
		Complex
	All-in-one module	Too high power Too high minimum speed
TOF Sensor Controller	All-in-one module	No dynamic optical power control Too low voltage

C. *Receiver Circuit*

Just like the transmitters, the receivers also need to have a rise and fall time below 73.8 ns, as shown by (2). The device must also be able to receive despite a relatively large angular error of 2 degrees at 1 m. This can be accomplished using photodiodes in photovoltaic mode, photodiodes in photoconductive mode, or phototransistors. In photoconductive mode, photodiodes are reverse biased, and it leaks current that is proportional to the light it receives. In photovoltaic mode, a photodiode is shorted with a resistor in series, and a small voltage that is proportional to the incident light is generated. The trade-off between these options is shown in Table 3. We decided to use a photodiode in photoconductive mode in the end. It is the fastest option by a wide margin and its support circuits consisting mostly of a transimpedance amplifier (TIA) are not overly difficult to implement.

TABLE 3. RECEIVER OPTIONS

Design	Pros	Cons
Phototransistor	Can output at any logic-level voltage Cheap	Slow
Photodiode - Photoconductive Mode	Fast	Output is in current, not voltage Low output current (µA)
Photodiode - Photovoltaic Mode	Outputs a voltage	Output voltage is low; not a logic level Slow

Now that we have decided on using photodiodes in photoconductive mode, the next trade deals with the type of photodiode. We mainly considered four options: PN, PIN, metal-semiconductor, and avalanche photodiodes. The pros and cons of each are listed below in Table 4. The PIN photodiodes do not have any major downsides and meet our speed requirements, which made it an ideal choice for this use.

Table 4. Photodiode Options

Design	Pros	Cons
PN Photodiode	Cheap	Slowest photodiode
PIN Photodiode	Fast	
	Cheap	
Metal-Semiconductor Photodiode	Fast	Bad performance at high wavelengths
	Cheap	
Avalanche Photodiode	Fastest	Expensive
		High gain noise

The size and the number of photodiodes was another choice to make. The options we considered included using a large array of photodiodes, individual photodiodes, or using a light diffuser that will cover the individual photodiodes. Market research showed that photodiode arrays are very expensive, and light diffusers reduce optical power that the photodiodes receive, making it perform worse. Meanwhile, individual photodiodes create an extremely small target. We decided to get around this using a slightly out-of-focus laser that will create a larger beam within our given target-size constraint.

The final trade-off to make for the receiver circuit was the type of the transimpedance amplifier (TIA) to use. The choices are using a discrete TIA, a fiber optic receiver, or a TOF sensor receiver. This trade-off is shown in Table 5. We elected to use a TOF sensor receiver because it contains most of the components internally, thereby requiring less external circuit design which could create more noise with its traces and components. It also has a single-ended output designed to be connected to an FPGA's digital pins rather than an ADC. Using an ADC would greatly reduce our operating speed because the communication to and from the ADC would need to be many times faster than our actual bit rate, and our communication speed is limited by the need to cross signals between the two boards.

Table 5. Transimpedance Amplifiers

Design	Pros	Cons
Discrete Transimpedance Amplifier	Input and output at any level we want	Expensive
	Can select individual components that meet our speed requirements	Complex
		More noise due to increased traces and components
Fiber Optic Receiver	Fast	Too high input current
		Differential output needs conditioning before going to FPGA
	Cheap	Too high minimum speed specification
		No ambient light filtering
TOF Sensor Receiver	Fast	
	Cheapest	
	No minimum speed	
	Ambient Light filtering	
	Single-ended output, designed for FPGA	

D. *Processor*

The processor onboard the PCB will need to handle the communication protocol used by the lasers. Our options for this are using an FPGA chip, a separate FPGA development board, or a microcontroller. We decided on using a separate FPGA development board, which will be interfaced to our PCB with GPIO headers as the final choice due to speed and processing reasons, explained below.

Initially, using an FPGA chip seemed to provide the best features: it can handle extremely high speed because it will be mounted on the same PCB as the lasers and receivers, and use hardware to handle logic making it extremely fast. This option puts an extremely high cap on the speed from the processor, and is reasonable to implement as well. When researching the chip, however, all of the recent Intel/Altera FPGA chips that our team is most familiar with were all out-of-stock, out of our budget, or were extremely outdated. Thus, this option became unrealistic.

A separate FPGA development board interfaced with our PCB is the second option that we considered. The main concern for this was the speed of the signals that can travel cleanly between the boards, as this project deals with high-speed communication. The FPGA provides 72 GPIO pins to interface with external sources, and we determined that at least a 12.5 MHz, or 25 Mbps signal, can cleanly be transmitted based on oscilloscope readings (see Fig. 2).

The last option is using a microcontroller. Due to prior experience, we looked into using an MSP430, which supports at most a 25 MHz clock. Due to software overhead on the MCU, the actual throughput that the MCU can handle is much lower. Even with the best-case scenario in which it outputs a signal at 25 MHz, it will not surpass the FPGA development board speed. Furthermore, a MCU will make it more difficult to transfer data in parallel, and transmission speed would be further limited if we do not use pre-existing protocols supported by the built-in hardware, leaving little room for customization and optimization.

Due to its speed, ease of handling data, and customization capabilities for the procool, we decided to interface our PCB with a separate FPGA development board. In the future, when the focus of this device gets shifted further towards secrecy, budget increases, and obtaining parts becomes easier, then the on-board FPGA chip will most likely become the best option.

E. *User-Device Interface*

Our device will need to interface with a user device that will send and receive data. We primarily considered three interface options for our devices: USB-A, USB-C, or a lightning port.

We initially planned to make an iPhone-compatible device, but as we redefined our use case, we found this feature to be unnecessary in addition to being difficult to work with due to Apple's hardware restrictions. Thus, we were left with the two USB options.

In terms of power, some USB-C ports provide a Power Delivery feature that allows a much-higher voltage and current as compared to typical USB ports. After confirming that Apple computers support the capability to be a Power Delivery source (not just a sink), we determined that receiving 9V3A through this mode will be sufficient to power the entire board and the FPGA board. This made it favorable over USB-A. In addition, since all USB-C ports are backwards-compatible with USB-A protocol, we do not need to worry about the added complexity that it may cause, leaving availability on older devices to be the only minor potential drawback.

We also needed to decide where to handle the USB protocol, which can happen either using our own code on an FPGA or using an external IC. Previously, one of our team members implemented a modified version of the USB protocol on an FPGA in a one-month, two-person effort. Since implementing this protocol handler is not novel to our project's purpose nor its complexity, we determined that it would be wasteful to spend a significant amount of time and effort on this. We instead elected to use a COTS USB transceiver. As these chips can handle the full speed of the USB, the only trade-off we would face is the added complexity and area on the PCB, which we determined to be reasonable.

F. *Laser Protocol*

Table 6 shows the theoretical performance of various protocols. For this purpose, we assume that the laser logic is driven by the FPGA with a 50 MHz clock and a max possible data rate of 25 Mbps across the GPIO pins. Furthermore, we assume that the laser hardware can handle at most 10 Mbps of transmission[1]. This value can reasonably go down in reality, which will further limit the speed of the protocol where the laser circuit is the limiting factor. The data rate column on the table shows the transmission speed of the combined laser lines, and the throughput indicates the number of usable bits sent per second. There are two UART protocols: one that does not oversample, always using the middle sample as its determined value (see Fig. 5 for timing detail), and one that oversamples 8 times, using a majority vote to make a decision on the bit.

All of the data transmission protocols meet the desired use-case requirement of 4 Mbps throughput. However, the SPI and Manchester Encoding protocols rely on the fact that the hardware can operate at 10 Mbps, which adds additional risk that cannot be verified until the hardware arrives. Because of this, we ruled out SPI and Manchester encoding.

Using the UART protocol allows us to have a large margin on time even when hardware transmission speed is limited. Furthermore, these protocols are not too complex to implement. Since hardware speed requirements are easily met with both, we plan to implement our initial version with UART oversampling for better accuracy.

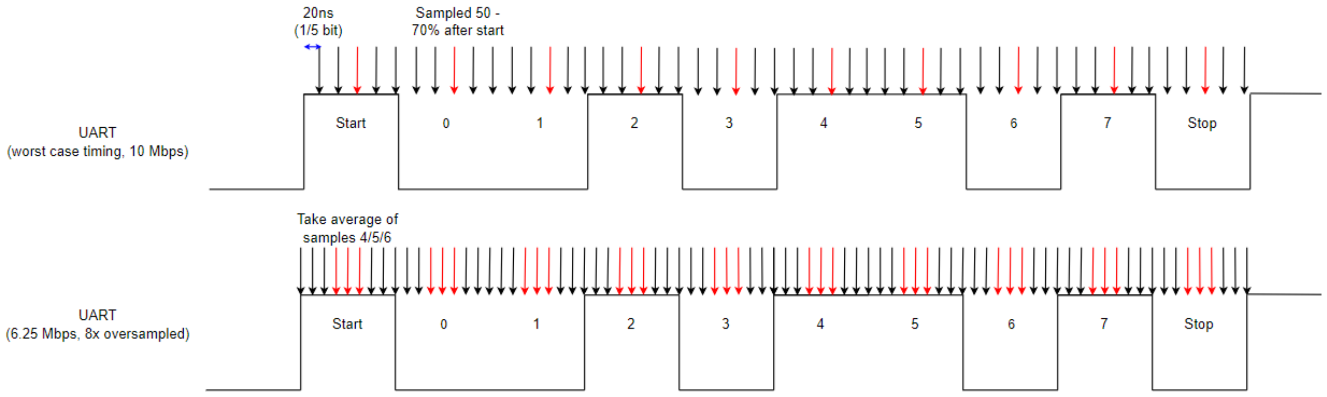

20ns
(1/5 bit)
Sampled 50 - 70% after start
UART
(worst case timing, 10 Mbps)
Start
0
1
2
3
4
5
6
7
Stop
Take average of samples 4/5/6
UART
(6.25 Mbps, 8x oversampled)
Start
0
1
2
3
4
5
6
7
Stop

Fig. 5. Different UART options. The red arrows indicate relevant samples that are recorded.

[1] Value estimated from component datasheets.

TABLE 6. LASER PROTOCOL OPTIONS

Protocol	Laser Use	Total Data Rate	Total Throughput	Limiting Factor
SPI	1 clock, 1 data	10 Mbps	10 Mbps	Laser Circuit
UART	2 data	10 Mbps	16 Mbps [a]	FPGA clock
UART (oversampled)	2 data	6.25 Mbps	10 Mbps [b]	FPGA clock
Manchester-Encoded Serial	2 data	10 Mbps	10 Mbps [c]	Laser Circuit

a. Refer to (3) below.
b. Refer to (4) below.
c. Refer to (5) below.

$$10\,Mbps \cdot \frac{8\,data}{10\,bits} \cdot 2\,lasers = 16\,Mbps \qquad (3)$$

$$\frac{50\,MHz\,Clock}{8x\,oversampling} \cdot \frac{8\,data}{10\,bits} \cdot 2\,lasers = 10\,Mbps \qquad (4)$$

$$10\,Mbps \cdot \frac{1\,data}{2\,edges} \cdot 2\,lasers = 10\,Mbps \qquad (5)$$

VI. SYSTEM IMPLEMENTATION

A. Laser Transmitter Circuit

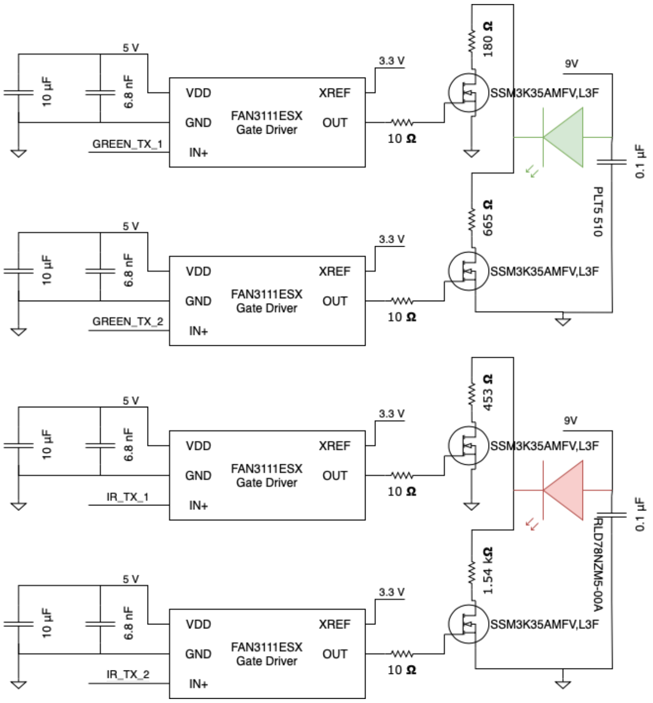


Fig. 6. Dual Transmitter Schematic

The laser transmitter circuit schematic is shown above in Fig. 6. The lasers selected are single-mode lasers, which have lower rise and fall times as compared to multi-mode lasers. Each laser is switched with two low-side NMOS switches that have different resistors in series to create three laser states: off, low power (logic 0), and high power (logic 1). The optical power of a logic 0 is 0.5 mW, and the optical power of a logic 1 is 2.5 mW. The equivalent capacitance of the gate driver is found in (6). This value is then used in (7) to find that the rise and fall time to reach 98% of the full-charge voltage on the MOSFET gates are 2.7 ns each (5.4 ns total), which is marginal compared to our speed requirement of 73.8 ns.

$$C_{eq} = \frac{Q_g}{V} = \frac{340\times10^{-12}\,C}{5\,V} = 68\,pF \qquad (6)$$

$$t_{98\%} = 4R_gC_{eq} = 4\times10\,\Omega\times68\,pF = 2.7\,ns \qquad (7)$$

The bypass capacitor for the gate driver is found with (8) using the rule of thumb of 100 times the equivalent gate capacitance:

$$C_{byp} = 100C_{eq} = 100\times68\,pF = 6.8\,nF \qquad (8)$$

Series resistors for the laser diodes are found using (9) through (14) and values from the datasheets. Resistors are swapped for the nearest values available at 1% precision.

$$R_{Glow} = \frac{V_{in}-V_f}{A_f} = \frac{9\,V-4.5\,V}{0.025\,A} = 180\,\Omega \qquad (9)$$

$$R_{Ghigheq} = \frac{V_{in}-V_f}{A_f} = \frac{9\,V-4.75\,V}{0.03\,A} = 142\,\Omega \qquad (10)$$

$$R_{Ghigheq} = 142\,\Omega = \frac{R_{Glow}R_{Ghigh}}{R_{Glow}+R_{Ghigh}} = \frac{180\,\Omega\times R_{Ghigh}}{180\,\Omega+R_{Ghigh}} \qquad (11)$$

$$\Rightarrow R_{Ghigh} = 673\,\Omega \approx 665\,\Omega$$

$$R_{IRlow} = \frac{V_{in}-V_f}{A_f} = \frac{9\,V-3.3\,V}{0.0125\,A} = 456\,\Omega \approx 453\,\Omega \qquad (12)$$

$$R_{IRhigheq} = \frac{V_{in}-V_f}{A_f} = \frac{9\,V-3.4\,V}{0.016\,A} = 350\,\Omega \qquad (13)$$

$$R_{IRhigheq} = 350\,\Omega = \frac{R_{IRlow}R_{IRhigh}}{R_{IRlow}+R_{IRhigh}} = \frac{453\,\Omega\times R_{IRhigh}}{453\,\Omega+R_{IRhigh}} \qquad (14)$$

$$\Rightarrow R_{IRhigh} = 1539\,\Omega \approx 1540\,\Omega$$

B. Laser Receiver Circuit

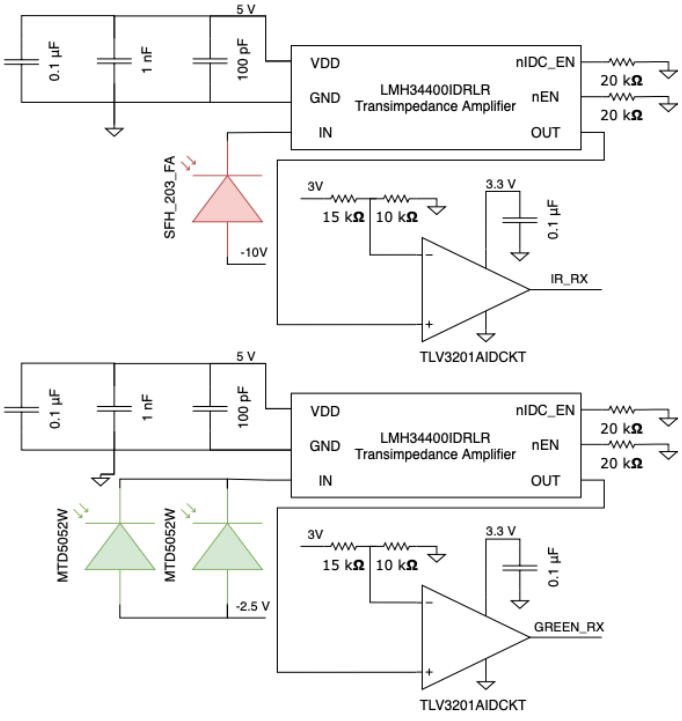


Fig. 7. Dual Receiver Schematic

The schematic of the laser receiver circuits are shown above in Fig. 7. The photodiodes were selected primarily for their timing characteristics, availability, and spectral response to ensure their ability to effectively filter out the opposing laser. Once we selected the photodiodes, the circuits were designed to match their voltage and current specifications. We use two green photodiodes in parallel to double the receiving current to better meet the transimpedance amplifier (TIA) specifications; this is needed only for the green photodiodes as they produce less current than the IR photodiodes. The nIDC_EN pin on the TIA is pulled down to enable the ambient light cancellation feature built into the IC. This pin is also connected to the FPGA so it can be disabled later if necessary. The nEN pin is pulled down which turns on the chip, and is also connected to the FPGA so it can be toggled. The output of the TIA is a maximum of 1 V_{P-P} centered at 1 V, so a comparator is used to convert this into a TTL-level digital signal. Later testing may reveal that thresholds of the comparators are set incorrectly; we plan to handle this problem by swapping out the resistors on the inverting inputs, which should be easy to do on the fly. The negative voltage biases on the photodiodes are based on specifications from their datasheets. The negative voltage rails are created by charge pumps which require very little current. We decided to use the MAX889RESA+ charge pump for creating –2.5V on the green receiver and the LM2682MM/NOPB charge pump for creating –10V on the IR receiver. Additionally, a 3V reference (MCP1501-30E_SN) is used to create the references for the comparator so they are stable.

C. USB Interface

We will be using an FTDI chip, FT232HPQ, for our USB transceiver. We decided on using an FTDI chip due to their widespread use. Our requirements made it ideal for us to find a chip that can interface with a single COM port from our laptop; meet our project's minimum speed while also working with the limited 50 MHz clock on the FPGA as well as the 25 Mbaud (12.5 MHz clock) that can communicate across the board to the IC; and support USB Power Delivery (USB-PD), which allows us to power all of our circuits using a single COM port.

FT232HPQ allows a speed of up to 12 Mbps through its Fast Serial mode to the FPGA, which is over double our required speed, and also allows us to request 9V3A via USB-PD. Additionally, this IC features eight different protocols for communicating with the FPGA, which gives us great flexibility. We are planning on using Fast Serial, which is a form of UART that uses a clock. However, we will connect all of the data bus lines from the FTDI chip to the FPGA GPIO as risk mitigation, which will enable us to switch to a different protocol later. The schematic of this IC as well as the USB-C connector and the connectors to the FPGA are shown in Fig. 14. The FPGA GPIO pins are all identical, so the exact pin mappings do not matter for implementation. The USB connector features TVS diodes to suppress ESD events while plugging and unplugging the device, and a ferrite bead to reduce noise from the power line. FT232HPQ uses the four standard USB 2.0 data lines for transmission using USB 2.0 protocol, and CC1/CC2 pins to request 9V on VBUS. We will not need the rest of the USB-C pins on the chip.

D. Power Circuit

All components on our PCB are powered by the 9V3A supplied via USB-PD, as requested by the FTDI chip. Our PCB will then generate a 5V and 3.3V line from the 9V supply by using LDOs (NCP1117IDT50T4G and NCP718BSN330T1G, respectively). The components on this board are powered by a combination of these three voltages. We also power the FPGA board from this 5V line, supplying power through its GPIO header with an ideal diode made with a PMOS (AO3415A) in series to protect against reverse current. This allows us to power the entire board off of the same port as communication, greatly simplifying the setup to use our device. This also prevents the need for a boost converter to create high voltage to power the lasers, which avoids a large source of noise from being present on the board. Schematics for the power circuit, along with a complete PCB schematic, are located at the end on Fig. 14.

E. PCB Layout

The PCB Layout was done with careful attention to signal integrity on high speed data lines while keeping fabrication costs to a minimum. There is a significant price hike from four layers to six, so we were limited to using a four-layer board. The best layer stackup for signal integrity given our power requirements and layer limitations is to use the outer two layers as signal layers with power planes poured, and the inner two layers as unbroken ground planes. The board has a 3.3V plane on the top signal layer and a 5V plane on the bottom. This gives all signals an unbroken return path on a neighboring layer and also ensures we have no via stubs. Common rule of thumb states that traces need to be treated as transmission lines if the total conductor length approximately exceeds the wavelength of the signal divided by twenty, which is only true for the USB data lines that can be connected to a long cable supplied by the user. Matched impedance, matched length, and differential routing are all used to make sure that the board meets USB data requirements. Other high-speed signals are also length-matched, but this is less important as the traces are all short relative to communication speed since they are only used on this small board. The final PCB is shown in Fig. 8.

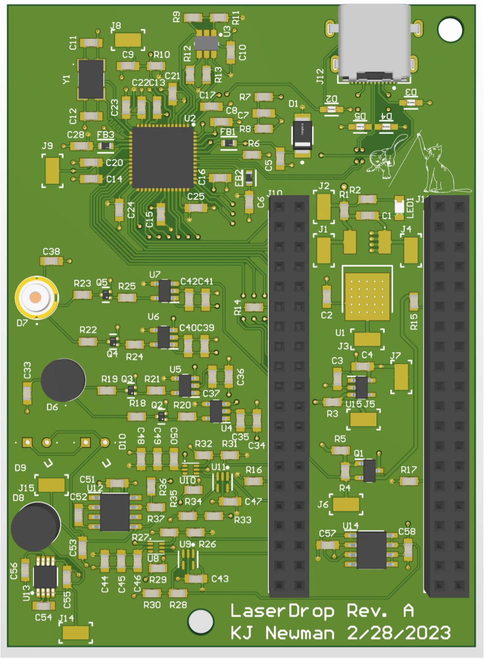


Fig. 8. Final PCB Layout[2]

F. *Communication Protocol*

The communication protocol begins with a handshaking procedure consisting of mutually-shared square waves. The sender device sends a square wave to the receiver who must reciprocate the square wave to begin data transmission. The packets are then transmitted, each containing a start byte, a packet tag byte, 60 data bytes, and 2 Hamming encoding bytes. Upon receiving the packet, the receiver sends an acknowledgement to signal that the next packet can be sent. If a packet is not received, a fail signal is sent by the receiver; the same packet will be resent repeatedly by the sender until an acknowledgement is received.

Once the data is transferred to the receiving computer/device, the packet is Hamming decoded. This method can correct 1-bit errors and detect 2-bit errors. If one or no errors are detected, the packet is added to a priority queue after correction. This queue uses packet tags as its priority, and later on reconstructs the entire file based on this. If the receiver computer/device detected a 2-bit error, it would add the faulty packet's tag to the error queue and send the tag back over the lasers. This will be received by the file sender, signaling to them that the packet must be re-transmitted.

When the last packet is received, the sender sends a stop message. The stop message consists of a stop byte, the final packet length, and the final packet tag, which should match the last packet sent. The final packet length is required because that is the only packet that has the potential of containing actual data less than 60 bytes. The tag will be used as a sanity check to ensure that data transmission is functioning appropriately. Meanwhile, the receiver will respond with a done signal once it receives the stop message and its error queue is empty.

The final phase of the communication is the termination phase, which mimics the start sequence: it requires an 8-bit duration square wave handshaking between the sender and the receiver. A timing diagram of the whole protocol is shown in Fig. 9.

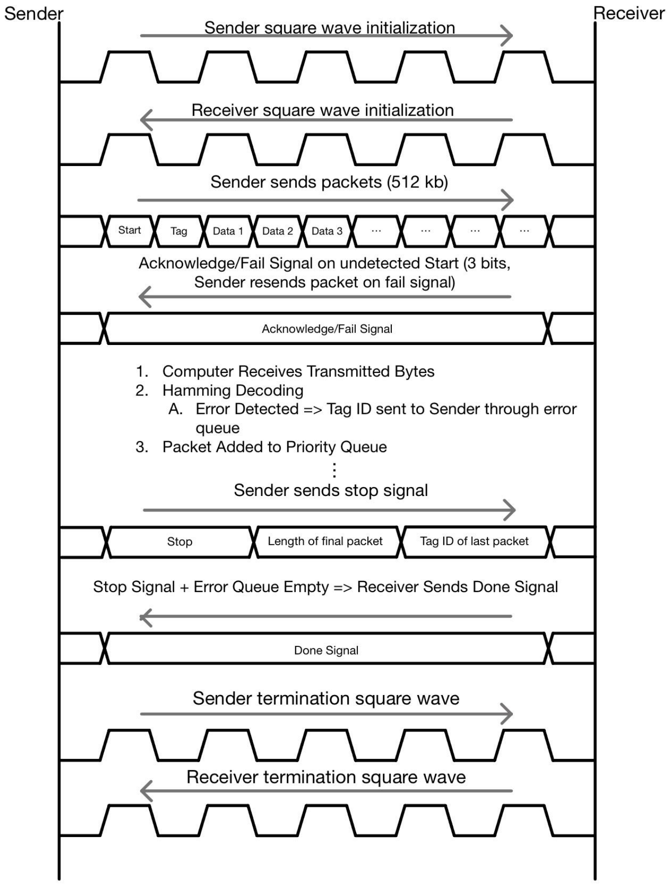

Sender
Receiver
Sender square wave initialization
Receiver square wave initialization
Sender sends packets (512 kb)
Start
Tag
Data 1
Data 2
Data 3
...
...
...
...
Acknowledge/Fail Signal on undetected Start (3 bits, Sender resends packet on fail signal)
Acknowledge/Fail Signal
1. Computer Receives Transmitted Bytes
2. Hamming Decoding
A. Error Detected => Tag ID sent to Sender through error queue
3. Packet Added to Priority Queue
⋮
Sender sends stop signal
Stop
Length of final packet
Tag ID of last packet
Stop Signal + Error Queue Empty => Receiver Sends Done Signal
Done Signal
Sender termination square wave
Receiver termination square wave

Fig. 9. Communication Protocol

G. *FPGA Implementation*

We will be using the Terasic DE0-CV for our FPGA development board, which contains a Cyclone V 5CEBA4F23C7N FPGA chip and 2 40-pin headers that will interface with our PCB. The header consists of 36 GPIO pins each that use 3.3V logic, and 4 pins that are used for the 5V and 3.3V power.

Fig. 10 shows the simplified FSM that the FPGA will be running on. Initially, the FPGA starts at the INIT state where it awaits for either data from the USB transceiver chip or a signal coming from the laser receiver. Upon seeing either, the device would initiate or respond to the handshake by sending a square wave through the laser transmitter. If this procedure is not successful, the receiver goes back to the INIT state, while the transmitter will continue trying until a timeout. After handshaking, the FPGA behavior will split depending on whether the device becomes a transmitter or receiver.

[2] GPIO pins are placed on the top of the board on this 3D rendering, but will be placed on the bottom on our actual board.

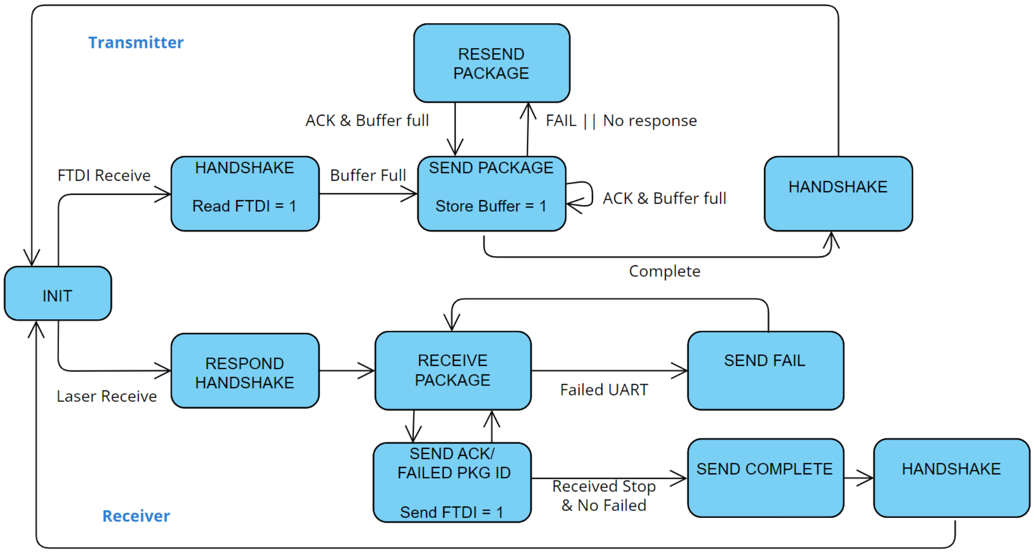


Fig. 10. Simplified FPGA FSM

The transmitter will send a packet to the receiver by reading data from the USB transceiver and sending it over via lasers. If it receives an acknowledge signal back, then it will continue to fetch the next packet from the transceiver , while it will remain in the same state to resend the current packet if it sees any other behavior. The FPGA will store the current message in a 512-bit register so the packets can be easily resent if needed.The simplified interface between the USB transceiver and the FPGA is shown in Fig. 11. The hardware will be split into two stages, one that will operate at 10 MHz to fetch 2 bytes from the USB transceiver, the other that will send the received byte simultaneously over its two lasers at a speed of 5 MHz or less. Once both stages finish their task, the first stage will proceed to load the next byte from the transceiver while the second stage will proceed to transmit the bytes from the first stage. As the first stage utilizes a clock twice as fast as the second, the pipeline should have minimal delays waiting for the stages to finish. The transmitter will also send any received tag packets over to the USB transceiver using the TX data line so the computer/device can resent those packets. The transmitter's behavior will continue this way until it sees a start sequence from the USB transceiver, in which they will send the short message and continue to wait for a response.

The receiver FPGA, meanwhile, will perform three parallel tasks: one for listening to its receiver and sending acknowledge/fail signals back, one that relays the complete, received data to the transceiver, and one that awaits for an erroneous packet tag from its computer/device. As all of these interactions run on separate lines (laser, transceiver TX, and transceiver RX), these tasks can be easily parallelized using the FPGA. The FSM will keep track of the state that the receiver is in, with both devices wrapping up its interaction with a handshaking at the end that will put the state back to INIT.

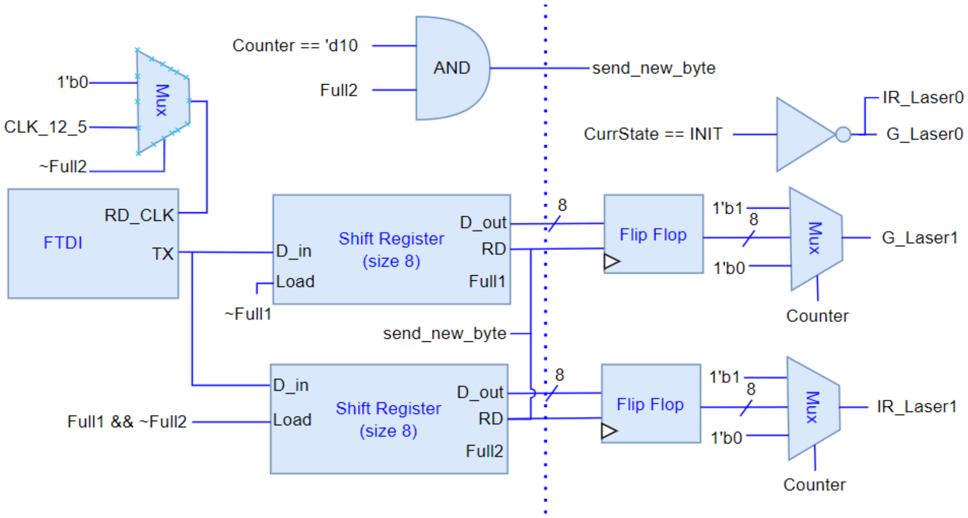


Fig. 11. Block Diagram of Transmitter

H. Sender Data Transmission Algorithm

Once a user tells the application to send over a file, the software will send a start transaction signal to the USB transceiver so that the FPGA can initialize the handshaking. During normal data transmission, the sending computer is responsible for providing tags and Hamming encoding for each packet. The next packet is provided after receiving an acknowledgement, so that there is always at least one packet in the USB transceiver buffer. In addition to providing this data for transmission, the sender must also be able to resend the packet in case the receiver detects an error in the Hamming code. In order to resend a packet that is requested from the receiver, the tag and the location within the file is necessary. A dictionary will be used to store this information. The tag will be used as a key, and the location within the file will be stored in the entry. Upon an erroneous message receipt, the corresponding packet will be reconstructed and resent. When the final packet is constructed, the packet will generally contain a smaller amount of data bytes than others. Thus, we plan on padding the final packet with zeros and send a stop signal that indicates the size of the relevant data bytes in the final packet. This allows the receiver to reconstruct the shortened packet to append at the end of the entire file. Additionally, the tag of the final packet will be provided in the stop signal as a form of error checking. After the FPGA receives a done signal back and completes the termination handshake, it will send a transaction complete signal through the USB transceiver that marks the end of the transaction. A block diagram showing the flow of information on the sender user device is shown in Fig. 12 below.

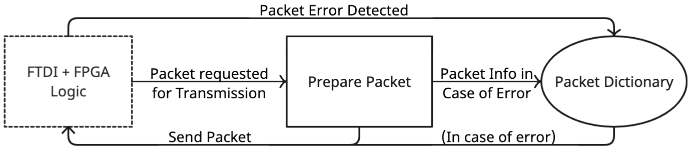


Fig. 12. Sender Block Diagram

I. Receiver Data Transmission Algorithm

Once the receiver FPGA responds to the handshake, it will send a signal to the user computer/device that will mark the start of transaction.The user computer/device is responsible for error checking through Hamming decoding. Upon detection of a 0 or 1-bit error, the packet is corrected if necessary and placed onto a priority queue that uses packet tags as its priority. If the packet tag wraps around and two messages have the same tag, it will then use the order that it is pushed onto the queue as its secondary priority. Upon the detection of a 2-bit error, the packet data is discarded, its tag will be pushed onto an error queue, and the tag will be sent to the USB transceiver for retransmission. The tag stays in the queue until a packet with the same tag is successfully retransmitted. If no errors are detected, the decoded packet is placed onto a priority queue as well. The priority queue is used for reconstruction purposes at the end of data transmission. Using the tag as priority allows packets to be reconstructed in the proper order. The receiver continues processing data in this fashion until the stop signal is received. Upon receiving this, the receiver must note that the final packet has been transmitted and store its length. Once the error queue is empty, the receivers should send a stop signal indicating that it has acquired all the data, and data reconstruction will begin. Items will be removed from the priority queue and reconstructed in that order. Each data packet will then be lined back-to-back to form the full file. A block diagram showing the flow of information for the receiver is shown in Fig. 13 below.

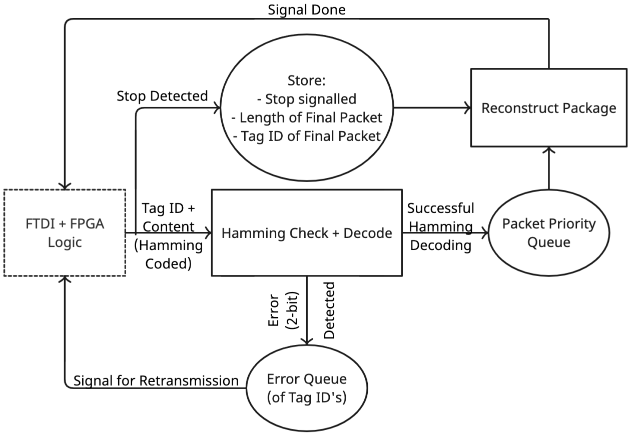


Fig. 13. Receiver Block Diagram

VII. Test, Verification, and Validation

System testing will address the laser-receiver signal link, USB connectivity, optical angular error, hardware power, error correction, full system transmission speed, distance functionality, ambient lighting performance, and full-system stress test using a large file injected with errors. These tests are designed to verify our use-case and design requirements and ensure that the overall system is functional, efficient, robust, and user-friendly.

A. Laser to Receiver Speed Test

Data transmission relies on the fact that a clean square wave is transmitted across the devices. We will evaluate the signal quality by transmitting a 1.36 MHz (2.71 Mbps) square wave on each laser between the two units at a distance of 1 m. We will measure the received wave using an oscilloscope on the receiver GPIO headers and evaluate the signal quality. We will record rise time and fall time to ensure that both values are below 73.8 ns. We will also verify that the transmitting square wave is sufficiently clean qualitatively. This will be performed early on so adjustments can be made in hardware if a problem arises in signal quality.

B. *USB Interface Test*

The USB-C connection from the computer to the FTDI chip to the FPGA must be tested to ensure accuracy and speed on the user-device interface side. To test USB connectivity, we will echo data at 5.42 Mbaud between the FPGA and the computer. We will verify that the data integrity is held and the speeds met.

C. *Optical Angular Error Test*

To avoid light leakage to the surrounding environment, we specified an optical angular range of 0.5 to 2 degrees at 1 m in our use-case requirements. This translates to a target radius size of 0.87 cm to 3.5 cm. Since laser beams may not form perfect circles, this test will verify that both the semi-major and semi-minor axis are between 0.87 cm to 3.5 cm. Because we are using an adjustable lens, if this test fails, the focus of the beam will be adjusted accordingly to optimize leakage and ease-of-aim.

D. *Power Test*

Our device will be powered with 9V3A via a USB-C Power Delivery mode, as stated in our use-case requirements. We will verify that our PCB's current draw is less than 3A during all modes of operation using a current clamp.

E. *Error Correction Test*

Our system is required to detect up to 2-bit errors, and correct 1-bit errors. Upon detection of a 2-bit error, data will be requested for the second time. This functionality will be verified by creating test packets that are injected with 1 and 2-bit errors to ensure that our system handles these errors appropriately.

F. *Transmission Speed Test*

The whole system will be tested once fully integrated. Two devices will be connected to the same computer on different ports, and data will be transmitted between them to determine baud rate and latency. Using the same computer ensures that the same clock is used to time the start and end of transmission on both sides. The average time will be measured for file sizes under and over 256 packets to ensure that our overall algorithm is robust and efficient for small and large files. As outlined in our use-case requirements, we will verify that both of these transmissions happen at a rate of at least 4 Mbps, or 50 PDF pages per second.

G. *Distance Test*

The whole system will be tested at a distance of 1 m to verify its functionality, as required by the use-case requirements. This test will pass if a file successfully transmits from one computer to another.

H. *Ambient Light Test*

As outlined in our use-case requirements, our device needs to perform in a variety of constant ambient-light environments. We will be testing our device in a variety of different rooms at a distance of 1 m. Our device should be able to handle both dark environments (e.g. lights off) and bright environments (e.g. lights on or windows open).

I. *System Stress Test*

Lastly, the whole system will be stress tested using a large file that deconstructs to at least 2560 packets, artificially injected with 1-bit and 2-bit errors amid signal disruptions (the light source will intermittently be blocked and unblocked), all at a distance of 1 m. Our system should be able to handle all of these anomalies and successfully transmit the file.

VIII. Project Management

A. *Schedule*

Fig. 15 shows the Gantt Chart with our project schedule and member assignment for each task.

B. *Team Member Responsibilities*

Each team member has the following specific responsibilities:

- Anju Ito: FPGA software
- Roger Lacson: User-device software
- KJ Newman: PCB design and 3D printing

Furthermore, all members have worked to design the overall protocol and architecture of the system, and will partake in the integration, testing, and verification at the end of the project.

C. *Bill of Materials and Budget*

Table 7 shows the bill of materials for creating this project, which includes enough components to make three PCBs. It is split into two sections: one showing the cost of parts to create our MVP, and another showing the cost of parts purchased solely for prototyping. The generic resistors and capacitors used on the PCB are included as a single line item for brevity; they are standard 0603 components.

D. *Risk Mitigation Plans*

We are planning on using an FTDI chip for this project, which we have not directly interfaced with before. This adds risk to the project since all aspects of our device must deal with the chip. Available reference schematics for the chip are not well-documented on their documentation sheet, adding concerns about its behavior. If the circuit does not work properly, we plan to power the board separately instead of USB-PD and use an external USB to UART adapter to interface with the FPGA. This will make it slower and more clunky, but will still deliver a functioning product. Furthermore, if the FTDI chip works properly on the circuits side, the FPGA must deal with the FTDI's custom protocol and interface, which has a risk of taking up large portions of time. We plan on mitigating this risk by connecting all possible interfaces from the FTDI chip to the FPGA in case one is easier to deal with than the other. Lastly, the FTDI chip will interact with an EEPROM, which may not be intuitive to program initially. The mitigation plan if the programming does not work out is to use the FTDI chip in the default mode of operation (UART) and power our board using an external power supply.

Our PCB will also contain many risk mitigation features. Because we will most likely only have the time and budget to order one revision of the board, there is substantial risk that any one part would not work, which will jeopardize our entire project. We mitigate this risk by creating as many configurable options as we can on our board. This includes connecting pull-up and pull-down resistor pads to all configuration lines, and placing many 0 Ω resistors and DNP pads, which will allow us to change power and signal paths as needed. We also connect many of the configuration lines to the FPGA, which allows us to change the settings of any of our ICs if our intended modes do not work. This essentially makes our board configurable, as components can be modified and/or disconnected later without needing a new revision.

Regarding software, we plan to implement priority queues and dictionaries to store information until error detection protocols confirm the correctness of a packet. Should these algorithms be too slow, especially across larger packets, an array of fixed size will be used to pre-allocate space so we can save time on data accesses in the hopes of speeding up the algorithm. The main drawback of this would be memory inefficiency, but this implementation should speed up the algorithm significantly.

IX. Related Work

Most existing implementations of laser communications are considerably higher power than our application. The only common use case is in space communications. This is done by organizations such as NASA [2], ESA [3], SpaceX [4], and the U.S. Space Force [5]. The lasers in these applications are many orders of magnitude higher than our application, which constrains itself to be eye-safe. Free-space optical communication using LEDs instead of lasers has been frequently experimented with for internet delivery, such as RONJA [6]. This is not replicable for us because it is not discreet norsecure, but the concepts behind it are the same. Finally, laser-based communications are very common in everyday life in the form of fiber optic communication. This works the same way as our system, but does not send the laser through free space; the transmitter and the receiver electronics are similar, but there are many differences in practicality, such as the fact that fiber optic lasers are far too strong to be eye-safe. However, fiber optic circuits do exactly the same thing that we need our circuits to do, so their components and reference designs proved very useful. LiFi is a commercially available system for running wireless data over IR LEDs and receivers at speeds of about 100 to 150 Mbps [7]. It is comparable to our project in that they also transmit data over light in free-space. However, the modules are very expensive and not open-source, so we were unable to draw much inspiration from it.

Students from the University of Central Florida created a project called LDT-AIR for their capstone with similar requirements as ours: they used an eye-safe IR laser and photodiodes to transmit data over a short range with a comparable budget [8]. Their analysis and component trade studies proved highly valuable in creating our design for LaserDrop. This project uses a fiber optic laser modulation IC, a fiber optic laser for transmission, and a photodiode and TOF-sensor TIA to receive. We realized that the fiber optic hardware will be too bulky and heavy for our use case, but the electrical subsystem designs were helpful in understanding what works best for low-power laser transmission.

A team from Dartmouth College created a system called AmphiLight for communicating between a flying drone and an underwater drone with a laser link [9]. While this project focused on the ability to track a moving target through the surface of the water (which refracts light), it also details the design of their laser communication link. The laser link is created with a high-power Thorlabs laser and a lot of optical components to aim the laser with high precision. This project is higher budget than ours, but the core concepts of their laser communication are still applicable. This project was able to achieve communication speeds of 5.04 Mbps.

Another Dartmouth College research team created a protocol for free-space optical communication described in the paper “The Darklight Rises” [10]. Darklight is a communication method that uses visible light LEDs at such a low duty cycle that it becomes invisible to humans. They do so by pulsing a green LED for 500 ns with a 0.007% duty cycle, and using the time of the pulse within the frame to determine its value. Their device never pulses light long or fast enough for the human eye to detect, and achieves a 1.6 kbps data rate. This does not meet our speed requirements, although we considered using a similar approach to make our lasers invisible, which would benefit us in our application of espionage.

X. Summary

The LaserDrop system is designed to transmit data via lasers to improve communication security and efficiency in covert operations. The device will be designed to work at a distance of 1 m for inconspicuousness, transmit data at 4 Mbps which would allow large documents to transfer in a short period, and utilize error detection and correction methods to ensure message integrity. Through careful design and continuous integration, we hope to mitigate any challenges to develop a system that maintains security and secrecy to benefit national security.

Glossary of Acronyms

COTS – Commercial-Off-The-Shelf
EEPROM – Electrically Erasable Programmable Read-Only Memory
DNP – Do Not Populate
ESA – European Space Agency
ESD – Electrostatic Discharge
SPI – Serial Peripheral Interface
TIA – Transimpedance Amplifier
TOF – Time-of-Flight
TTL – Transistor-Transistor Logic
TVS – Transient Voltage Suppression
UART – Universal Asynchronous Receiver/Transmitter
USB-PD – Universal Serial Bus Power Delivery
VCP – Virtual COM Port

References

[1] "ANSI Z136.1-2022: Safe Use of Lasers," ansi.org. https://blog.ansi.org/ansi-z136-1-2022-safe-use-of-lasers/ (accessed Feb. 28, 2023).

[2] "Laser Communications Relay Demonstration (LCRD)," nasa.gov. https://www.nasa.gov/mission_pages/tdm/lcrd/index.html (accessed Feb. 28, 2023).

[3] "Alphasat's laser terminal on target," esa.int. https://www.esa.int/Applications/Telecommunications_Integrated_Applications/Alphasat/Alphasat_s_laser_terminal_on_target (accessed Feb. 28, 2023).

[4] "SpaceX adds laser links to Starlink satellites to serve Earth's polar areas," arstechnica.com. https://arstechnica.com/information-technology/2021/01/spacex-adds-laser-links-to-starlink-satellites-to-serve-earths-polar-areas/ (accessed Feb. 28, 2023).

[5] "US Will Try Using Lasers to Send Data From Space to Drones," defenseone.com. https://www.defenseone.com/technology/2021/06/pentagon-will-try-using-lasers-send-data-space-drones/174810/ (accessed Feb. 28, 2023).

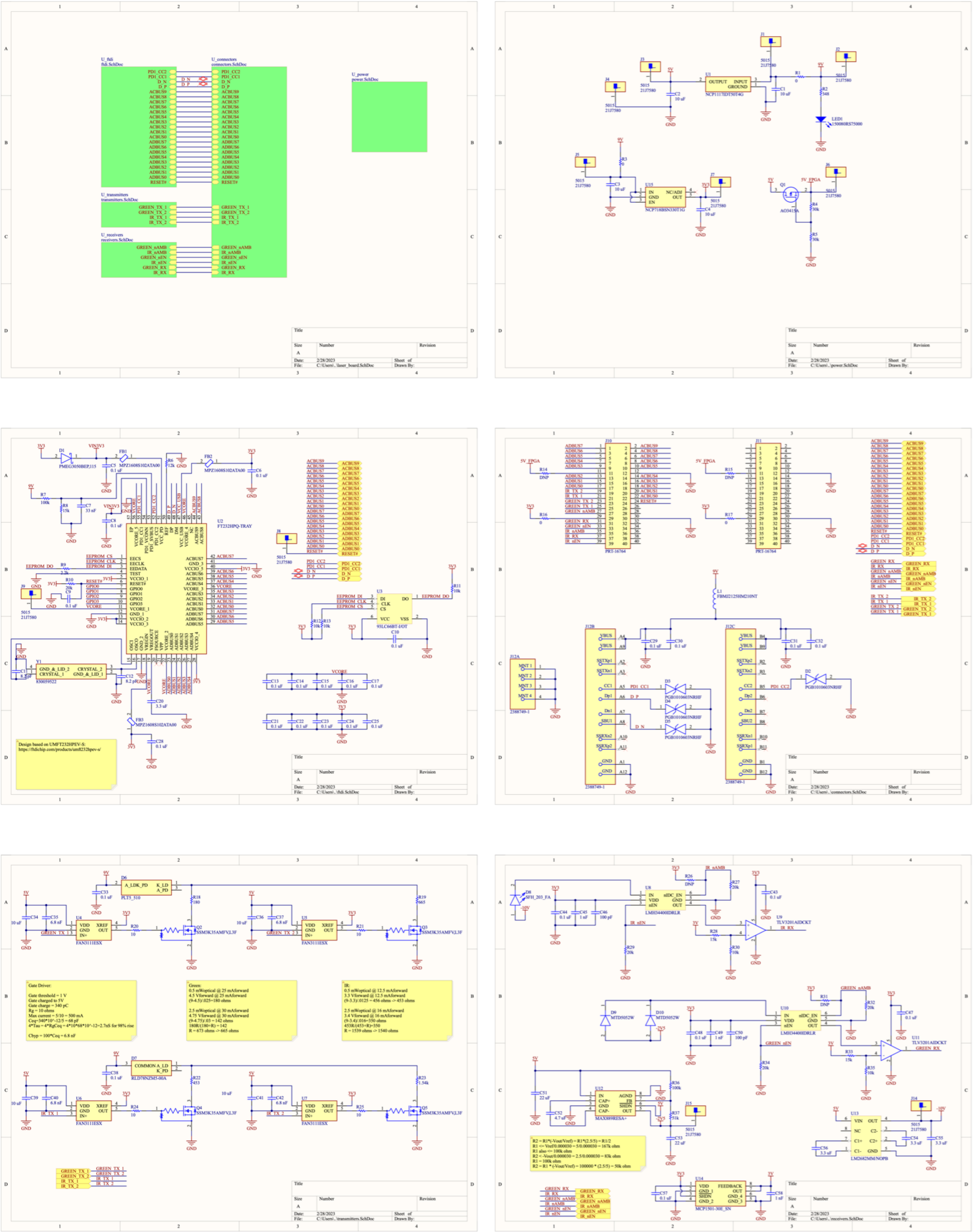
[6] "About Ronja," twibright.com. http://ronja.twibright.com/about.php (accessed Feb. 28, 2023).

[7] "What is Lifi?" lifi.co. https://lifi.co/what-is-lifi/ (accessed Feb. 28, 2023).

[8] "LDT-AIR," D. Clark, K. Figueiredo, and J. Jordan. https://www.ece.ucf.edu/seniordesign/fa2015sp2016/g32 (accessed Feb. 22, 2023).

[9] C. J. Carver, Z. Tian, H. Zhang, K. M. Odame, A. Quattrini Li, and X. Zhou, "Amphilight," GetMobile: Mobile Computing and Communications, vol. 24, no. 3, pp. 26–29, 2021.

[10] Z. Tian, K. Wright, and X. Zhou, "The Darklight rises: Visible Light Communication in the dark," Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, 2016.



Design based on UMFT232HPEV-S:
https://ftdichip.com/products/umft232hpev-s/

Gate Driver:

Gate threshold = 1 V
Gate charged to 5V
Gate charge = 340 pC
Rg = 10 ohms
Max current = 5/10 = 500 mA
Ceq=340*10^-12/5 = 68 pF
4*Tau = 4*RgCeq = 4*10*68*10^-12=2.7nS for 98% rise

Cbyp = 100*Ceq = 6.8 nF

Green:
0.5 mWoptical @ 25 mAforward
4.5 Vforward @ 25 mAforward
(9-4.5)/.025=180 ohms

2.5 mWoptical @ 30 mAforward
4.75 Vforward @ 30 mAforward
(9-4.75)/.03 = 142 ohms
180R/(180+R) = 142
R = 673 ohms -> 665 ohms

IR:
0.5 mWoptical @ 12.5 mAforward
3.3 Vforward @ 12.5 mAforward
(9-3.3)/.0125 = 456 ohms -> 453 ohms

2.5 mWoptical @ 16 mAforward
3.4 Vforward @ 16 mAforward
(9-3.4)/.016=350 ohms
453R/(453+R)=350
R = 1539 ohms -> 1540 ohms

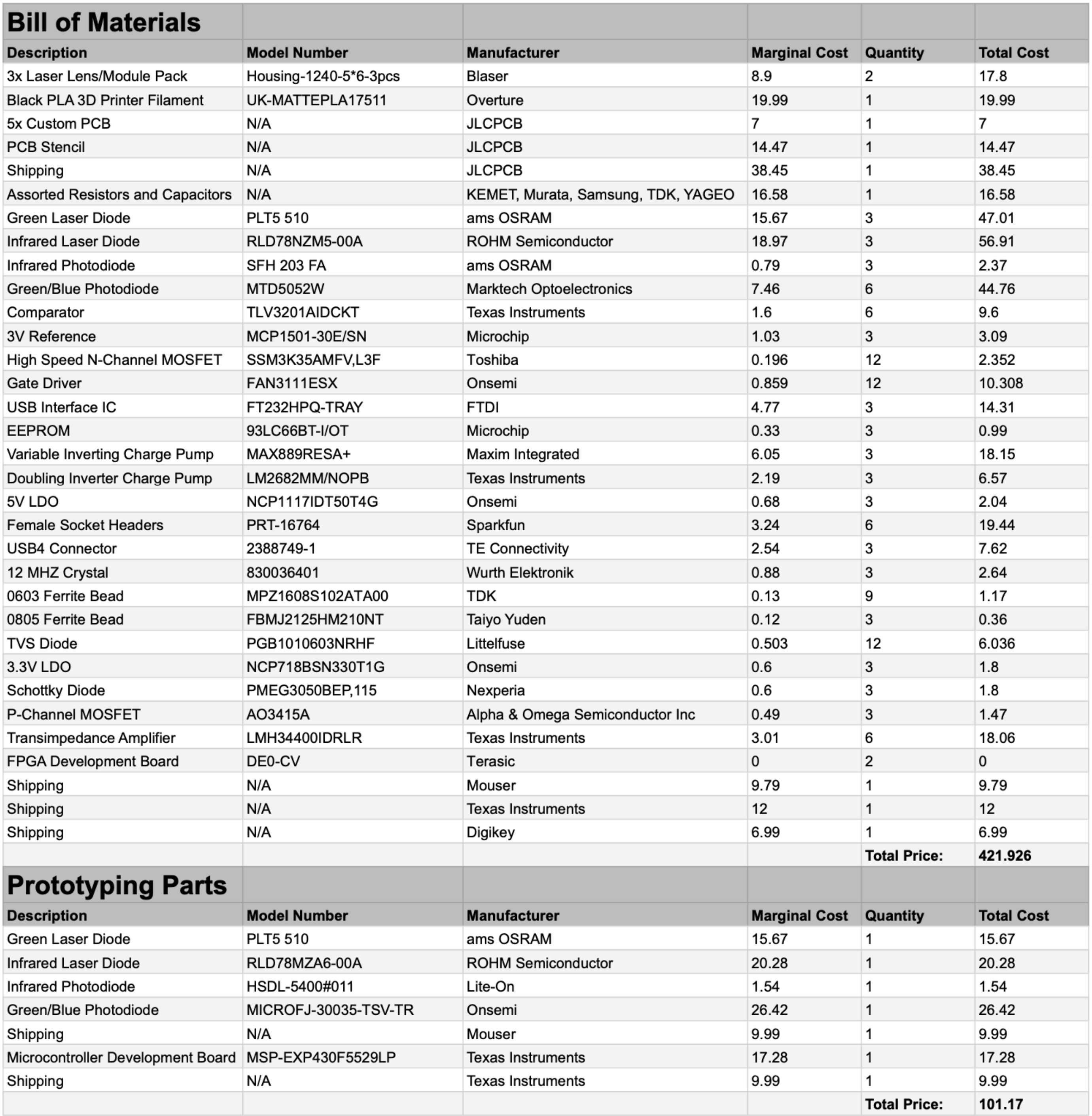
R2 = R1*(-Vout/Vref) = R1*(2.5/5) = R1/2
R1 <= Vref/0.000030 = 5/0.000030 = 167k ohm
R1 also <= 100k ohm
R2 < -Vout/0.000030 = 2.5/0.000030 = 83k ohm
R1 = 100k ohm
R2 = R1 * (-Vout/Vref) = 100000 * (2.5/5) = 50k ohm

Fig. 14. Board Schematic

Tasks	Wk1 (2/6)	Wk2 (2/13)	Wk3 (2/20)	Wk4 (2/27)	Wk5 (3/6)	Wk6 (3/13)	Wk7 (3/20)	Wk8 (3/27)	Wk9 (4/3)	Wk10 (4/10)	Wk11 (4/17)
Hardware					Spring Break						
Pick parts for preliminary testing	All										
Test preliminary parts		All									
PCB schematic design & simulation			KJ								
Test laser lens				Anju							
PCB layout				KJ							
PCB & parts ordering				KJ							
3D Print Laser Housing						KJ					
Assemble & test PCB						KJ					
Firmware/FPGA											
Set up FPGA dev environment		Anju									
Develop detailed FPGA design			Anju								
FPGA Implementation				Anju							
Test & refine						Roger/Anju					
FPGA compilation onto PCB							Anju/KJ				
Software											
Setup CPU USB environment			Roger								
Implementation - CPU to FTDI				Roger							
Develop UI						Roger					
Design a test suite							Roger				
Systems											
SW/HW integration								All			
Testing & Validation											
Speed/latency test									All		
Range test									All		
Error correction test									All		
Power consumption test									All		
Slack										All	All

Fig. 15. Gantt Chart

Table 7. Bill of Materials & Budget



Bill of Materials					
Description	**Model Number**	**Manufacturer**	**Marginal Cost**	**Quantity**	**Total Cost**
3x Laser Lens/Module Pack	Housing-1240-5*6-3pcs	Blaser	8.9	2	17.8
Black PLA 3D Printer Filament	UK-MATTEPLA17511	Overture	19.99	1	19.99
5x Custom PCB	N/A	JLCPCB	7	1	7
PCB Stencil	N/A	JLCPCB	14.47	1	14.47
Shipping	N/A	JLCPCB	38.45	1	38.45
Assorted Resistors and Capacitors	N/A	KEMET, Murata, Samsung, TDK, YAGEO	16.58	1	16.58
Green Laser Diode	PLT5 510	ams OSRAM	15.67	3	47.01
Infrared Laser Diode	RLD78NZM5-00A	ROHM Semiconductor	18.97	3	56.91
Infrared Photodiode	SFH 203 FA	ams OSRAM	0.79	3	2.37
Green/Blue Photodiode	MTD5052W	Marktech Optoelectronics	7.46	6	44.76
Comparator	TLV3201AIDCKT	Texas Instruments	1.6	6	9.6
3V Reference	MCP1501-30E/SN	Microchip	1.03	3	3.09
High Speed N-Channel MOSFET	SSM3K35AMFV,L3F	Toshiba	0.196	12	2.352
Gate Driver	FAN3111ESX	Onsemi	0.859	12	10.308
USB Interface IC	FT232HPQ-TRAY	FTDI	4.77	3	14.31
EEPROM	93LC66BT-I/OT	Microchip	0.33	3	0.99
Variable Inverting Charge Pump	MAX889RESA+	Maxim Integrated	6.05	3	18.15
Doubling Inverter Charge Pump	LM2682MM/NOPB	Texas Instruments	2.19	3	6.57
5V LDO	NCP1117IDT50T4G	Onsemi	0.68	3	2.04
Female Socket Headers	PRT-16764	Sparkfun	3.24	6	19.44
USB4 Connector	2388749-1	TE Connectivity	2.54	3	7.62
12 MHZ Crystal	830036401	Wurth Elektronik	0.88	3	2.64
0603 Ferrite Bead	MPZ1608S102ATA00	TDK	0.13	9	1.17
0805 Ferrite Bead	FBMJ2125HM210NT	Taiyo Yuden	0.12	3	0.36
TVS Diode	PGB1010603NRHF	Littelfuse	0.503	12	6.036
3.3V LDO	NCP718BSN330T1G	Onsemi	0.6	3	1.8
Schottky Diode	PMEG3050BEP,115	Nexperia	0.6	3	1.8
P-Channel MOSFET	AO3415A	Alpha & Omega Semiconductor Inc	0.49	3	1.47
Transimpedance Amplifier	LMH34400IDRLR	Texas Instruments	3.01	6	18.06
FPGA Development Board	DE0-CV	Terasic	0	2	0
Shipping	N/A	Mouser	9.79	1	9.79
Shipping	N/A	Texas Instruments	12	1	12
Shipping	N/A	Digikey	6.99	1	6.99
				Total Price:	**421.926**
Prototyping Parts					
Description	**Model Number**	**Manufacturer**	**Marginal Cost**	**Quantity**	**Total Cost**
Green Laser Diode	PLT5 510	ams OSRAM	15.67	1	15.67
Infrared Laser Diode	RLD78MZA6-00A	ROHM Semiconductor	20.28	1	20.28
Infrared Photodiode	HSDL-5400#011	Lite-On	1.54	1	1.54
Green/Blue Photodiode	MICROFJ-30035-TSV-TR	Onsemi	26.42	1	26.42
Shipping	N/A	Mouser	9.99	1	9.99
Microcontroller Development Board	MSP-EXP430F5529LP	Texas Instruments	17.28	1	17.28
Shipping	N/A	Texas Instruments	9.99	1	9.99
				Total Price:	**101.17**