Drone Monitoring of Power Lines

ECE 480 Senior Design

Jake Hersha

Mitch Johnson

Ian Meredith

Dan Pittsley

Faisal Tameesh

Cody Wilson

Facilitator:

Dr. Tongtong Li

Sponsored by:

Consumers Energy

Count on Us

Executive Summary:

Consumers Energy has been researching a more efficient solution for the inspection of their power lines. This project has accomplished that objective by employing a drone to perform remote power line inspections. Thermal image inspection of power lines is an industry-wide method for detecting present or potential faults in power lines and other electrical equipment. Electric utility companies currently perform their inspections by either using handheld thermal cameras or using thermal cameras mounted to helicopters. These methods are both problematic as helicopter operation is expensive and using handheld thermal cameras in remote areas is dangerous and difficult. To address this issue, the team's goal was to create a drone-mountable system to be used in the detection of defective power lines. Detection is performed using a thermal camera and microprocessor mounted on a drone. Video from the thermal camera is transmitted to a base station for human viewing. When a problem is detected, the operator instructs the system to record the thermal video and a GPS location of the problem area. All problem areas are subsequently plotted on Google Maps for further inspection. Each plot-point will contain the video recorded at that location.

Acknowledgements:

We would like to thank Amanda Monette and Andy Bordine from Consumers Energy for their support throughout the project. We sincerely appreciate their prompt assistance throughout the semester. We would also like to show our gratitude to Dr. Tongtong Li for her valuable insights and guidance.

Table of Contents

I. Introduction and Background

Power line maintenance for electric utility companies is expensive, dangerous, and time-consuming. Therefore, Consumers Energy has been researching alternative methods for detecting faults on power lines. Their main method for fault detection is thermal hot-spot recognition, which uses a thermal camera to inspect power lines for high-impedance areas, also called “hot-spots” or “problem areas”. These high-impedance areas generate more heat than the surrounding equipment and are thus visible on the thermal camera. Also, these areas signal that there is a fault on the power line. It is often practical for a person on the ground to use a thermal camera to monitor the power lines, but this can be laborious and time-consuming. Additionally, this method is impractical when power lines are located above rugged terrain. Consumers Energy currently solves this problem by using a thermal camera in a helicopter. Helicopter use is also problematic as it is very expensive - costing $1,200 per hour. This cost is exceedingly prohibitive if routine monitoring is desired. In addition, helicopters generate too much noise and cannot be safely flown above power lines that provide power to residential areas. Therefore, Consumers Energy employed Team 14 to explore the feasibility of using a drone to remotely monitor their power lines.

The goal of Team 14’s design project was to create the components and software that would be used by the drone to make power line monitoring more effective and efficient. The objective of the design is to simultaneously stream and analyze video from two different cameras - a thermal camera and a regular video camera. When a problem area is detected in the video feed, the operator can signal the system to record video of the problem area. The video is subsequently uploaded to a back-end and plotted on a Google Map. This design is shown in Figure 1.1.

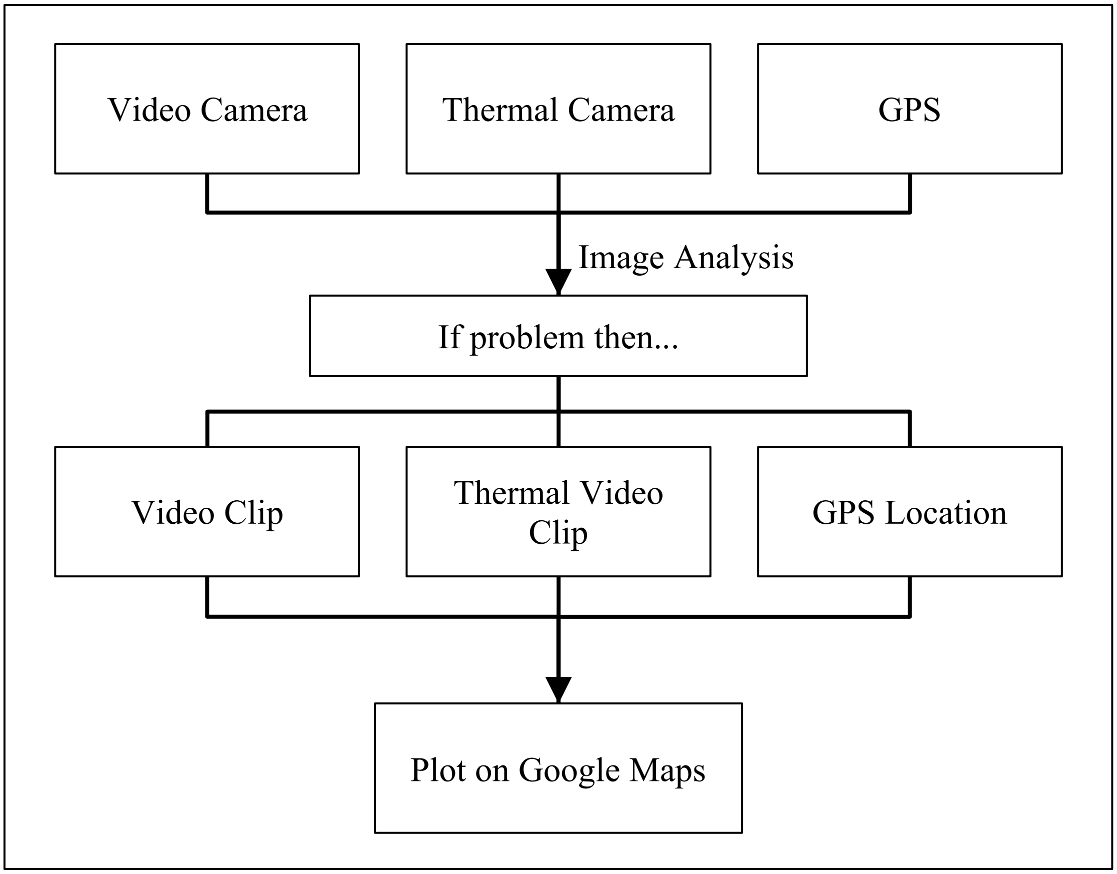


Figure 1.1 Design Objectives.

As drones are becoming more prevalent, utility companies can benefit advantage of the latest technologies. The system would make effective use of drone technology while eliminating the expense of using helicopters. Unlike previous power line monitoring systems, this design will be capable of uploading all of the information to a back-end database for future analysis. This will allow utility companies to review the findings and locations of all of their recorded problem areas. The database of these recorded problem areas could then be used for future detection methods. With the front-end software communicating to the utility companies servers, they will be able to instantly deploy crews to high priority fault sites. Implementation of this design would allow crews to assess problems at a greater distance, creating a safer environment for workers.

II. Exploring the Solution Space and Selecting a Specific Approach

For this project, the main goal was to plot power line faults on Google Maps. To do so, it was necessary to record video, record location, and detect power line faults. The drone must capture video by hovering over the power lines. For fault detection, a human operator must analyze the video feeds coming from the drone. The video feeds are transmitted from the drone to a base station via analog transmitters and receivers. After gathering the videos, the operator's console at the base station will query the GPS on the drone to extract the GPS data, which will then be sent to the back-end. After completing these three steps, it will be possible to plot all power line faults on a Google Map. A FAST Diagram of these functions is shown in Figure 2.1.

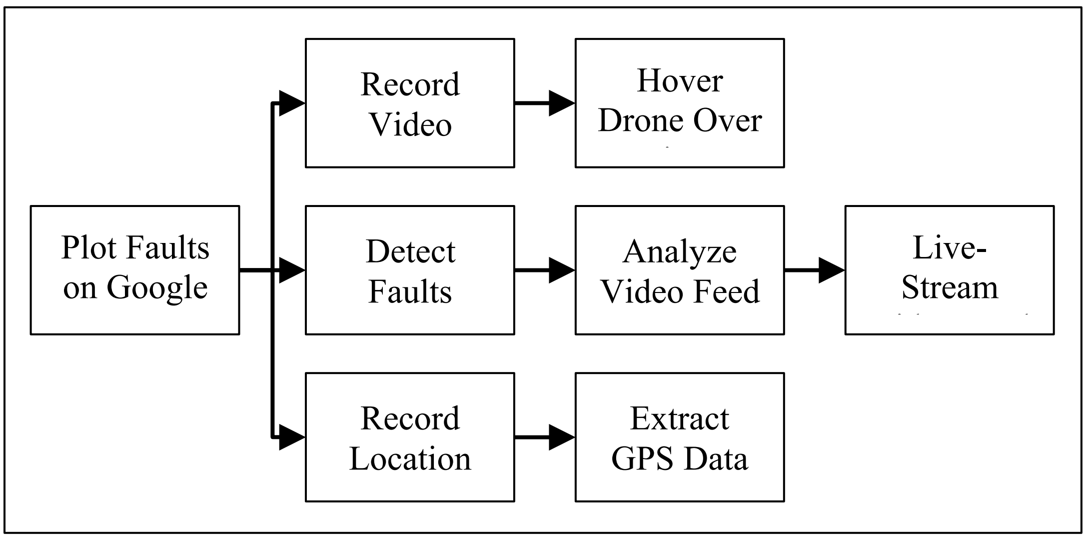


Figure 2.1. FAST Diagram.

For each of the three basic functions (record video, record location, detect faults), it was possible to implement them in many unique ways. This required the team to make design decisions using the sponsor's specified requirements. The following sections describe the team's design approach for each basic function.

Recording Video

The process of recording video involves utilizing a thermal camera for infrared video and a GoPro camera for regular, non-thermal video. The solution space for this function did not involve a choice of cameras, as these devices were chosen and provided by the sponsor. Controlling the drone to hover over power lines was outside the scope of the design.

Detecting Faults

In order to detect faults on power lines, it was necessary to live stream video from the drone to the base station. The team conceived two separate methods to accomplish this function. The first method involved connecting each camera on the drone to a microprocessor, and transmitting both video feeds to the base station over Wi-Fi. The second method was to transmit the video feeds through analog transmitters. These two methods are shown in Figure 2.2.

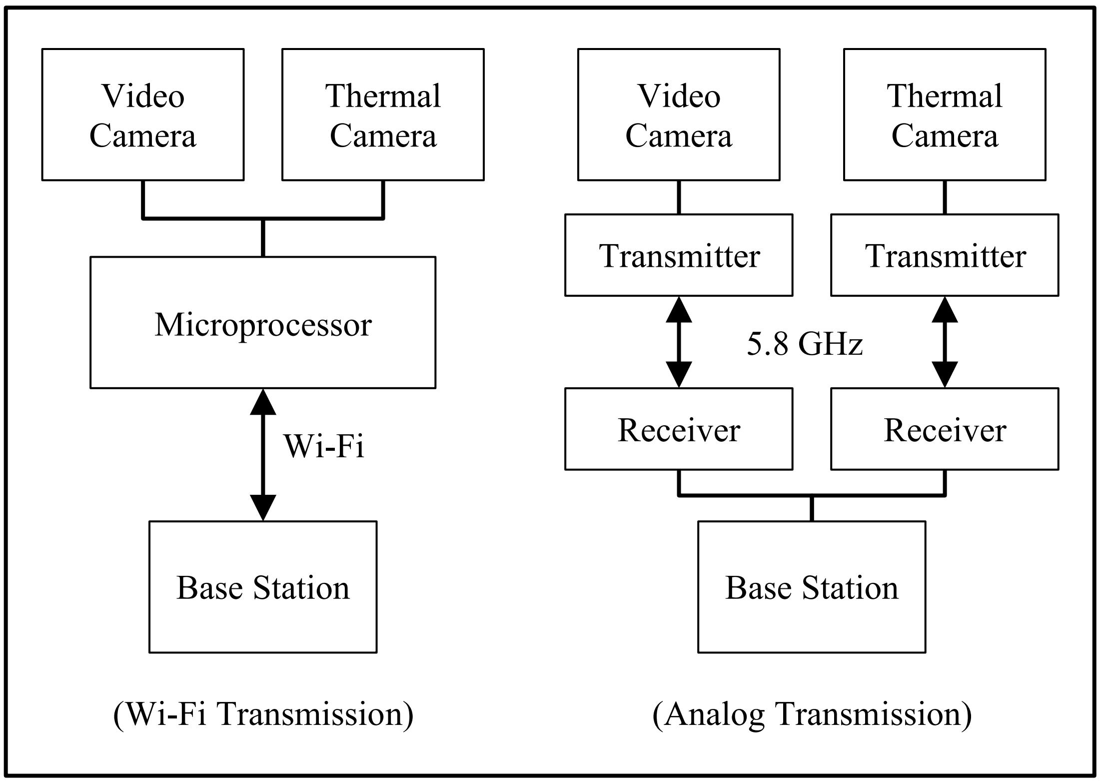


Figure 2.2. Possible Designs for Video Transmission.

The team originally thought that Wi-Fi transmission would be the optimal design in order to eliminate the cost and weight of transmitters on the drone. However, this choice proved to be impractical, as Wi-Fi transmission required the microprocessor on the drone to simultaneously convert both analog video feeds to digital. This capability required a very powerful processor; it was discovered that incorporating such processing power on the drone would not be practical given the weight and power constraints. The team therefore modified the design to eliminate analog-to-digital conversion on the drone, and instead incorporate analog transmission.

In the analog transmission design, the video streams are transmitted from the drone using analog transmitters and receivers. Analog-to-digital converters at the base station transfer information from the receivers to a Linux-based computer - the base station. A human operator who monitors the videos for hot-spots then analyzes the video feeds at the base station. When a hot spot is detected, the operator records the video feeds and uploads them to the back-end. The two videos are then stored in a database with their associated GPS coordinates. This design allows for faults to be detected and recorded from the base station while maintaining reasonable weight and power requirements. The problem areas can then be plotted on a Google Map for further analysis.

Recording Location

Another basic function of the project was to record location of the detected problem areas. To accomplish this, it was necessary to relay location information from a GPS on a drone to the base station. The drone provided by the sponsor already had a stock GPS attached to the drone, but using the provided GPS would present some issues. Thus, the team was faced with a design decision - choosing between using the provided GPS or obtaining another GPS. Each design option had advantages and disadvantages.

The main benefit of using the provided GPS was to avoid extra weight and power consumption on the drone. However, there were several disadvantages of using the provided GPS. First, the GPS operated on proprietary software from the drone manufacturer, which makes programming more difficult than a non-proprietary GPS. Second, the team had limited access to the drone during the semester as the drone was in Jackson, Michigan while the team operated in East Lansing, Michigan. This situation would cause difficulty with testing. The third disadvantage dealt with modularity; because the software on the provided GPS was proprietary, additional drones purchased in the future would need to be from the same manufacturer in order to operate with the designed software.

For these reasons, the team chose to obtain a separate GPS to mount on the drone. This decision caused the additional cost, weight, and power of a GPS as well as a microprocessor for the GPS to connect to. However, using a new GPS has many advantages. First, this design method allowed the team to easily test functionality of the system. Second, it was much easier to program the GPS as the output stream was not proprietary and could easily be read. The most important advantage is that, when using an off-the-shelf GPS, the entire system can be mounted on any drone with no compatibility issues

The GPS was used to record the drone's latitude and longitude while recording a problem area. These locations would later be plotted on a Google Map. This was achieved by using the GPS receiver in conjunction with the microprocessor, which constantly reads the GPS data. When the operator selects the upload button on the front-end, the microprocessor is pinged for a coordinate pair. Afterwards, the two videos and GPS coordinates are uploaded to the back-end, where the coordinates are recorded in a SQLite database, and the videos are uploaded to the server with their file locations stored in the same database entry. Completing this process will successfully record the GPS data, which will ultimately be used to plot the problem areas.

Component Selection

After the overall design was outlined, it was necessary to select specific hardware components to be used for the drone and base station. The team had several options with regard to the type of processor, GPS and power converter to be used. The list of options was reduced to a single component using several design criteria. The potential processors, GPS's, and types of power converters, along with the project's design criteria and relative importance rankings can be seen in Table 2.1. From this figure, it can be seen that the Raspberry Pi processor and BU-53S4 GPS were chosen. Also, a switching power converter proved to be the most optimal for the project.

Engineering Criteria	Importance	Possible Solutions							
		Processor				GPS		Power Converter	
		Pi	BBBlack	Laptop	Pi2	BU-353S4	Spark Fun	Switching	Linear
Power Consumption	5	9	9	1	9	9	9	9	1
Processing Power	3	3	9	9	9				
Software Availability	4	9	3	9	3				
Weight	4	9	9	1	9	9	9	9	3
Size	5	9	9	1	9	9	9	9	1
Cost	3	9	3	1	9	3	3	3	9
Programmability	5	9	3	9	3				
System Compatibility	5					9	3		
Durability	3					3	1		
Total		243	189	125	207	189	153	135	49

Table 2.1. Solution Selection Matrix.

Project Schedule

In order to achieve timely completion, it was necessary to separate and plan the comprising elements of the project; this was done using a Gantt chart. The Gantt chart divided the project into three main stages - project definition, research, and design. Each main task was divided into several subtasks. As the team progressed and became more familiar with possible design solutions, the Gantt chart evolved to become more appropriate. This is discussed further in Chapter 5. The originally devised Gantt chart tasks are shown below in Table 2.2.

		Task Mode	Task Name	Duration	Start	Finish	Predecessors
1			**Project Definition**	**2 days**	**Fri 1/23/15**	**Mon 1/26/15**	
2			Meet with Company Sponsor	1 day	Fri 1/23/15	Fri 1/23/15	
3			Meet with Faculty Advisor (Dr. Li)	1 day	Mon 1/26/15	Mon 1/26/15	2
4			Buy materials (R Pi, GPS) + Delivery	4 days	Mon 1/26/15	Thu 1/29/15	2
5			Create/Start webpage	5 days	Mon 2/2/15	Fri 2/6/15	
6			**Research Stage**	**9 days**	**Mon 1/26/15**	**Thu 2/5/15**	
7			Analyze input stream from GoPro	9 days	Mon 1/26/15	Thu 2/5/15	2
8			Analyze input stream from GPS	9 days	Mon 1/26/15	Thu 2/5/15	2
9			Research Power Converter	5 days	Mon 1/26/15	Fri 1/30/15	2
10			Analyze input stream from thermal camera	9 days	Mon 1/26/15	Thu 2/5/15	2
11			**Design Stage**	**65 days**	**Mon 2/2/15**	**Fri 5/1/15**	
12			Test Power converter	1 day	Fri 2/13/15	Fri 2/13/15	
13			Make power converter	11 days	Mon 2/2/15	Mon 2/16/15	9
14			Design program to compile data streams	15 days	Fri 2/6/15	Thu 2/26/15	6
15			Test data stream compiler	5 days	Fri 2/27/15	Thu 3/5/15	14
16			Report hot spots from data stream	20 days	Fri 2/27/15	Thu 3/26/15	14
17			Test hot spot reporting	5 days	Fri 3/27/15	Thu 4/2/15	16
18			Report only hot spots on power lines	16 days	Fri 3/27/15	Fri 4/17/15	16
19			Test power line isolater	5 days	Mon 4/20/15	Fri 4/24/15	18
20			Integrate modular programs	5 days	Mon 4/20/15	Fri 4/24/15	18
21			Prepare for design day	5 days	Mon 4/27/15	Fri 5/1/15	18,20,19
22			Complete Final Proposal	4 days	Mon 2/16/15	Thu 2/19/15	
23			Finish Final Report	5 days	Wed 4/22/15	Tue 4/28/15	

Table 2.2. Original Gantt Chart Tasks.

Expected Budget

Figure 2.3 shows the expected cost of each component in the project; together, they total \$182.72. Note that the two cameras (thermal and video cameras) are not included in the budget, as the sponsor provided these. If other systems are to be made in the future, the budget should include an additional \$700 and \$300 for the thermal and video cameras, respectively. Also, the cost of the drone is \$5,000. Thus, the project had an expected budget of less than \$200, while an entire system should cost about \$6,200.

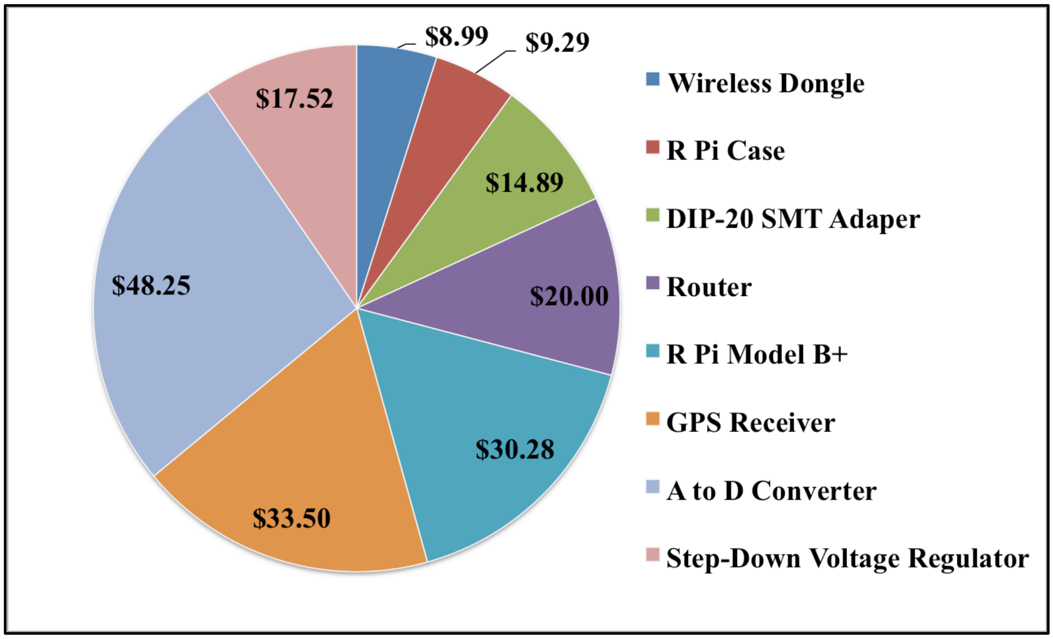


Figure 2.3. Expected Budget.

III. Technical Description of Work Performed

Hardware Design

Figure 3.1 depicts the all of the hardware involved in this project, except for a Linux-based computer that can interface with the analog-to-digital converters. The system's architecture allows for the swapping of any hardware components with higher quality parts. This flexibility allows for a smaller prototyping budget and a more modular design.

The components to be mounted on the drone, pictured on the platform in the front of Figure 3.1, consist of the following:

- FLIR Tau 2 thermal camera
- GoPro Hero3 camera
- Raspberry Pi 2
- Wi-Fi dongle
- GPS Receiver
- Two analog video transmitters
- USB Hub
- Power Converters (pictured in Figure 3.2)

The components at base station are pictured on the platform in the back of Figure 3.1. That platform contains the following components:

- Two analog video receivers
- Two analog-to-digital converters
- Router to keep the system under the same network
- Linux-based computer (not pictured)

Figure 3.1. Prototype Components.

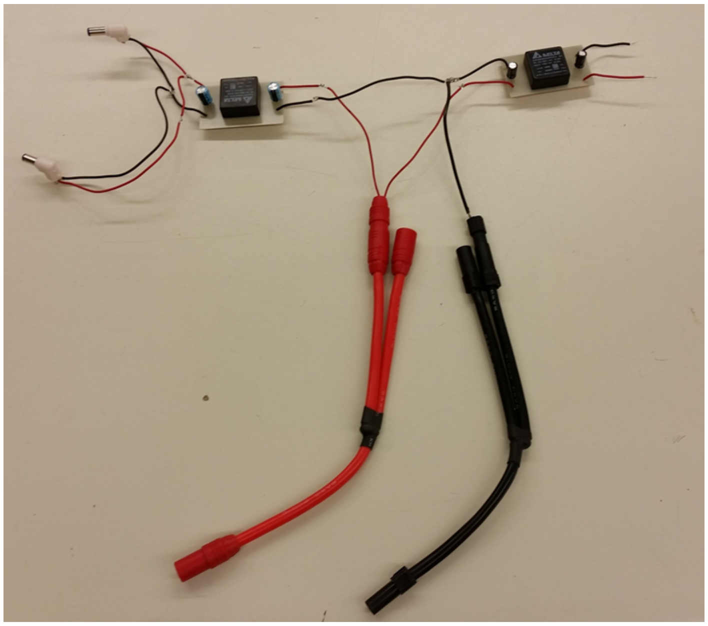


Figure 3.2. Power Converter System.

FLIR Tau 2 Thermal Camera

The FLIR Tau 2 thermal camera is the main component used to detect faults on power lines. It is an uncooled thermal camera with dimensions of 1.75 in. by 1.75 in. by 1.18 in. The camera outputs an analog video stream. Specifications of the camera can be found in Appendix III. The sponsor provided this camera.

GoPro Hero3

GoPro Hero3 was used to get a video feed of the area surrounding faults detected by the thermal camera. It also gives the operator a better understanding of the fault type, as it can be difficult to deduce from the thermal image. The GoPro Hero3 has an internal battery, which makes it a more power efficient camera, as it does not draw power from the drone battery. The sponsor provided this camera.

Raspberry Pi 2

The Raspberry Pi 2 takes the input location from the GPS and sends it to the base station through a wireless dongle. The Raspberry Pi 2 will power the GPS. The team originally planned to use the Raspberry Pi 2 to transmit both video feeds through the wireless dongle. However, this approach was ultimately infeasible because the Raspberry Pi 2 could not process the bandwidth of both video feeds. The live streams were choppy due to frame losses. This led to using analog transmitters to transmit the video feed.

GPS

The GPS used is the GlobalSat BU-353S4. It is a USB powered GPS receiver. The BU-353S4 is a serial device and outputs GPS information in NMEA format.

Analog Video Transmitters/Receivers

Monoprice 5.8Ghz Wireless Audio/Video Transmitters/Receivers were used to send an analog video stream to the base station. There are multiple channels to reduce interference between the two transmitters. These transmitters operate at 12 V and 250 mA. The cameras feed into the two analog video transmitters, which transmit the analog signal to the receivers at the base station. The transmitters have a range of about 200 ft. with clear line of sight.

USB Hub

The USB hub used is the Plugable USB 2.0 10-port high-speed hub. The USB hub will power the FLIR TAU 2 thermal camera and the Raspberry Pi 2. The GPS will be indirectly powered by the USB hub through the Raspberry Pi 2.

Analog to Digital Converters

The Hauppauge 610 USB-Live 2 Analog Video Digitizer and Video Capture Device was used as the analog-to-digital converter (ADC). There are two ADCs, because there are two video feeds. Each ADC connects from the video receiver to the base station.

Router

The components on the drone are all operating under the router's subnet, which is located on the base station. The router acts as the central hub where all data passes. The front-end software connects to the back-end through the router.

Linux Computer (Base Station)

At the base station, the laptop is connected to the two analog-to-digital converters, which are connected to the two analog video receivers. The video feed is interpreted by the computer using various open source software that will be discussed in the next section.

Power Converters

The drone is powered by the TBM Ultra Power DJI S1000 21000mAh 6S1P 30C 22.2 V Li-Po Battery Pack. This system uses the battery as the power source, which outputs roughly 22 V depending on the charge of the battery. The Raspberry Pi, GPS, and thermal camera operate at 5 V while the analog transmitters operate at 12 V. Therefore, the battery voltage must be stepped down using two different DC-DC buck converters. In order for the converters to work for the project they must meet the following criteria:

- Wide input voltage range (18 V to 25 V)
- Overcurrent protection
- Stable output
- High efficiency
- Wide operating temperature
- 2 A output for 5 V converter
- 500 mA output for 12 V converter

The converters must tolerate variations in input voltage without affecting the output voltage due to the fact the battery will not output a constant voltage. Overcurrent protection and stable output ensures that the system is not damaged due to a malfunction or short circuit. Efficiency is important because the system is drawing power from the drone battery. The more power the system uses, the less flight time the drone has. It is necessary to note the drone will fly during both the summer and the winter, so the power converters need to be operational in hot and cold temperatures.

Using the above criteria, the Linear Technology LT3697 was chosen as the 22 V to 5V DC-DC buck converter. The LT3697 was designed for USB applications which made it an ideal choice for the project. The main drawback to this converter is the complexity of implementing it. Additional components are needed to complete the converter circuit. The chip comes in a 16-lead MSOP package. This package is small and difficult to solder. From the datasheet found in Appendix III, it can be seen that the package size is 3 mm by 4 mm. The pin size is 0.17 mm wide so it takes a precise hand to solder only one pin and not short pins together. While attempting to solder the chip to the MSOP-16 to DIP-20 SMT adapter board used for prototyping, multiple chips were overheated and damaged. In order to avoid spending more time or money, the team decided to find a new chip that would be easier to implement.

After careful research using the criteria listed above, the Delta S24SE05003 was selected for the 5 V DC-DC converter. This converter is an all-in-one chip, which has six pins. The package allows for quick prototyping and through-hole soldering. It also comes with features such as overcurrent protection, overvoltage protection, over temperature protection, an operating temperature of -40 °C to +85 °C, and a wide input voltage range of 9 V to 35 V. In addition, it has a maximum output of 3 A which meets the design criteria. More information about the chip can be found in Appendix III.

As stated previously, the Monoprice 5.8Ghz Wireless Audio/Video Transmitters operate at 12 V and 250 mA. All of the same specifications are needed for this new 12 V converter. The Delta S24SE12001 was chosen as the buck converter for the transmitters. This converter is in the same package as the Delta S24SE05003 so no additional design work was needed to implement this new converter. The Delta S24SE12001 outputs 12 V and has a maximum output of 1.25 A. All other specifications are the same as the Delta S24SE05003. Figure 3.3 shows the power converter connected to the PCB. More information can be found in Appendix III.

Figure 3.3. Delta S24SE12001

Printed Circuit Boards (PCB) were designed using the CadSoft EAGLE PCB design software. The converter needed to be custom designed as the converter was not in the parts library. A PCB is a more robust and reliable solution as opposed to a breadboard. It also is lighter than a breadboard. The PCB Layout can be seen in Appendix III. The approximate size of the PCB is 2.3 in. by 1.35 in.

In order to connect the system to the drone's battery, an adapter was needed. The drone's battery uses an AS150 and XT150 connecter. A parallel connector was used to connect the system to the battery. This is the best option as it can be easily removed when the system is not needed on the drone.

The USB hub and video transmitters use DC power ports for their power source connection. The USB hub uses a size "H" power port and the video transmitters use a size "M" power port. These power ports were soldered to the output of the power converters.

Hardware Implementation and Photo Documentation

Figure 3.4 gives a block diagram of the wireless communication in the system. As stated before, the GoPro camera, thermal camera, GPS, Raspberry Pi, and analog transmitters are mounted on the drone. The analog receivers, analog to digital converters, and router are located at the base station.

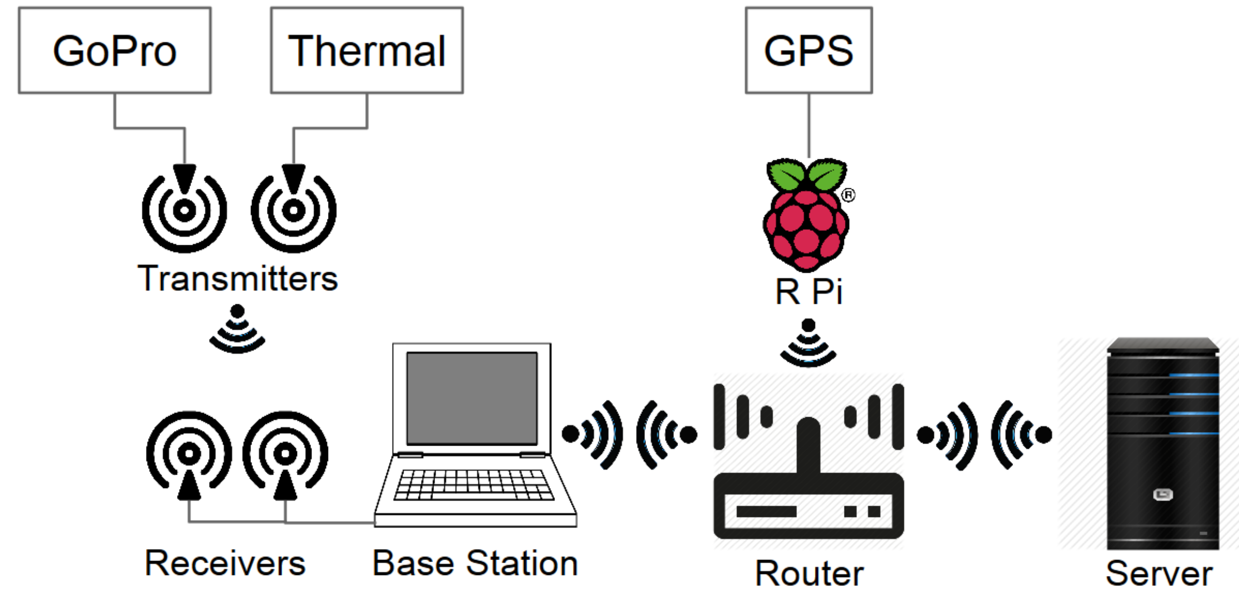


Figure 3.4. Block Diagram of Wireless Connectoins

Figure 3.5 gives a block diagram of how the system is powered. The battery is connected to the power converters through the AS150 and XT150 parallel connector. Both analog transmitters are connected to the 12 V power converter. The USB hub is connected to the 5 V power converter. The thermal camera and Raspberry Pi 2 are powered through the USB hub. The Raspberry Pi 2 powers the GPS.

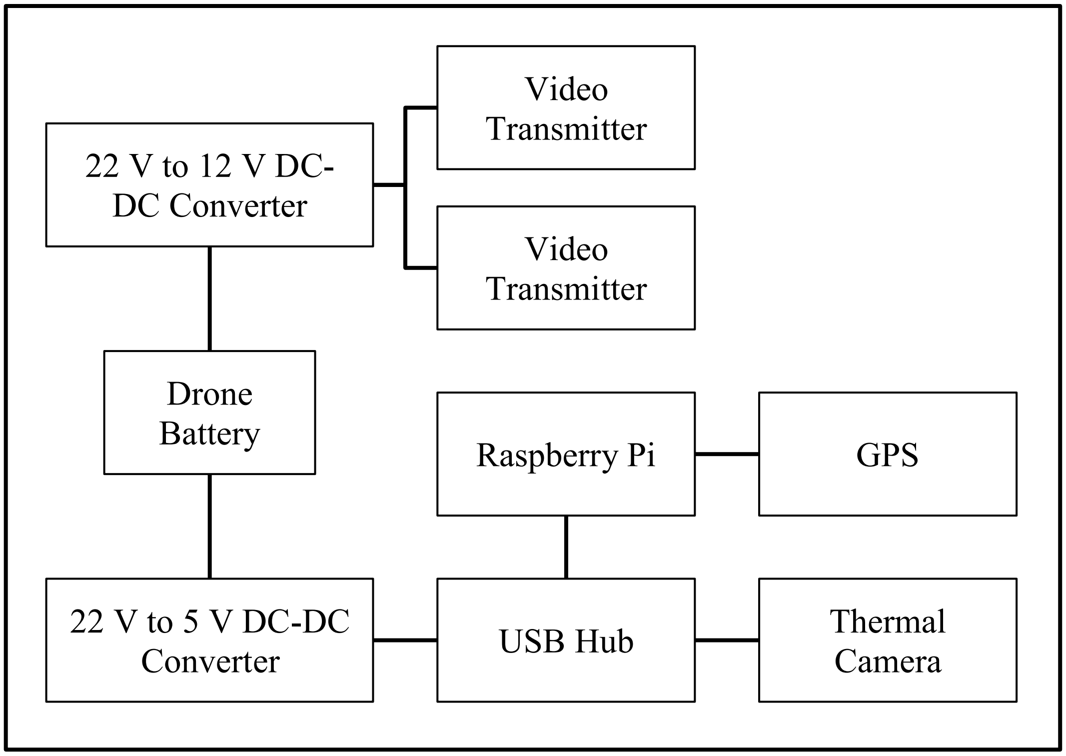


Figure 3.5. Diagram of the Power System

Software and Interface Design Requirements

The sponsor's design requirements originally entailed the capacity to record data on the ground station and also the ability to automatically detect any failures or problem areas on the power line. The team investigated the capacity to automatically detect failures and determined that in order to do so, a pattern recognition algorithm must be implemented. However, in order to train the algorithm, a large training data set must be available. Since the sponsor did not have such a training set, the team had to resort to using the human element to detect failure on the power line.

The investigated pattern recognition algorithm involved using an artificial neural network to analyze the video feed. In order to train the network, the team considered using a heuristic approach to the training methodology. The heuristic solver would require a large training set, which would modify the connections in the artificial neural network to achieve the most efficient solution.

Another approach was to use evolutionary algorithms in order to evolve the artificial neural network. This method is highly experimental and would require immense amounts of computing power. A suggested incubator for the evolutionary system would be

Michigan State University's High Performance Computing Center. The fitness function of such an algorithm would consist of the ratio of accuracy between correct identification and false alarms. Over time, and depending on the fitness function and landscape, the algorithm would improve.

Since both of those methods require a training set, the team proceed to design a system that would allow a human operator to determine whether the given feed was indicative of a problem area. This path of action would still satisfy the sponsor's requirement of identifying problem areas. In fact, since a reliable training set would be difficult to compile, the human's accuracy in determine power line failure excels beyond that of an automatic algorithm that has not been sufficiently trained.

Software Implementation

The aforementioned method of human observation of problem areas requires the development of two systems: a front-end and a back-end. In the scope of this project, the front-end is the software that the operator will run on the Linux computer at the ground station, whereas the back-end is the software that will run on a server located somewhere on the internet, most likely part of the sponsor's domain.

The front-end software consists primarily of a Python program. The choice of using Python as the programming language was due to the fact that the language is highly flexible, and because there are many open-source science-based libraries available to use. Planning for any future pattern recognition capacities, the team determined that the language's flexibility and potential for growth was an objective reason to utilize Python as the programming language. There are some drawbacks to this method, however. Python is a scripting language, which is inherently slower than a compiled language, like C++. On the other hand, programming in C++ requires explicit management of memory, which would detract from the high-level essence of the primary objective of coding a GUI to capture streamed video.

After choosing the language, the team was still in need of a reliable and flexible Graphical User Interface module to use with Python. After significant research, the team concluded that using TKinter was the right tool for the job. The module was highly flexible and allowed for quick changes to the User Interface. Also, the GUI ran in its own root thread, allowing other processing to take place by using other threads.

Since there are multiple videos streams and communication to the Raspberry Pi must take place, the team was in need of a library to handle multi-threading. The module “multiprocessing” was a popular, well-documented option. This module allows for synchronous data sharing between threads by the use of a custom multiprocessing queue and the class, Value.

Afterwards, the team dealt with the video processing issue using the gargantuan OpenCV library, which is a massive library dedicated to video processing. It should be noted that this library is available for both, Python and C++. This library allows the python program to extract frames from the USB ports, which come in from the Analog-to-Digital Converters. The library stores files in .avi format to ensure that no compression takes place that might reduce from the videos’ quality.

Having discussed the front-end, the back-end is just as important in the mapping of the problem areas. The back-end consists of a Ruby on Rails instance running under the sponsor’s domain. Ruby on Rails is a Model-View-Controller supportive framework. In the application, there is a primary model for the recordings, which stands for the database table scheme that the user can communicate with. This allows for the application’s controller to access the database without concern that the model’s specifics might change in the future. Had this type of framework not been used, the language would have to craft custom queries to the database every time the table’s architecture changes.

The database itself is a SQLite database. The team decided to go with this option, because it was the simplest to deploy and manipulate. SQLite does not have a server running to interface with it; the database exists simply as a file on the operating system. Some disadvantages to using SQLite exist. Since the database is simply a file, the operating system will lock the file in the event of writing to the database. This means that only one instance of writing to the database is allowed at a time. For prototyping and small-scale purposes, this is perfect; however, for distributed applications with many operators uploading to the database simultaneously, a database like MySQL or PostgreSQL must be implemented. These databases run as a server and can handle concurrent writing to the database.

There are two primary views in the back-end, the Google Maps page and the page that displays the site recording data for each makers on the Google Maps page. The former of the views is primarily a Javascript endeavor, although it uses embedded Ruby markup to fill in the variables of which coordinates to get from the database. The latter of the pages simply displays the database table and relevant files based on the ID passed into the page via HTML parameters.

Problems

In order to correctly implement the OpenCV module in Python with the two incoming video streams, the team had to manage the limited bandwidth of the laptop's USB hub. This required the Python program to multiplex the hub in order to avoid video lag. Although this method introduced frame drops in the live streams, the recordings of the videos are smooth. The team determined that video lag was worse than frame drops, because the operator requires a live feed of the immediate vicinity under the drone at a given moment in time; a loss of frames is less detrimental to the recordings than a lag in the live video, because the operator controls the starting and stopping of the recording.

Another problem that was encountered earlier in the software design implementation was limited Wi-Fi bandwidth on the USB dongle. The team was originally planning on converting the analog signal to a digital stream on the drone, rather than at the base station. After much experimentation, it was concluded that the Raspberry Pi could not handle streaming via Wi-Fi, as there was not enough bandwidth to smoothly stream two videos of the .avi extension. Furthermore, the team attempted to use a Raspberry Pi 2 rather than the original model to no avail. At that point, the team made a strategic decision, which cut down further experimentation time, to transfer the video feeds in an analog manner rather than digitally, and then multiplex the signal at the base station. This solution worked, although it required the purchase of two sets of analog transmitters and receivers.

IV. Test Data with Proof of Functional Design

The prototype created is completely functional and can be simply mounted onto the drone for use. As previously described, Figure 3.1. showcases the hardware components of the design. This equipment will be mounted onto the drone or placed at the base station. During operation the base station will have an operator that will be viewing the front-end of the system, shown in Figure 4.1. The operator will be presented with the video feeds of both cameras. Each video feed will be a live stream from each camera. This figure shows the thermal camera displaying the thermal signature of a toaster being used. The red circle displayed on the thermal was manually put on for this example to show the hot-spot. The operating system will not contain this. When the operator detects a problem area, he/she will press "START" and the system will begin recording both video feeds. If the operator feels enough video has been recorded then the "STOP" button should be pressed to end the recording. Afterwards, pressing the "UPLOAD" button will upload both video feeds to the back-end with the corresponding GPS coordinates. These coordinates are the exact latitude and longitude coordinates of the GPS device when "UPLOAD" is pressed.

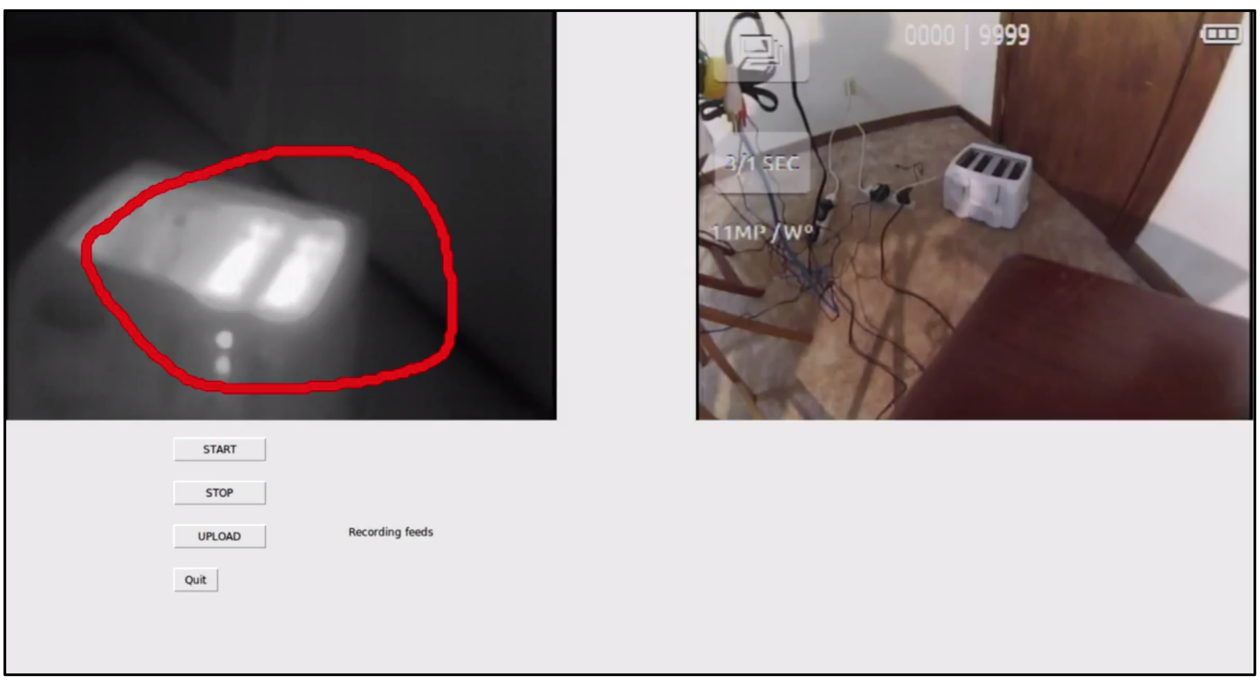


Figure 4.1. Front-End

The back-end can be accessed by any person who has access to the network the back-end is placed on. Accessing the back-end simply presents the Google Maps application with markers of where each recording was executed on the front-end. Shown in Figure 4.2 below is the landing page of the back-end. To view the recording done at this location the operator will click on the marker shown.

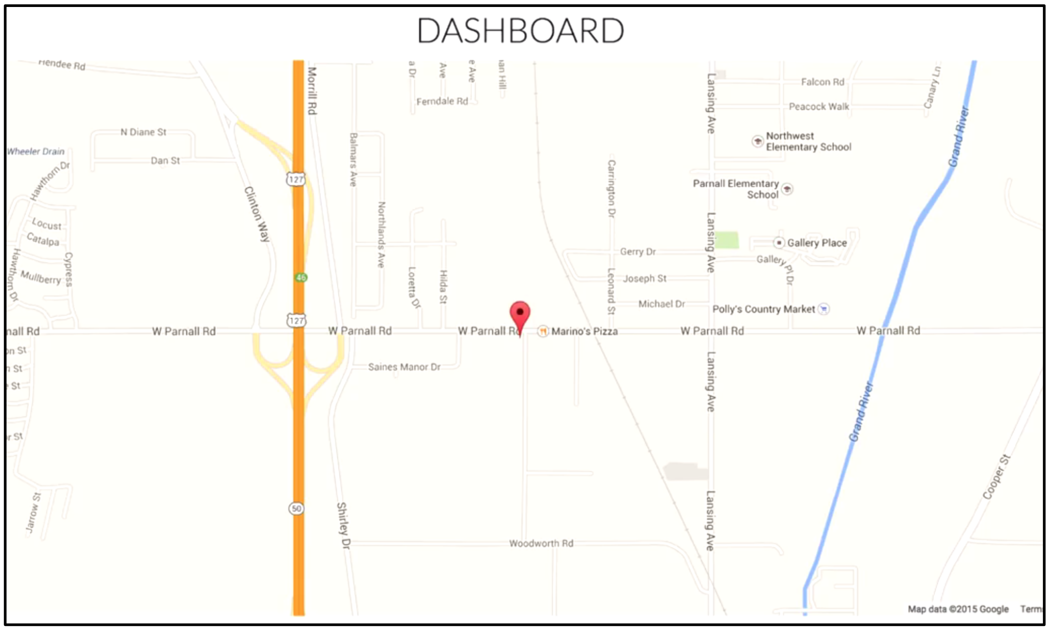


Figure 4.2. Back-End.

Upon clicking the marker on the dashboard, the user will be directed to the site recording data page. Figure 4.3. below contains an example of this page. This page contains the specifics of the problem area recorded. These details include the exact time the problem area was recorded, the GPS coordinates of the problem area, and the two video feeds that were recorded. To view any of the video feeds that were recorded, the user will click on “Thermal” or “Regular” (for the GoPro) to download the specific video feed onto their machine.

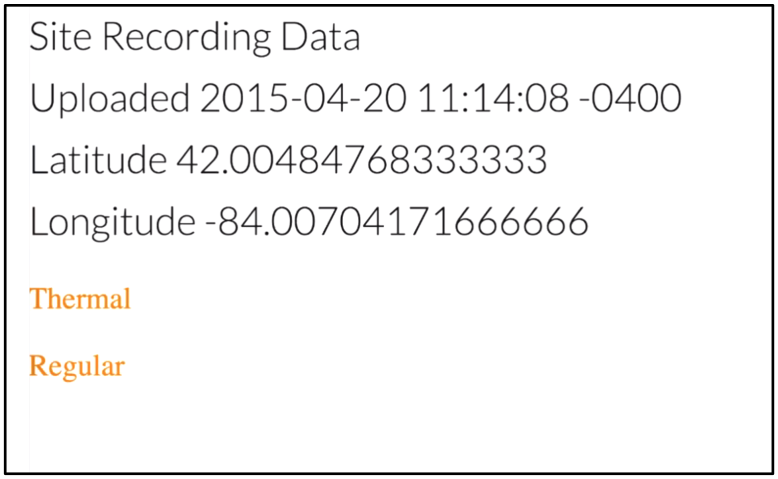


Figure 4.3. Back-End Video/GPS.

Each problem area that is recorded will be placed on the back-end that can be viewed at time while it is stored in the network. This allows for multiple users to view the recordings at any time. In order to have a system that consistently works, much testing was performed on various aspects of the prototype.

The cameras were tested individually before attaching them to the system. The cameras were connected to an Analog to Digital Converter (ADC) connected to a laptop via USB. Testing the GoPro was trivial as it is a normal video feed. To test the thermal camera, the team pointed it at various objects to see if there was a thermal difference. A thermal image of power lines taken with the thermal camera can be seen in Figure 4.4 below.

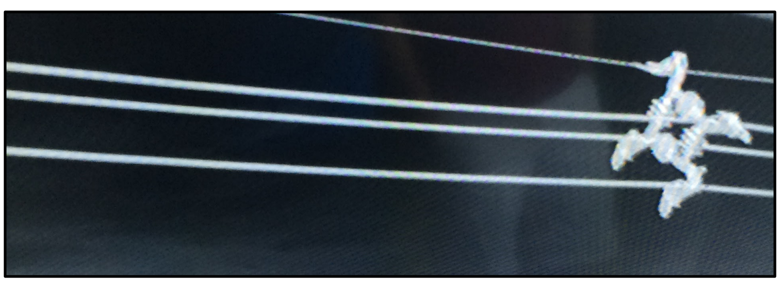


Figure 4.4. Power Line Test.

To test the system without the power converters, the individual components were connected, as they would be on the drone, except the power was supplied via AC adaptors that the various components came with. The Raspberry Pi, base station laptop, and back-end laptop were then configured to the proper IP addresses and ports. The software was then started on the base station laptop, and the back-end software was started on the server laptop. The software and hardware functioned as expected on the first test.

While the system was able to display video from both cameras, record and send that video to the back-end, the back-end application, as seen in Figure 4.2 was placing the Google Maps pin in the wrong location. The pin was being placed miles away from where it should have been. It turned out to be a string parsing error in the program that runs on the Raspberry Pi to get the GPS coordinates. The program gets the coordinates as a string of characters and this string is parsed and manipulated to get the coordinates in a format that can be used with the Google Maps API. The problem was corrected and the system began to function properly.

Another issue was discovered involving the GPS. When testing the design in the Engineering Building at MSU, the system was not uploading the videos. It was discovered that inside a large building, such as the Engineering Building, the GPS could not receive information from satellites and therefore could not output location coordinates. The front-end requests GPS coordinates from the Raspberry Pi when the operator clicks the upload button. When the GPS is not receiving a satellite signal, the Raspberry Pi cannot get the coordinates, so it does not respond to the front-end request. Thus, the front-end program will perpetually wait for the Raspberry Pi's response. This problem was overcome by manually sending GPS coordinates to the back-end. This GPS reception issue is only problematic for testing; it would not exist in the outdoors environment for which the system is designed.

With those issues resolved the system worked well. The system was tested by setting up objects with varying temperatures, such as a toaster as can be seen in Figure 4.1 above, recording the videos, and uploading them to the back-end. From there, the team was able to see the pins on Google Maps and by clicking the pins were able to see the GPS information as well as links to download the videos as shown in Figure 4.3 above.

Testing the Power converters

The Delta S24SE05003 and Delta S24SE12001 were tested using a similar procedure. The first step was to build the circuit using a breadboard. This can be seen in Figure 4.5. Using the Agilent E3611A power supply in the lab, the circuit was energized with no load connected. The output and input voltages were measured using the Agilent Infiniium DSO-9064A Digital Storage Oscilloscope. A resistor bank, which is equivalent to 10 Ω, was then connected to the 5 V power converter. The output voltage can be seen in Figure

4.6. A resistor bank, which is equivalent to 24 Ω, was then used as a load for the 12 V power converter. The result can be seen in Figure 4.7. These same steps are repeated while the power converters soldered to the PCBs to ensure the operation is the same.

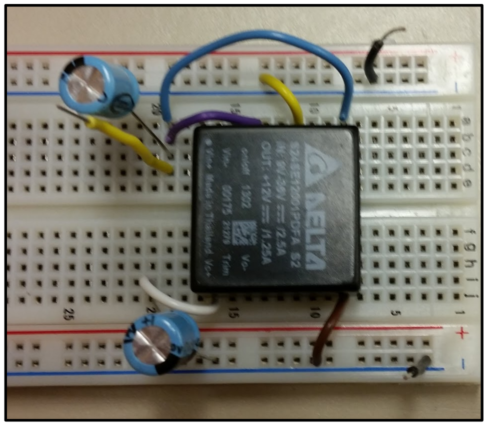


Figure 4.5. Breadboard Prototype.

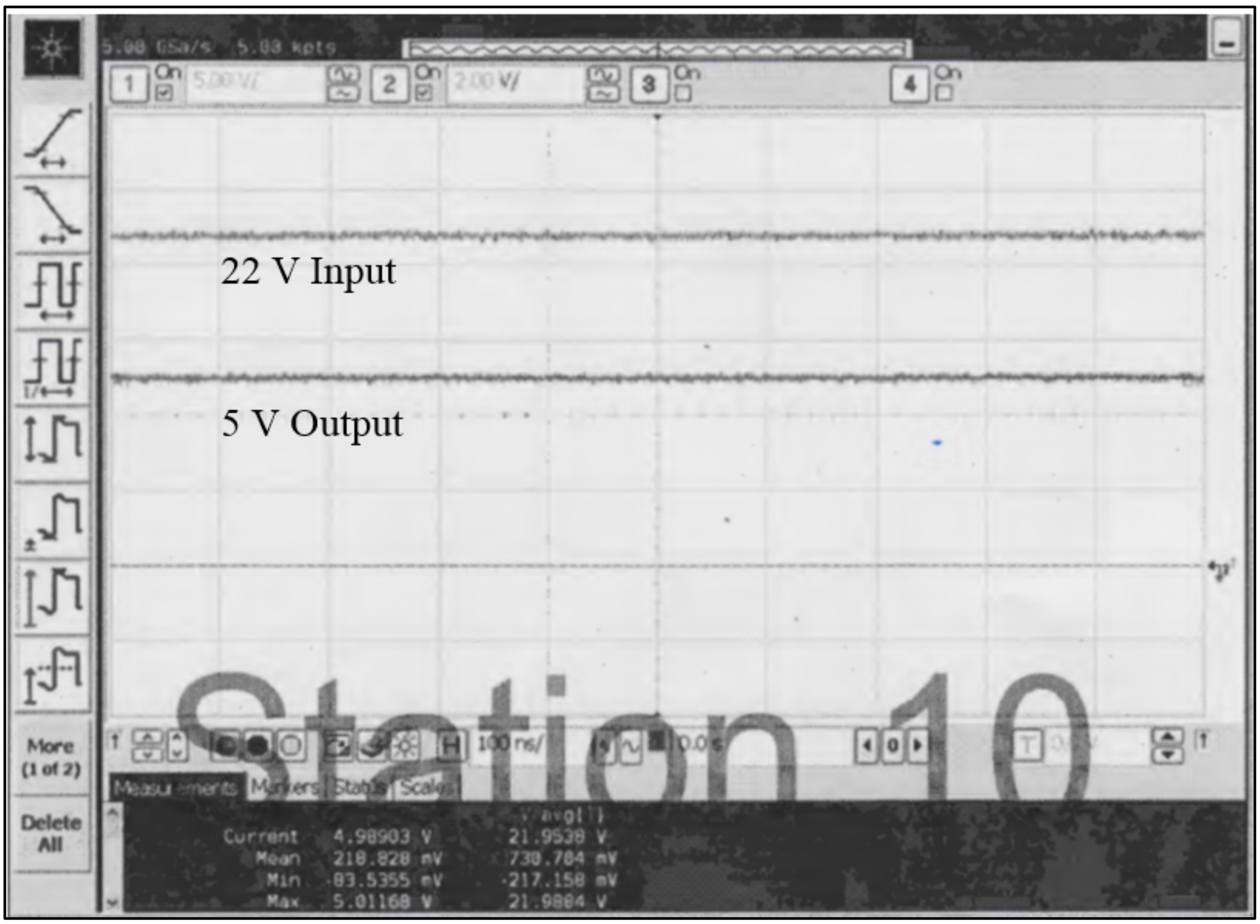


Figure 4.6. 5 V Converter Output.

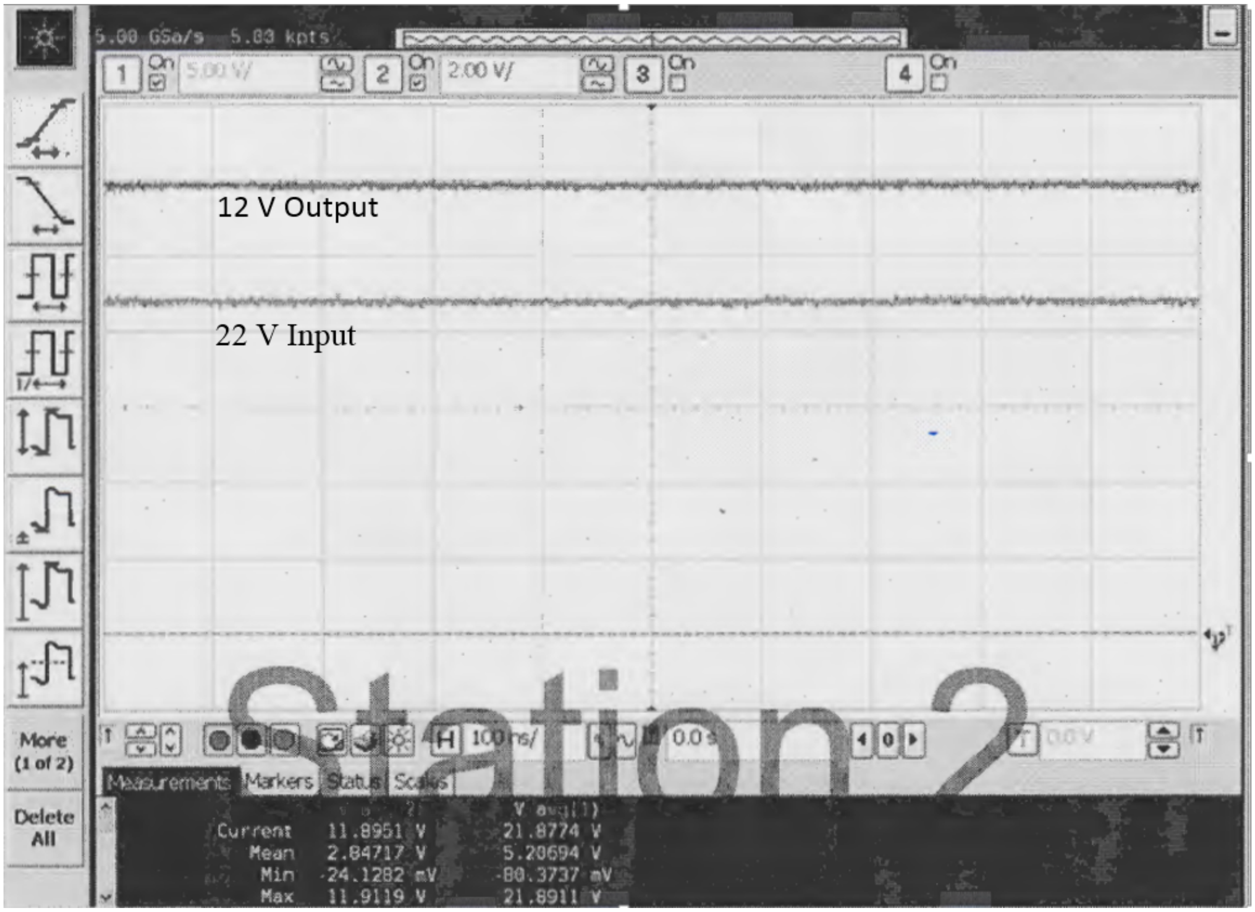


Figure 4.7. 12 V Converter Output.

V. Final Cost, Schedule, Summary and Conclusion

Conclusion

In conclusion, Team 14 completed the design specifications. However, this was not without its struggles. One failure of the project came from overheating and damaging the LT3697. This did not affect progress due to early failure in the timeline of the project. Consequently, a better option was found for a power converter and only a small amount of money was the tradeoff. The team originally planned to use the Raspberry Pi to transmit all of the video provided by the GoPro and the thermal camera. Another setback was encountered due to the lack of processing power on the Raspberry Pi. To resolve the situation, the team tried to upgrade to the Raspberry Pi 2; unfortunately, there was a lack of enough processing power with this device as well. The team made a design choice of keeping the video files completely analog and used transmitters for a faster more efficient live stream. Another deemed failure of the project was that the sponsor initially required the team to create automatic hot-spot recognition software. After some consideration, it became clear that this method was unrealizable, because the team did not have access to a database of hot-spot images or videos.

Ultimately, the design project was a huge success. The design and time requirements were met. The primary success of the project is a manifestation of software's solid architecture. The completion of this project demonstrated the team's ability to stream live videos from two cameras and be able to pinpoint the problem areas using GPS. The team was also able to keep the weight under 22 lbs. and be able to power the entire system from the drone's power supply. The sponsor was highly pleased with the team's performance.

Future Improvements

Based on the success of this prototype, there are a few features that can be implemented to make it more applicable for the sponsor. A key feature of the thermal camera is the ability to calibrate the detectable temperature range. Compensating for ambient temperatures or even scenarios such as clouds casting a shadow on an object is key for accurate readings. Currently, the camera may be calibrated by connecting it to a computer running on a Windows operating system. As the base station will operate under a Linux-based machine, the camera cannot currently be calibrated while the drone is in the air. To address this issue, a software development kit (SDK) may be purchased. The SDK is rather expensive, but the main reason this feature was not implemented in the prototype is due to the risk of damaging the camera. Sending the wrong set of instructions to the camera can permanently destroy the hardware, as it has no user-safe mode. With careful implementation of the SDK, the sponsor will be able to calibrate the camera wirelessly.

As previously mentioned, automatic detection of a hot-spot was not feasible for this project. In order to implement automatic detection, the software analyzing the video feed must be presented with a data set. This data set will contain images of power lines and frequently sampled areas being inspected. Since the software currently does not have enough data to create said data set, automatic detection cannot currently be implemented. With enough use of this prototype, the software will eventually gain this data set as the database grows. Afterwards, when the program receives images, it can compare the captured image with images from the data set. If the program is used to detect enough problem areas, the system can then automatically signal a hot-spot.

These two improvements will be the main factors to create a robust system that will greatly benefit the sponsor. This prototype was designed with a low prototyping cost in mind. Creating a prototype with a low-cost solution ensures that the architecture is operational. Higher quality transmitters and wireless devices can easily replace the components to create a more robust system that can operate at greater distances.

With the video feeds being displayed in "avi" format they are presented as large data files. This prototype can be improved by implementing a way to compress these files. Compressing the video data will create less frame drops on the front-end while also saving money on preserving space on the client's servers.

Further, the user interface on the front-end may be upgraded. With enough use the drone operators may feel that more options should be presented upon hot-spot detection. The options may be comment fields or drop-down menus to assign a priority field to a hot-spot. Much of these improvements will depend on what the sponsor feels necessary after enough use of the prototype.

Project Execution

Week	Activity
Jan 26 - Feb 1	Brainstormed prototype ideas
Feb 2 - Feb 8	Met with sponsor
Feb 9 - Feb 14	Specified project objectives Delivered proposal to sponsor
Feb 15 - Feb 21	Finalized project research
Feb 22 - Feb 28	Met with hardware compatibility setbacks Discovered automatic detection is unfeasible
March 1 - March 7	Discussed alternative methods with sponsor
March 8 - March 14	Obtained live stream from cameras, separately Selected new 5 V power converter
March 15 - March 21	Incorporated GPS functionality Began back-end programming Began front-end programming
March 22 - March 28	Discovered digital transmission was impossible Explored alternative solutions
March 29 - April 4	Ordered analog video transmitters receivers Ordered analog-to-digital converters Ordered new 12 V power converters
April 5 - April 11	Finished front-end
April 12 - April 18	Finished back-end Finished project
April 19 - April 25	Presented final product to Consumers Energy
April 26 - May 1	Design Day

Table 5.1. Project Execution.

Budget

The project cost is based on the final design components, and not the total cost of prototyping. The total comes out to be $423, with the most expensive parts being the transmitters. The manufactured cost has a few slight differences than that of the team's project comes out to a cost of $467. Most utility companies' maintenance vehicles already have internet and routers on the vehicles; therefore, the sponsor would not need to purchase a router. Also, for manufacturing purposes of the product, it would be necessary to purchase better transmitters for the design. Although the chosen transmitters work well with the prototype, they are directional transmitters and may cause connectivity issues when mounted on a drone. It should also be noted that the cost of the design does not include the price of the two cameras that were provided, at an equivalent cost of $1000. These charts also do not include the price of the drone or mounting unit that have an equivalent cost of $5,000. That would make the total cost of implementing this design $6,500. Although the design may appear to be expensive, it is much cheaper than the alternative of flying a helicopter; the drone approach is significantly more cost-effective.

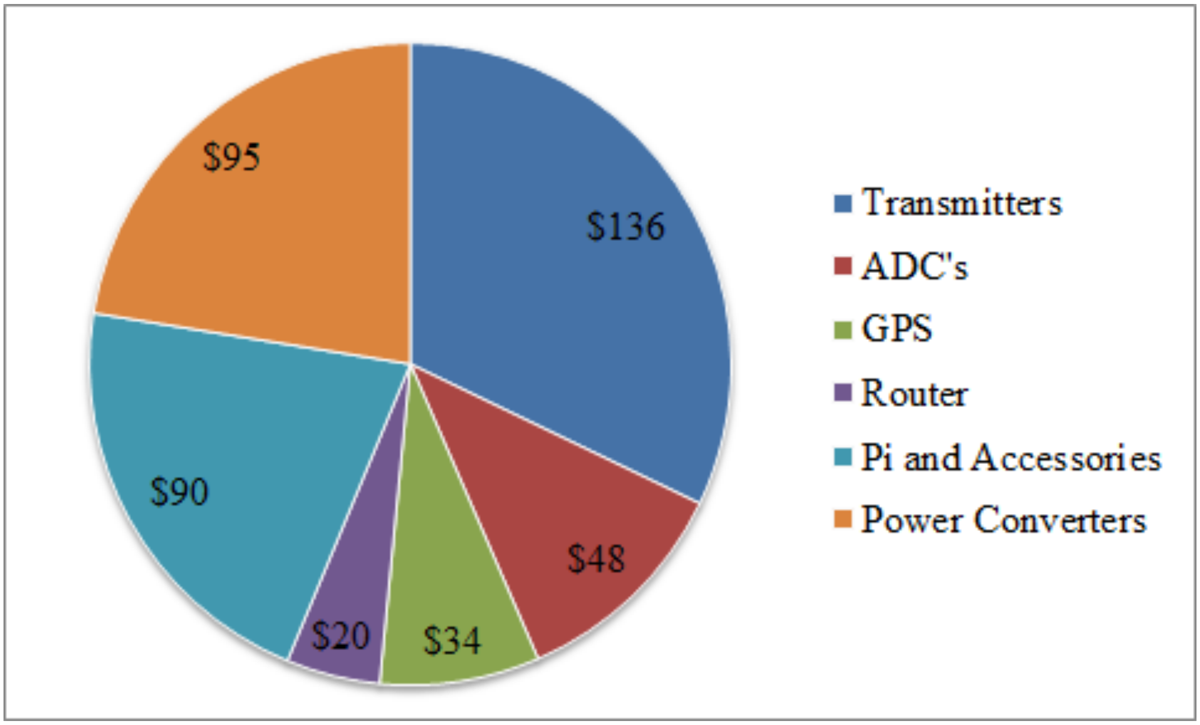


Figure 5.1. Project Cost.

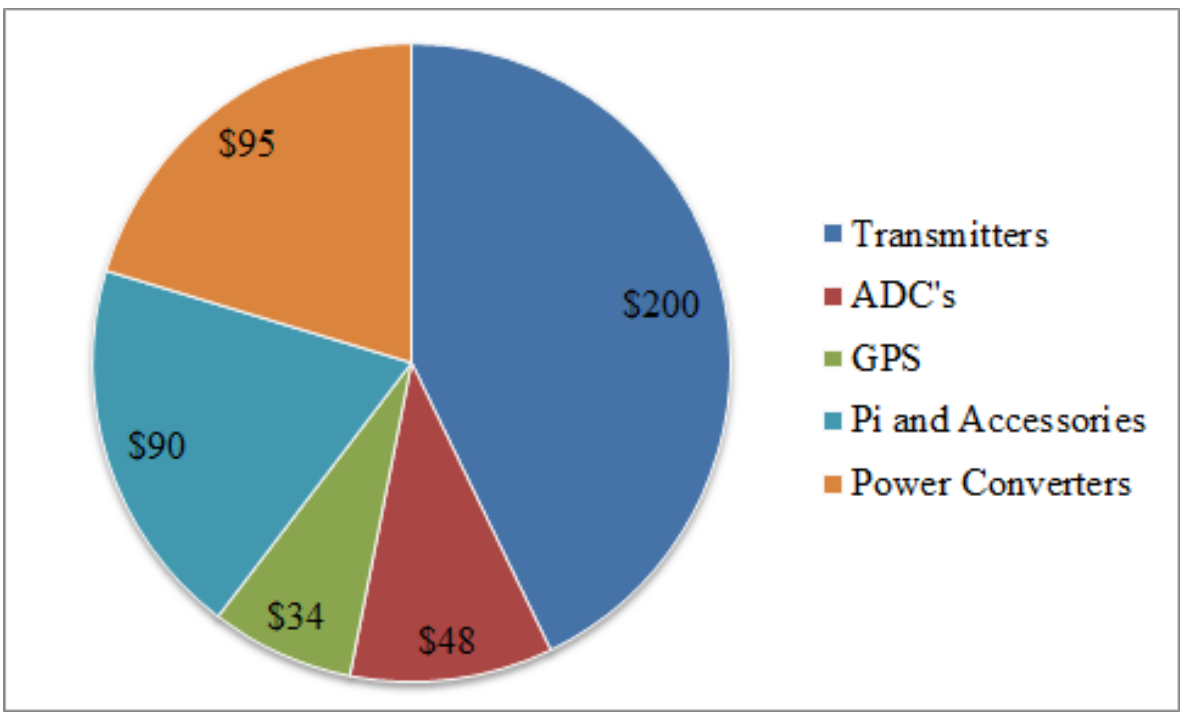
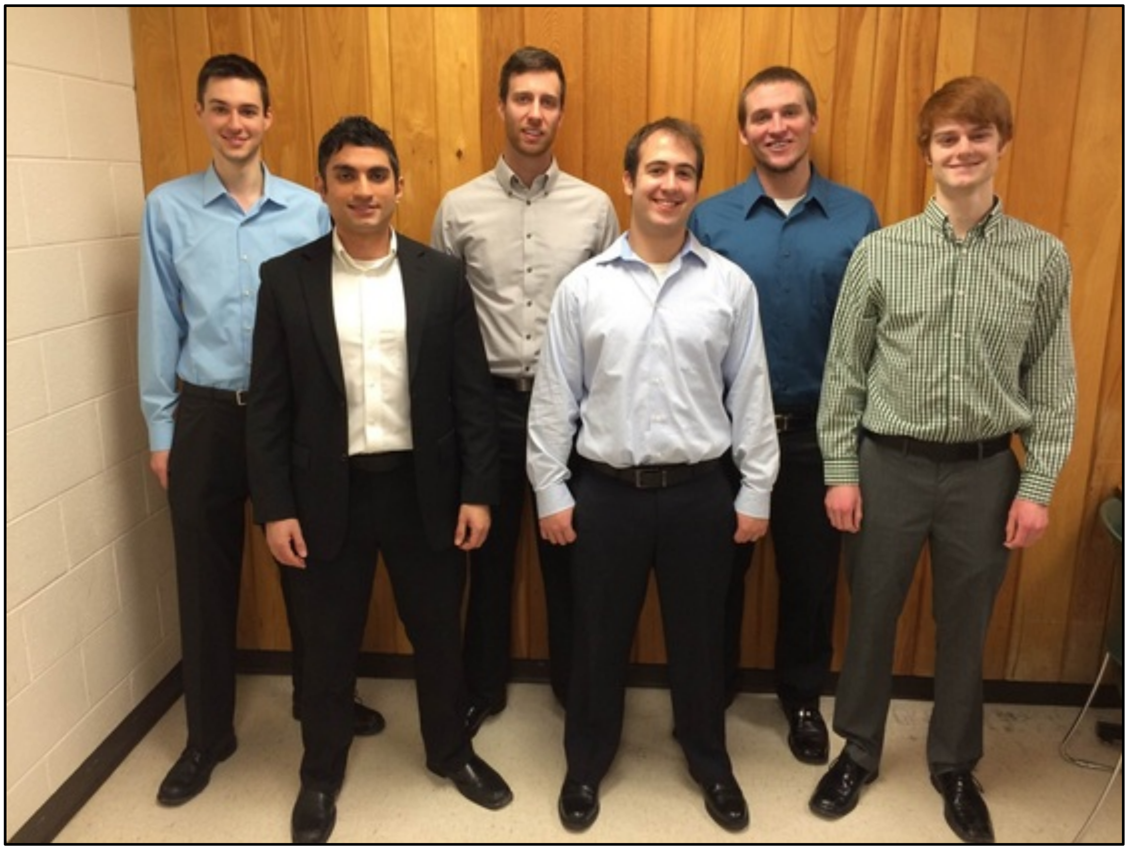


Figure 5.2. Manufactured Cost.

Appendix I - Technical Roles, Responsibilities, and Work Accomplished



Pictured (left to right): Cody Wilson, Faisal Tameesh, Dan Pittsley, Mitch Johnson, Jake Hersha, Ian Meredith

Cody Wilson - Lab Coordinator

The primary work for Cody was writing the software to retrieve the coordinates from the GPS. Cody had familiarity with Python and the Linux operating system. He was able to code parts of the project from the early stages of the project.

Writing the software for the GPS meant Cody needed to know a great deal about how the GPS worked, such as the what type of device it is, serial or block, the output rate, format, etc. This meant Cody initially had to spend the bulk of his time researching how the GPS worked and properly interpreting the data that he did find.

Once Cody understood the GPS, he began by running tests to connect the GPS to a linux system. He then needed to read data from it. Cody had previous experience with Linux. He was able to research the proper terminal commands to set a USB port speed and read from a serial device.

At this point Cody understood what the program needed to do and began to write the Python code. The first program he wrote used the serial library in Python to read the input from the GPS just like the Linux commands, so that he knew his method worked.

From there Cody began to write the code for the program that would read from the GPS and parse its output for the two coordinates that were needed. As Cody finished his code, he found a Python library, pynmea. This library allowed Cody to simplify his code as it was built to handle the output from GPS devices that use the NMEA format, like the one used in the project.

When done revising his code, Cody worked with Faisal on getting the Raspberry Pi to run the GPS code. This was done in order to communicate with the front-end software that would be requesting the GPS coordinates.

Faisal Tameesh - Presentation Preparer

Faisal was the architect behind how the software systems operated and communicated with one another. After studying the requirements with the team, Faisal was responsible for delegating various parts of the software to the team members. Faisal stressed the usage of version control and tests, and also reviewed merged code to ensure that the system was operational from a big-picture perspective. Faisal's pragmatism was central to the project's success.

The primary parts of the software that Faisal worked on were the front-end and the low-level details of the back-end. The front-end consisted of utilizing Python to code the Graphical User Interface, and the back-end work consisted of coming up with a database scheme to store the relevant information for the project, while allowing the database to operate quickly. An example of how this was done was to use file paths to store in the database instead of the video files themselves. This allowed for much faster access to the database, as it was significantly less cluttered.

Thanks to Faisal's prior knowledge of Python, Javascript, and Ruby on Rails, the system was fairly divided into the various components amongst the team. Investigation of which modules or gems to use with the frameworks was also conducted by Faisal, as his previous experience of what to use was tremendously useful to ensure that project was on track, and that no part of the software was lagging behind.

Dan Pittsley - Team Manager

The final system designed by Team 14 used many different programming languages such as Ruby, Python, and JavaScript. Dan's fundamental responsibility was to research, design, and implement the Google Maps JavaScript API used for the back-end. Dan completed a lot of preliminary research and practice in order to become familiar with the technology. Fortunately, Dan had previous experience with object-oriented programming languages and this step progressed smoothly. An important consideration with regard to this portion of the project was the implementation of model architecture as part of the Rails project; Dan worked to ensure that this was accomplished. To ensure correct operation, it was necessary to become familiar with the chosen database - SQLite. This required more independent research and collaboration with Faisal, who was developing the database scheme. Dan also took great care to follow relevant standards during development, which was important in order to ensure an effective product, and a product that can be easily used and understood by other engineers in the future.

For all components of the project, it was required to match the customer's specifications, but it was particularly so for the Google Maps view, as it may be used by non-technical people. Dan worked to ensure that this portion of the project matched the sponsor's descriptions. It was also necessary to fully understand the technology in order to communicate with the sponsor about possibilities and limitations of the final design. For example, the original sponsor specifications called for all data to be accessible in a single, monolithic Google Maps view. However, after attaining a sufficient understanding of the Google Maps API, it was communicated to the customer that this was a feature that could easily be added in the future. But, having a good knowledge of the Google Maps API, Dan was able to provide several alternative approaches - such as linking a plot-point to a new page - and implement the sponsor's chosen design.

Mitch Johnson - Assistant

Mitch was primarily involved in hardware development in the design process. Specifically he helped with creating the power converters with Ian and Jake. All of the power had to come from the drone therefore the only source that was available for the system was the drone battery. It was important for the created power converters to be efficient for the system so that the drone could have more power to work with. Mitch also contributed to testing of the system.

Ian did most of the design on the power converters and Mitch laid out the converters on Eagle. Eagle is a software that is used to layout a circuit design onto custom designed PCB. Since Mitch had little prior knowledge of the software he had to teach himself the software language in order to create a useable PCB board. He taught himself how to create unique components on the software because the converter needed to be a custom design. With the help of the MSU ECE shop Mitch was able to print out the designed PCB that was created on Eagle. Once the PCB was created, Mitch assisted Jake and Ian with soldering the components onto the created board.

Mitch worked with Jake and Ian to pick components that would work best for Cody, Faisal and Dan who were programming the bulk of the software. This includes the selection of the transmitters and also the Raspberry Pi, which initially was going to do most of the processing.

Once the project was completed Mitch helped Jake and Dan to create the display the project on. This includes selecting cost effective and durable materials that have the ability to be visually appealing. Overall Mitch learned a lot about drones and their hardware specifications over the course of this project.

Jake Hersha - Webmaster

Jake had the responsibility of working with the hardware for the system. More specifically, he aided in powering the system and determining which hardware components would fulfill the project's requirements. Since the system will be mounted onto the drone, the incorporated components must be supplied power other than a power outlet. In order to do this, all components must be provided power from the drone's battery. Jake initially began the project by determining specifications of all the components used for the system. The components used for the system all operate at either 5 V or 12 V so the 22 V provided from the drone would have to be stepped down.

After inspection of the power specifications, Jake assisted Ian in providing initial designs of power converters for the system and researching current converters to purchase. Once the power converters were successfully designed and tested, it was necessary to understand how to physically connect the drone battery, power converters, and components all together. A simple adapter was found to connect the battery and the power converter. Soldering the wire from the adapter to the converters had to be done once purchased and then could be connected to the necessary DC power ports such as the USB hub and transmitters.

Jake also did intensive research into selecting the hardware components to use for the whole system. In order to get real time GPS coordinates he found an easy to use GPS device. There are many GPS devices available on the market so Jake compared multiple devices and found one to suit the project that was sturdy and compatible with the Linux-based machine.

Originally, it was proposed to have the Pi perform video analysis directly, but this was found to be problematic as mentioned earlier in the report. Jake aided in determining that transmitting these signals to a machine is actually much more efficient. Analog systems are much easier to transfer data with, as they are less tolerant to noise and make good use of bandwidth. Therefore, low cost analog transmitters and receivers were selected to send the video data to the base station machine. Since this analog data must be interpreted by software created by the team, Jake found simple devices to convert the analog data into digital streams with the analog to digital converters.

After completion of the prototype, Jake worked with Dan and Mitch on packaging the prototype to make it portable and ready for presentation. This entailed cutting wood and putting together a nice looking base for the equipment to sit on. The base station box has a compartment to hide some of the wires of the system so it can show case the actual components being used while hiding the clutter of all the wires.

Ian Meredith - Document Preparer

The primary responsibility for Ian was powering the system. This includes selecting components, prototyping, and testing. Since power was being drawn from the drone battery, this included selecting DC-DC buck converters and the components needed to implement the converters. Selecting the converters meant working with Jake to get specifications on all of the components selected for the system. Initially, the components used all operated at 5 V so only one converter was necessary. As the semester progressed, new components were used which operated at 12 V so Ian selected a second converter.

Ian worked to prototype and test both converters first using a breadboard. All of the wiring was carefully checked to ensure accuracy. He tested the converters at multiple input voltages and loads to ensure the converters could power the system in all situations.

He then worked with Mitch to create the printed circuit boards using the Eagle software. The converters were then soldered to the printed circuit boards and tested again to ensure the converters operated as designed. Heat Sinks were added by Ian to ensure the converters would not overheat while in operation. This included selecting the correct thermal adhesive to maximize thermal contact.

Ian next worked with Jake to select the components, which would be used to connect the system to the battery through the power converter. This involved selecting an adapter, which would connect the drone's battery to the power converters. Careful research was used to find a reliable connector that will not affect the performance of the drone. The transmitters and USB hub are powered through DC power ports. The ports needed to be sized correctly, which involved taking measurements. The full system was then connected through these components and tested to the fullest extent possible as the team was not able to access the drone's battery.

Appendix II - Literature and Website References

"Power Line Infrared Thermography Inspection." Martechnical. Martechnical, n.d. Web. 05 Feb. 2015.

"Raspberry Pi" Raspberry Pi Foundation Web. 20 Feb. 2015 <http://www.raspberrypi.org>

GPS Receiver. GlobalSat: Worldcom Group, n.d. Web. Feb. 2015. <Http://www.globalsat.com.tw/products-page_new.php?menu=2&gs_en_product_id=2&gs_en_product_cnt_id=76&img_id=563&product_cnt_folder=3>.

"HERO3 Black." *GoPro*. N.p., n.d. Web. 20 Feb. 2015. <http://shop.gopro.com/cameras/hero3plus-black/CHDHX-302-master.html>.

"Hauppauge 610 USB-Live 2 Analog Video Digitizer and Video Capture Device" <http://www.amazon.com/gp/product/B0036VO2BI/ref=od_aui_detailpages00?ie=UTF8&psc=1>

Appendix III - Detailed Technical Attachments

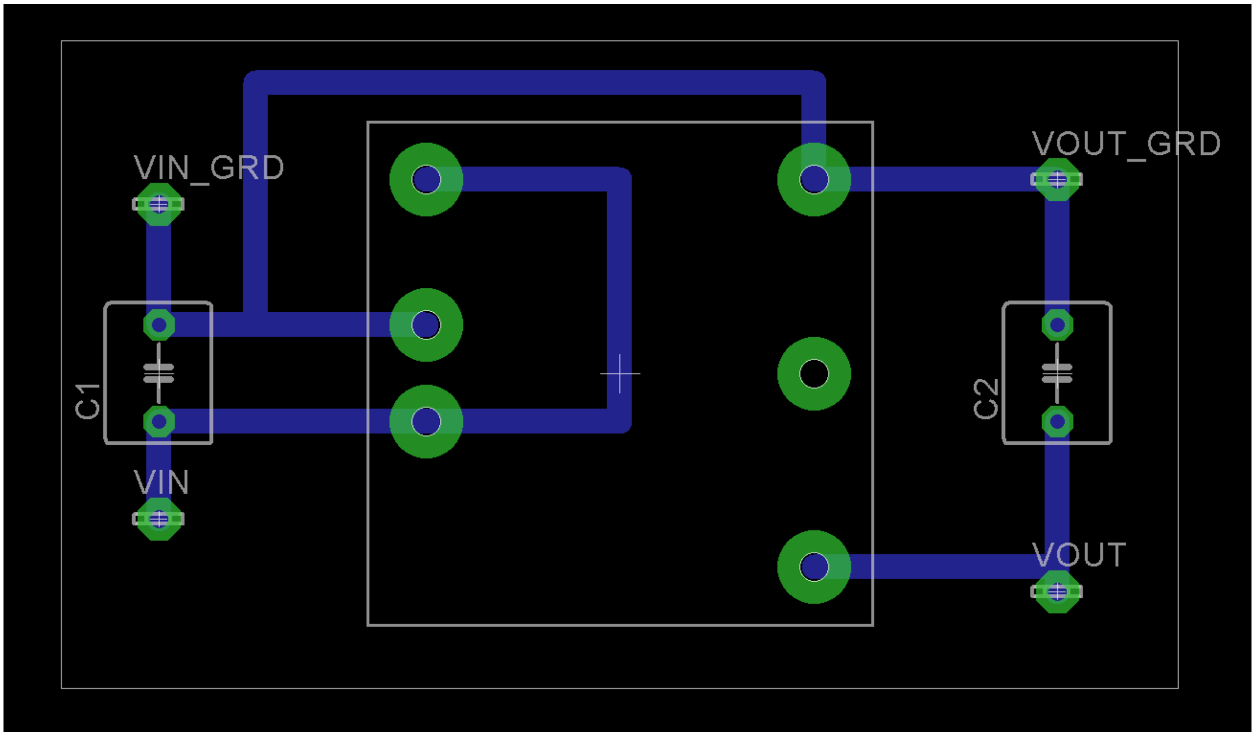


Figure A3.1. PCB Layout for Delta S24SE Series.

S24SE/S24DE series

15W Single/Dual Output DC/DC Converter

FEATURES

- Efficiency up to 89%
- Wide input range, 9V-36V
- Package with Industry Standard Pinout
- Package Dimension:
  25.4 x25.4 x10.2mm (1.0" x1.0" x0.40")(No HSK)
  31.0 x25.4 x17.5mm (1.22" x1.00" x0.69")(HSK)
- Over voltage protection, hiccup mode
- Over current protection, hiccup mode
- Positive or Negative Remote ON/OFF
- Without tantalum capacitor inside module
- Operating Temperature range - 40°C to +85°C
- Input to Output Isolation: 1600VDC
- RoHs Compliant
- 3 Years Product Warranty
- Heat-sink is option
- UL/cUL 60950, with Amendment 1 :2009 (US & Canada) recognized

The S24SE/S24DE series is miniature, isolated 15W DC/DC converters with 1600VDC isolation. The S24SE/S24DE family comes with a host of industry-standard features, such as over current protection, over voltage protection, over temperature protection and remote on/off. An optional heatsink is available for more extreme thermal requirements . All models have an untra-wide 4:1 input voltage range (9V to 36V). With operating temperature of -40°C to +85°C, it is suitable for customers' critical applications, such as process control and automation, transportation, data communication and telecom equipment, test equipment, medical device and everywhere where space on the PCB is critical.

Model List

Model Number	Input Voltage (Range)	Output Voltage	Output Current		Input Current (typ input voltage)		Load Regulation	Maxcapacitive Load	Efficiency (typ.)
			Max.	Min.	@Max. Load	@No Load			@Max. Load
	VDC	VDC	mA	mA	mA(typ.)	mA(typ.)	mV	uF	%
S24SE3R305	24 (9 ~ 36)	3.3V	4500	0	720	35	±10	10000	87%
S24SE05003		5.0V	3000	0	710	35	±10	10000	89%
S24SE12001		12V	1250	0	720	30	±12	470	87.5%
S24SE150R9		15V	1000	0	720	25	±15	470	87.5%
S24DE120R6		±12V	625	0	720	20	±120	±470	87.5%
S24DE150R5		±15V	500	0	720	30	±150	±470	87.5%

Input Characteristics

Item	Model	Min.	Typ.	Max.	Unit
Input Surge Voltage (100 msec)	All Models			50	VDC
Input Turn-On Voltage Threshold	All Models	8	8.5	9	VDC
Input Turn-Off Voltage Threshold	All Models	7	7.5	8	VDC
Input Under-Voltage Lockout Hysteresis	All Models	0.4	1	1.7	VDC
Off-Converter Input Current	All Models		6		mA
Input reflected ripple current	All Models,with 12uH, 20MHz		5	20	mA
Reverse Polarity Input Current	All Models	---	---	0.3	A
ON/OFF Control, Logic High	All Models	2.4		10	VDC
ON/OFF Control, Logic Low	All Models	-0.7		0.8	VDC
Input Filter	All Models	Internal PI Filter			

DS_S24SE/S24DE 15W _05232013

E-mail:DCDC@delta.com.tw
http://www.deltaww.com/dcdc

Figure A3.2. Delta S24SE Datasheet

S24SE/S24DE series

15W Single/Dual Output DC/DC Converter

Output Characteristics

Item	Conditions	Min.	Typ.	Max.	Unit
Output Voltage Accuracy		---	±1.0	±2.0	%Vo
Output Voltage Balance	Dual Output, Balanced Loads	---	±1.0	±2.0	%Vo
Line Regulation	Single output		±0.1	±0.2	%Vo
	Dual output		±0.1	±0.5	%Vo
Cross Regulation	Dual output,Asymmetrical Load 25%-100% Full Load		±2	±3	%Vo
Total Outupt Voltage Range	Over Load, Line and Temperature	---	---	±3	%Vo
Ripple & Noise	12V, 15V, ±12V, ±15V	---	50	---	mV p-p
Ripple & Noise	3.3V, 5.0V		50	---	mV p-p
Dynamic load response	50%-75% full load, 0.1A/uS		3		%Vo
Output Over Current Protection	Output Voltage 10% Low, Hiccup	110		160	%Iomax
Short Output Protection	Long Term, Auto-recovery				
Output Over-Voltage Protection	Hiccup, Auto-recovery	115		150	%Vo
Output Trim Range	Single Output	-10		+10	%Vo

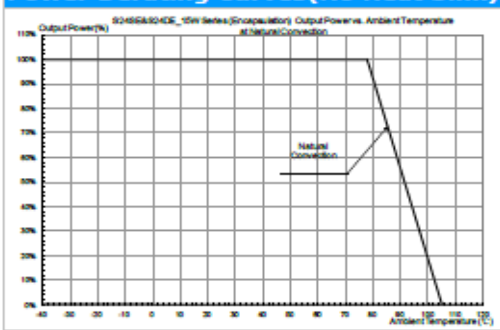
General Characteristics

Item	Conditions	Min.	Typ.	Max.	Unit
I/O Isolation Voltage (rated)		---	---	1600	VDC
I/O Isolation Resistance		10	---	---	MΩ
I/O Isolation Capacitance			1100		pF
Switching Frequency			450		KHz

Environmental Specifications

Parameter	Conditions	Min.	Max.	Unit
Operating Temperature Range (with Derating)	Ambient	-40	+85	℃
Case Temperature		---	+105	℃
Storage Temperature Range		-50	+125	℃
Humidity (non condensing)		---	95	% rel. H
Cooling	Free-Air convection			

Power Derating Curves(No Heat Sink)



Notes

1 Specifications typical at Ta=+25℃, resistive load, nominal input voltage and rated output current unless otherwise noted.
2 Ripple & Noise measurement bandwidth is 0-20MHz, with 10μF, tantalum capacitor and 1μF ceramic capacitor.
3 All DC/DC converters should be externally fused at the front end for protection.
4 Specifications are subject to change without notice.

E-mail:DCDC@delta.com.tw
http://www.deltaww.com/dcdc

Figure A3.2. Continued.

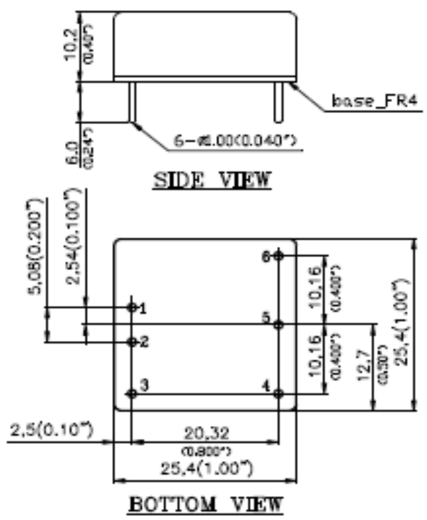
DELTA

S24SE/S24DE series

15W Single/Dual Output DC/DC Converter

Mechanical Drawing

Mechanical Dimensions



Pin Connections

Pin	Single Output Function	Dual Output Function
1	Vin+	Vin+
2	Vin-	Vin-
3	On/off	On/off
4	Vout-	Vout-
5	Trim	Common
6	Vout+	Vout+

Physical outline

Case Size: 25.4*25.4*9.5(1.0"*1.0"*0.38")

Case material: Al alloy, anodize black

Baseplate material: Non-coductive FR-4

Pin material: Brass; finish: Matte Tin plating and Nickel under plating

Pin length: refer part numbering system

Weight: 17.5grams

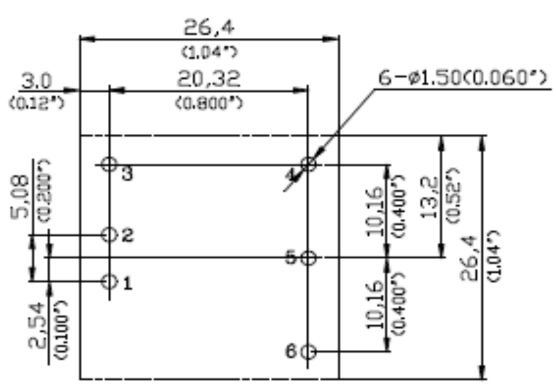
- All dimensions in mm (inches)
- Tolerance: X.X±0.5 (X.XX±0.02)
  X.XX±0.25 (X.XXX±0.010)
- Pins Diameter : ±0.10(±0.004)

Application notice:

For modules with through-hole pins, they are intended for wave soldering assembly onto system boards; please do not subject such modules through reflow temperature profile.

Recommened layout refer below

RECOMMENDED LAYOUT



PIN	Single	Dual
1	Vin(+)	Vin(+)
2	Vin(−)	Vin(−)
3	ON/OFF	ON/OFF
4	Vout(−)	Vout(−)
5	Trim	Comm
6	Vout(+)	Vout(+)

E-mail:DCDC@delta.com.tw
http://www.deltaww.com/dcdc

Figure A3.2. Continued.

LT3697

USB 5V, 2.5A Output, 35V Input Buck with Cable Drop Compensation

FEATURES

- **Accurate 5V Output**
- **Programmable Cable Drop Compensation**
- **Programmable Output Current Limit**
- **Adjustable Output from 5.0V to 6.1V**
- **Dual Input Feedback Permits Regulation on Output of USB Switch**
- **Active Load Reduces Output Overshoot**
- $\overline{FLT}$ Flag Indicates Overcurrent on the USB Output
- 1.5ms $\overline{FLT}$ Flag Delay Filters Hot Plug Events
- USB Output Current Monitor
- Wide Input Rage: Operation from 5V to 35V
- Withstands Input Transient to 60V
- 2.5A Maximum Output Current
- Survives Output Short to GND and Car Battery
- Adjustable Switching Frequency: 300kHz to 2.2MHz
- Synchronizable from 300kHz to 2.2MHz
- Small, Thermally Enhanced 16-Lead MSOP Package

APPLICATIONS

- Automotive USB
- Industrial USB

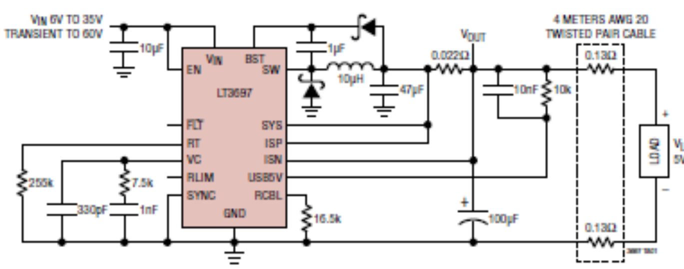
DESCRIPTION

The LT®3697 is a 35V, 2.5A step-down switching regulator designed to power 5V USB applications. A precise output voltage and programmable cable drop compensation maintain accurate 5V regulation at the USB socket connected to the end of a long cable. The accurate, programmable current limit can eliminate the need for a USB power switch and improve system reliability. The provided 180mA active load reduces output overshoot during load transients. Dual feedback allows regulation on the output of a USB switch and limits cable drop compensation to a maximum of 6.1V output, protecting USB devices during fault conditions. A separate 5V output can be taken from the SYS terminal to power auxiliary circuitry such as a USB hub controller. The LT3697 also provides a load current monitor output and an overcurrent fault indicator.

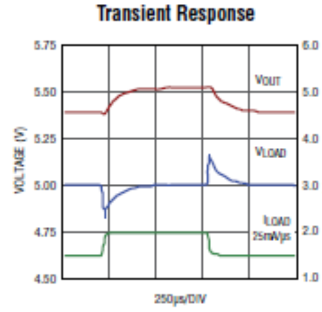
The LT3697 operates from 300kHz to 2.2MHz and withstands input voltage transients up to 60V. The device's output survives shorts to ground and to the battery. A current mode topology is used for fast transient response and good loop stability. Shutdown reduces input supply current to less than 1µA. The LT3697 is available in a 16-lead MSOP package with an exposed pad for low thermal resistance.

TYPICAL APPLICATION

5V Step-Down Converter with Cable Drop Compensation and Output Current Limit



Transient Response



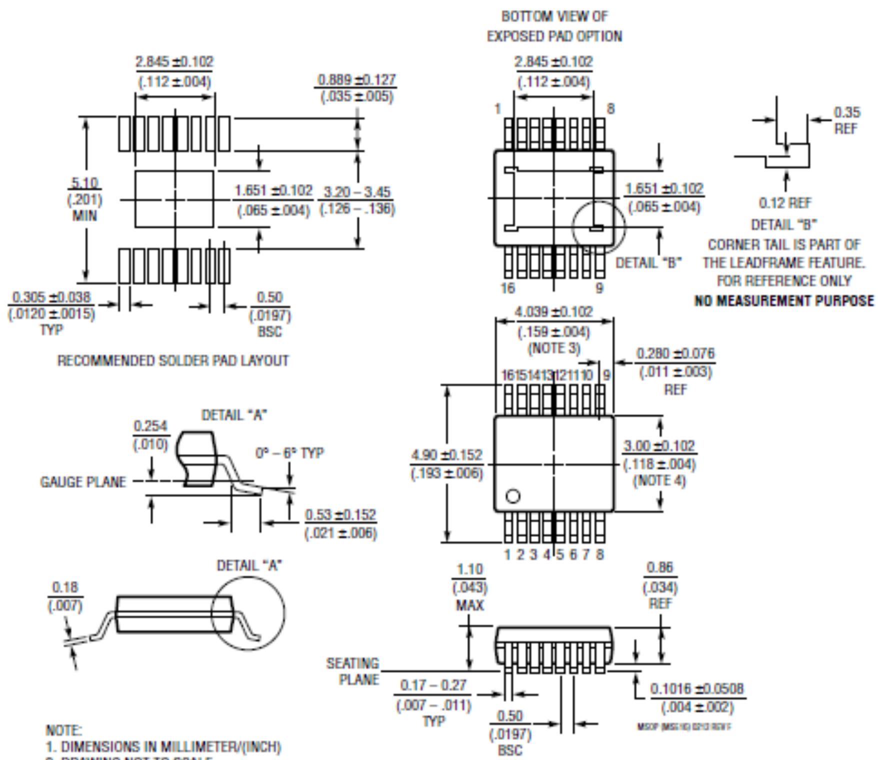
3697f

Figure A3.3. Linear Technologies LT3697 Datasheet.

PACKAGE DESCRIPTION

Please refer to http://www.linear.com/designtools/packaging/ for the most recent package drawings.

MSE Package
16-Lead Plastic MSOP, Exposed Die Pad
(Reference LTC DWG # 05-08-1667 Rev F)



NOTE:
1. DIMENSIONS IN MILLIMETER/(INCH)
2. DRAWING NOT TO SCALE
3. DIMENSION DOES NOT INCLUDE MOLD FLASH, PROTRUSIONS OR GATE BURRS. MOLD FLASH, PROTRUSIONS OR GATE BURRS SHALL NOT EXCEED 0.152mm (.006") PER SIDE
4. DIMENSION DOES NOT INCLUDE INTERLEAD FLASH OR PROTRUSIONS. INTERLEAD FLASH OR PROTRUSIONS SHALL NOT EXCEED 0.152mm (.006") PER SIDE
5. LEAD COPLANARITY (BOTTOM OF LEADS AFTER FORMING) SHALL BE 0.102mm (.004") MAX
6. EXPOSED PAD DIMENSION DOES INCLUDE MOLD FLASH. MOLD FLASH ON E-PAD SHALL NOT EXCEED 0.254mm (.010") PER SIDE.

Figure A3.3. Continued.

FLIR TAU 2

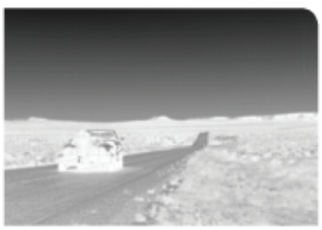
Longwave Infrared Thermal Camera

FLIR® Tau® 2 thermal imaging cameras offer an unmatched set of features, making them well-suited for demanding applications, such as unmanned vehicles (UVs), thermal weapon sights, and handheld imagers. Improved electronics now give Tau 2 even more capabilities, including radiometry, increased sensitivity, a 60Hz frame rate, and powerful image processing modes that dramatically improve detail and contrast.

IMPROVED IMAGE PROCESSING

For clearer imagery, edge sharpening, and contrast

- Second generation Digital Detail Enhancement (DDE)
- Active Contrast Enhancement (ACE)
- Smart Scene Optimization (SSO)
- Information Based HEQ (IBHEQ) automatically adjusts AGC
- Silent Shutterless NUC for continuous image improvement



Tau image Without ACE

Tau image with new ACE feature

ACCURATE TEMPERATURE MEASUREMENT

Supports radiometry, analytics and telemetry

- TLinear output places temperature data in each pixel
- Adjustable isotherm thresholds colorize temperatures of interest
- Rugged and reliable in all terrain

COMMON FEATURES ACROSS MODELS

Fosters improved OEM integration

- 640, 336 and 324 resolutions
- Multiple lens and FOV options
- 60Hz or 30mK frame rates
- Mechanical / electrical compatibility
- FLIR brand and support

www.flir.com/Tau2

FLIR | The World's Sixth Sense™

Figure A3.4. FLIR Tau 2 Datasheet.

Imaging Specifications

System Overview	
System Type	Uncooled LWIR Thermal Imager
Tau 2 640	640 x 512 VOx Microbolometer
Tau 2 336	336 x 256 VOx Microbolometer
Tau 2 324	324 x 256 VOx Microbolometer
Pixel Size	17 µm (Tau 2 640, 336); 25 µm (Tau 2 324)
Spectral Band	7.5 - 13.5 µm
Performance	<50 mK @ f/1.0
Outputs	
Analog Video	Field-switchable between NTSC and PAL
Tau 2 640	30/60Hz (NTSC); 25Hz/60Hz (PAL); <9Hz option for export (factory set)
Tau 2 336, 324	30/60 Hz (NTSC); 25/50 Hz (PAL) ; <9Hz option for export (factory set)
Digital Video	8- or 14-bit serial LVDS; 8- or 14-bit parallel CMOS; 8-bit BT.656
Operation & Control	
Image Control	Invert, revert, continuous digital zoom, dynamic zoom & pan, digital zoom presets,polarity, false color or monochrome, isotherms, AGC, second generation digital detail enhancement (DDE), image optimization (BPR, NUC & AGC'd video),Active Contrast Enhancement (ACE, Information Based Histogram Equalization (IBHEQ), Smart Scene Optimization (SSO), settable splash screens
Camera Control	Manual via SDK & GUI, dynamic range switching (Tau 2 324 only)
Signal Interface	Camera Link (Expansion Bus Accessory Module), discrete I/O controls available, RS-232 compatible (57,600 & 921,600 baud), external sync input/output, power reduction switch (removes analog video)
FFC Duration	<0.5 sec
Physical Attributes	
Size	1.75" x 1.75" x 1.75" (less lens)
Mounting Interface	6 attach points in lens mount, M2 x 0.4 on 3 sides, 2 per side (sealable bulkhead mounting feature on lens barrel [M29 x 1.0], WFOV only)
Power	
Input Voltage	4.0 – 6.0 VDC
Primary Electrical Connector	50-pin Hirose
Power Dissipation	~ 1.0 W (Tau 2 324 & 336); <1.2 W (Tau 2 640); <1.3W (Tau 2 640/60Hz)
Time to Image	<5 seconds (Tau 2 640); <4 seconds (Tau 2 336 and 324)
Environmental	
Operating Temperature Range	-40° C to +80° C external temp
Storage Temperature Range	-55° C to +95° C external temp
Scene Temp Range	High gain: -40°C to +160°; Low gain: -40°C to +550°
Shock	200 g shock pulse with 11 msec sawtooth
Temperature Shock	5°/min
Vibration	4.3 g 3 axes, 8 hours each
Humidity	5 - 95% non-condensing
Operational Altitude	+40,000 feet
ROHS, REACH, and WEEE	Compliant

Applications:

Unmanned Airborne Vehicles
Handheld Imagers
Security Cameras
Maritime cameras
Thermal weapon sights

SANTA BARBARA
FLIR Systems, Inc.
70 Castilian Drive
Goleta, CA 93117
USA
PH: +1 805.690.5097

PORTLAND
Corporate Headquarters
FLIR Systems, Inc.
27700 SW Parkway Ave.
Wilsonville, OR 97070
USA
PH: +1 866.477.3687

EUROPE
FLIR Systems
Luxemburgstraat 2
2321 Meer
Belgium
PH : +32 (0) 3665 5100

CHINA-SHANGHAI
FLIR Systems Co.,Ltd.
K301-302, No 26 Lane
168, Daduhe Road,
Putuo District, Shanghai
200062, P.R.China
PH: +86-21-5169 7628

www.flir.com
NASDAQ: FLIR

www.flir.com/Tau2

FLIR | The World's Sixth Sense™

Figure A3.4. Continued.