The Illuminator

Authors: Claire Kim, Gleb Rybatsev, Sidharth Srivastava
Affiliation: Electrical and Computer Engineering Department, Carnegie Mellon University

***Abstract*—Lidars and time-of-flight cameras sensors are widely used in land and air robotics. However, their application in underwater robotics is limited because water blocks infrared light. We design a visible-light time-of-flight 3D imaging system for use in underwater robotics. In this design report, we evaluate the use-case requirements, and describe the architecture of our system.**

***Index Terms*—3D imaging, high speed illumination, LED, LED driver, time-of-flight, ToF camera, underwater imaging, visible light sensing**

1 Introduction

Autonomous underwater vehicles (AUVs) are increasingly relied upon for tasks like marine research, environmental monitoring, and underwater inspection (for infrastructure, ships, etc). A critical challenge for these systems is perceiving and mapping their surroundings in 3D, especially in murky or low-light environments where traditional stereo-vision approaches fail. The use case for this project is to equip AUVs (specifically the TartanAUV platform at Carnegie Mellon University) with a compact, low cost, 3D imaging system that enables close range navigation and manipulation. By providing accurate scene depth information underwater, the system allows robotic arms or manipulators to interact precisely with objects on the seafloor or in confined environments.

Current underwater volumetric sensing technologies each present significant drawbacks. Sonars are often the sensors of choice for long-range perception, but suffer from low resolution and multi-path effects. This is especially problematic for close-range perception for manipulation. Furthermore, phased-array 3D imaging sonars are often prohibitively expensive, with entry-level models priced above $30,000. Stereo-vision offers a more economical alternative, but also suffers from limitations. Stereo-matching algorithms rely on visual features in the images to compute a disparity map. However, underwater scenes often have low contrast, poor illumination, and simple geometry, which makes stereo-matching challenging. In the air, lidars, time-of-flight cameras or structured light cameras[1] often better performance than traditional stereo cameras. However, commercially available sensors operate in the infrared wavelength range, which is blocked by water. Therefore, they cannot be used on an AUV.

In this project, we design a time-of-flight camera that can be used underwater. We make it possible by using a **visible light source** rather than an infrared one. Our goal is to create an **integrated sensor module** that can be installed on a wide range of underwater vehicles.

2 Use-Case Requirements

2.1 Applications of AUVs

Autonomous underwater vehicles are widely used in inspection and maintenance of critical subsea infrastructure such as oil and gas pipelines, offshore platforms, and undersea communication cables. They enable routine monitoring of mechanical integrity, corrosion, and biofouling, as well as precise localization of faults or damage without relying on human divers. In addition to industrial use, AUVs support environmental and scientific surveys by mapping seafloor topography, inspecting marine habitats, and collecting visual and acoustic data for long-term studies. These missions often require the vehicle to operate close to structures or the seabed, where accurate, high-resolution 3D perception is essential for safe navigation and manipulation. This motivates the integration of compact time-of-flight (ToF) imaging systems to provide dense, reliable depth information in short-range underwater operations.

2.2 TartanAUV

Given the broad range of use-cases of underwater robots, it is impossible to produce a single set of design requirements that would fit every application. For example, inspecting an large dam for signs of structural failure would require a very different camera than inspecting a ship's propeller. In the first case, one would need wide field-of-view and long range to cover a large area quickly. In the second case, a high-resolution close-range sensor is preferable to detect small fractures, dents, corrosion, or biological buildup. Moreover, defining form-factor or electrical interface requirements is difficult because most commercial AUV designs are proprietary.

Instead of trying to tailor our sensor to a specific *commercial* application, we take a different route. We are collaborating with and taking inspiration from TartanAUV, an undergraduate student organization building underwater robots to compete in the international RoboSub competition. The competition involves a variety of navigation and manipulation tasks that must be completed by an AUV fully autonomously. RoboSub is sponsored by various companies engaged in underwater robotics, as well as the U.S. Office of Naval Research. The competition tasks are designed to be representative of typical problems solved in

[1] Eg., *Intel RealSense* products.

the industry. Thus, a sensor that can be used effectively in the RoboSub competition scenario, will likely be suitable for numerous commercial applications.

There are two tasks that stand to benefit the most from a close-range volumetric sensor.

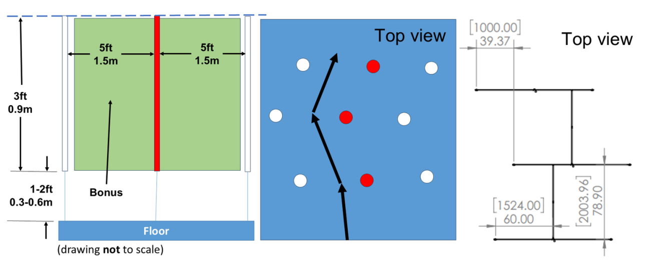

Figure 1: RoboSub 2025 navigation task (view from above). Image courtesy of RoboNation.

The **navigation task** requires the robot to pass through a narrow channel of poles without touching them. The gap between the poles is 1.5 m. (See Figure 1.) To accomplish this task, a 3D image sensor must be able to localize the poles quickly to compute and track an optimal trajectory.

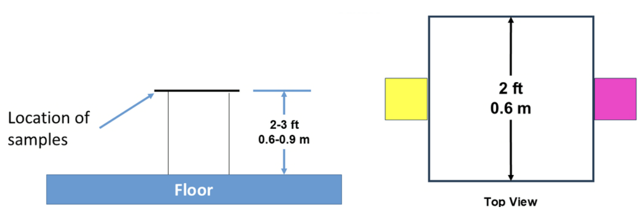

Figure 2: RoboSub 2025 manipulation task (side view). Image courtesy of RoboNation.

The **manipulation task** involves picking up objects resembling plastic waste items from a 2 ft-by-2 ft table, and placing them in bins. This task requires the 3D camera to localize the objects initially, as well as to track the objects as the vehicle and/or the manipulator is guided towards them. The objects are bottle-shaped, around 2 inches wide and 7 inches tall.

Note that the competition is held in a standard Olympic swimming pool with clear fresh water.

2.3 Use-case requirements

The use-case requirements are dictated by the RoboSub competition tasks above, as well as by the physical characteristics of TartanAUV's vehicle, *Osprey* (Figure 3). Osprey moves in the water using eight thrusters and reaches a maximum speed of around 1 m/s. To pick up objects for the manipulation task, Osprey is equipped with a suction tube that is shown in Figure 4. The inner diameter of the suction tube is 4 inches.

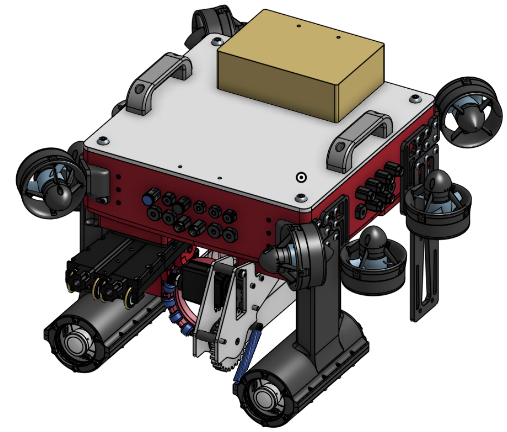

Figure 3: Osprey AUV overview. Image courtesy of TartanAUV.

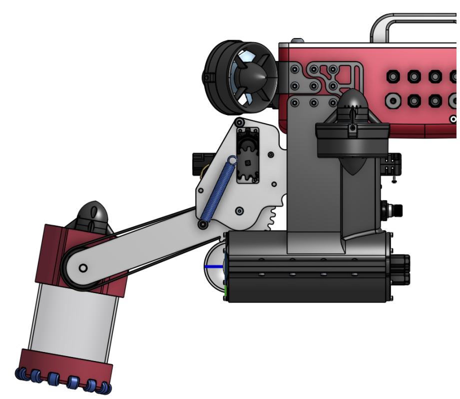

Figure 4: Osprey AUV with the suction tube deployed. Image courtesy of TartanAUV.

We list the use-case requirements below.

- **Maximum range: 2 m.** This is the maximum distance at which the sensor should be able to detect an object. This requirement is derived from the navigation task (see Figure 1). When passing between a pair of obstacle poles, the robot must be able to see the next pair of poles to determine its trajectory. The poles are 2 meters away.

- **Minimum range:** 25 **cm.** This is the minimum distance that the sensor can measure. This is approximate distance from the front of the vehicle (where the ToF will be mounted) to the suction tube inlet, where an object will be picked up during the manipulation task.

- **Horizontal field of view: 90 degrees.** This requirement comes from the navigation task. An angle of 90 degrees is required to see the next row of obstacles.

- **Range accuracy: ±2.5 cm.** This requirement specifies how accurately the ToF camera can measure distances to objects. The requirement originates from the manipulation task. The objets that must be picked up for the task are 5 cm wide, while Osprey's suction tube inlet is 10 cm wide. If we center the object in the tube, our position estimate must be within 2.5 cm of the ground truth to ensure that the object can be grabbed.

- **Lateral accuracy: ±2.5 cm** (Measured at 30 cm range.) This requirement specifies how accurately the ToF camera can resolve the position of a target along an axis orthogonal to the optical axis of the camera. Motivation is the same as for range accuracy.

- **System frame rate: 10 frames per second.** An overly slow camera would limit the speed with which tasks can be performed. The rate of 10 FPS matches the speed[2] of the onboard imaging cameras, ensuring that the depth images are synchronized with RGB images.

3 Principle of Operation

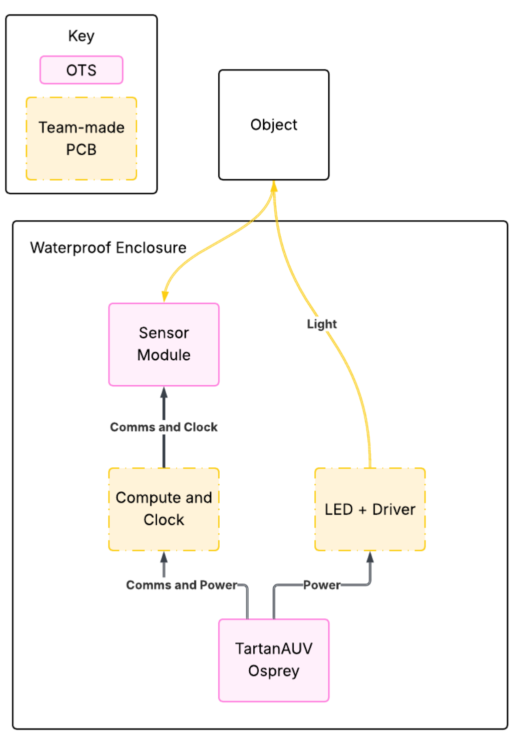

Figure 5: High Level Block Diagram

At a high level, our system consists of two main components: an illumination module PCB and a sense module PCB. The excitation module consists of a high power LED and peripheral driver circuitry to illuminate the scene in front of it. The sense module is an off-the-shelf image sensor which will interface through a connector to our compute.

The illumination module requires high frequency, high precision modulation of the LED output to the scene. The modulation frequency is calculated based on the desired distance to be measured. During the "off" stage of the LED, the sensor module will take its pixel measurement. The precise calculation for this frequency is shown later in the Design Requirements section.

The sensing module consists of an image sensor with an array of pixels that can read back the intensity of the reflected light from the illumination module. These image sensors can be difficult to integrate with PCBs that we assemble ourselves, so we attempted to look for OTS solutions/carrier boards with peripheral circuitry that are easy to integrate into custom boards. Since members of our team had experience with the EPC660 image sensor/carrier board and it had been used in similar research projects, we decided to build our system around this particular unit.

There were a couple tradeoffs regarding the split of the modules: since the modulation signals might have required communication or other components, we wanted to keep this circuitry as close to our compute unit as possible. In addition, we chose to separate out the sensing and illumination modules to ensure that we could vary the distance between the two functionalities to change the overlap of the illumination cone and the sensor FOV.

4 Design Requirements

Our use-case requirement translate into design requirements on different components of the system. In this section, we will derive these design requirements.

Many of our requirements drive component choices, and those choices affect other design requirements. Therefore, we started by choosing the components with smaller number of alternatives, most importantly the ToF sensor.

4.1 ToF Sensor IC Requirements

The ToF sensor is in many ways similar to a regular camera sensor, and shares a lot of characteristics with it.

Resolution The sensor resolution design requirement is derived from our required lateral resolution and the sensor FOV. With an HFOV of 90, the imaging plane at 40 cm is 80 cm wide. To achieve a 2.5 cm resolution at this distance, we need at least

$$\frac{80}{2.5} = 32$$

[2]This rate is limited by the throughput of CV and ML inference pipelines running on the main onboard computer.

horizontal pixels. Vertical resolution is less important becasue of how the suction tube operates.

Frame rate The sensor must support the required 10 FPS frame rate.

Modulation frequency and phase resolution A continuous wave ToF camera works by modulating the illumination light at a high frequency, and measuring the phase shift of the returned light. The phase is measured by the sensor IC for each pixel. The upper limit on the modulation frequency comes from the maximum range constraint, as the roundtrip time of the light must be less than one modulation period to avoid spatial aliasing. The lower limit comes from the range resolution. A given difference in range will correspond to smaller difference in phase at a lower frequency.

For the maximum frequency, we have

$$f_{max} = \frac{c}{k_w \cdot d_{\max}} = \frac{3 \cdot 10^8 m/s}{1.33 \cdot 2m} \approx 110 \text{ MHz},$$

where k_w is the index of refraction of water and c is the speed of light. For the minimum frequency, we assume conservatively that the sensor has 8-bit phase resolution. Then, we want

$$\frac{2\pi}{2^8} < \frac{4\pi f k_w \Delta d}{c},$$

where Δd is the range resolution. We get

$$f_{min} = \frac{c}{2 \cdot 2^8 \cdot k_w \Delta d} \approx 17 \text{ MHz}.$$

The above formulas can be easily derived by considering the spatial wavelength of the modulation wave.

ToF sensor selection We chose the EPC660 time-of-flight sensor, with $f_{modulation} = 24$ MHz and the resolution of 320 x 240 pixels.

4.2 Optics requirements

Image circle EPC660 pixel field has a diagonal of 8mm. We need a lens with an image circle around 8mm.

HFOV By use-case requirements, our lens should provide 90° horizontal field of view.

Depth of field The lens should be able to deliver sharp images of objects at 25cm to 2m range.

4.3 Illumination requirements

Brightness It is intractable to calculate the required illumination brightness theoretically. Any estimate will likely be significantly off due to dissipation effects, unknown reflectivities of target materials, unknown LED attenuation at high frequencies, thermal derating of illumination components, and numerous other factors.

Therefore, we approach this requirement empirically. We know from previous experimentation at TartanAUV that a combined 2000 lumen light source is sufficient to illuminate a scene within 1-2 m, which matches our range requirements. (This was tested in a pool with indoor lighting.)

To account for possible adverse effects of high-frequency modulation, we add a safety factor of 3, and target 6000 lumen as our peak brightness requirement.

Frequency The light source must be able to be modulated at 24 MHz.

5 Design Trade Studies

A few major subsystems of our solution required significant thought to decide between potential options. Chief among these choices include our compute, image sensor, light driver, and modulation clock source.

5.1 Compute

We considered a mix of microcontrollers and SBCs for our compute unit. Our criteria included support for parallel communication with an image sensor (as most image sensors use a similar parallel protocol), RAM size for frame buffer storage, output clock capabilities for modulation, and board footprint/ease of integration. The STM32F4, F7, H7, and H5 series were considered for MCUs, and the Nvidia Jetson Nano was considered for SBCs. Ultimately, we chose the STM32H563 as it had the highest clock output modulation combined with the smallest footprint. The SBC was ruled out as it did not have parallel communication support and was difficult to integrate into a single PCB module, as well as not having a designated timer output. A design matrix of the options is shown below:

Table 1: Compute Design Matrix

Compute	RAM	Clock (MHz)	Integration	Comms.
H563	640 KB	**250**	**Easy**	**Yes**
H725	640 KB	**550**	Medium	**Yes**
F413	320 KB	96	**Easy**	**Yes**
Jetson	**4 GB**	≥ 1000	Difficult	No

5.2 Light Driver

For our light driver, we considered two main options: a high speed MOSFET driver like the TPS2816 or a dedicated LED driver chip like the EPC21603. We considered a few different factors: peripheral circuitry, modulation speed, and ease of integration onto the PCB. The EPC21603 is more difficult to integrate due to its BGA solder package, but it can handle modulation frequencies of up to 100 MHz, while the TPS2816 can only handle up to 40 MHz. In addition, the EPC21603 takes in a differential clock signal directly. As discussed later in the System Implementation section, a differential clock is crucial for signal integrity when travelling between boards. The EPC21603 also does not require any external MOSFETS, which relieved some part selection burden on us. Ultimately, we ended up going with the EPC21603 to fully meet our technical requirements. A design matrix of the options is shown below:

Table 2: Light Driver Design Matrix

Light Driver	Circuitry	Integration	Modulation
EPC21603	**Low**	Difficult	**96 MHz**
TPS2816	High	**Easy**	40 MHz

5.3 Clock Source

We had two main options when considering the modulation clock source: either generating it off of the STM32, or trying to find an off-the-shelf chip to generate the clock. We did a lot of experimentation with both the STM32 TIM and MCO functionalities to output clock signals on the GPIO pins of the MCU. However, we found that at the speeds we needed (up to 96 MHz for the EPC660), the clock division capabilities of the STM32 are severely limited due to the limitations on the ARR register values, which made it difficult to precisely control phase offsets on the two different clocks if necessary on the STM32. Thus, we ended up overwhelmingly deciding on an external clock chip, and we were able to find powerful parts that allow for very fast, 96 MHz speeds and picosecond-precision phase offsets, more than good enough for our application.

6 System Implementation

See the end of this report for a large, system level block diagram.

6.1 Image Sensor Module

The image sensor PCB will consist of all of the peripheral hardware necessary to communicate with our chosen image sensor, the EPC660, and transmit the obtained frames to the outside world. The EPC660 has a 320x240 range and can support up to 78 FPS output, which is more than enough to meet our design requirements. The STM32H563 compute unit lives on this board along with all of its decoupling and oscillator circuitry. The STM32 communicates with the image sensor carrier board through the DCMI protocol, which is a parallel, 12-wire data bus with sync lines. All of the image sensor interface lines run through a 60-pin right angle card edge connector that is compatible with the carrier board. In addition, the modulation clock frequencies are generated on this board by the Si5338Q chip. This chip communicates through I2C with the STM32. Two in-phase clock signals are generated from this chip, one for the EPC660 and one for the LED modulation on the other board. The EPC660 clock signal is single ended, as it goes directly to the image sensor through the edge connector. The LED modulation clock signal is differential, as it travels quite a bit farther to the LED driver PCB. This clock will run through a differential RJ45 jack for signal integrity purposes to the other LED Driver. Finally, there is an Ethernet PHY chip and another RJ45 connector on this board to communicate with the TartanAUV Osprey submersible. 10V, 5V, and 3V3 power for the STM32 and EPC660 will be derived from the submarine and routed through a Molex Nanofit connector.

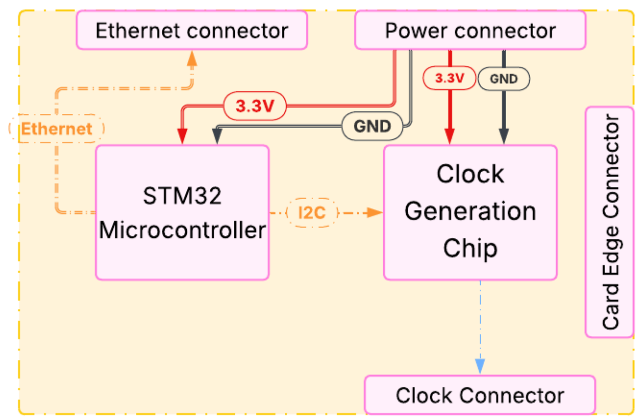

Figure 6: Sense PCB Diagram

6.2 LED Driver Module

The LED driver PCB takes in the differential clock signal generated from the image sensor PCB. We utilize a high-speed, low-side LED driver chip, the EPC21603, specially designed for LIDAR and ToF applications. This board will take in a 12V input voltage through a Nanofit connector and pass this voltage into a constant current driver to ensure stable illumination of the LED. We will follow TI application notes on the use of passive components and routing for these sensitive modulation circuits. The EPC21603 will switch the output of the constant current signal, and it will be driven by the differential clock signal from the sense PCB.

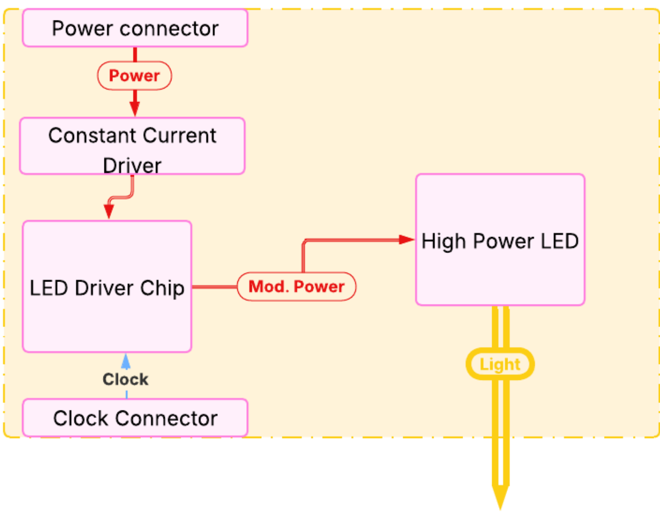

Figure 7: LED Driver Diagram

6.3 Power Distribution Module

While the TartanAUV Osprey can provide all of the power supplies we need, a stretch goal is to manufacture a power board that can decouple our system from the submarine. The power module will have multiple on-board buck converters to generate the 12V, 10V, 5V, and 3V3 signals we need for both the LED driver and the sensor board. It will route all of these power supplies out through a Nanofit connector as well. It is important to note that this module is not currently part of our MVP, and the other boards will still function completely with power from the sub.

6.4 Firmware

Our firmware implementation for the STM32 is based on interrupts and direct memory access (DMA). On boot, we will initialize the EPC660 and Si5338Q chips through I2C commands, starting the modulation clock. Then, we will link the DMA controller and set up interrupts for the DCMI peripheral. We will also set up the Ethernet peripheral with a UDP client. Most of the actual code logic will be event-driven, as this frees up CPU time and allows for quick responses to communication. The DCMI interrupt will trigger the DMA controller to transfer a frame into SRAM. Using DMA means that the CPU will not be involved with memory transfer at all, allowing it to focus on the Ethernet transmits. Each frame from the EPC660 is 320x240 pixels with 12-bit resolution, yielding a frame size of 115 kB. The STM32H563 has an SRAM size of 640 kB, which allows for a buffer of 4 frames, as this is the maximum amount of frames we can hold while accounting for program size. We will also have a "half-full" buffer interrupt that will trigger the Ethernet transmits. Thus, we can be transmitting out the top half of the buffer while the DMA controller writes frames to the bottom half of the buffer simultaneously, enabling fast and efficient data transfers.

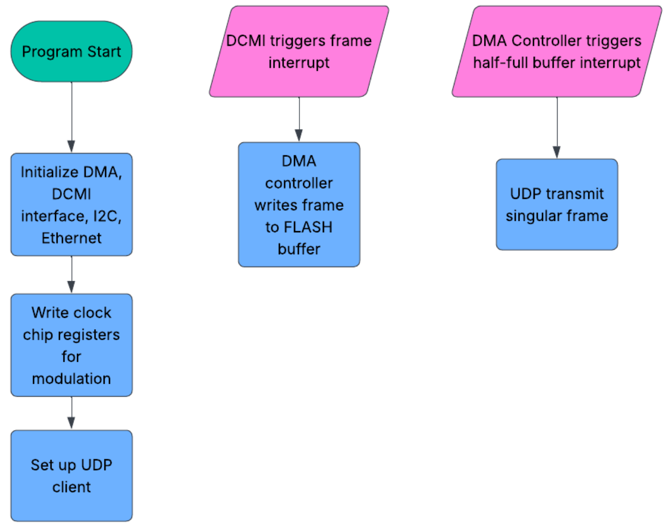

Figure 8: Firmware Flowchart

6.5 Mechanical Integration

We plan on using an off-the-shelf enclosure, bandpass filter, and diffuser for the PCBs. Each of the PCBs will be housed in a waterproof Blue Robotics tube with a filter and diffuser attached. The PCBs will be mounted to a cut 2x4 fixture inside the tube to make sure that the distance between them can be varied easily.

7 Test & Validation

7.1 Unit/Functionality Testing

Before we can fully test our design goals, we have a comprehensive set of unit tests that will ensure that each component functions properly on its own. Firstly, we plan to power and short test each PCB by measuring resistance between every power line and ground using a multimeter. In addition, we plan on hooking up each line to a power supply individually and making sure that current limits are not exceeded. To individually test firmware, we have bought an STM32H563 Nucleo breakout board to unit test the firmware. Since the board comes with an Ethernet PHY interface, we can use it to validate the output data format and functionality. Finally, to unit test the LED modulation, we will use a discrete photodiode connected to a separate Nucleo board. This Nucleo will be polling the photodiode at 2x the modulation frequency of the LED (by Nyquist) and the resulting photodiode values will be plotted. We will then be able to see how "square" the modulation of the LED actually is, and determine any rise-time delays on the output signal.

7.2 Tests for FPS Requirement

To ensure that our system's throughput is sufficient to meet the TartanAUV vision task frequencies, we plan on conducting a large-scale stress test over a 2 hour period. We will write a barebones ROS driver on the TartanAUV Osprey Jetson compute to ingest frames from Ethernet/UDP. We will then average throughput over the whole 2 hour test period and ensure that we get ¿=10 frames per second. In addition, we will also ensure that the throughput never drops below 5 FPS over a 30 second period, as this will ensure there are no temporary buffering issues that could interfere with the submarine for a short period of time.

7.3 Tests for Range Requirement

To ensure that the submarine can construct a map of its surroundings well enough, we want to test the maximum range of the sensor and the consistency of these range measurements. We also care about the mapping between raw sensor output and actual distance, as we want to make sure this is a linear relationship to accurately reflect the real value. We will place an object 5 cm x 5 cm (since this is the size of the smallest objects in the manipulation task for the sub) at different distances up to 2 meters in front of the camera. We will then plot the average pixel intensity for this object vs the real distance, and ensure this is a linear fit. We will repeat this test 5 times to ensure reliability. In addition, we will run the same 5 trials in a dark, non ambient light environment to demonstrate consistency in different environments.

7.4 Tests for Resolution and FOV Requirements

To test resolution, we want to ensure that the smallest object (5 cm x 5 cm) required for manipulation can be seen at a 2 m range. To ensure that these objects can be identified, we will require that the object shows up as at least a 2x2 grid of pixels when situated 2 meters away from the sensor. The object will be placed against a flat, constant depth background to limit any noise around it. We will also move the object around horizontal range of 80 cm while it is 40 cm away from the sensor to ensure that we hit the 90 degree FOV requirement.

7.5 Tests for Waterproofing

Once we place the PCBs inside their waterproof enclosure and the optics are fixured, we will take the assembly to the Robotics Innovation Center in Hazelwood, which has a tank meant for AUV operation. We will place the assembly in the tank and ensure that no water enters the tank. We will also place an object in front of the assembly to ensure that the distance values from the sensors change.

8 Project Management

8.1 Schedule

The schedule is shown in Fig. 10.

8.2 Team Member Responsibilities

We split team member responsibilities broadly based on our previous experience and the large subcomponents of our whole assembly. Gleb is working on the analog driver PCB for the excitation LED due to his previous experience in high frequency switching power design. Claire is designing the digital logic and sensor PCB to interface with the EPC660 given her past experience with STM32 boards, and Sid is writing the firmware for DCMI and Ethernet interfacing. Unit testing of each of the subsystems is the responsibility of the person responsible for it, while integration testing and mechanical integration is a collective responsibility.

8.3 Bill of Materials and Budget

Table 3: Bill of Materials

Item	Price	Source
EPC660-CC Carrier Board	$ 92	purchased
EPC21603 LED Driver	$ 5 x 4	purchased
Passive Components	$ 0	TartanAUV supply
TPS92515 LED Driver	$ 4 x 4	purchased
PCB manufacturing	$ 50	purchased
Waterproof housing	$ 0	TartanAUV supply
Cree XPE-GR LED	$ 2 x 12	purchased
CIL height		

8.4

TechSpark Usage Plans Since we have access to both the TartanAUV and Carnegie Mellon Racing workspaces, our usage of Techspark will be minimal, as we can solder and rework the boards in our shops. We may use Techspark in case we need to examine the BGA packages on our LED driver chip with a PCB X-Ray in case there is a short or other issue during reflow.

8.5 Risk Mitigation Plans

A large risk factor that our team needs to deal with is the optical interplay of the modulation and sensing elements. This was one of the main reasons we split the modulation and sensing components into two different PCBs, as this allows us to empirically test the optimal distance between the FOV of the illumination and the FOV of the image sensor. In addition, there is an inherent risk associated with routing high frequency signals like our modulation clock off board. To remedy this, we are using a known physical interface in the form of a CAT-5 Ethernet cable + connector with twisted-pair wiring, as this standard can handle switching signals of much higher frequency than our modulation clock.

9 Related Work

There has been some research related to visible-light ToFs, and is some of where we got our inspiration for this project. There was an IEEE Oceans Conference paper that used this image sensor unit that we are using. However, their design was very clunky and built around existing OTS compute units, so we wanted to build a similar, yet simpler system from the gorund up.

10 Summary

In summary, The Illuminator is a low-cost, close-range, underwater time-of-flight (ToF) imaging solution designed for efficient and reliable operation in challenging aquatic environments. It will offer a sensing range of over meters and a frame rate exceeding 10 frames per second. The design incorporates visible light illumination and supports a 10 Mbps Ethernet interface with DMA and flash buffering for smooth data handling. Its modular design and waterproofing will enable it to be an effective solution for underwater robotics.

Glossary of Acronyms

- AUV - Autonomous Underwater Vehicle
- DCMI – Digital Camera Messaging Interface
- DMA – Direct Memory Access
- FOV – Field of View

References

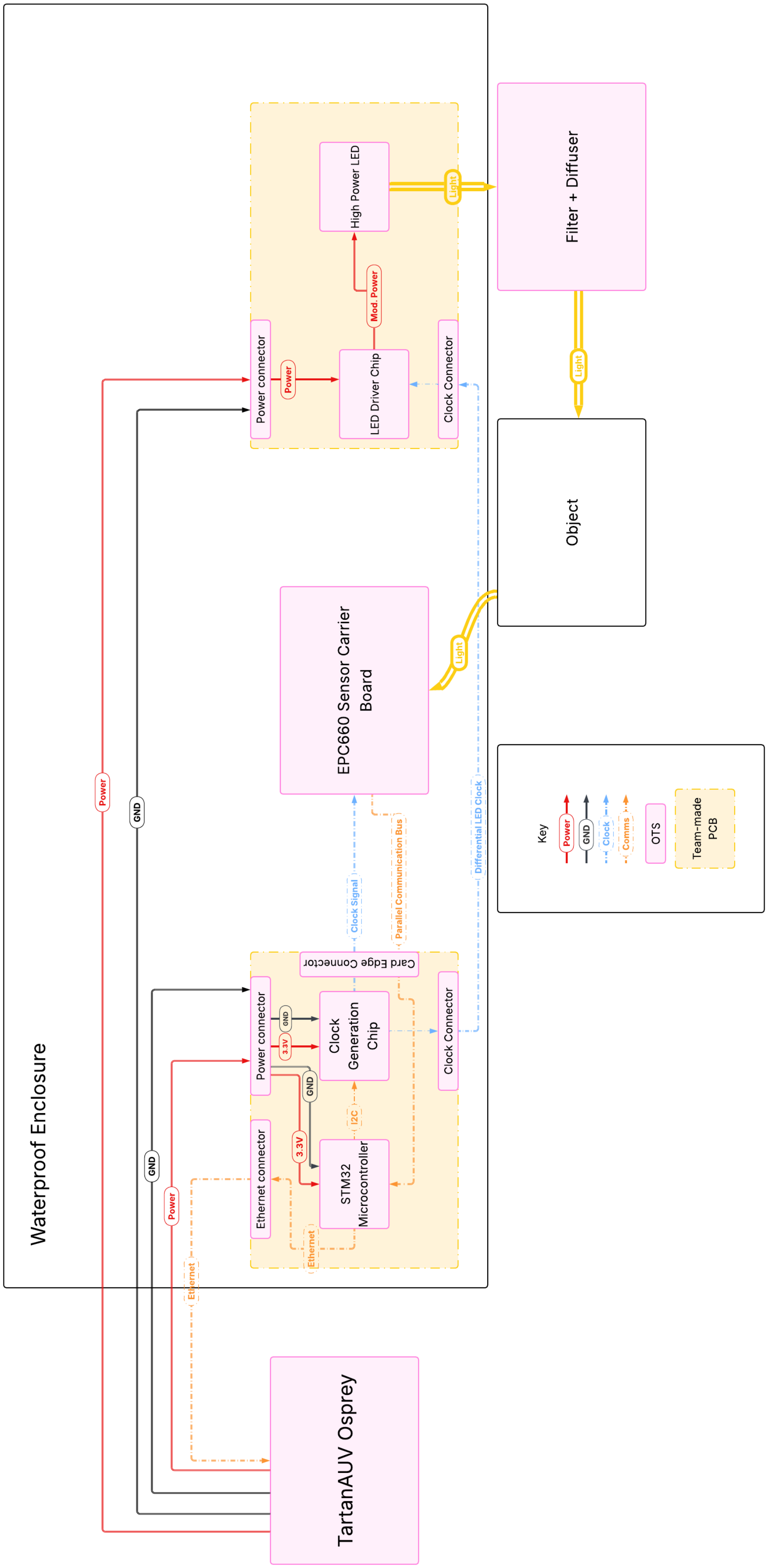

Figure 9: Overall System Block Diagram

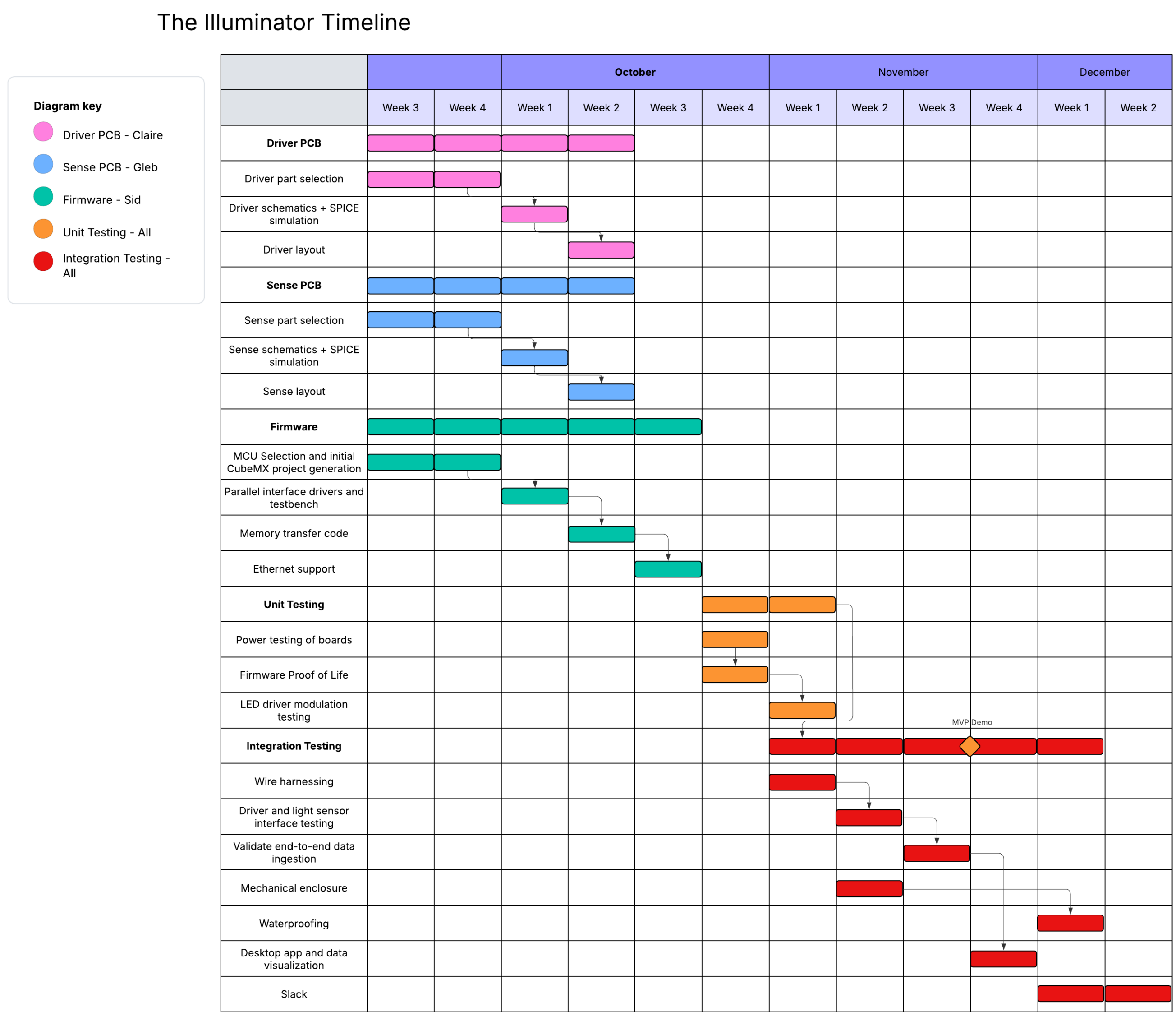

Figure 10: Gantt Chart