Lidar-Based Physical Security Sensor

Calvin Lindberg
Elijah Orozco
Michael Yoon

CONCEPT OF OPERATIONS

REVISION – 2
5 October 2025

Concept of Operations
for
Lidar-Based Physical Security Sensor

Team <09>

Approved by:

____
Project Leader Date

____
Prof. Kalafatis Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	9/17/2025	Calvin Lindberg		Draft Release
2	10/5/2025	Michael Yoon		Updated Figures

Table of Contents

List of Figures

1. Executive Summary

Sandia National Laboratories is a laboratory that specializes in technology that is meant to support U.S national security. The laboratory has asked to develop a detection system that can be used to guard an area from intrusion. Current technologies such as image detection, infrared detection, and ultrasonic wave detection have some limitations. The sponsor has asked to develop a solution that utilizes lidar to mitigate these limitations. We intend to develop a system that will use LiDAR time-of-flight technology to detect movement and signal an alarm when motion is detected. The system will create a 3D point-cloud environment to represent the spatial configuration of an entire room. The system will also meet the specifications of high physical reliability, low false alarm rate, low power demands, and be robust against defeat methods. The goal of the project is to provide the Sandia National Laboratories with a working prototype which Sandia engineers can continue to work upon.

2. Introduction

This report details the intended scope and function of a LiDAR-Based Physical Security Sensor (LBPSS), sponsored by Sandia National Laboratories. LiDAR stands for Light Detection and Ranging and uses the Time-of-Flight (ToF) technology to measure the distance between the sensor and nearby objects in its line of sight. The sensor will be used for high consequence applications and will therefore be robust and capable of lasting maintenance free for at least 10 years. Additionally, the False Alarm Rate (FAR) will be kept as low as reasonably possible. Upon detecting motion, the LBPSS will output a simple Boolean signal indicating an intrusion.

2.1. Background

As demonstrated by our sponsor, Sandia National Laboratories, current implementations of security systems have some limitations. The capabilities of traditional cameras are significantly impeded in the dark. Even if they are in bright environments, these cameras generally need human monitoring which can be costly. To eliminate the need for human intervention, some cameras have been connected to smart image processing algorithms to guard certain regions of the border, but these have been defeated by miliary professionals using rudimentary methods such as doing cartwheels or hiding behind bushes. Ultrasonic detection systems can be bypassed simply by moving at an extremely slow pace. Infrared detection cameras seem useful for security in dark environments, but an individual can simply bypass them by masking their own thermal signature through clothing or shielding. The proposed solution to these weaknesses that we will be demonstrating with the LBPSS is the use of LiDAR. LiDAR operates by measuring the time of flight that it takes for a light signal to travel from the sensor to an object and back. Using this data, one can create a 3D mapping of a room—called a 3D point cloud—and use any measured disturbances in this mapping to detect intruders. This approach guarantees no reduction of efficacy because of dark environments, and it cannot be beaten by hiding behind physical objects. Additionally, solid state LiDAR sensors have the potential to draw very small amounts of power, making battery implementation feasible.

2.2. Overview

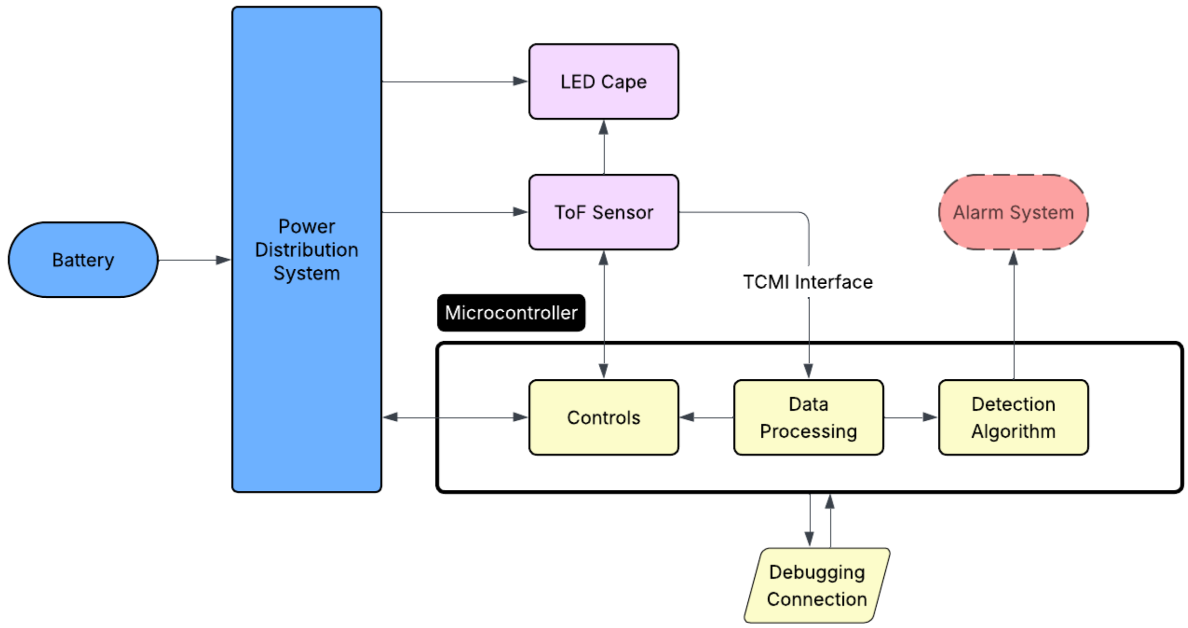

Figure 1: LBPSS System Block Diagram

Our system will be generally composed of three major components, those being the power system, the LiDAR Time of Flight sensor, and the microcontroller. The entire device will be battery powered, and it will be necessary to implement multiple DC to DC converters to supply the proper voltages to the LiDAR sensor and microcontroller. The microcontroller will be the primary control device for the entire system and will communicate with all other subsystems. The node exiting the detection algorithm in the microcontroller will be the main external point of communication with the greater security system in which the LBPSS is contained.

2.3. Referenced Documents and Standards

Document Number	Revision/Release Date	Document Title
Datasheet_epc660-V2.20	2023	Datasheet-epc660 3D TOF imager 320x240 pixel
Datasheet_epc660-CC-Carrier-V1.10	2018	DATASHEET-epc660 cc 3D TOF imager 320x240 pixel on mounting carrier
Datasheet_DME_660-V1.06	2018	DATASHEET-DME 660-108º/10m Distance measurement camera engine
DS12110 Rev 10	March 2023	STM32H742xI/G STM32H743xI/G

3. Operating Concept

3.1. Scope

The LBPSS will be corner mounted in a room and will have a complete 108° by 77° field of view. It will be capable of monitoring a small to medium sized room by providing accurate data of any motion up to ten meters radially outwards from the sensor. The raw data provided by the LiDAR sensor will be captured by a microcontroller and processed to eliminate noise to increase the robustness of the system and decrease the likelihood of false alarms. If movement is detected, the LBPSS will activate an alarm signal that can be used to activate a separate alarm system. This separate alarm system is outside the scope of this project and will not be addressed.

3.2. Operational Description and Constraints

The LBPSS will be designed to operate fully autonomously without the need for human intervention; therefore, it will be capable of functioning maintenance free (excluding battery replacements) for at least ten years. The battery will allow the LBPSS to operate for a year without replacement. Since the sensor is Time-of-Flight based, it will also be capable of operating in the dark without negatively affecting the performance. The nature of ToF sensing, however, creates certain technical constraints. Firstly, the room in which the LBPSS is installed must have relatively reflective surfaces. Objects that are highly nonreflective will fail to be recognized by the sensor unless at a short range. Secondly, it will not be designed to operate outdoors since changing sunlight can affect the accuracy of the sensor. Lastly, the sensor will have a maximum distance rating of 10m and a specified field of view that must be observed.

3.3. System Description

The *Power Distribution System* will receive power from the battery and will perform necessary voltage and current transformations or regulations to supply both the LiDAR sensor as well as the microcontroller unit. The *LiDAR* sensor will emit infrared light and sense the reflection off nearby objects, measuring the time between emission and detection. The *Data Processing* system will receive raw data from the LiDAR sensor, cleaning and filtering it to remove outliers and noise to prevent any false alarms. It will then digitally format the data into a 3D point cloud representation of the room. This module will feed data into both the power controls and the detection algorithm systems. The *Power Controls* are essential for achieving the desired one-year battery life. If no disturbances are detected, this system will signal the microcontroller unit to enter a low-powered sleep mode. Additionally, it will monitor the current battery health for the LBPSS. The *Detection Algorithm* will receive the processed data and compare it to reference data. If the 3D readings of the room experience enough of a disturbance compared to readings when the room is known to be secure, then this system will activate the alarm signal. The *Alarm System* is not in the scope of this project, and it will be the user's responsibility to implement an alarm that is suitable for the situation.

3.4. Modes of Operations

The LBPSS by itself will only have two primary modes of operation: armed and disarmed. Once armed, any detection of motion will immediately trigger the alarm signal. It will continue to operate even after an alarm has been signaled until the user disarms the system. Once disarmed, the LBPSS will be completely removed from power and inactive until armed again. Additionally, the LBPSS will have a low power mode that will be automatically enabled should the battery charge fall below a certain point. This mode would decrease framerate and potentially the LED illumination power (depending on configuration) to enable the LBPSS to conserve power. It is important to note, however, that the LBPSS is only a sensor and is not intended to be used in place of a standalone security system. Instead, it should be one component in the larger security system as a whole. This may include more than one LBPSS or even other security sensors.

3.5. Users

Since the LBPSS is only a sensor and not an isolated system, the types of users may vary depending on how the sensor is implemented. Initially the LBPSS will be used by security and tech personnel who will be responsible for installing the sensor and making sure that it is fully integrated with the broader security system as a whole. This could involve having it connected to a clock that arms the system after hours each day. In this case, after installation, the LBPSS will be generally automated, not requiring intervention except by maintenance teams and security personnel in the case of irregular operation. On the other hand, the LBPSS could be configured to operate manually only, in which case authorized personnel would be responsible for arming and disarming the sensor as needed.

3.6. Support

The files of pcb design and code base will be provided so that Sandia engineers can iterate on the design. Other supporting documentation will also be provided. This includes flow diagrams, code structures, and documentation used to implement the design. Collateral needed for the operation of the device will also be provided.

4. Scenario(s)

4.1. Equipment Storage

The lidar system could be installed in areas with equipment or materials that should be only accessible during specific times or by certain people. The use of the system can help prevent theft, misuse, or damage of harmful and expensive equipment or materials. Since this system can be used in dark environments, it would perform well in areas such as these where there is not normally a constant source of light. Automatic detection can save time and money by eliminating the need for constant surveillance in an area that normally has very low activity.

4.2 Research Laboratory Monitoring

Due to the sensitive nature of a laboratory environment, a highly effective security system is required. Lidar could be used in electronic, biomedical, or chemical labs to increase security and safety. The primary use for the lidar system would be after hours when there is little to no normal activity. It could be used to detect break-ins, spills, or any movement caused by materials or equipment that were not properly stored or powered off. This system could be especially useful in labs where cameras could create privacy concerns.

4.3 Server and IT Rooms

The lidar detection system can be utilized in server rooms and IT closets to monitor any suspicious or unauthorized movement. Spaces like these are often critical to facility operation and need to have a very high level of security. The 8–10-meter range of the sensor will likely be more than enough for most of these rooms and since the power system is locally contained, it can be installed without altering existing cabling or infrastructure. Also, since many of these spaces are dimly lit, lidar is an excellent solution to provide high levels of security at all times.

5. Analysis

5.1. Summary of Proposed Improvements

- The LBPSS will be resistant to all traditional methods of avoiding detection and will not lose effectiveness in dark environments
- The system will scan at a frequency of 2-8 Hz to balance frequent monitoring with extended battery life.
- The LBPSS will be optimized for use at a range of 8-10m with a field of view of 108° by 77°.
- All data processing will be locally performed with efficient change-detection algorithms.
- The low power demand will allow the entire system to be locally powered by a battery sized for roughly 1 year of continuous operation.

5.2. Disadvantages and Limitations

The system proposed will have some drawbacks:

- Lidar technology tends to be much more expensive than most other security systems such as ultrasonic or cameras.
- The current design will not provide information on object classification.
- Surfaces that are highly reflective, transparent, or absorbent such as glass, mirrors, or matte black surfaces can interfere with lidar and cause inaccurate readings.

5.3. Alternatives

Some alternatives to the proposed lidar detection system are:

- Other forms of motion detection such as infrared or ultrasonic
- Camera-based system with image processing/machine learning for motion detection
- Using Wi-Fi or Bluetooth for communication to be able to connect to the system rather than a hardwired connection.
- Using a system that has a battery backup, but relies on building power for most normal operations
- Utilizing pressure sensors in the monitored space or staff to patrol the area

5.4. Impact

- The relatively small scale and low power consumption of this project will have a very small environmental impact.
- The system will help improve the safety and security of areas that it is monitoring.
- The automated nature of the design will reduce operational costs and decrease the need for staffing.
- Since point cloud detection does not capture images like a camera, there is little concern about privacy.
- All testing done for the system should be conducted in a way that is ethical, especially if done in a public area or involves participants who are not on the project team.

Lidar-Based Physical Security Sensor

Calvin Lindberg
Elijah Orozco
Michael Yoon

FUNCTIONAL SYSTEM REQUIREMENTS

REVISION –
4 October 2025

Functional System Requirements
for
Lidar-Based Physical Security Sensor

Team <09>

Approved by:

____
Project Leader Date

____
Prof. S. Kalafatis Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	10-4-2025	Elijah Orozco		Draft Release
1	11-5-2025	Calvin Lindberg		Subsystem Update

Table of Contents

List of Figures

1. Introduction

1.1. Purpose and Scope

Our sponsor Sandia National Laboratories has observed that current implementations of security systems have some limitations. The proposed solution to these weaknesses that we will be demonstrating with the LBPSS is the use of LiDAR. LiDAR operates by measuring the time of flight that it takes for a light signal to travel from the sensor to an object and back. Using this data, we can create a three-dimensional point cloud which can be used to determine changes in the environment. We will be achieving this with a time-of-flight (TOF) sensor, an illumination module, and a microcontroller.

Figure 1. Project Conceptual Image. Reproduced from "DATASHEET-DME 660-108º/10m Distance measurement camera engine" by Espros Photonics Corporation

We will be basing our design on the existing DME 660 shown above in *Figure 1*. We will recreate the illumination infrastructure and power distribution of the DME 660, but our design will implement a different computing/data processing unit to achieve low power specifications.

1.2. Responsibility and Change Authority

The team leader, Elijah Orozco, will be responsible for making sure the requirements are met. The requirements can only be changed with the approval of the team leader and Matthew McDonald.

Subsystem	**Responsibility**
TOF Test Board	Elijah Orozco
Embedded Boot Code	Elijah Orozco
Power System	Calvin Lindberg
Lidar and TOF Design	Micheal Yoon

2. Reference Documents

2.1. Reference Documents

The following documents are reference documents utilized in the development of this specification. These documents do not form a part of this specification and are not controlled by their reference herein.

Document Number	**Revision/Release Date**	**Document Title**
Datasheet_epc660-V2.20	2023	Datasheet-epc660 3D TOF imager 320x240 pixel
Datasheet_epc660-CC-Carrier-V1.10	2018	DATASHEET-epc660 cc 3D TOF imager 320x240 pixel on mounting carrier
Datasheet_DME_660-V1.06	2018	DATASHEET-DME 660-108º/10m Distance measurement camera engine
DS12110 Rev 10	March 2023	STM32H742xI/G STM32H743xI/G

2.2. Order of Precedence

In the event of a conflict between this Functional System Requirements (FSR) document and any other applicable document, the requirements of this FSR shall take precedence. All documents identified as *applicable* in this FSR are incorporated by reference. References contained within applicable documents are provided for information and guidance only, unless they are Interface Control Documents (ICDs), which are considered applicable and incorporated by reference.

3. Requirements

3.1. System Definition

The LiDAR-Based Physical Security Sensor will be a highly reliable security camera with a low false alarm rate and minimal maintenance requirements. It will operate using Time-of-Flight technology, allowing it to capture depth data of a room. This depth data will be used to detect any physical disturbances, and the LBPSS will signal an alarm if any are detected. The camera will also be configurable to provide different imaging data that could be used to recreate a 3D point cloud image, a 2D depth map, or a 2D greyscale image.

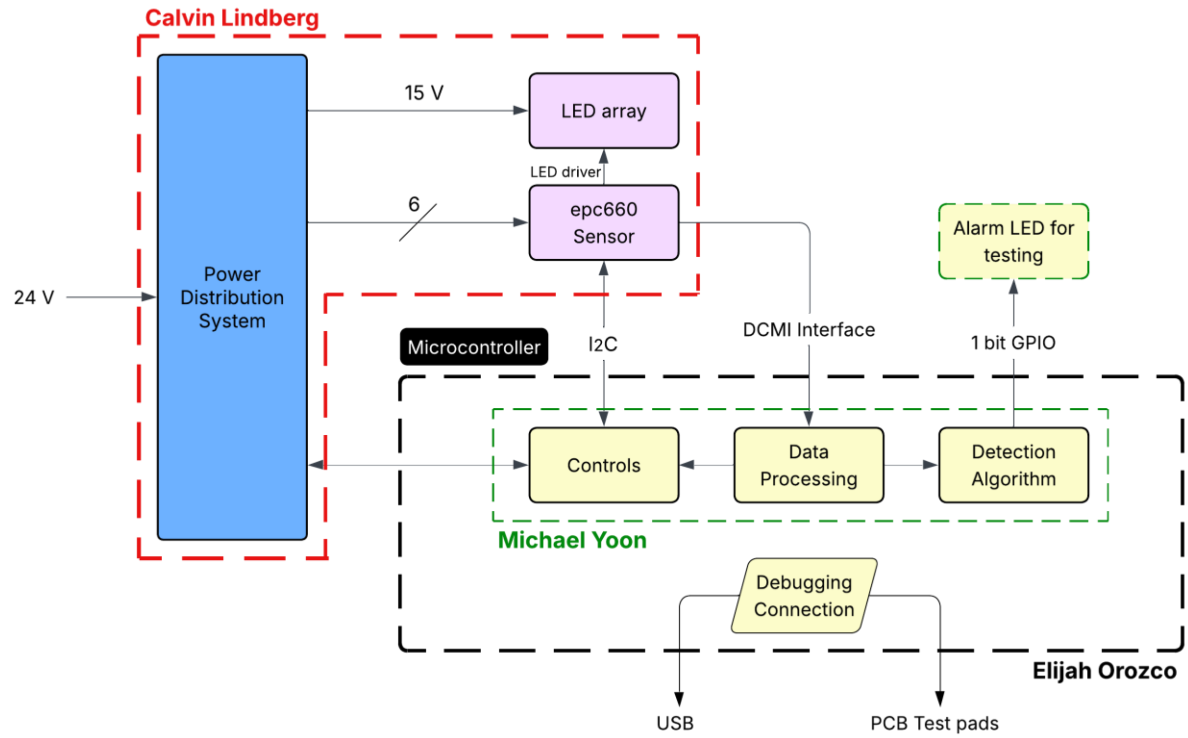

Figure 2. Block Diagram of System

The LBPSS will be comprised of three primary subsystems. Those include the power/LED subsystem, the microcontroller subsystem, and the software subsystem.

The Power and LED Subsystem is responsible for generating the regulated voltage rails required by the epc660, the MCU, and the LED driver circuit. This subsystem converts the 24 VDC input into the five stable power rails specified by the epc660 datasheet, ensuring that each rail meets the necessary tolerance and ripple requirements for reliable sensor operation. In addition to supplying power, the subsystem includes the external LED driver circuit used to generate the illumination needed for depth imaging. The epc660 provides control signals that drive this LED circuit, enabling synchronized light output during frame capture. Together, the regulated power rails and LED driver allow the epc660 to operate within specification, produce clear images, and support the processing and communication tasks handled by the other subsystems.

The microcontroller subsystem will be essential for providing control of the sensing chip, data transfer, data processing, security output signals, and an interface for debugging. The microcontroller (MCU) will provide I2C writes to the register map of the epc660. This will allow for the sensor's configuration. The data transfer aspect will be handled by the MCU's digital camera module interface (DCMI). This will serve as a parallel bus in which the MCU can take in the data coming from the epc660. There will also be GPIO pins for some controlling of other peripherals. The next part will be data processing. The MCU will process the data so that a 3D point cloud can be generated from the frames that the epc660 provides. From this point the security algorithm can detect changes in the cloud and provide security output signals. The subsystem will also allow for debugging of the MCU with an external connection to a computer.

The software subsystem will integrate all the hardware from the microcontroller, power, and LED subsystems. It will be comprised primarily of the firmware on the microcontroller as well as python scripts written on the host PC. Through C code written on the MCU, the epc660 will receive proper voltage and current supplies from the power subsystem as well as essential control signals from the microcontroller. These signals will control the epc660's integration time, modulation frequency, data readout, shutter, and many other settings. During integration time, the sensing chip will simultaneously provide the modulation frequency for the LEDs as well as the demodulation signal for the pixel array. After completing one differential correlation sample (DCS) frame, pixel data will be transmitted to the microcontroller through the parallel 12-pin ToF Camera Module Interface (TCMI) connection at 48MHz. After four DCS samples have been collected, the MCU will process them into one depth frame, calculate if there was any displacement (or intruders), then send the completed depth data to the host PC for display. On the host PC side, a python script will receive the depth frame, make sure that it is aligned, then display it on the screen using a color scale to represent depth.

3.2. Characteristics

3.2.1. Functional / Performance Requirements

3.2.1.1. Maximum Sensing Range

The LBPSS shall have a maximum sensing range of not less than 6.25 meters and may be capable of sensing objects up to 12.5 meters depending on configuration settings, ambient light, and reflectivity of surfaces in range.

> *Rationale: A 6.25 meter range (~20.5 feet) when paired with a horizontal field-of-view of 108 degrees and proper placement will allow the LBPSS to monitor a 4.4 by 4.4 meter room (~14.5 by 14.5 feet). This should accommodate many practical applications for a LiDAR security sensor.*

3.2.1.2. Frame Rate

The LBPSS shall have a configurable frame rate of 1 – 10 frames per second. Higher frame rates will necessarily cause higher power consumption and decreased battery life for the system.

> *Rationale: Depending on the specific application of the LBPSS, a higher frame rate may be necessary to ensure security. On the other hand, a lower frame rate may be necessary to ensure a longer battery life. It is therefore important to allow the user to choose among a range of frame rates.*

3.2.1.3. Low False Alarm Rate

Given proper installation and usage, the false alarm rate of the LBPSS will be less than one false alarm per year.

Rationale: For the high consequence applications that Sandia Labs may utilize the LBPSS for, false alarms will be very costly and disruptive, therefore they shall be limited to as low as reasonably possible.

3.2.1.4. Resistance to Common Defeat Methods

The LBPSS shall be resistant to common defeat methods that are typical to camera sensors.

Rationale: For cameras mounted in rooms there cannot be simple hacks that intruders can use in order to bypass the security system. These hacks include moving at a slow speed, placing a box/cover in front of the camera etc....

3.2.2. Physical Characteristics

3.2.2.1. Volume Envelope

The volume envelope of the senor shall not exceed dimensions where mounting the device becomes unfeasible.

Rationale: This is a requirement specified by our customer due to constraints of their system in which the Search and Rescue System is integrating.

3.2.2.2. Mounting

The device may be mounted to a the conner of a wall and have access to an outlet.

Rationale: The placement of the sensor in the conner of a wall will provide the greatest amount of area to be monitored. The access to an outlet will ensure that power is available for situations where the battery in not required.

3.2.3. Electrical Characteristics

3.2.3.1. Inputs

a. The presence or absence of any combination of the input signals in accordance with ICD specifications applied in any sequence shall not damage the LBPSS, reduce its life expectancy, or cause any malfunction, either when the unit is powered or when it is not.

b. No sequence of command shall damage the LBPSS, reduce its life expectancy, or cause any malfunction.

Rationale: By design, should limit the chance of damage or malfunction by user/technician error.

3.2.3.1.1 Power Consumption

a. The maximum peak power of the system shall not exceed 6 watts. This value may be significantly lower once the system is running at the desired frame rate and integration time.

Rationale: Low power consumption was a secondary requirement that was desired by the sponsor. This will not be a focus during the design of individual subsystems but will be a concern in later into the project.

3.2.3.1.2 Input Voltage Level

The input voltage level for the Lidar security system shall be +24 VDC.

Rationale: Specified input voltage by the evaluation board.

3.2.3.1.3 Noise and Ripple Input

The LBPSS shall operate to specification with up to 0.25 Vpp ripple superimposed on the input, over the frequency range DC to the maximum practical AC test frequency, measured at the system input terminals.

Rationale: Some ripple should be tolerated, but a steady voltage source is needed for operation

3.2.3.1.4 External Commands

The LBPSS shall document all external commands in the appropriate ICD.

Rationale: The ICD will capture all interface details from the low level electrical to the high-level packet format.

3.2.3.2. Outputs

3.2.3.2.1 Data Output

The LBPSS shall include an interface compatible with the data system.

Rationale: The LBPSS information passes directly to the customer's system.

3.2.3.2.2 Diagnostic Output

The LBPSS shall include a diagnostic interface for control and data logging.

Rationale: Provides the ability to control things for debugging manually and a way to view/download the node map with associated potential targets.

3.2.3.2.3 Raw TOF Video Output

The LBPSS shall include an interface that allows users to view the raw time-of-flight (ToF) video data in real time. This interface will provide a visual representation of the sensor output for testing and analysis purposes. It will be designed to support data inspection, debugging, and performance evaluation within the laboratory environment.

Rationale: Will be useful for troubleshooting and inspecting quality of output.

3.2.3.3. External Connectors

The LBPSS shall have an Ethernet and micro-USB port to interface with the system

Rationale: These are the connections supported by the NUCLEO-H743ZI2.

3.2.4. Environmental Requirements

The LBPSS shall be designed to operate in indoor laboratory environments with controlled temperature and humidity. It should operate reliably under ambient temperatures between 20°C and 25°C and relative humidity below 60%, without exposure to excess dust, moisture, or direct sunlight. The system is not designed for outdoor or harsh environmental conditions.

Rationale: Our client indicated eventual use in outdoor scenarios but stated that the first goal is to make an indoor design.

3.2.5. Failure Propagation

The LBPSS is isolated from all other systems and will not allow for failure of the system of cause any external system to fail.

4. Support Requirements

The files of pcb design and code base will be provided so that Sandia engineers can iterate on the design. Other supporting documentation will also be provided. This includes flow diagrams, code structures, and documentation used to implement the design. Collateral needed for the operation of the device will also be provided

Appendix A: Acronyms and Abbreviations

AC	Alternating Current
DC	Direct Current
DCMI	Digital Camera Module Interface
DCS	Electromagnetic Compatibility
EMI	Electromagnetic Interference
I2C	Inter-Integrated Circuit
ICD	Interface Control Document
LED	Light Emitting Diode
MCU	Microcontroller Unit
MHz	Megahertz
PCB	Printed Circuit Board
TCMI	Time-of-flight Camera Module Interface
TOF	Time-of-flight
VDC	Volts Direct Current
VPP	Volts Peak-to-Peak

Lidar-Based Physical Security Sensor

Calvin Lindberg
Elijah Orozco
Michael Yoon

INTERFACE CONTROL DOCUMENT

REVISION – Draft
25 January 2018

INTERFACE CONTROL DOCUMENT
FOR
Lidar-Based Physical Security Sensor

TEAM <09>

APPROVED BY:

____
Project Leader Date

____
Prof. S. Kalafatis Date

____
T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	10-4-2025	Elijah Orozco		Draft Release

Table of Contents

List of Tables

List of Figures

1. Overview

The document will cover how the Time of Flight (ToF) sensor and Microcontroller (MCU) will interface. It will list all of the input data and the output data. It will also show how each subsystem will manage each.

2. References and Definitions

2.1. References

Refer to section 2.2 of the Functional System Requirements document.

2.2. Definitions

CCA	Circuit Card Assembly
mA	Milliamp
mW	Milliwatt
MHz	Megahertz (1,000,000 Hz)
TBD	To Be Determined
TTL	Transistor-Transistor Logic
VME	VERSA-Module Europe
Tof	Time of Flight

3. Physical Interface

3.1. Weight

Component	Weight	Number of Items	Total Weight
EPC660 Carrier Board	NA	1	NA
TOF Cape	TBD	1	TBD
NUCLEO-H7A3ZIQ	NA	1	NA

3.2. Dimensions

Component	Length	Width	Height
EPC660 Carrier Board	37.25mm	36.00mm	3mm
TOF Cape	TBD	TBD	TBD
NUCLEO-H7A3ZIQ	133.34mm	76.00mm	5mm

3.3. Mounting Locations

The LBPSS shall be mounted based on an application basis. The user will decide in what direction they wish the camera to monitor.

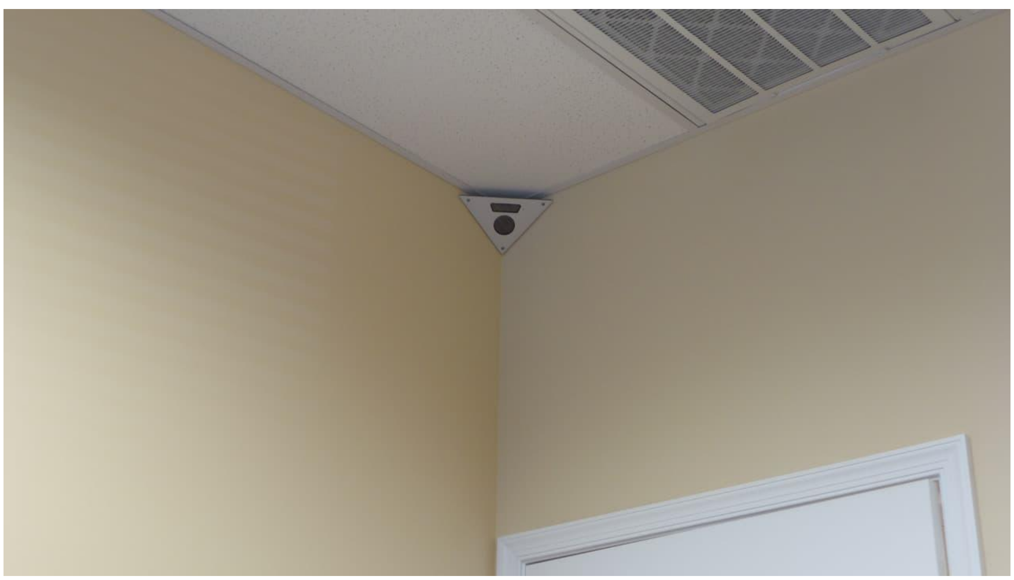

Figure 1: Example Location of LBSS

4. Thermal Interface

There will be no thermal interface.

5. Electrical Interface

5.1. Primary Input Power

The primary power will be supplied from a 24VDC wall adapter.

5.2. Voltage, Current, and Power levels

Component/Input	Voltage [V]	Current (Typical) [mA]	Power[mW]
EPC660 (V_{BS})	-10	3.8	38
EPC660 (V_{DD}, V_{DDPLL}), STM32H7	1.8	18	32.4
EPC660 (V_{DDIO}), STM32H7	3.3	8	26.4
EPC660 (V_{DDA}, V_{DDPXM})	5	126	630
EPC660 (V_{DDPXH})	10	13	130
LED Cape (LED Switching)	5	86	434
LED Cape (LEDs)	15	100	1500

The current and power calculations were based on the datasheets for these various components which did not give clear power consumption information for the frame rate and integration time that is planned. Most likely these values can be lowered once basic functionality is achieved.

5.3. Signal Interfaces

5.3.1 ToF cape MCU interface

The ToF cape and MCU will interface using the morpho connectors of the NUCLEO board. This will essentially be a board-to-board connector. The main pins interacting will be integrated circuit communication (I2C) and Digital Camera Module Interface (DCMI).

5.3.2 MCU to Computer interface.

The MCU will connect to an external computer for validation and checking to see if the frames being generated are desirable. This connection will be done using a universal serial bus (USB).

5.4. Video interface

5.4.1 External Frame Viewer

The MCU will transmit the frames captured from the ToF sensor to an external computer using the MCU to Computer interface. On the computer, the frames will be displayed using processing code and code that is able to make a visual representation of the frames.

5.5. User Control Interface

The user will not have a user control interface since the control code will be flashed to the MCU and will not need updating during use.

6. Communications / Device Interface Protocols

6.1. Wired Communications

6.1.1 Universal Serial Bus 2.0 Full-Speed

A micro-USB connection from the NUCLEO board will be used to connect the MCU to an external computer.

6.1.2 Ethernet RMII

An ethernet cable will be used to send the camera data to an external computer due to its larger load capacity than USB.

6.1.2 Wires from board

Wires will be run from the ToF board to the MCU. These wires will include I2C and DCMI communications.

6.2. Wired Communications Protocol

6.2.1 ST-LINK over USB

The programming software, STM32CubeIDE will, uses ST-LINK over USB to flash compiled code to the MCU.

6.2.2 User Datagram Protocol (UDP)

The ethernet cable will use UDP to send frames captured from the ToF sensor to the external computer. This will be used because of an average need of 1.5MB/s of data needed to be transmitted.

6.2.3 Digital Camera Module Interface (DCMI)

The ToF senser will send the sensor data over DCMI. This will involve a parallel 12 bit line for data, HSYNC line, VSYNC line and DCMICLK line.

.

Lidar-Based Physical Security Sensor

Calvin Lindberg
Elijah Orozco
Michael Yoon

SCHEDULE AND VALIDATION

REVISION – 1
4 October 2025
25 January 2018

Revision - 1

Execution Timeline

Week of:	8/24	8/31	9/7	9/14	9/21	9/28	10/5	10/12	10/19	10/26	11/2	11/9	11/16	11/23	11/30
TASK															
Project/Team Assignment												**Status:**			
Brainstorming												Completed			
ToF Sensing Module Selection												Delayed			
CONOPS Report												In Progress			
MCU Selection/Order															
ToF Module Study															
PCB Design Study															
Midterm Report															
Midterm Presentation															
Order DMCI Test Module and PCB Components															
Configure DCS Acquisition and processing															
ToF PCB Board Design/Order															
MCU PCB Board Design/Order															
Status Update Presentation															
Build MCU PCB															
Build ToF PCB															
Write debug code for the STM32															
Begin Debugging/Iterating PCB Designs for ECEN 404															
Tune epc660 to Optimize Performance															
Validation and Final Demo of Subsystems															
Final Presentation															
Final Report															

Validation Plan

Test Name	Success Criteria	Methodology	Status	Responsible Engineers
Important configurations are permanently stored in non-volatile memory	Framerate, mode, and error threshold settings persist despite losing power	Apply specific settings such as a max framerate of 2, then disconnect and reconnect power. Check whether the framerate continues to be set to 2.	TESTED	Michael Yoon
At least 1 FPS during full 4 DCS capture.	When capturing 4 DCS frames for maximum accuracy, the sensor should record at least 1 frame per second.	The sensor will be configured to capture 4 DCS frames and the framerate will be measured	TESTED (not on epc660)	Michael Yoon
Intruder detection	Motion and displacement within the frame immediately trigger the sensor	Map the alarm signal to an LED. Simulate motion with reference to the calibration frame, and make sure the LED turns on.	TESTED	Michael Yoon
Resilience to noise	The STM32 when properly calibrated will not report false alarms due to noise	Apply moderate artificial noise (test different levels). Calibrate the STM32, then make sure that intruders are still detected whilst the noise doesn't trigger the sensor by itself.	TESTED	Michael Yoon
Robustness to incomplete frames	The STM32 continues to function if the epc660 gets interrupted during frame transfer	Manually force the sensor to stop sending data mid frame and ensure that the software handles it gracefully	UNTESTED	Michael Yoon
Software compensation of temperature drift	The epc660 accurately measures depth accurate to 1cm in a wide temperature operating range	Capture distance measurements in the freezer and in the oven (set to ~150 degrees Fahrenheit)	UNTESTED	Michael Yoon
Important configurations are permanently stored in non-volatile memory	Framerate, mode, and error threshold settings persist despite losing power	Apply specific settings such as a max framerate of 2, then disconnect and reconnect power. Check whether the framerate continues to be set to 2.	TESTED	Michael Yoon
Voltage Spikes in Power Supplies	The Vout stays constant over a range of input values	Use a signal generator to create a voltage sweep or spike and measure outputs	PARTIALY TESTED	Calvin Lindberg
Voltage Level and Enable Signal	The output voltage levels stay within an acceptable range during steady state and enable	Using the test headers and benchtop multimeters/power supply test all power output and enable signals	PARTIALY TESTED	Calvin Lindberg
MCU Voltage spikes	MCU remains functional after power is removed	Unplug the mcu from the USB power supply	TESTED	Elijah Orozco
MCU output signals	Output signals for data lines match the dev board	Measure the output signals with an oscilloscope and compare.	TESTED	Elijah Orozco
Sensing Range	The sensor will unambiguously detect depth up to 6.25m or 12.5m depending on configuration	Move objects to just under the unambiguous range and check if the depth is accurately measured	UNTESTED	Full Team
Sensing FoV	The system will have a functional FoV of 108 degrees	Place objects separated by ~105 degrees and move each to see if the motion is detected	UNTESTED	Full Team
Fast moving objects	The sensor will detect reasonably fast objects	Quickly throw an object through the sensing field and attempt to detect it using the sensor	UNTESTED	Full Team
Unreflective surfaces	The sensor will be capable of detecting the depth of reasonably nonreflective surfaces	Place black paper on a wall and take measurements half the maximum unambiguous range and at the maximum range	UNTESTED	Full Team

Lidar-Based Physical Security Sensor

Calvin Lindberg

Elijah Orozco

Michael Yoon

SUBSYSTEM REPORTS

REVISION –

5 December 2025

SUBSYSTEM REPORTS

FOR

Lidar-Based Physical Security Sensor

TEAM <09>

APPROVED BY:

____

Project Leader Date

____

Prof. S. Kalafatis Date

____

T/A Date

Change Record

Rev.	Date	Originator	Approvals	Description
-	12-6-2025	Elijah Orozco		Draft Release

Table of Contents

List of Figures

List of Equations

List of Tables

1. Introduction

The Lidar-Based Physical Security Sensor (LBPSS) will effectively and autonomously monitor a room or space for intruder using Time-of-Flight (ToF) technology. It will gather depth information using the epc660 sensor, and this data will be processed using an STM32H7A3 microcontroller to detect intrusions. The depth information will then finally be transmitted to a host PC for visualization and debugging purposes. The LBPSS will have a low false-alarm rate and a high rate of detection. The LBPSS is separated into three subsystems: the power and LED subsystem, the microcontroller subsystem, and the software subsystem. These three subsystems were independently tested and validated with moderate success. While there still remains work to be done regarding the construction of hardware and the integration of software with the actual epc660 chip, throughout this report, it will be shown that the subsystems are clearly outlined, developed, and will soon be ready to integrate together to produce the full system desired by our sponsor.

2. Power and LED Subsystem

2.1. Subsystem Introduction

The Power and LED Subsystem is responsible for delivering stable power to the EPC660, the MCU, and the LED driver circuitry. This subsystem is currently supplied by a wall-connected 24 VDC source, which is regulated through a series of converters to meet the needs of each component. The LED circuit is designed to be controlled by the EPC660 while being powered through these regulated outputs. Although some challenges were encountered during assembly, the subsystem was validated through simulations and through testing of the portions that were successfully assembled.

2.2. Subsystem Details

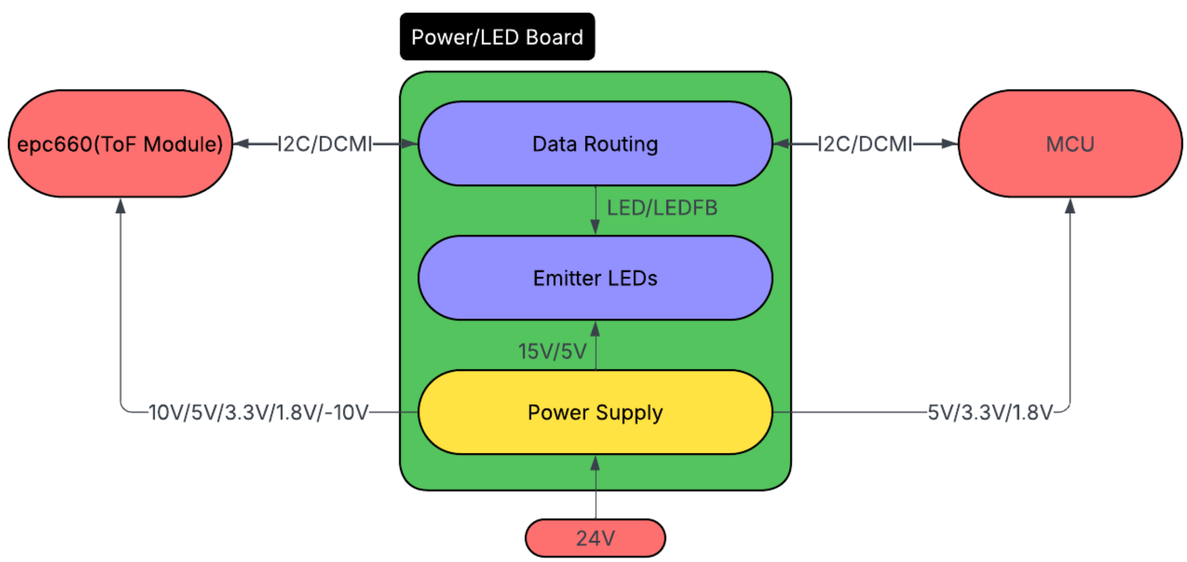

Figure 1: System Block Diagram

The first challenge for this subsystem was understanding the needs and requirements of the EPC660. After consulting the datasheet, it was determined that the EPC660 requires five different voltage levels with specific error margins and allowable ripple. The voltages must also be brought up to level in a specific order as to not damage the device. Using this information, the schematics from the EPC660 evaluation module were used as a reference for designing the LED driver circuit. Although the datasheet provides an example LED circuit that uses the EPC660's LED signal path to drive the LEDs, this path can only handle 200 mA, which is below what is required to produce a clear image. Therefore, an external driver circuit was designed to supply the necessary LED current. For this iteration, the MCU received power from a laptop using a micro-USB connection.

Once the system needs were defined, a PCB was designed and laid out in Altium. TI's WEBENCH Power Designer was used to determine the appropriate converter topologies and to select components that meet the EPC660's power requirements. WEBENCH was also used alongside PSpice for TI to validate the converters and confirm successful operation under load.

For the layout, each converter was first routed individually and then positioned on the PCB near the header or circuit it was intended to power. Additional layout considerations included placing decoupling capacitors close to their respective ICs, carefully positioning inductors to reduce mutual inductance, and selecting appropriate trace widths for each power rail. Design review meetings were held with our sponsor to go over layout decisions.

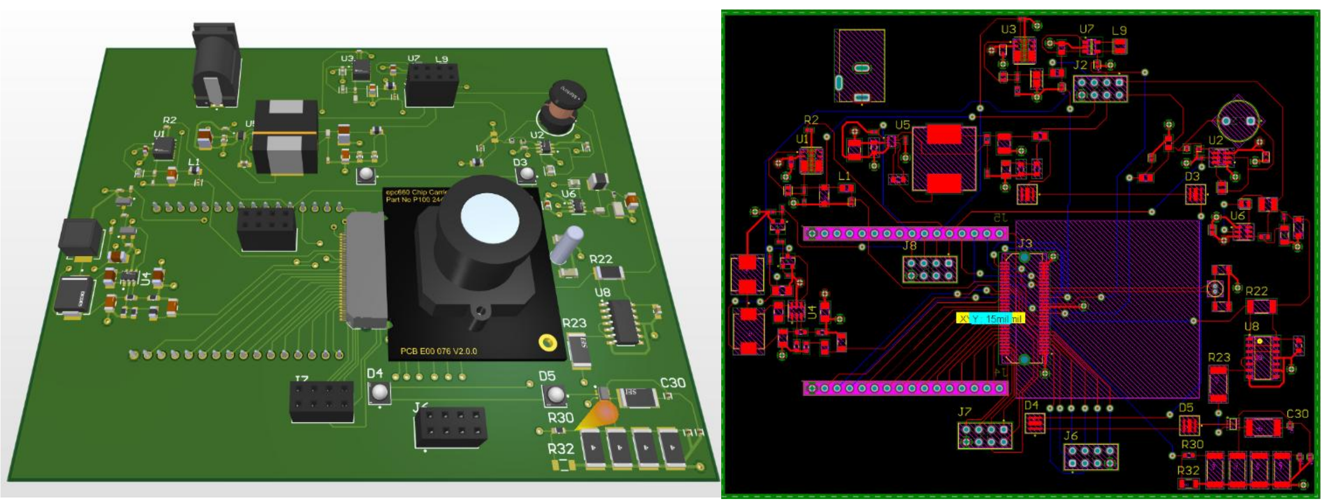

Figure 2: Schematic

2.3. Subsystem Validation

The Power and LED Subsystem was validated using simulation and hardware testing. Because the primary function of this subsystem is to provide stable and reliable voltage rails to the EPC660 and associated circuitry, validation focused on confirming that each converter met its expected output voltage, load capability, and ripple specifications.

All converters were first simulated using WEBENCH Power Designer or PSpice for TI. These simulations verified that each design produced the correct output voltage under load, maintained stability, and met the required ripple limits defined in the EPC660 datasheet.

Voltage	**Epc660 Ripple**	**Max Current**	**Acceptable Ripple**	**Voltage Maintained Under Load**
-10V	≤50mV	17mA		
1.8V	≤20mV	24mA		
3.3V	≤50mV	8mA		
5V	≤50mV	356mA		
5V (LED)	NA	150mA		
10V	≤20mV	13mA		
15V (LED)	NA	200mA		

Table 1: Voltage Values

During assembly, a soldering mistake resulted in a short between the power and ground planes. This prevented full hardware validation of all converters. However, the 10 V converter which was soldered before the short occurred was successfully tested on the assembled PCB. The measured output matched the expected voltage, remained stable under load, and was able to be controlled using the enable input.

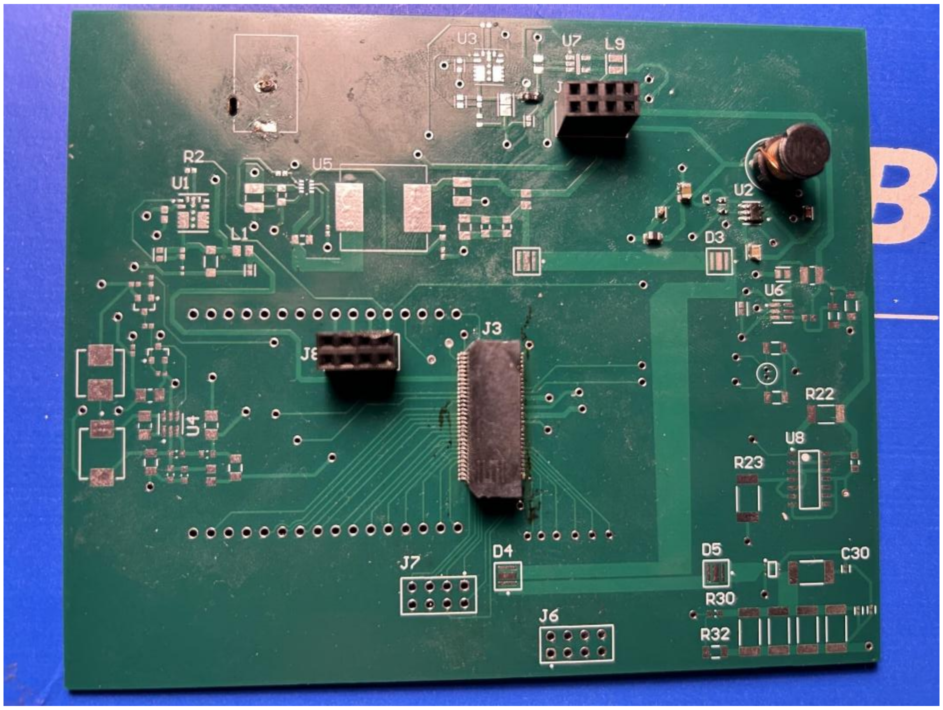

Figure 3: Partially soldered Power and LED

2.4. Design Revisions and Continued Validation

Since full hardware validation could not be completed within the expected timeframe due to soldering issues and other time constraints, additional testing and refinement will be carried out during the break between semesters. This work will also include the design updates necessary for the Power and LED board. The following tasks will be completed using available lab space at RELLIS provided by our sponsor, as well as personally sourced equipment:

- Test the converters that can be assembled with the tools currently available
- Reselect or redesign converter topologies that cannot be assembled or do not meet performance requirements
- Evaluate alternative LED options that meet illumination requirements and provide more solder-friendly packaging
- Update schematics and PCB layout to incorporate all design changes

2.5. Subsystem Conclusion

The Power and LED Subsystem was successfully designed to meet the voltage requirements of the EPC660 and associated circuitry. Simulation results confirmed that all converters operate within their expected performance ranges, and hardware testing of the 10 V rail further validated the design approach. Although full hardware verification could not be completed this semester, the subsystem provides a solid foundation for reliable sensor operation once the remaining converters are validated and the updated PCB is assembled.

3. MCU Subsystem

3.1. Subsystem Introduction

The MCU subsystem is the hardware that will combine the ToF, power, and LED subsystems. It will be used in the configuration process of the EPC chip, the continuous data capture, and the processing required.

3.2. Subsystem Details

The MCU is a STM32H7A3VIT6 which has a 32 bit, 2MB flash, 100 LQFP, and 280 MHz ARM microprocessor. This microprocessor was chosen because it has the necessary IO, sufficient ram, and has enough processing power for the detection algorithms.

3.2.1. IO Communications

The main IO communications used for communicating with other subsystems include DCMI, I2C, and GPIO pins. DCMI communication is the main protocol used for the main data acquisition. The ECP sensor captures the depth data and transmits it over DMCI. The I2C communication is used to write the registers of the EPC chip. These register writes involve configuration values needed for things such as frame rate, capture mode, etc... The GPIO pins are used to configure the enable signals of the various converters used to power the EPC chip. This is because the chip requires a specific starting sequence of voltages.

3.2.2. Hardware Implementation

The MCU has a dedicated PCB that exposes all of the communication pins. The exposed pins are designed to connect to the ToF PCB. The MCU PCB also has an LDO, micro-USB, and reset button. The LDO is used to convert the 5V supplied by the usb cable and convert it to 3.3V to power the PCB. The reset button is used to manually reset the MCU for any reason.

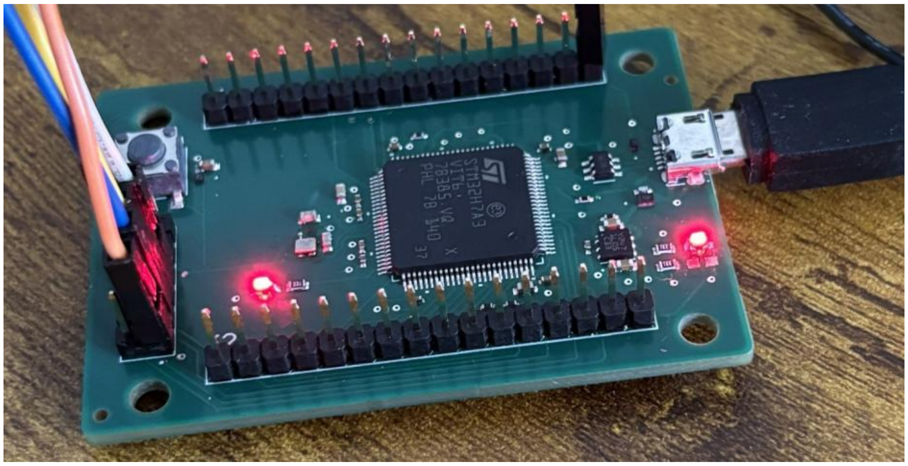

Figure 4: MCU PCB

3.3. Subsystem Validation

The validation of the MCU involved testing all the features needed to make the MCU work. This involved the power, SWD connection, communication pins, and reset capability.

It is important to note that all communication protocols could not be tested due to time constraints. The communications not able to be fully validated and tested are I2C and DCMI. The I2C code was written, but further debugging was required in order to produce a successful result. The DCMI needs to be tested with an appropriate peripheral which could reasonable event that requires so. The PCB also has serial wire debug pins exposed so that the board can be programmed and debugged in an easy manner. not be acquired before the final demo. That being said, further testing will be conducted over the winter break in order to be successful in the next semester.

3.3.1 Power

The power that supplies the MCU comes from an LDO that is located on the PCB. The voltage that the LDO provides is connected to a power plane that is on the PCB. A pin that exposes the power plane was used to test the voltage. The test procedure involved the board being disconnected from the usb cable then being connected. From the graph, it can be seen that the measured voltage is about 3.3V, which is expected.

Figure 5: MCU Voltage Graph

3.3.2 GPIO Pins

The GPIO pins provide a controllable signal that can be used to control other peripherals. It was very important that these pins work so that the system can function. The GPIO pins were tested with a logic analyzer and a scope. The logic analyzer was used to test the GPIOs to see if they were able to be toggled. From the figure below, it can be seen that the pins were able to be toggled, and the scope was able to pick them up. The second test involved the power that the pins were able to provide. This test was done by measuring the voltage at certain locations across two test resistors in series. The graph below shows the voltage at the start of resistor 1 and then the voltage after resistor 1. Using these voltages and the resistance, the current was calculated to be about 15.1 mA. This current was well within the expected value.

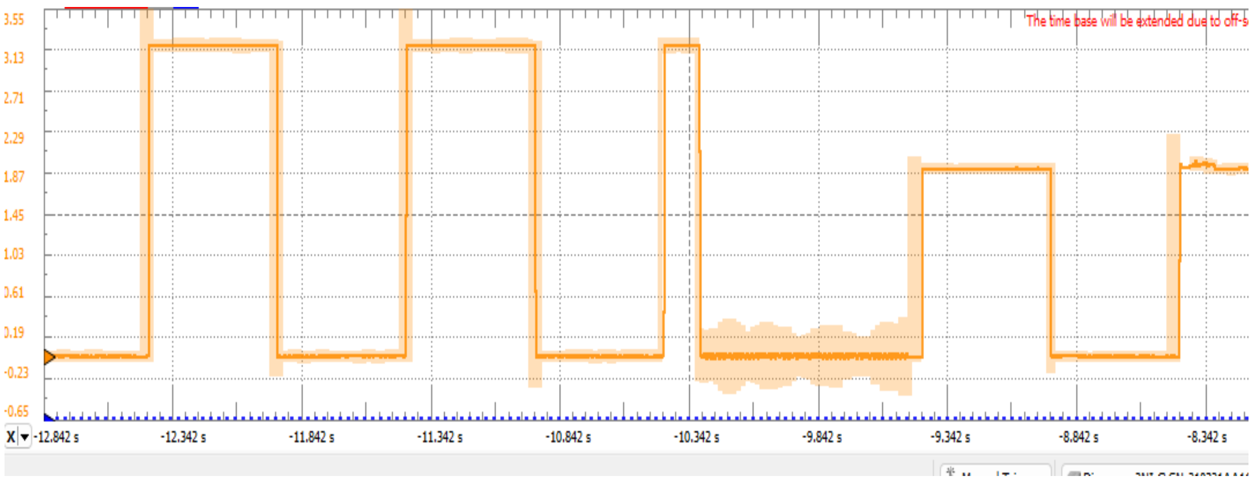

Figure 6: GPIO Test Voltage Graph

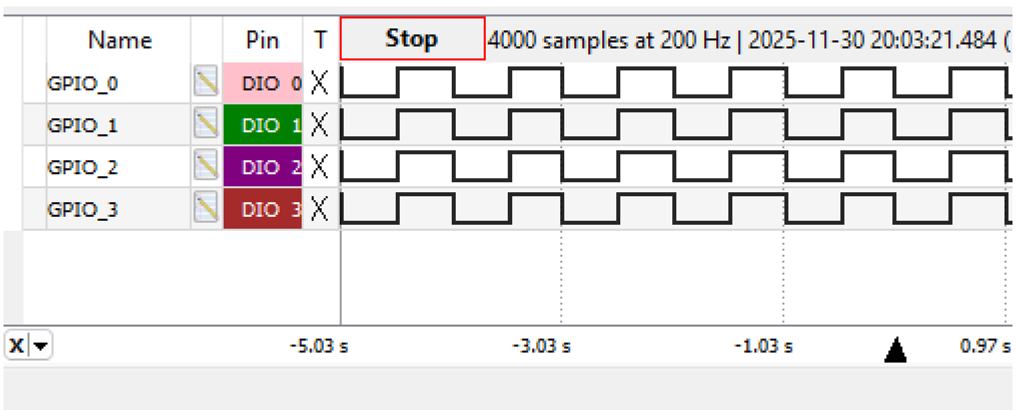

Figure 7: GPIO Digital Analyzer Graph

3.3.3 Miscellaneous Hardware

Other hardware components that were validated include the reset button, SWD pins, and cable disconnection.

The reset button was tested by programing the LEDs on the board to be toggled on. The reset button was then click and the LEDs response was observed. As expected, the LEDs turned off then turned back on.

The SWD pins were tested by simply trying to program the board. Since code was able to flashed, the pins were valid. Another test involved output coming from the SWD output pin to the debugging device. Since the print statements were able to be captured on an external device, this test was valid.

The validation plan also involved the less-than-ideal condition of the micro usb cable being disconnected during the operation of the board. The test was performed similarly to the reset button in which LEDs were toggled on and the cable was disconnected. Once the cable was reconnected the LEDs turned on which made the test valid.

3.4 Subsystem Conclusion

The MCU subsystem was successfully shown to be programable and provided necessary output for the communication protocols that were tested. Further testing needs to be done for the other key communication protocols. The base functionality and knowledge of implementing hardware will be very vital in the future when changes and further prototyping are needed. The integration of the firmware subsystem will be easier in the future due to the MCU being the same as the development board the subsystem is using.

4. Software Subsystem

4.1 Subsystem Introduction

While the power, LED, and MCU subsystems provide all the necessary hardware for a LiDAR sensor to function, The software subsystem integrates all these components together, making sure that they properly communicate with one another. This subsystem fundamentally consists of the MCU firmware (the C code written to directly interface with the epc660) and the python scripts written on the host PC to receive image frames, data logs, and provide a debugging interface with the MCU.

4.2 Subsystem Details

While ultimately the software will be designed to communicate with the epc660, since the hardware to make the epc660 function is still in development, the software was tested and developed using an OV7670 standard RGB camera. This camera module was selected because it uses the same DCMI video communication interface as the epc660, and it also requires the same amount of memory per frame.

4.2.1 OV7670 Testing Module

The OV7670 is a standard camera module with a resolution of 640 x 480 pixels, but it is configurable to have a resolution of 320 x 240 pixels which emulates the epc660. It uses an 8-bit DCMI interface whereas the epc660 uses a 12-bit DCMI interface, however, the STM32 microcontroller being used supports both modes. For both systems, a single pixel’s data is packed into one 16-bit halfword, leading to an identical frame size of 153.6 KB.

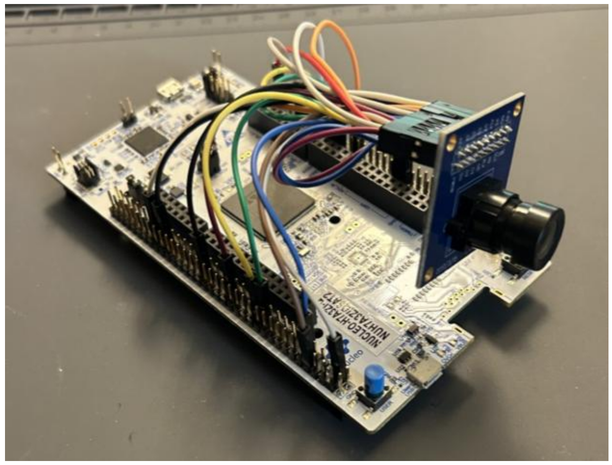

Figure 8: OV7670 Camera Module Wired to the Nucleo Board

4.2.2 STM32 Firmware

The majority of the software subsystem resides within the STM32 firmware. This code first initializes the microcontroller, enabling interrupts, configuring the internal clock, and enabling all the required peripherals. It then implements the boot-up of the OV7670, writing to registers via I^2C and toggling GPIO pins where necessary. While these configurations will have to change when migrating over to the epc660, the general infrastructure is already implemented and functional.

The STM32 then begins its DCMI capture via DMA (Direct Memory Access). The DMA pipelines the frame data automatically from the camera module into RAM, allowing other processes to continue while this happens. Once one set of 4 DCS (Differential Correlation Sample) frames are stored in memory, the STM32 processes these frames using the phase equation:

$$\theta = \arctan\left(\frac{A - C}{B - D}\right) \qquad ToF\ Phase\ Equation$$

Equation 1: ToF Phase Equation

Where A, B, C, and D are DCS frames. The phase values are stored in a 16-bit integer scale and represent processed depth data.

Before transmitting the depth data to the host PC for visualization and debugging, the STM32 compares this depth data pixel by pixel to a reference in memory which stores the original, calibrated depth data of the room. A displaced pixel is one that differs from the calibrated frame by more than the configured noise tolerance. If the number of displaced pixels exceeds the configured displaced pixel tolerance, then the STM32 will enable the “alarm signal” GPIO pin which powers an LED for demonstration purposes.

The depth data is then transmitted to a host PC via USB full speed. A python script receives the data and displays it to the screen, representing different depths either through different color or through greyscale (depending on user preference).

Given the restrictions in hardware, there is also a mode to fully simulate the DCS frames locally on the STM32, thereby bypassing the DCMI connection with the OV7670. More detail on the simulation will be provided in section 4.2.3.

4.2.3 Frame Reception Software

The host PC’s frame reception software is written in Python and is designed to interface with the STM32 over two connections, one being the USB full speed and the other being a command-line debugging link via UART. Through the USB, it continuously monitors the video port, searching for a specific four-byte header (0xAA55AA55) to synchronize the start of a new frame. Once synchronized, it reads the raw data into a 240 x 320 NumPy array. It uses the OpenCV library to display the depth data in real time and allows the user to cycle between different display modes (such as grey scale vs a color representation of depth).

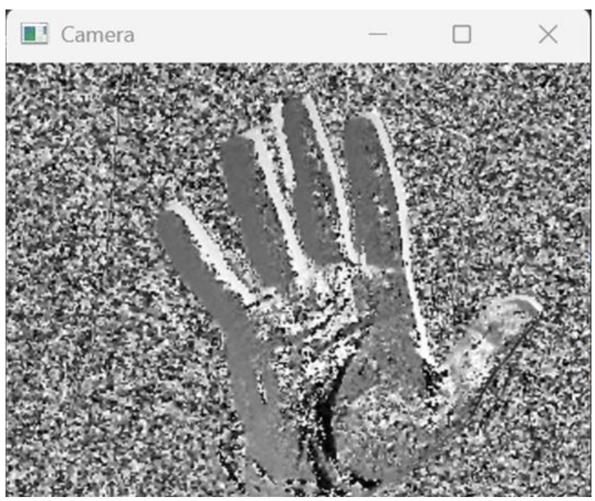

Figure 9: Processed “Depth Data” from the OV7670

Figure 9 was gathered via the OV7670 which doesn’t capture DCS frames for depth data. As a result, the processed image does not convey much information. In order to test the pipeline, a simulation was implemented. DCS frames were generated locally on the STM32, bypassing the DCMI capture.

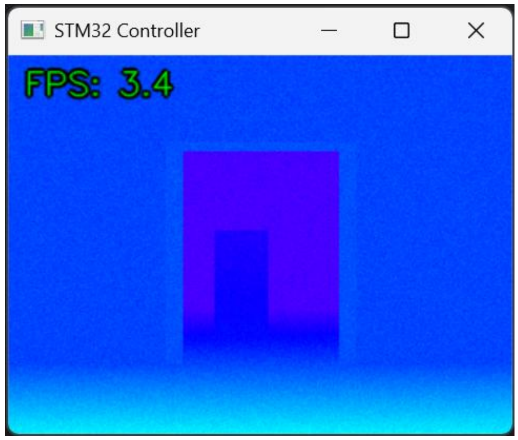

Figure 10: Simulated depth frame

In this simulation, we see a doorway with an “intruder” passing across the gap in the room behind. The phase is represented by color, and the distance that the phase refers to is dependent on the calibration of the sensor. The light blue pixels at the bottom represent a phase of nearly 0° which would be the closest to the camera. The far wall behind the doorframe has a phase of 180° which (in a calibrated unambiguous range of 12.5m) would refer to 6.25m. Everything else in the scene lies within this range. The “intruder” only enters the scene after a time and passes from the left to the right side. When the intruder is in frame, the STM32 detects the pixel displacement and enables the alarm GPIO pin which is connected to an LED. More on the detection and alarm LED will be shown in the validation section of this report.

4.2.4 Debugging and Configuration Interface

Lastly, coded into both the STM32 firmware and the host PC software is a command-line debugging interface. The STM32 has interrupt flags set up, which recognizes when the external software is trying to communicate with it via UART. It parses the command, then enacts the debugging or configuration action that the user requested. On the PC side, the software runs a separate thread to listen for user input in the terminal, allowing the commands to be sent back to the STM32. Some example commands are as follows:

```
--- TYPE COMMANDS BELOW AND HIT ENTER ---
>> MODE_CAPTURE
[STM32]: Command: MODE_CAPTURE
[STM32]: Switched to MODE_CAPTURE
>> CALIBRATE
[STM32]: Command: CALIBRATE
[STM32]: Calibration request queued
[STM32]: Reference frame CALIBRATED.
>> MAX_FRAMERATE=1
[STM32]: Command: MAX_FRAMERATE=1
[STM32]: MAX_FRAMERATE set to 1
>> DISPLACED_THRESHOLD=100
[STM32]: Command: DISPLACED_THRESHOLD=100
[STM32]: DISPLACED_THRESHOLD set to 100
>> DATA_LOG
[STM32]: Command: DATA_LOG
[STM32]: Avg DCMI capture (10 frames): 1210095.375 ms
[STM32]: Avg depth calc (10 frames): 75498.742 ms
[STM32]: Avg USB TX (10 frames): 137416.484 ms
[STM32]: Avg displaced pixels (10 frames): 72338.6
>>
```

Figure 11: Command-Line Debugging Interface

The user is able to type commands at any time and press enter. The STM32 dynamically handles the command and prints out the appropriate responses. The complete list of current commands are listed below.

Command	Function
MODE_CAPTURE	Switches the firmware into camera capture mode and restarts the DCMI/DMA pipeline if needed. Setting is stored in flash so it persists after reset.
MODE_SIMULATE	Enables the synthetic scene generator instead of the live sensor. This flag is stored in flash so the board reboots in the same mode.
MAX_FRAMERATE=<value>	Sets the maximum number of frames processed per second (1-120). Use 0 for unlimited. The chosen value is written to flash.
CALIBRATE	Copies the next processed frame into the reference buffer to establish a new baseline. Executes once per command and is not stored in flash.
DATA_LOG	Prints the running averages (last 10 frames) for DCMI capture time, depth calculation time, USB transfer time, and displaced pixel count.

DISPLACED_THRESHOLD=<value>	Sets how many displaced pixels are required (1-76800) before the intrusion LED turns on. Stored in flash and applied immediately.
ERROR_THRESHOLD=<value>	Adjusts the per-pixel phase-difference tolerance (0-65535) used when counting displaced pixels. Stored in flash and applies to future frames.

Table 2: Available User Commands

4.3 Subsystem Validation

It was difficult to validate certain aspects of the software subsystem due to a lack of hardware. For instance, it was impossible to validate the field-of-view, or the detection of intruders at the edge of the field-of-view when I don't have a functioning epc660 and lens. Given these constraints however, the validation was largely successful.

4.3.1 Robustness to Disconnects and Non-Volatile Storage

The software is designed to handle various types of disconnects, whether it be power or communication. Additionally, all configurations that the user applied should not be lost upon reset or disconnection. To validate this, the USB full speed connection was disconnected and reconnected. The frames did not desynchronize, and the video stream resumed as normal. Then, the max framerate was set to be 1 via the command-line interface.

Then, the power was disconnect and reconnected to the STM32. Upon regaining power to the microcontroller and regaining connection to the host PC, the result is shown below:

```
>> MAX_FRAMERATE=1
[STM32]: Command: MAX_FRAMERATE=1
[STM32]: MAX_FRAMERATE set to 1
```

Figure 12: Command Used to Reduce the Max Framerate

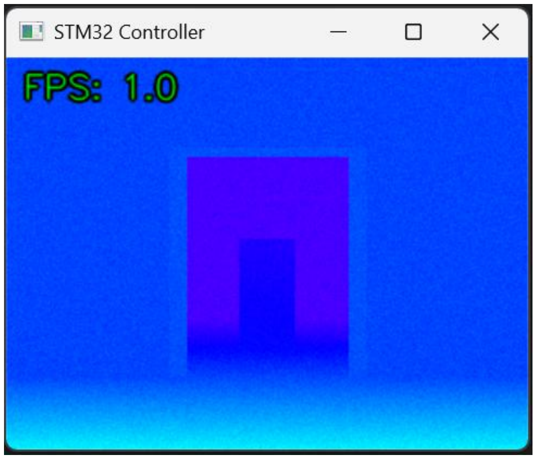

Figure 13: Video Feed After Power Reboot

As can be seen, the STM32 maintained its settings despite the reboot.

4.3.2 Minimum Required Framerate

Our sponsor requested a minimum framerate of 1 FPS for the security sensor. During simulation, the STM32 was regularly able to achieve 3.4 FPS, and from section 4.3.1 it obviously can maintain 1 FPS. During capture using the OV7670 module, the results were slightly disappointing.

```
>> DATA_LOG
[STM32]: Command: DATA_LOG
[STM32]: Avg DCMI capture (10 frames): 1210095.375 ms
[STM32]: Avg depth calc (10 frames): 75498.742 ms
[STM32]: Avg USB TX (10 frames): 137416.484 ms
[STM32]: Avg displaced pixels (10 frames): 72338.6
>> |
```

Figure 14: Timing Data Using the OV7670

The USB transmission takes place in parallel with the DCMI capture meaning that it has no impact on the frame rate since it is about 10 times faster than the DCMI capture. From the data retrieved, we can see that each frame is taking about 1.285 seconds, which is about 0.8 FPS; however, it must be remembered that this is the OV7670 camera module and not the epc660. The epc660 operates its DCMI at double the framerate of the OV7670, and while the OV7670 requires two clock cycles to transmit a single pixel, the epc660 will transmit a pixel in a single clock cycle, leading to a clean improvement of 4x the speed. Therefore, we can confidently say that the epc660 would produce more than 1 FPS during normal operation.

4.3.3 Intruder detection

Below shows some of the validation tests performed using the depth simulations

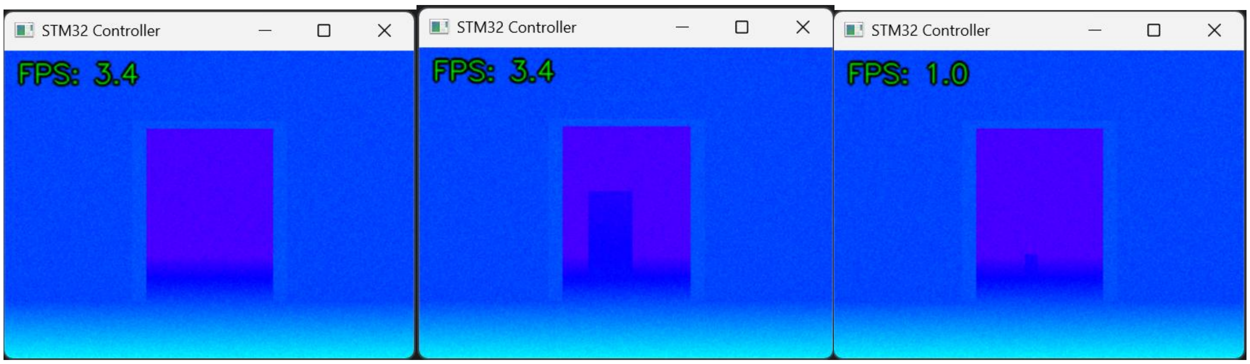

Figure 15: Simulated depth frame with no intruder (left), with a large intruder (middle), and with a small intruder at reduces framerate (right)

On the left is the scene with no intruder, and in the middle, the intruder has emerged. The STM32 easily detects the large intruder and lights the alarm LED as shown below in **Figure 15**. Even when the intruder is significantly reduced in size to only being 10 x 20 pixels and when the framerate is limited to just 1 FPS (shown on the right – it may be necessary to zoom in to see the small intruder), the STM32 is still able to make the detection and turn on the alarm LED.

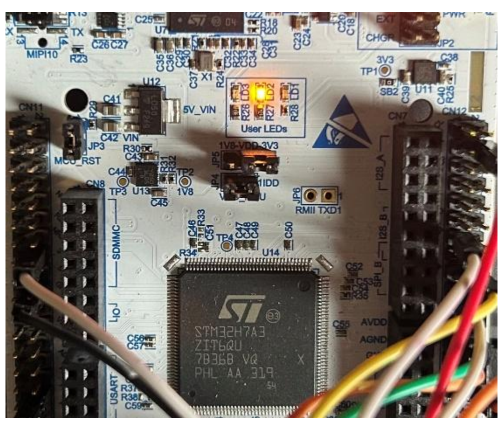

Figure 16: Alarm LED when the intruder is in frame

4.3.4 Resilience to Noise

As is true with any sensor, the signal-to-noise ratio (SNR) is absolutely crucial and largely depends on the environment and the noise-filtering capacity of the hardware. However, given a decent SNR, the software is able to be resilient to noise via the configuration of the ERROR_THRESHOLD parameter while still detecting intruders effectively. Here is an example given the empty, noisy simulation (with no intruder).

```
--- TYPE COMMANDS BELOW AND HIT ENTER ---
>> CALIBRATE
[STM32]: Command: CALIBRATE
[STM32]: Calibration request queued
[STM32]: Reference frame CALIBRATED.
>> ERROR_THRESHOLD=0
[STM32]: Command: ERROR_THRESHOLD=0
[STM32]: ERROR_THRESHOLD set to 0
>> DATA_LOG
[STM32]: Command: DATA_LOG
[STM32]: Avg DCMI capture: N/A (no capture frames)
[STM32]: Avg depth calc (10 frames): 157449.078 ms
[STM32]: Avg USB TX (10 frames): 136072.297 ms
[STM32]: Avg displaced pixels (10 frames): 76505.8
>> ERROR_THRESHOLD=1000
[STM32]: Command: ERROR_THRESHOLD=1000
[STM32]: ERROR_THRESHOLD set to 1000
>> DATA_LOG
[STM32]: Command: DATA_LOG
[STM32]: Avg DCMI capture: N/A (no capture frames)
[STM32]: Avg depth calc (10 frames): 157443.953 ms
[STM32]: Avg USB TX (10 frames): 136020.922 ms
[STM32]: Avg displaced pixels (10 frames): 3090.1
>> ERROR_THRESHOLD=2100
[STM32]: Command: ERROR_THRESHOLD=2100
[STM32]: ERROR_THRESHOLD set to 2100
>> DATA_LOG
[STM32]: Command: DATA_LOG
[STM32]: Avg DCMI capture: N/A (no capture frames)
[STM32]: Avg depth calc (10 frames): 157433.266 ms
[STM32]: Avg USB TX (10 frames): 136421.594 ms
[STM32]: Avg displaced pixels (10 frames): 0.0
```

Figure 17: Impact of Different Error Threshold Values

This data is compiled in the table below.

Error Threshold	Displaced Pixels
0	76506
1000	3090
2100	0

Table 3: Displaced Pixels at Different Error Thresholds

From this, we can see that an Error Threshold of 0, nearly all the 76,800 pixels were considered “displaced” and the alarm LED was constantly on. However, as we increase the error threshold, it begins to filter out the noise. Once we hit an error threshold of 2100, there is no noise. Even with the small intruder at this noise level and error threshold, the STM32 is able to make the detection and turns on the alarm light.

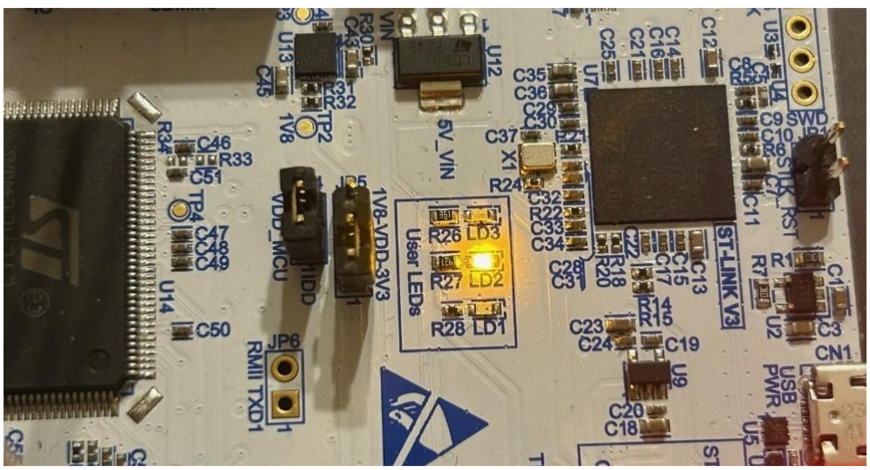

Figure 18: Detection LED for Small Intruder with Noise Filtering

4.4 Subsystem Conclusion

The software subsystem successfully integrates the DCMI camera module, STM32 microcontroller, and host PC. It correctly receives DCS frames, processes them within the STM32 firmware, performs a detection algorithm, and establishes a connection between the STM32 and host PC for purposes of debugging and visualization. The command-line user interface allows configuration settings to be adjusted and stored in non-volatile flash memory onboard the microcontroller, such that the settings persist between resets and power cycles. All of this was successfully tested using the OV7670 camera module and using generated simulations. While the epc660 has not been interfaced with directly yet, this software framework was built and tested such that it can be ported over to the epc660 with as minimal edits as possible.