TRUE SIGHT
ABSOLUTE VISION SYSTEMS

November 12, 2015

Dr. Andrew Rawicz
School of Engineering Science
Simon Fraser University
Burnaby, British Columbia
V5A 1S6

Re: ENSC 440W Design Specifications Document for TrueSight

Dear Dr. Rawicz,

Please find enclosed the design specifications document for TrueSight, a portable real time laser mapping device. At Absolute Vision Systems, our goal is to design and implement a laser mapping system which would help emergency responders navigate through low visibility environments.

Our design specifications document goes into detail about how the general system, hardware and software requirements are met for TrueSight. The design choice for each component is explained and reflects the functional specifications required for the product. We also include a test plan which helps us determine if our desired functionality has been met. This design specifications document will be utilized by our team to efficiently implement and test the TrueSight product.

Our company, Absolute Vision Systems, consists of five motivated and experienced electrical, biomedical and systems engineering students. Our team includes Don Labayo, Curtis Rietchel, Tomasz Szajner, Samson Tam and Jim Tu. Please contact us at dlabayo@sfu.ca if there are any questions or concerns.Sincerely,

Don Labayo

CEO

Absolute Vision Systems

Enclosure: *Design Specification Document for TrueSight*

TRUE SIGHT

ABSOLUTE VISION SYSTEMS

DESIGN SPECIFICATION FOR REAL-TIME 3D LASER SCANNING DEVICE

PROJECT TEAM

Don Labayo
Curtis Rietchel
Tomasz Szajner
Samson Tam
Jim Tu

SUBMITTED TO

Andrew Rawicz - ENSC 440W
Steve Whitmore - ENSC 305W

School of Engineering Science
Simon Fraser University

PRIMARY CONTACT

Jim Tu
jimt@sfu.ca

ISSUED DATE

November 12 2015

REVISION

Rev 01

Abstract

The design specifications for the TrueSight contain a detailed description of the design and development of the proof-of-concept model. Therefore, the discussed system will only include the functional requirements of the proof-of-concept model detailed in the document *Functional Specification for Real-Time 3D Laser Scanning Device* [1].

The document gives the reader a detailed description of the hardware, firmware, and software systems, as well as justification of the design approach. Hardware design focuses on the power management system printed circuit board (PCB), embedded processor, and sensor. The power management system converts the battery voltage into the various voltage rails required by TrueSight's subsystems. The heart of the TrueSight platform lies in the Jetson TK1 embedded processor. It is used to control all of the TrueSight's subsystems as well as process the incoming sensor data to be displayed back to user. Firmware design focuses on the communication between subsystems. Firmware will take control of the GPIO pins and I^2C communication standards. Software design focuses on processing sensor data and creates three distinct image types to suit different environmental conditions.

The final section of this document details the testing procedures to assess the proof-of-concept's functionality. Testing is split up into two main phases - modular testing and integration testing. The test plan will be used as a benchmark for the proof-of-concept's success.

Table of Contents

Abstract ii
Table of Contents iii
List of Figures v
List of Tables v
Glossary v
1. Introduction 1
1.1 Scope 1
1.2 Intended Audience 1
2. System Overview 2
3. Hardware Design 5
3.1 Microsoft Kinect v2 Sensor 5
3.2 NVIDIA Jetson TK1 7
3.3 Display Module 8
3.4 Mechanical Design 9
3.5 Power PCB 10
3.5.1 Battery 11
3.5.2 12V Boost Converters 11
3.5.3 5V Regulator 12
3.5.4 1.8V and 3.3V LDOs 13
3.6 User Inputs & Outputs 14
4. Software Design 18
4.1 General Software Design 18
4.1.1 Boot-up & General Scripts 18
4.2 Image Processing 19
4.2.1 Mode Toggle Design 20
4.2.2 Image Filtering 20
4.2.3 Edge Detection 21
4.2.4 Zoom Calibration Design 22
4.2.5 X and Y Calibration Design 23
4.2.6 Image Translation Design 23

4.3 Firmware 24
4.3.1 I2C 24
4.3.2 GPIO 24
5. Test Plan 25
5.1 Schedule 25
5.2 Phase 1 Testing 25
5.2.1 Hardware 25
5.2.2 Firmware 26
5.3 Phase 2 Testing 26
5.3.1 Functional Tests 26
5.3.2 Integration Tests 27
6. Conclusion 28
7. References 29

List of Figures

Figure 1 System Top Level Overview 2
Figure 2 Embedded Computing System and Mounting Design 3
Figure 3 Display HUD Design 3
Figure 4 Kinect v2 Sensor 4
Figure 5 Hardware Design Block Diagram 5
Figure 6 Jetson TK1 and Power PCB Enclosure 9
Figure 7 Power Management System PCB 10
Figure 8 Jetson 12V Boost Converter 11
Figure 9 Kinect 12V Boost Converter 12
Figure 10 5V Buck Regulator 13
Figure 11 1.8V LDO 14
Figure 12 3.3V LDO 14
Figure 13 Jetson TK1 Expansion Header 15
Figure 14 Potentiometer Biasing Circuit 16
Figure 15 Switch Biasing Circuit 16
Figure 16 LED Driving Circuit 17
Figure 17 General Software Architecture 18
Figure 18 Filtering Flow Chart 19
Figure 19 Toggle State Diagram 20
Figure 20 Un-equalized and Equalized Image Intensity Range [5] 21
Figure 21 Zoom Algorithm 23
Figure 22 Image Geometric Transform Example [6] 24
Figure 23 Project Timeline 25

List of Tables

Table 1 Kinect v2 Technical Specs 6
Table 2 Jetson TK1 Development Board Technical Specs 7
Table 3 TrueSight Signal Header Pin 15

Glossary

ABS	Acrylonitrile butadiene styrene
ADC	Analog to Digital Converter
CCM	Continuous Conduction Mode
CPU	Central Processing Unit
CUDA	Compute Unified Device Architecture

DCM	Discontinuous Conduction Mode
e-test	Electrical Test
GPIO	General Purpose Input Output
GPU	Graphical Processing Unit
HUD	Heads up Display
IC	Integrated Circuit
I^2C	Inter-Integrated Circuit
IR	Infrared
LCD	Liquid Crystal Display
LDO	Linear Dropout
LED	Light Emitting Diode
Lexan	Type of Polycarbonate Plastic
nits	Measurement of display brightness (candela/m^2)
OpenCV	Open Source Computer Vision
OS	Operating System
RGB	Red Green Blue
SoC	System on Chip
Sysfs	A virtual file system provided by the Linux kernel that is used to access low-level hardware
USB	Universal Serial Bus

1. Introduction

TrueSight, by *Absolute Vision Systems*, is a real time laser mapping system intended to enhance and supplement human vision in low visibility environments. TrueSight is a helmet-mounted device which grabs IR depth data and projects this information into the user's field of view in real time. Our goal at *Absolute Vision Systems* is to create a unique visual system which is a robust and simple-to-use solution for emergency response teams. TrueSight improves efficiency for first responders in low visibility environments while simultaneously reducing the risk they take.

1.1 Scope

This document describes the main design specifications for the product. The design choices are made with respect to the functional specifications outlined by *Absolute Vision Systems*. The document contains an overview of the software and hardware design choices. Most design considerations are aimed at creating a prototype. However, there is also design discussion with respect to the final product. A test plan for each component of the system is devised, which will help evaluate the performance and success of the product.

1.2 Intended Audience

The design specification will be used as a reference for the *Absolute Vision Systems* team. Team members can refer to this document as a guideline for prototype implementation and testing. Test engineers will refer to this document to ensure functional requirements are met.

2. System Overview

A top level functional block diagram of the TrueSight system is shown in Figure 1**.** The TrueSight product is based on three main components: the Jetson TK1 embedded computer, the LCD HUD, and the Kinect v2 sensor. Model diagrams of each of the components are illustrated in Figure 2**,** Figure 3**,** and Figure 4**.**

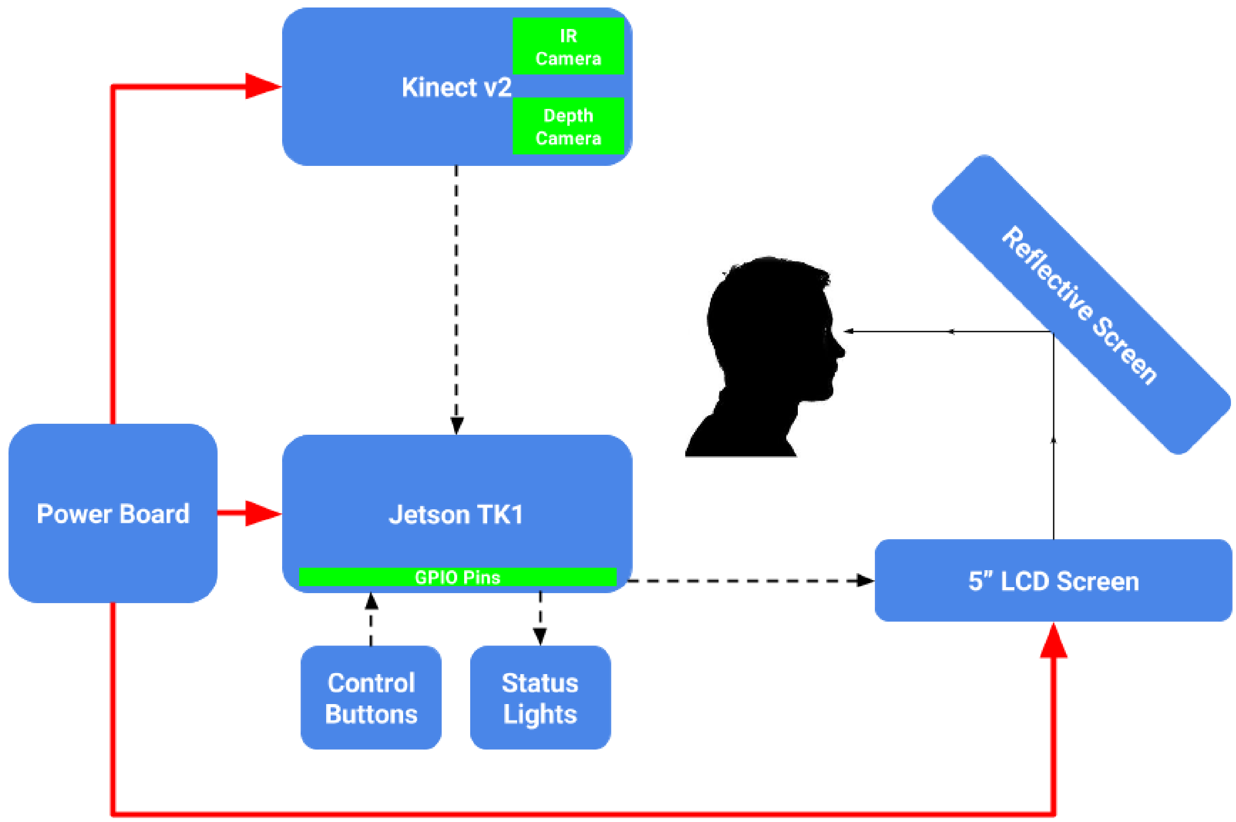

Figure 1 System Top Level Overview

Figure 2 shows the embedded computer system, which will be used for functionality control via switches, processing incoming image data, and sending images to the display system. It also has a clamp-like mounting system, which would be attached to a firefighter's oxygen tank. The box enclosure contains both the Jetson TK1 and the Power PCB, which powers various modules. Figure 3 represents the LCD Display HUD, which will take processed images from the embedded computer and reflect them into the user's field of view. The display system will be mounted on a helmet, with the display in front of the user's eyes. Figure 4 shows the Kinect v2 sensor. Its purpose is to grab depth and IR data from the surrounding environment. The sensor unit will also be mounted on the user's helmet. Each of these components work in harmony to create a unique visual experience for the user.

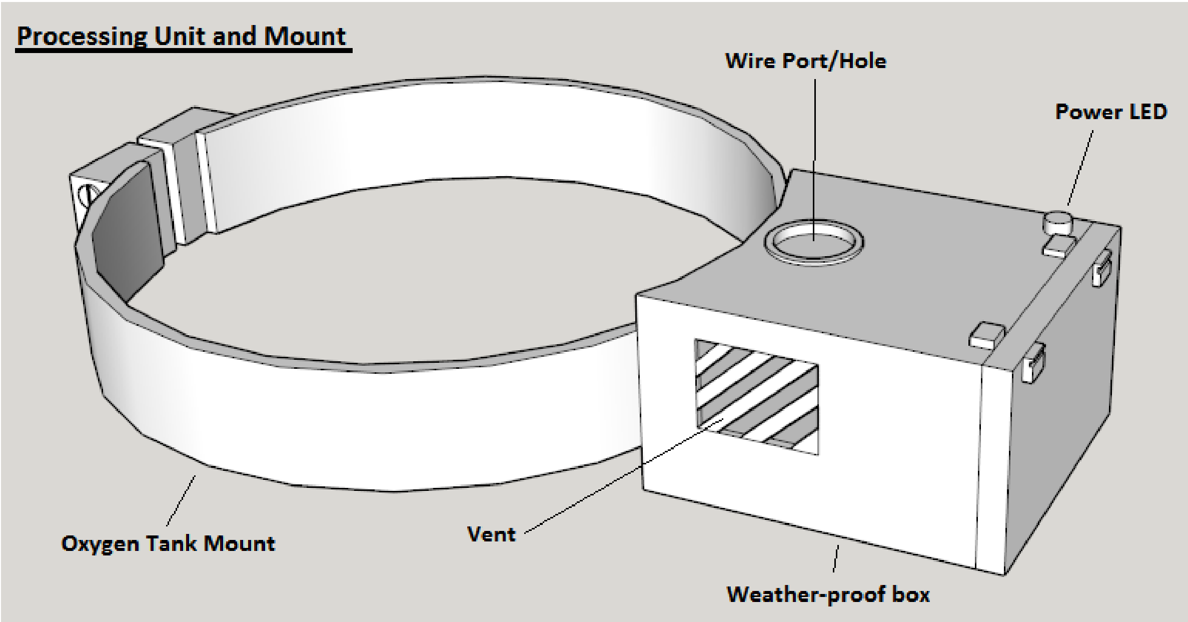

Figure 2 Embedded Computing System and Mounting Design

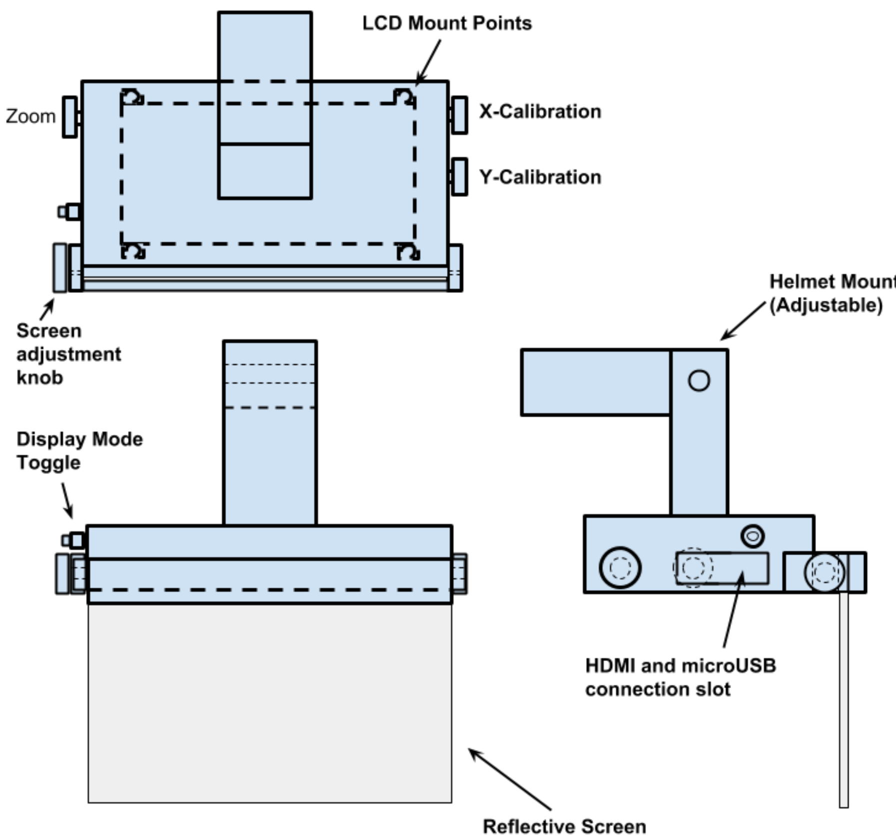

Figure 3 Display HUD Design

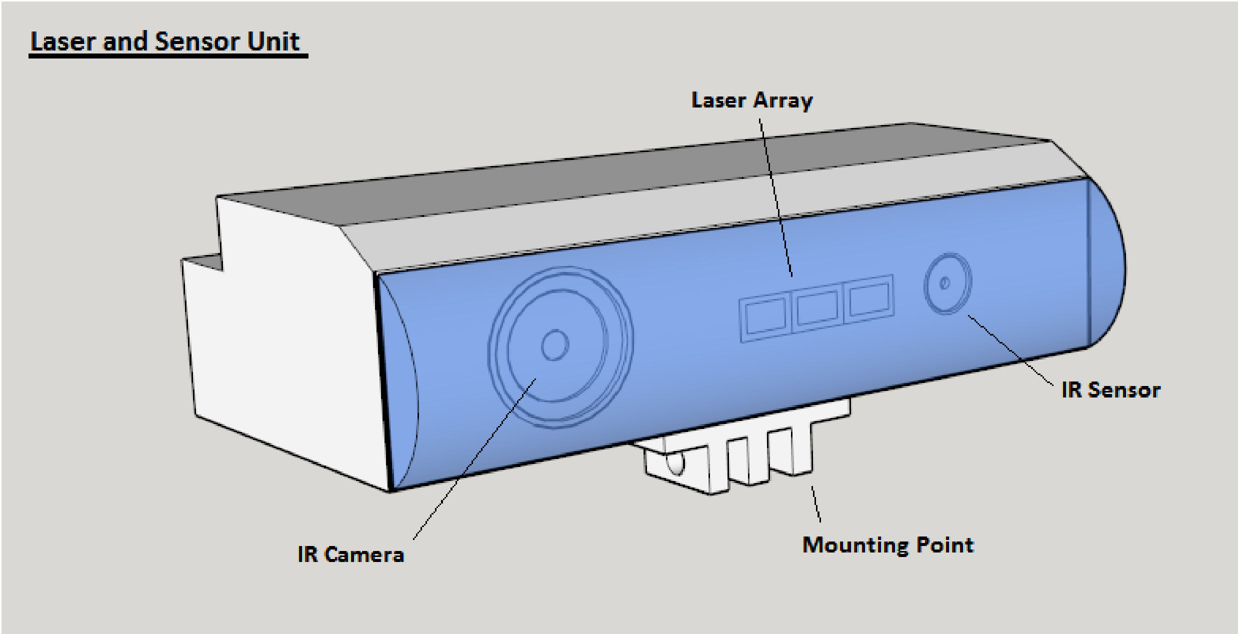

Figure 4 Kinect v2 Sensor

3. Hardware Design

The following section will detail the selection and design of the major hardware components used in the TrueSight platform. The modules designed and utilized in the system include:

1. Microsoft Kinect v2
2. NVIDIA Jetson TK1
3. Display Module
4. Power PCB
5. User Inputs & Outputs (Switches & LEDS)

A brief overview of each component integrated within the overall system is seen in Figure 5 below.

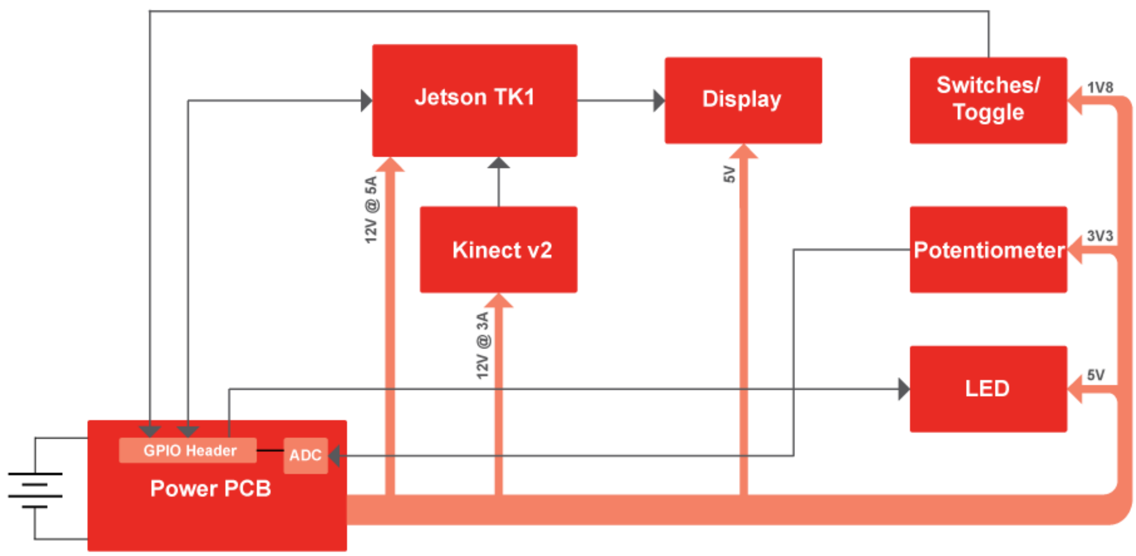

Figure 5 Hardware Design Block Diagram

3.1 Microsoft Kinect v2 Sensor

For TrueSight to operate in low-visibility environments, the most efficient solution is to adapt and program Microsoft's Kinect v2 sensor. The Kinect v2 was chosen for its multiple sensors and general fitment with our prototype requirements **[R039-A, R040-A]**. The choice of the Kinect v2 was also based on availability and a reasonable price point. This allowed for more time to design the system that surrounds the sensor and implement a more comprehensive, multi-faceted product. Below is Table 1, which shows general characteristics of the Kinect v2 sensor.

Table 1 Kinect v2 Technical Specs

Frame Rate	~30
FOV	70° x 60°
Resolution	512 x 424
Depth range	0.5 - 4.5 m
Communication Interface	USB 3.0
Voltage Requirement	12V
Power Consumption	32 W

The main purpose of the Kinect v2 is to utilize its IR camera and depth sensor as both modes will have the option to be displayed in the TrueSight product. It is important to note that the Kinect v2 also has a regular RGB camera, which will not be used as part of the prototype. It is possible that the RGB camera could be utilized in final product as a way to record the movements of the user.

While not the optimal wavelength, the laser diodes included in the Kinect are within the infrared wavelength bandwidth, which gives us the ability to see in the dark, and possibly in low visibility situations such as smoke **[R043-A]**. A notable design limitation of the Kinect sensor is its range and field of view. While the depth sensor has an approximate range of 4.5 meters, the best solution for the final product would have a slightly longer range, more suitable for larger rooms **[R044-A]**. However, this problem is slightly alleviated due to the fact that the IR camera has somewhat greater range. The option to toggle between depth and IR modes allows for more visibility depending on the size of room and the conditions of the surrounding environment. From Table 1, it is also notable that the framerate of the Kinect sensor is sufficient for us to create a real time view of the environment surrounding the user. However, due to the high amount of data coming from multiple sources, an appropriate embedded computer with sufficient processing power is required. The Kinect depth sensor also uses class 1 laser diodes, which satisfies our desired safety requirements **[R046-A]**.

From a software perspective, the Kinect v2 has a healthy open source community, which helps us create our own application for the Kinect. However, the sensor requires a separate power source to interface with an USB enabled device, and on top of this, requires a USB 3.0 connection to maintain a high throughput data stream for the captured data.

For a final product design, more expensive options could be considered such as SWIR camera or longer wavelength laser diodes (wavelengths of up to 1000 - 1600 nm). Due to their high price point and general unavailability, the Kinect v2 was chosen.

3.2 NVIDIA Jetson TK1

To control each module of the TrueSight system, the Jetson TK1 was chosen to take the role of the embedded computer. The Jetson TK1 development board was chosen for primarily 3 reasons: it uses a NVIDIA SoC which has an integrated GPU, it has a USB 3.0 Port, and it comes with a wicked expansion header. These qualities make it ideal as an embedded development board for our project.

The Jetson TK1 development board uses an NVIDIA Tegra K1 SoC as its main processing unit. The Tegra K1 is a System on Chip, meaning that it contains both a CPU and GPU, and is used on smartphones, tablets, and Chromebook [2]. The SoC is important to our project because we will be doing intensive image processing on our video streams. The NVIDIA GPU, Kepler GK20a, has 192 CUDA cores enables parallel computing, and supports hardware-accelerated APIs for image processing and computer vision **[R050-A]**. The Tegra K1 is built using the NVIDIA Kepler architecture, a GPU architecture that NVIDIA uses for both mobile and desktop products.

USB 3.0 is needed for our project because the Kinect v2 transmits its data over USB 3.0 and is not backwards compatible with older version of USB. This is because USB 2.0 does not have sufficient bandwidth to support the Kinect's video streams, while USB 3.0 is capable of up to ten times the bandwidth of its predecessor [3].

Additionally, the Jetson TK1 development board is an embedded Linux development platform, and thus contains various interfaces and expansion ports. These interfaces and expansion ports include GPIO pins that will be used for system integration, I^2C support (an industry standard for serial communication) and several other standard interfaces **[R047-A, R049-B]**. Table 2 below details some of these interfaces.

With all the functionality the Jetson TK1 development board is able to provide, it is still able to keep its power consumption on a 12V rail relatively low. Using protocols from the OpenCV library and activating the use of all graphics processing CUDA cores, the Jetson board consumes on average of 8-10W of power keeping it below the requirement **[R051-A]** [4].

Table 2 Jetson TK1 Development Board Technical Specs

Dimensions	5" x 5"

Processor	Tegra K1 SoC
RAM	2GB DDR3L
Storage	SD slot, 16GB eMMC
PCIe	1 mini-PCIe
Audio/Video	1 HDMI, 1 Mic in and 1 Line out
Serial Communication	USB, RS-232, UART, SPI, I^2C
Networking	1 Gigabit Ethernet port
GPIO	7 pins
SATA	1 full-size
Power	12V DC barrel power jack

3.3 Display Module

To display the information grabbed by the Kinect and processed by the Jetson TK1, two options were to utilize a small projector or a LCD display. Our initial choice was to purchase a pico-projector and simply project the image onto a visor / screen mounted to the existing equipment. However, we could not find a pico-projector whose cost, focal distance, and image resolution met our functional requirements. The best option was to create a Heads-Up Display, similar to the technology used in cars, by utilizing a small LCD display and a reflective and semi-transparent screen **[R052-A]**. A TFT display module produced by Adafruit integrating a 5" LCD panel by On Tat Industrial Company was chosen. The LCD display will be positioned such that the image will be reflected to the user, as seen in Figure 3. Imperfections in the reflected image's size / position are adjusted via control knobs.

The display module purchased consists of both a driver board and mating LCD panel. The LCD is capable of display at 200nits brightness while consuming approximately 600mA at 5V. The panel used is 5" across in diameter and has a resolution of 800x480. This is more than sufficient in outputting all the data that the Kinect is able to capture to the user. Tuning the zoom and position of the video stream can be done on the fly via knobs to ensure the outlined edge detection image fits correctly around objects in the user's field of view. The chosen LCD module more than satisfies the functional requirements for TrueSight outlined by **[R052 - R056].**

3.4 Mechanical Design

The goal for TrueSight was to design a product which is lightweight and adaptable to current firefighting and emergency response systems. The mechanical design consists of the processing/power enclosure, the helmet mounted Kinect, and the HUD enclosure. For the prototype, we chose mostly ABS plastic materials as it is simpler to machine into the desired shape. For the final product, the enclosures will mostly be designed out of metal, which will resist heat and the environment much better than plastics.

The processing and power enclosure will house the Jetson TK1 as well as the power PCB and battery. The enclosure also includes a PC fan and ventilation to cool the circuits inside. Since the size of the Jetson Dev Board is 5" x 5" and the power PCB is 1.77" x 5.89", we chose a slightly larger box with dimensions of 2.5" x 8" x 7", while still remaining within our desired functional requirements **[R007-A]**. Figure 6 shows the isometric view of the enclosure. This allows extra space for the battery, as well as power cables and ribbon cables throughout the enclosure. Holes in the enclosure allow for power and data lines to connect between both the HUD enclosure and the Kinect v2. One main rocker switch will be used to turn on and off the device, and will be placed on the side of the enclosure. A clamp system shown in Figure 2 will allow the option for attaching to a firefighter's air tank.

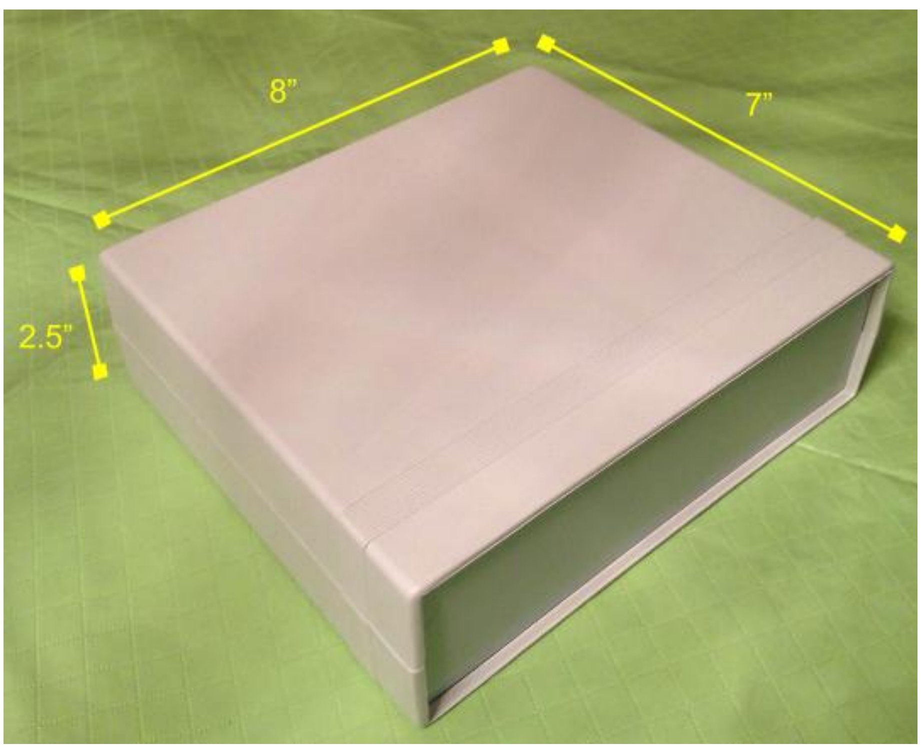

Figure 6 Jetson TK1 and Power PCB Enclosure

The Kinect v2 will most likely be mounted with a single machine screw to the top of the helmet. Further implementation in the final product would require a separate housing to resist heat and weather, since the current housing for the Kinect would be impractical for environments where the user would need a robust and durable product. For the TrueSight initial design, it is still important that we achieve a light weight on the user's head, and the Kinect v2 satisfies the initial functional requirement **[R009-A]**.

The HUD enclosure design is carefully thought out, as the goal is to create a relatively small enclosure which is not too obstructive to the user's view **[R008-A]**. Since the display is already 5”, the HUD enclosure is of size 1.5” x 5” x 7”. This enclosure is designed such that it will attach via an adjustable arm to the front of the user's helmet, and around the forehead level. The Lexan plastic screen is 3.5 x 4.5 inches and is positioned 45 degrees to the enclosure. This ensures a good reflection off the LCD screen. The plastic screen is adjustable so the use can fine tune the screen position to their liking. Figure 3, shows the general design of the HUD enclosure. In the final product, the preferred design would be smaller, but since we are restricted to using a larger LCD display over a projector, a larger enclosure is designed.

3.5 Power PCB

In order to ensure the portability of the TrueSight system, we chose to design our own power management system. The power management system takes power in from the 11.1V battery and outputs 12V at 3A, 12V at 5A, 5V, 3.3V, and 1.8V to the required subsystems. The power management board can be seen in Figure 7 below.

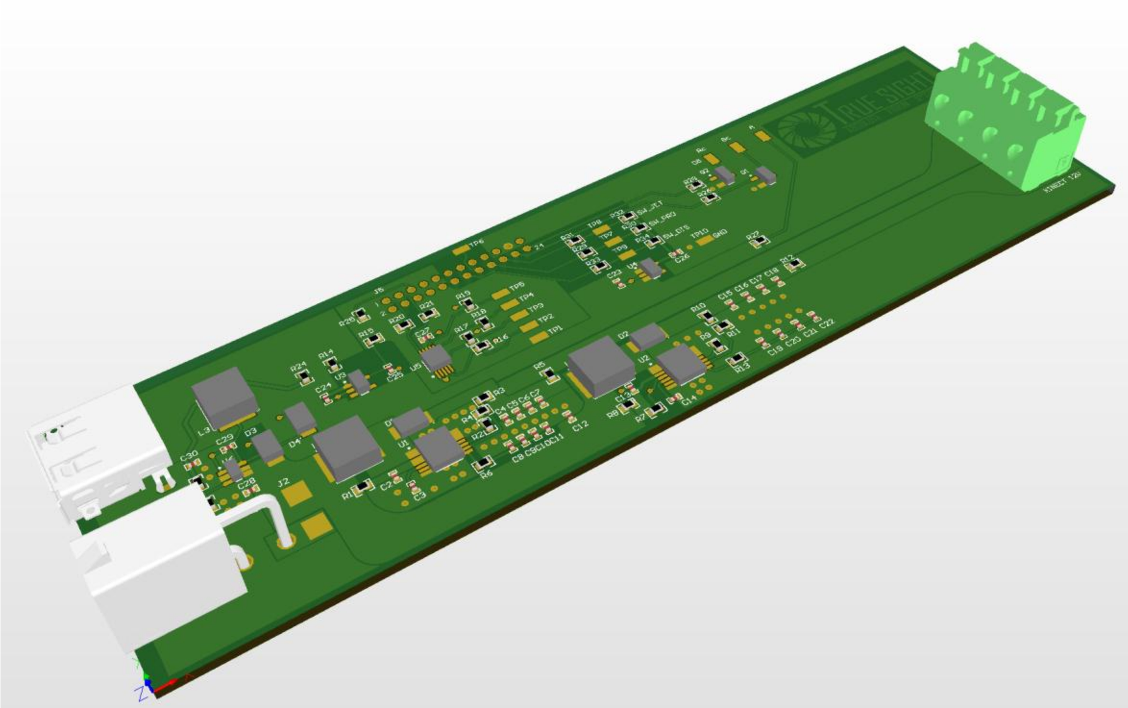

Figure 7 Power Management System PCB

3.5.1 Battery

To power the entire TrueSight system, we needed a battery that could support the entire system for a minimum of 30 minutes, as specified in **[R052-A, R059-A]**. Under max load, the entire TrueSight will consume upwards of 95W of power. The battery chosen is the Duratrax 11.1V LiPo battery, capable of delivering 5000mAh at a max constant discharge current of over 25A. This battery is designed for use in radio controlled vehicles which requires it to be both light and powerful while adhering to safety regulations that are also required for the TrueSight system **[R035-B]**. The battery can be charged by an AC adapter and can be easily swapped on-the-fly to minimize down time of the system **[R060-A]**. If the components of TrueSight were operating at max capacity, this battery will able to deliver 35 minutes of continuous use. Fortunately, we will not be running the platform at max capacity and will reduce its consumption, thus giving TrueSight a longer time of operation.

3.5.2 12V Boost Converters

Figure 8 below is the schematic design of the 12V boost converters.

The power management board uses the TPS55340 chip as a boost converter. When the switch is turned off, the inductor's reluctance to changes of current creates a voltage spike which turns on the diode and charges the capacitor. Depending on the voltage desired, the circuit can be designed for DCM or CCM operation. Because of the small step up, 11.1V to 12V, the chip operates in DCM with an 11% duty cycle.

In the event that the boost converters fail, the 0R resistors R1 & R8 can be depopulated. In doing so, the Jetson/Kinect would be drawing power directly from the battery.

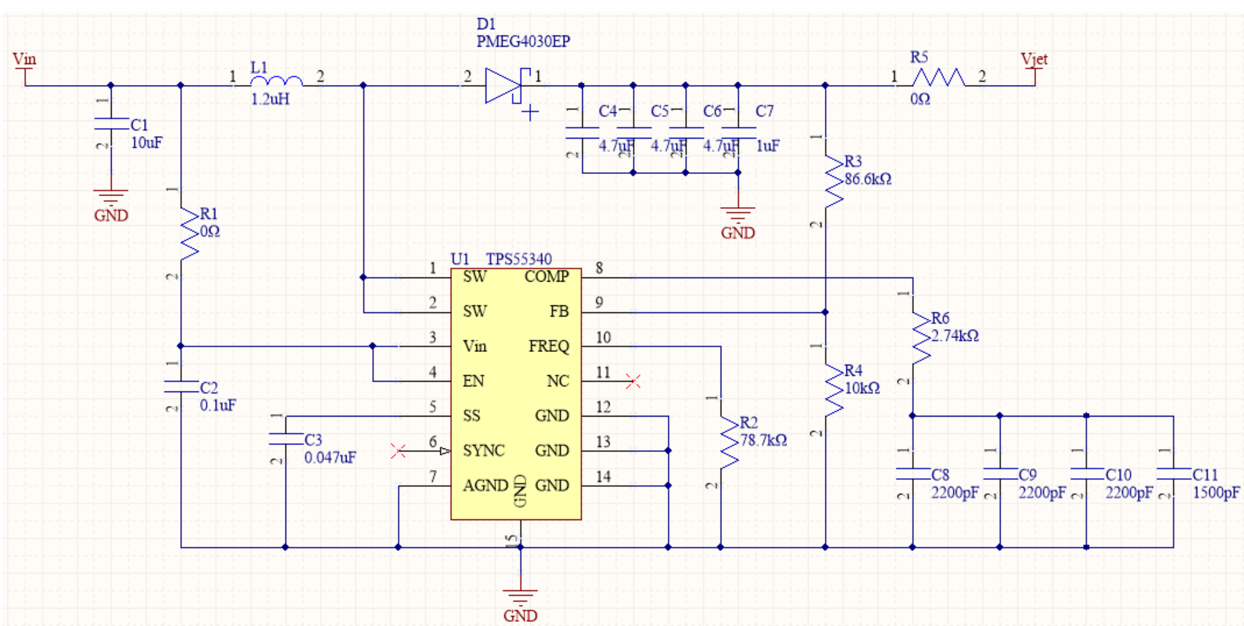

Figure 8 Jetson 12V Boost Converter

The boost converter for the Kinect uses the same design as above but also has the enable pin connected to a GPIO pin, as seen in Figure 9 below.

This allows us the option of controlling the Kinect boot up via the Jetson TK1. In the event that the Kinect v2 enable is not able to be controlled via software, R7 can be populated with a 0R jumper to tie the enable pin back up to the rail voltage.

The 12V converters are designed to handle a maximum of 5A which can be used by the Jetson TK1 at full operating load. However, the Kinect only requires a maximum of 2.67A. Our circuit design is flexible enough to ignore the minor changes in resistor values in redesigning the circuit for a 3A maximum.

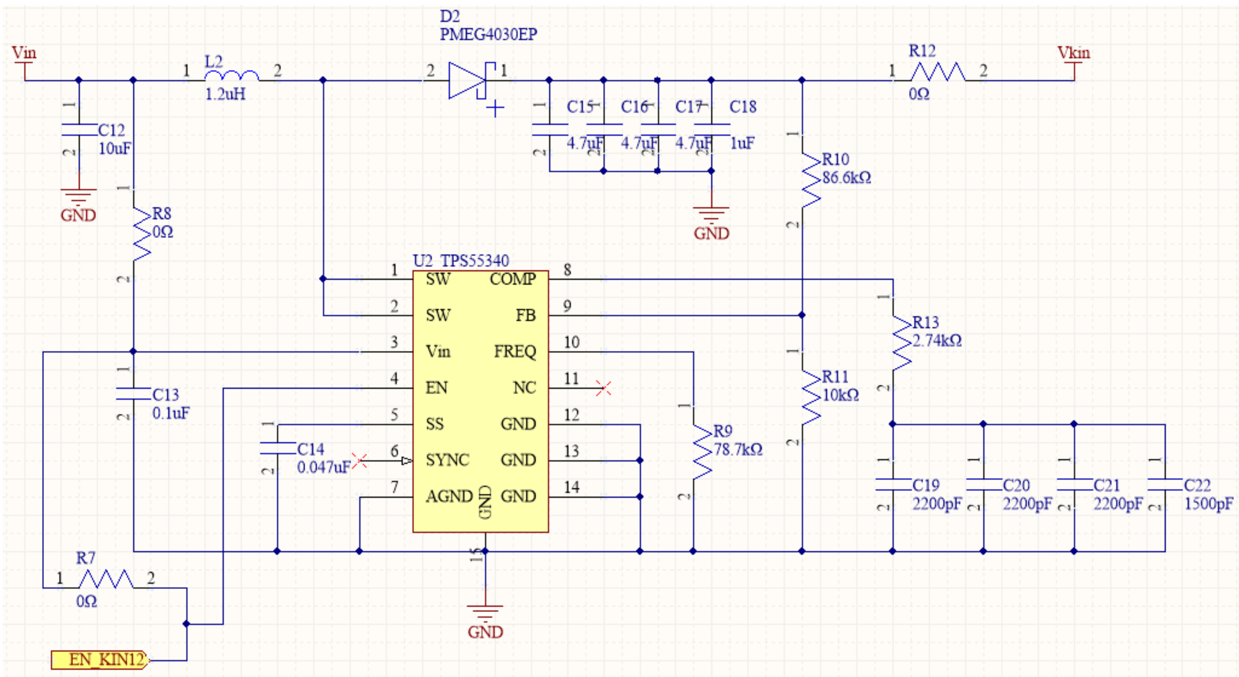

Figure 9 Kinect 12V Boost Converter

The HTSSOP package was chosen due to budget restrictions and ease of prototyping. Future revisions of the power management board will utilize pick and place machines which will make the board more compact.

Power is delivered from a terminal connector to the Jetson & Kinect v2.

3.5.3 5V Regulator

The 5V rail is provided by a buck regulator IC. Figure 10 below is the schematic design of the 5V regulator.

The 12V boost converters shown above, boost the supply voltage from 11.1V to 12V; the buck converter will take the battery voltage 11.1V and output 5V. We chose to use a switching IC instead of a LDO to ensure sufficient current can be deliver as it not only drives the LCD module but also the LEDs. This stage ensures that we deliver a stable DC voltage to the USB powering the display module and remaining subsystems on the board.

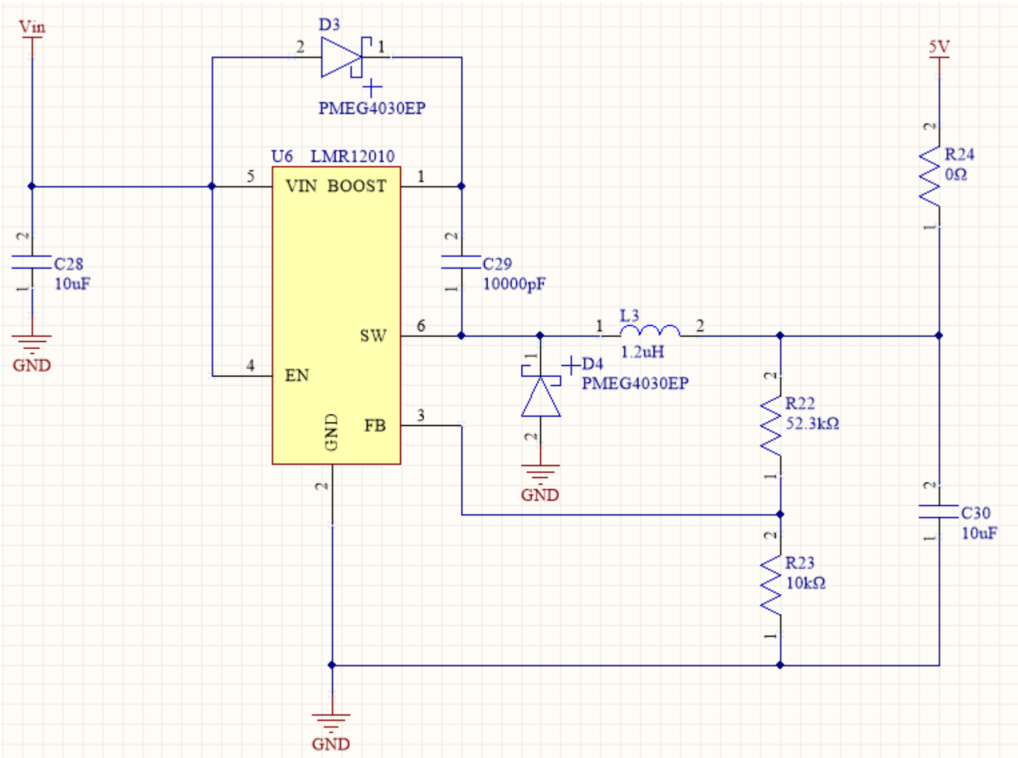

Figure 10 5V Buck Regulator

3.5.4 1.8V and 3.3V LDOs

The 3.3V and 1.8V rails bias the switch, potentiometer inputs and power the ADC. These functions require very little current to operate. Therefore an LDO regulator was used in our design which can provide up to 300mA of continuous current at a steady voltage level of 1.8V or 3.3V. By implementing LDO regulators, we maximize the efficiency when dropping the voltage from 5V to 1.8V and 3.3V. The design implemented for the 1.8V and 3.3V LDO is shown below in Figure 11 and Figure 12**.**

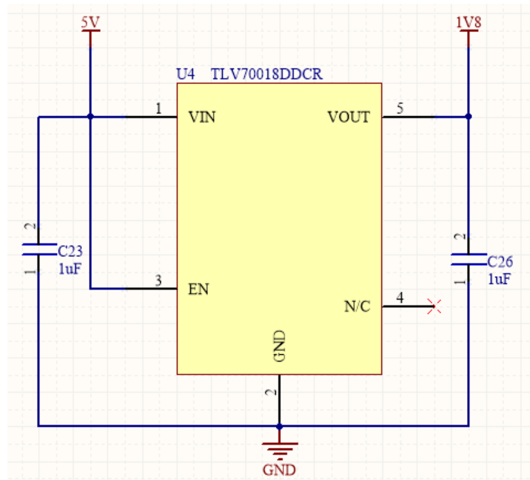

Figure 11 1.8V LDO

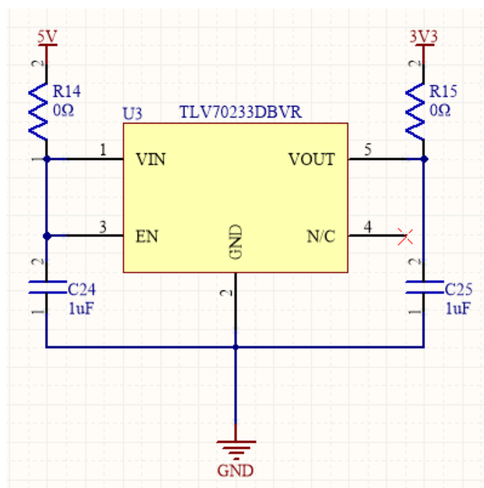

Figure 12 3.3V LDO

3.6 User Inputs & Outputs

The functionality of the TrueSight system is controlled by the user via control knobs and switches **[R064-A, R065-A]**. In order for the user to change between image types, a toggle button is implemented. There are also buttons to turn off and on the HUD, the Jetson TK1, and the entire system. To provide visual feedback to the user, there are LEDs to indicate when the system is booting up and when it is ready. These input/outputs are routed from the Power PCB to the Jetson TK1 Development Board Expansion Header via a ribbon cable. Figure 13 shows the location of the ribbon cable header on the Expansion Header. Table 3 outlines the pinout of the signal header used to connect to the power PCB.

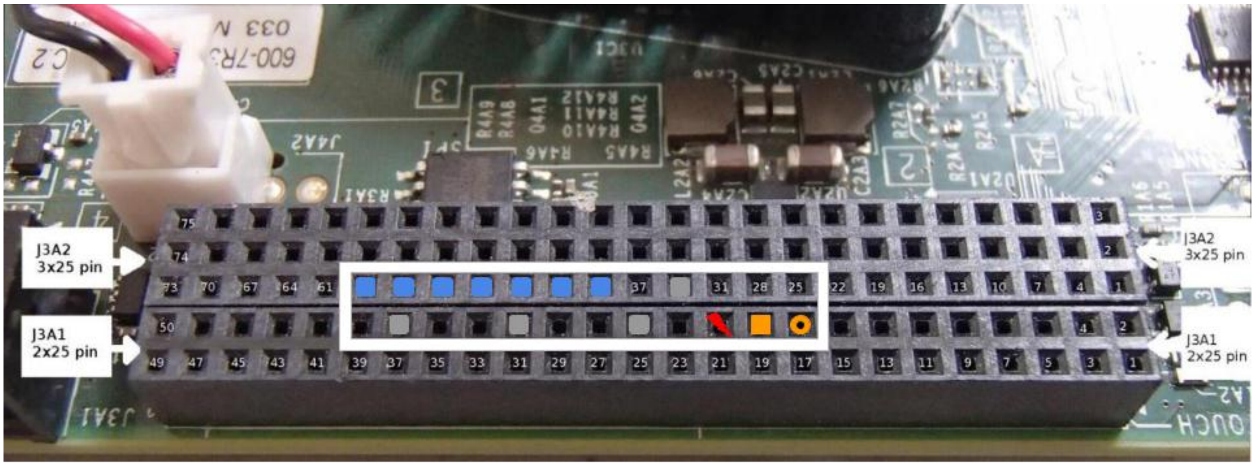

Figure 13 Jetson TK1 Expansion Header

Table 3 TrueSight Signal Header Pin

TrueSight Signal Header	
Symbol	Functionality
	GPIO pins, order from left to right : Bootup LED, Ready LED, Jetson Power Switch, HUD Power Switch, Unused, Display Toggle Switch, Kinect Power
	3.3 V
	Ground
	I^2C Data Signal
	I^2C Clock Signal

In order for the Jetson TK1 to detect the various inputs and control the LED outputs, proper biasing is required. To detect tuning inputs a potentiometer is implemented in a voltage divider configuration and biased to 3.3V shown below in Figure 14. An ADC is used to monitor the change in resistance as a change in voltage between the potentiometer and fixed resistor. The ADC makes the data available via the I^2C bus which is read by the Jetson TK1. Detection of the switches is done directly by the GPIO pins of the Jetson development board. Therefore the switches must be biased to the operating voltage of the GPIO pins of 1.8V as shown in Figure 15**.** To ensure that sufficient brightness is generated by the indicator LEDs, a transistor is used to sink about 26mA of current. The usage of a transistor is essential since the GPIO pin on the Jetson TK1 development board can only provide 4mA of current and a very low voltage. The leaded LEDs used in the prototype have a typical dropout voltage of 2.2V which requires it to be driven by the 5V rail to allow for a reasonable resistor value to provide the needed current shown in Figure 16.

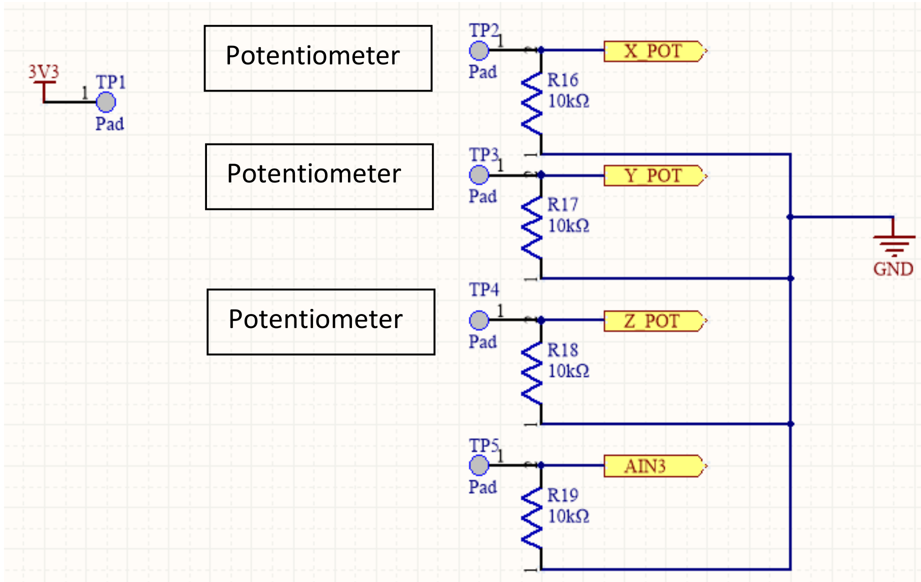

Figure 14 Potentiometer Biasing Circuit

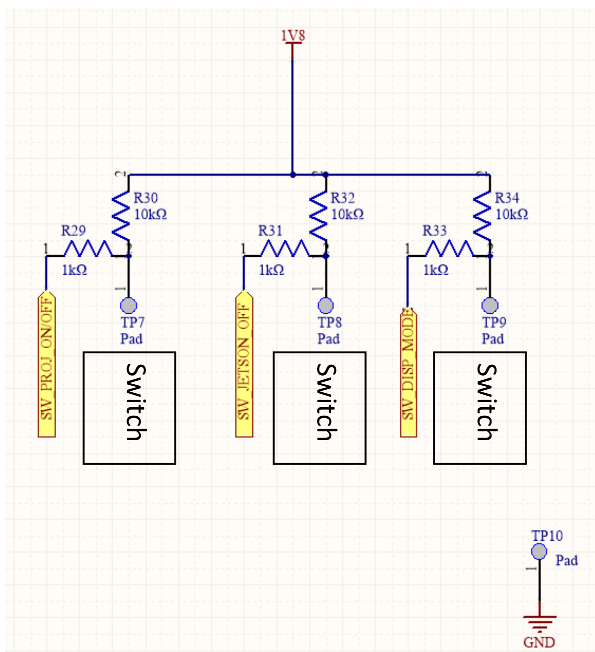

Figure 15 Switch Biasing Circuit

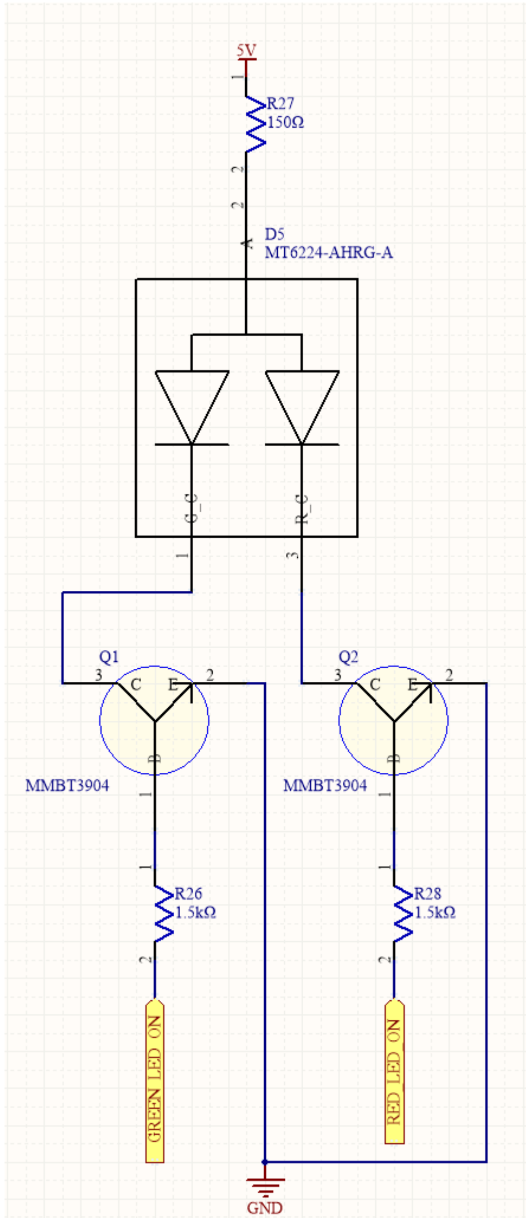

Figure 16 LED Driving Circuit

4. Software Design

4.1 General Software Design

The general purpose software is designed to help each module of the TrueSight product interact with each other. Since the Jetson TK1 was specifically designed for image based applications, its Tegra SoC are utilized in order to create responsive images that are effectively real time by leveraging the hardware-accelerated APIs such as CUDA and OpenCV4Tegra **[R002-A]**. The software design includes the use of low-level GPIO pins as well as the I^2C protocol, which serve as the communication between the user inputs and the software. In order for the Jetson TK1 to interface with the Kinect v2 sensor, we utilize the open source library libfreenect2. The libfreenect2 library acts as the device driver for the Kinect v2 on Linux. The OpenCV library is utilized so that the image grabbed by the Kinect can be output onto our display. OpenCV has many useful properties and functions which help create a clean image and filter out distortion and noise from the Kinect v2's sensors. Figure 17 shows the general software architecture used in TrueSight.

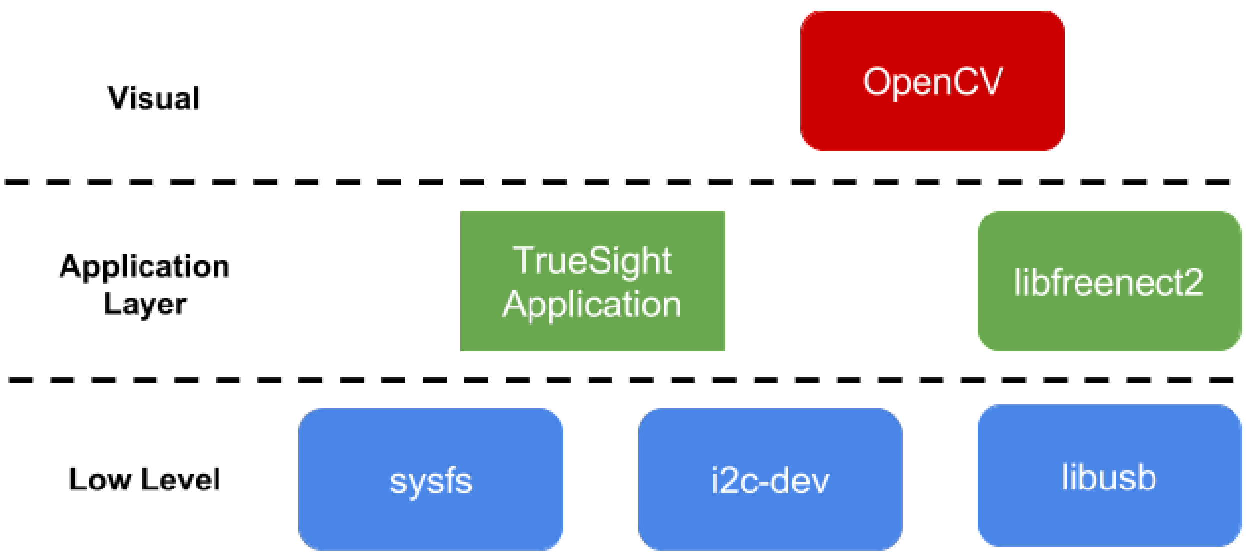

Figure 17 General Software Architecture

4.1.1 Boot-up & General Scripts

To create a seamless product, the software is designed such that it is enabled upon boot-up. This is designed such that the user will not interface with the Linux OS when the display is active. To do this, we will have scripts running to check whether the TrueSight Process *is alive*. If it is not alive, we will simply (re)start it. This will have two benefits: it will enable the application upon boot-up in order for the product to be an embedded system rather than a general purpose computer, and will also take care of possible software crashes.

The display on/off functionality will be controlled through software, rather than cutting power to the display. This will be done by simply enabling/disabling the HDMI port. These scripts will be run in parallel with the main TrueSight program.

4.2 Image Processing

One of the main design features of TrueSight is the ability to create three distinct image types and output these to the user in real time. The three image types chosen from functional specifications, are infrared, depth and edge detection images **[R072-A, R073-A]**. To do the image processing on the data received by the Kinect v2 sensor, we utilized the OpenCV library for its usefulness in image processing. Our goal for the TrueSight product was to design a software architecture which would toggle between each of the image types stated above and provide methods to calibrate the images in order to match the user's field of view. Below is Figure 18, which shows a general flowchart of how each image type is processed and ultimately displayed to the user.

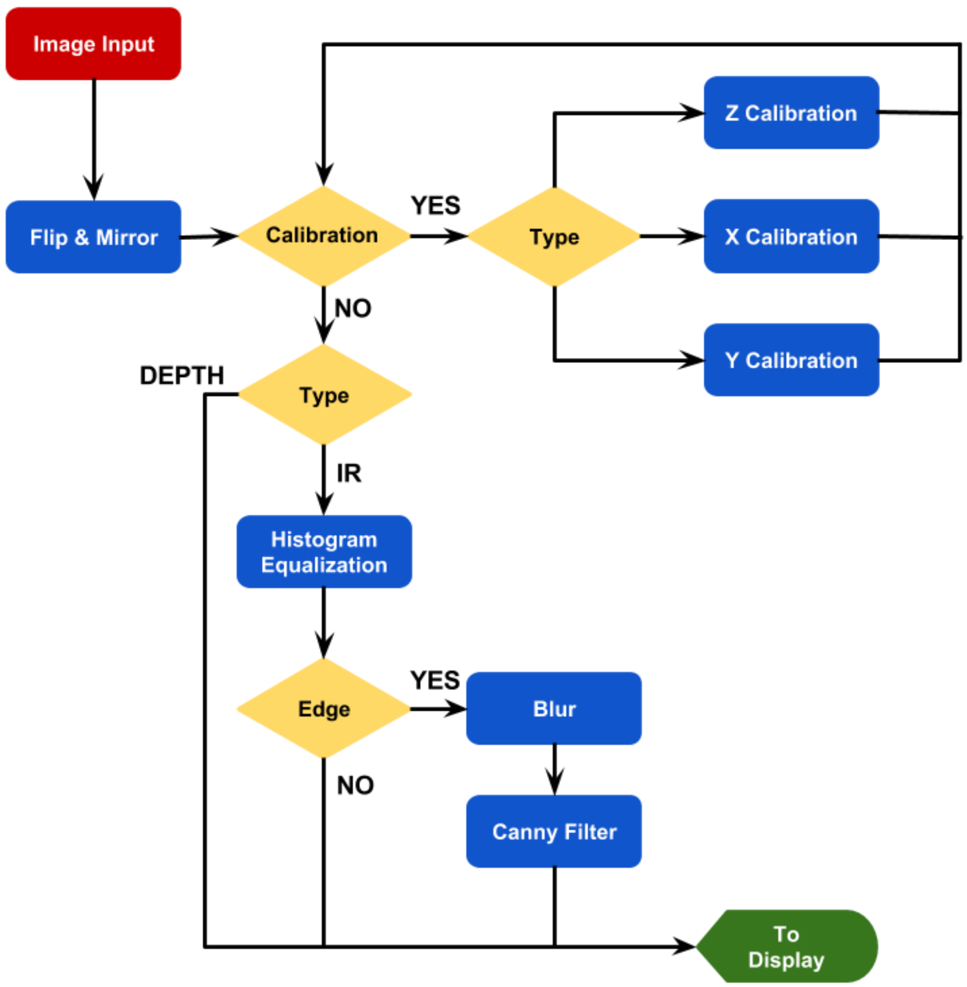

Figure 18 Filtering Flow Chart

4.2.1 Mode Toggle Design

Since a primary function for TrueSight is the ability to change between different image types, a toggle-mode system is designed through software **[R075-A]**. A toggle button signals the software to change the image type each time the button is pressed. Figure 19, shows a state diagram of the mode toggle system. The toggle button is controlled through GPIO pin 43, as seen in Figure 13.

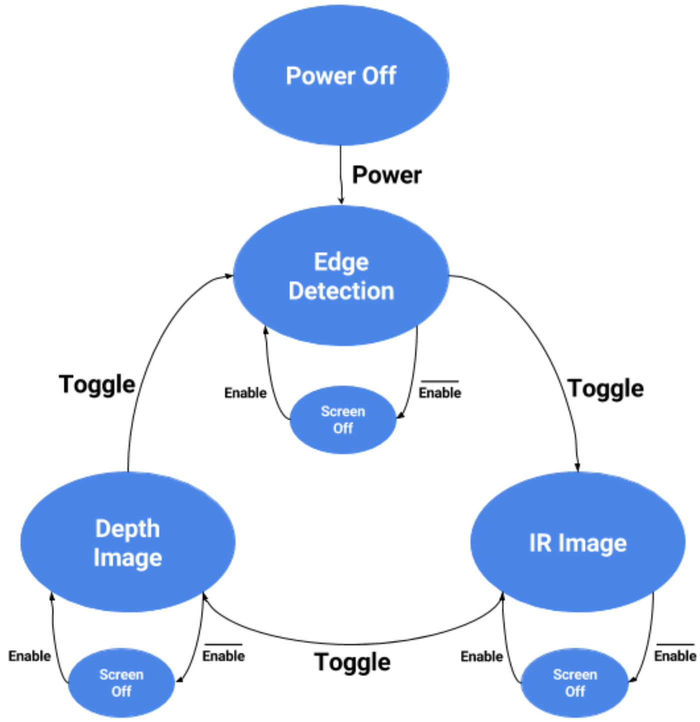

Figure 19 Toggle State Diagram

4.2.2 Image Filtering

In order to ensure the best quality stream for the user, several filters are applied each time a new frame arrives from the sensor. The flowchart in Figure 18, highlights the necessary steps which are taken to achieve the final filtered image. By starting with a base IR image, histogram equalization and blur filters are applied to reduce noise in the image. Histogram Equalization improves the contrast of the IR image by stretching pixels over a larger range of intensities. An example of an equalized image vs non-equalized image is shown below in Figure 20. Though some preliminary testing, the equalized image has proven to increase view range in the edge detected image, as well as produce slightly cleaner edges. Each image type (IR, Edge and Depth) are flipped and mirrored since our display is mounted upside down and reflected to the user on a screen.

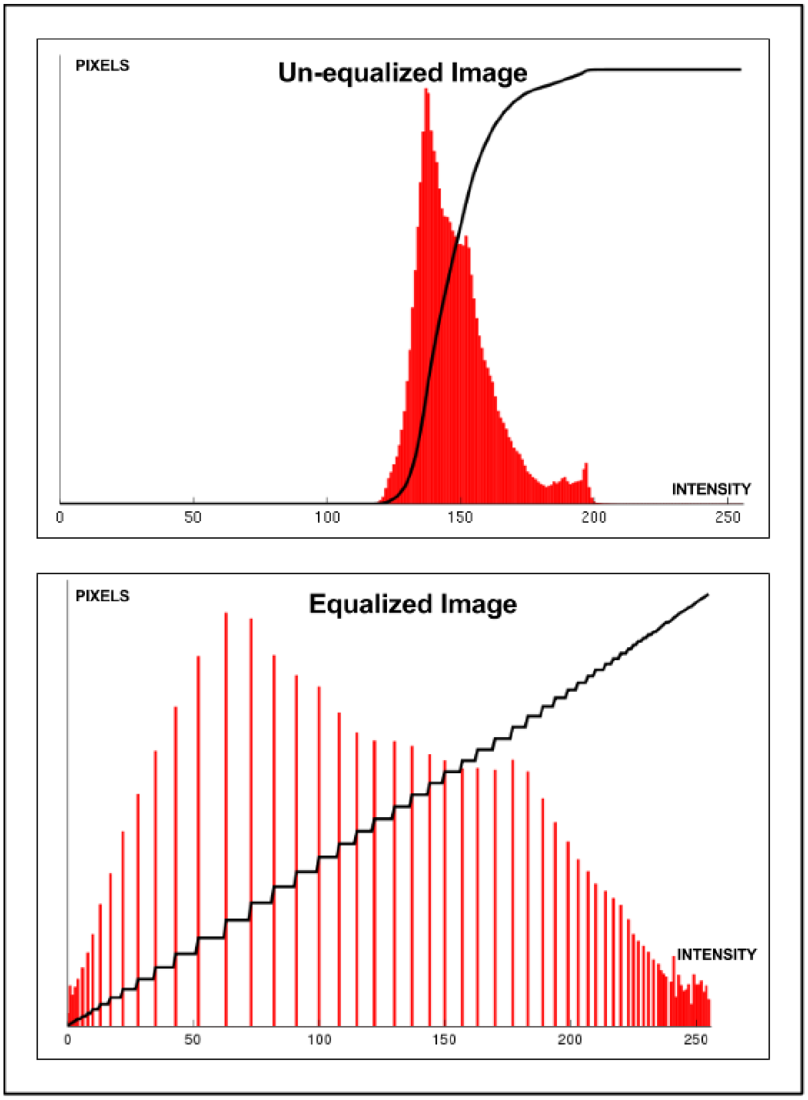

Figure 20 Un-equalized and Equalized Image Intensity Range [5]

4.2.3 Edge Detection

To create a crisp outline of the observable world, multiple steps of image filtering are applied to create a final edge detected image. The principal behind creating an edge detected image is to utilize the Canny edge detection algorithm, named after John Canny. The Canny algorithm is composed of three main steps

1. Produce an intensity gradient of the image
2. Filter to find high *changes* in intensity
3. Hysteresis to intelligently create lines/edges

The OpenCV library provides a Canny filter which combines these steps into one function. This canny filter is then applied to the equalized and blurred IR image.

4.2.4 Zoom Calibration Design

To overlay each image perfectly with the user's current field of view, it is necessary to design several calibration methods so that the image can be fine-tuned. Designing this zoom calibration requires taking a subset of the original image, then scaling that subset to the original image size. By changing a scaling factor through a turn knob potentiometer, the zoom of the image is changed. Figure 21 shows a diagram of the zoom feature. From the diagram, the subset image width and height is controlled by a scaling factor ***z***, where

$$Width = \frac{columns \times z}{100}$$

$$Height = \frac{rows \times z}{100}$$

The width and height of the subset image will act as the "zoomed" image. However, the zoomed image needs to be scaled back up to the original image resolution, so the size isn't constantly changing. To grab the correct subset, the range of the image must be determined, which can be observed as range from x_1 to x_2, and y_1 to y_2 in Figure 21. The equations to find central location of the subset image are as follows.

$$x_1 = \frac{columns - width}{2}$$

$$x_2 = columns - \left(\frac{columns - width}{2}\right)$$

$$y_1 = \frac{rows - height}{2}$$

$$y_2 = rows - \left(\frac{rows - height}{2}\right)$$

Once the central image range is obtained, it is then stretched to the original size. The reason this design works is because each frame is grabbed in rapid succession. This means the zoom can be continuously applied to the previous subset frame.

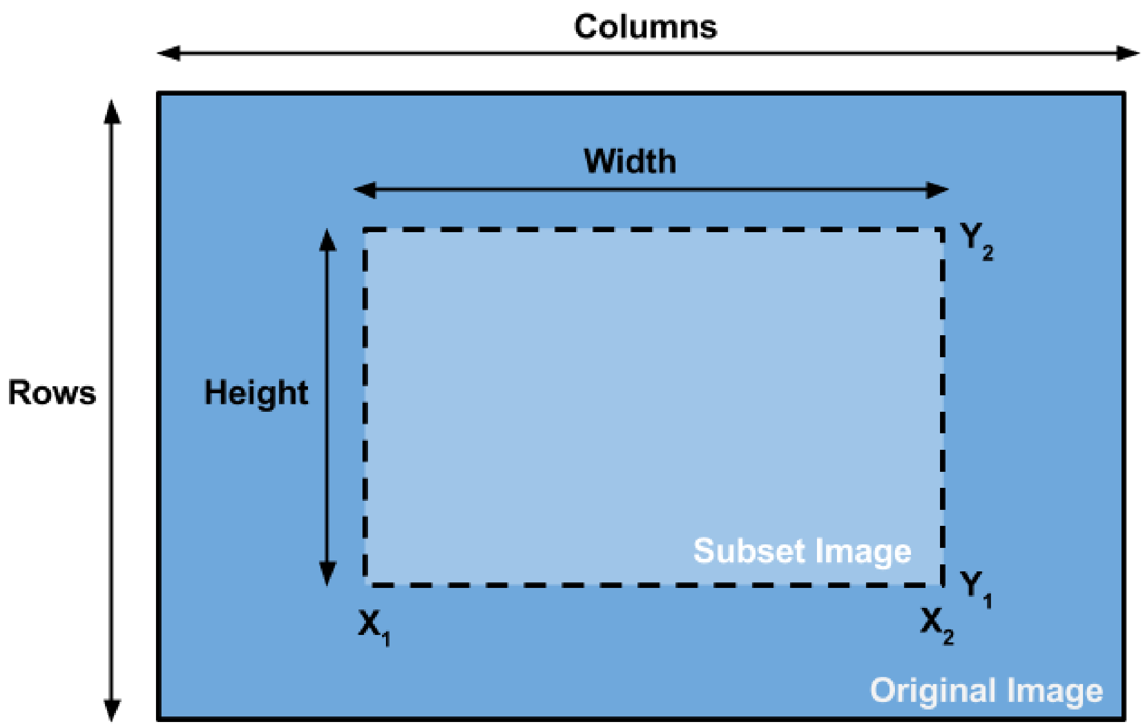

Figure 21 Zoom Algorithm

4.2.5 X and Y Calibration Design

When the user equips the display, the position of the screen will need to be calibrated relative to their field of view. Since the Kinect v2 field of view is limited compared to a human's field of view, the displayed image will need to be zoomed as well as positioned properly in the XY plane. To move each image in the X and Y direction on the screen, the software will poll the control turn knobs, via I^2C, which calibrates the image in the X and Y position. Since the size of the displayed image is smaller than the total screen size, the image is surrounded by a "border" of black, blank space. This extra space can be used to move the image to the correct location for the particular user, as well as be used to display miscellaneous data.

4.2.6 Image Translation Design

Image translation is required to reorient the image. Since the Kinect sensor position is on-top of the user's helmet, its line of sight is higher than the user's normal line of sight. The image processing library OpenCV allows us to apply several geometric transforms by deforming the pixels of the original image and placing them into a map of the correctly oriented image **[R076-B]**. Figure 22 which shows an example of a simple geometric transform using OpenCV.

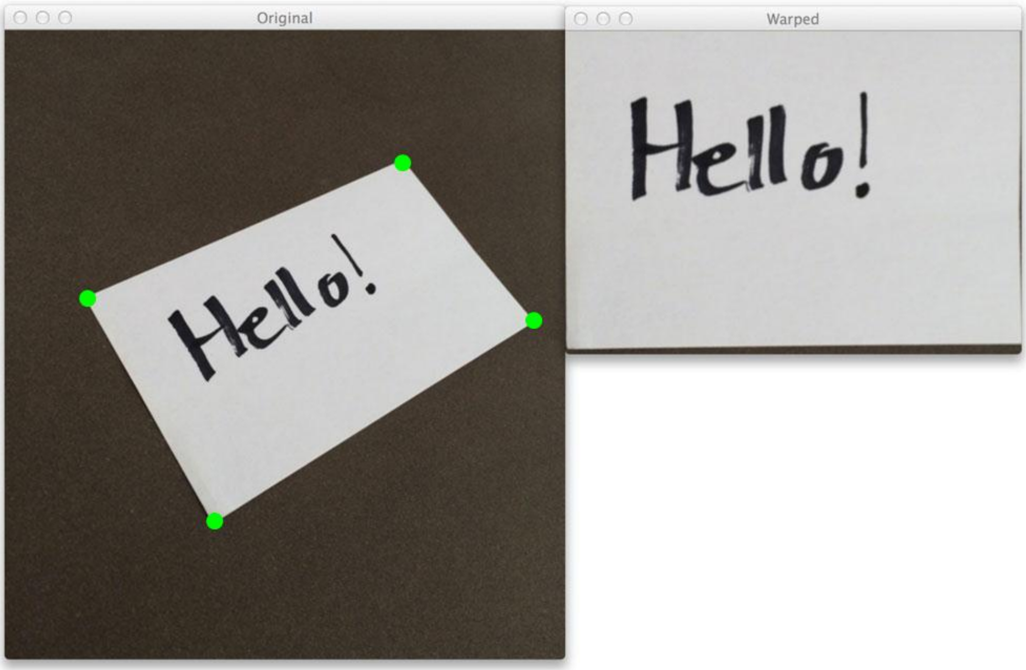

Figure 22 Image Geometric Transform Example [6]

4.3 Firmware

4.3.1 I2C

I^2C protocol is used to communicate with our ADC - the ADS1015. The Jetson TK1 comes with I^2C busses available via the expansion header. The ADC provides us with the potentiometers', or control knobs', values so that we can adequately calibrate/control our visual display. The Linux kernel provides an interface, I^2C -dev, that eases the integration of the I^2C protocol. Our software will poll the 4-input ADC for 3 analog inputs that will correspond to the X, Y, and Zoom calibration values. The fourth analog input is left unconnected and may be used to solve any unforeseen problems at this point.

4.3.2 GPIO

The Jetson TK1 development board comes with GPIO pins accessible in software via sysfs. The GPIO pins are located on the expansion header and will be connected via a ribbon cable to status LEDs and control switches. Figure 13 shows the GPIO pins used on the expansion header. The GPIO switches will be used to control various aspects of the TrueSight application, including the state flowchart in Figure 19, as well as toggling power to various modules of the True Sight Product **[R079-A]**.

5. Test Plan

Absolute Vision System's engineers have developed a two phase test plan to ensure proper functionality of the TrueSight prototype. The first phase will test the main functionality of each of the components. The second phase will evaluate the integrated hardware and software system. Individual components will be tested to ensure safety, reliability, and that they meet the expected functional specification from the datasheets. The prototype will be tested on integration between the separate components, ensuring all systems are running at the optimal level.

5.1 Schedule

Figure 23 details our current project timeline. The test plan will be carried out for 25 days total from November 20 to December 5, giving us ample time to troubleshoot and address any unforeseen issues.

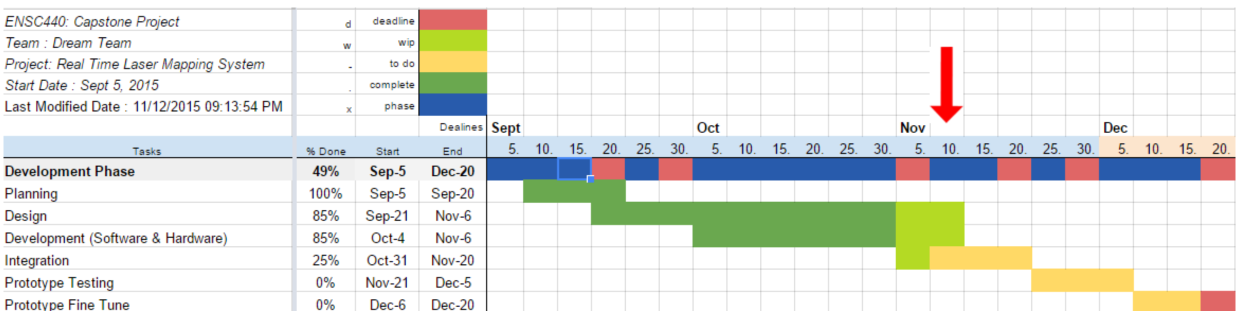

Figure 23 Project Timeline

5.2 Phase 1 Testing

5.2.1 Hardware

The hardware test plan will focus on the power management board. Omni, the PCB manufacturer tasked with making the boards, will perform a standard e-test to ensure the boards are free of shorts, opens, and broken traces. Several tests must be performed prior to integration to ensure the safety of the rest of our components.

1. Solder joints do not connect to separate nets
    a. Free of cold solder joints
2. Open & short testing
3. Jetson 12V rail verification
    a. Level accuracy
    b. Voltage ripple
4. Kinect 12V rail verification
    a. Level accuracy
    b. Voltage ripple
    c. Kinect enable
5. 5V rail verification
    a. Level accuracy
    b. Voltage ripple
    c. 5V USB output

   d. 3.3V LDO input
   e. 1.8V LDO input
   f. LED input
6. 3.3V rail verification
   a. Level accuracy
7. 1.8V rail verification
   a. Level accuracy
8. GPIO pins match Jetson header pinout

5.2.2 Firmware

The firmware test plan will be used to determine whether the interfaces used in system integration are functioning properly and can be controlled in Linux on the Jetson TK1. The following interfaces will be thoroughly tested.

1. GPIO LEDs
2. GPIO Switch
3. I^2C bus
4. USB 3.0
5. HDMI

5.3 Phase 2 Testing

5.3.1 Functional Tests

To ensure the software functionality of TrueSight is working as intended, the following tests will be performed.

1. Image type text is viewable in blank space surrounding video stream
2. Changing between image type changes the image type text accordingly
3. Multiple cycle test - image text
   a. Cycle through image types via toggle button 25 times.
   b. Correct text is displayed whenever image type is switched
4. Multiple cycle test - image type
   a. Cycle through image types via toggle button 25 times
   b. Verify that image type is correct for each toggle.
   c. Verify that the image order is Edge Detect -> IR -> Depth
5. Startup test
   a. Verify that upon startup of the TrueSight software that the edge detected image is displayed
6. Mode toggle responsiveness
   a. Cycle between image types 3 times
   b. Verify that between each toggle press, there is less than a maximum of 1 second delay between button press and image toggle
7. Z-Calibration test
   a. Adjust the Z axis potentiometer from 0 - 100% zoom

    b. Verify that the image zooms, while not changing the screen size
8. X- and Y-Calibration test
    a. Adjust the X axis potentiometer from 0 - 100% pan
    b. Verify that the image pans, while not disrupting the image resolution or size
    c. Repeat steps a-b for Y calibration potentiometer
9. Calibration responsiveness
    a. Adjust the Z calibration potentiometer a small amount
    b. Measure time it takes to update the image stream
    c. Verify the responsiveness is less than a maximum of 1 second
    d. Repeat steps a - c for X calibration potentiometer
    e. Repeat steps a - c for Y calibration potentiometer
10. Simultaneous Adjustment - XY
    a. Adjust the X and Y knobs simultaneously, from 0 to 100% pan
    b. Measure time it takes for image stream to update
    c. Record the end position of the screen, with the X and Y knobs panned fully
    d. Individually adjust the X potentiometer to 100% pan
    e. Individually adjust the Y potentiometer to 100% pan
    f. Record the position of the screen.
    g. Verify that the position of the screen from position c and f match
    h. Repeat steps a-g with XZ and YZ combinations
11. Boot-up
    a. Switch on main power to device
    b. Verify that boot-up LED is on
    c. Record time it takes from switching on to displaying edge detected stream
    d. Verify that time to boot is approximately 1 minute
    e. Verify that once the TrueSight application is running, the Ready LED is on
12. Projector HDMI OFF and Calibration Potentiometers
    a. With projector HDMI port off, change calibration potentiometers
    b. Turn project HDMI port on
    c. Verify that the image zoom and/or pan has changed

5.3.2 Integration Tests

Once each subsystem has passed modular testing, the TrueSight system, as a whole, will be tested to ensure that it performs as described. The proof of concept will focus on the following:

1. User mobility
    a. User movement does not interfere with system processes
2. Battery durability test
    a. Battery powers system at least 30 minutes
    b. Battery pack is easily replaced
3. Power switch test
    a. System power switch
    b. Jetson power switch
    c. HUD power switch
4. I^2C test
    a. Master/Slave configuration and control
    b. Serial Data Communication

5. GPIO
    a. Able to write to Status LEDs
    b. Able to read from control switches
6. Display toggle switch controls the software functionality
7. The system boots up in under 1 minute
8. Image is displayed on HUD

6. Conclusion

Throughout the design specification document, we have discussed the design choices made to create a product which will ultimately enhance human vision in low visibility environments. We have focused on why we chose particular modules and components, as well as the theory and motivation behind our design choices. Our design took into account future improvements and changes which could be added to create a polished final product. Also, a two phase test plan was devised to test TrueSight's desired functionality with our design solutions. We hope that True Sight will be a unique and innovative addition to the firefighting arsenal.

7. References

[1] Absolute Vision Systems, "Functional Specification for Real-Time 3D Laser Scanning Device," 19 10 2015. [Online]. Available: http://www2.ensc.sfu.ca/~whitmore/courses/ensc305/projects/2015/hfunc.pdf.

[2] NIVIDA Corporation, "Next-Gen Smartphones, Tablets Devices | NVIDIA Tegra | NVIDIA," 2015. [Online]. Available: http://www.nvidia.com/object/tegra-phones-tablets.html.

[3] Intel Corporation, "Intel Universal Serial Bus (USB) Frequently Asked Questions (FAQ)," 2015. [Online]. Available: http://www.intel.com/content/www/us/en/io/universal-serial-bus/universal-serial-bus-faq.html. [Accessed 2015].

[4] eLinux, "Jetson/Computer Vision Performance," 2015. [Online]. Available: http://elinux.org/Jetson/Computer_Vision_Performance#Power_draw_during_computer_vision_tasks.

[5] Konstable, "Equalized Histogram - Histogram Equalization," 2015. [Online]. Available: https://en.wikipedia.org/wiki/Histogram_equalization#/media/File:Equalized_Histogram.svg.

[6] PyImageSearch, "4 Point OpenCV getPerspective Transform Example," 2015. [Online]. Available: http://www.pyimagesearch.com/wp-content/uploads/2014/08/getperspective_transform_01.jpg.